

IN THE COURT OF APPEAL (CIVIL DIVISION)
ON APPEAL FROM THE HIGH COURT OF JUSTICE
CHANCERY DIVISION (PATENTS COURT)
The Hon Mr Justice Floyd
[2012] EWHC 1789 (Pat)

Royal Courts of Justice
Strand, London, WC2A 2LL

Date: 03/05/2013

Before:
LORD JUSTICE RICHARDS
LORD JUSTICE LEWISON
and
LORD JUSTICE KITCHIN

Between:
HTC Europe Co Ltd

Appellant
in action
2043 (the
'022 Patent)

-and-

Apple Inc
(a company incorporated under the laws of the
State
of California)

Respondent
in action
2043 (the
'022 Patent)

-and between-

Apple Inc
(a company incorporated under the laws of the
State of California)

Appellant
in action
2044 (the
'948 Patent)

-and-

HTC Corporation (a company incorporated
under the laws of the Republic of China)

Respondent
in action
2044 (the
'948 Patent)

Mr Guy Burkill QC and Mr Joe Delaney (instructed by Freshfields Bruckhaus
Deringer LLP) appeared for Apple Inc
Dr Justin Turner QC (instructed by the Treasury Solicitor) appeared
for the Comptroller-General of Patents

Hearing dates: 12/13 March 2013

Approved Judgment

Lord Justice Kitchen:

Introduction

1. This appeal concerns a judgment of Floyd J dated 4 July 2012 and his consequential order following the trial of four actions between HTC Europe Co Ltd and HTC Corporation (together “HTC”) and Apple Inc (“Apple”). The actions involved four patents owned by Apple but on this appeal, brought with the permission of the judge, we are concerned with only two of them, namely European Patents No. 2 098 948 (the “948 patent”) and No. 1 964 022 (the “022 patent”).
2. The 948 patent relates to computer devices with touch sensitive screens which are capable of responding to more than one touch at a time. The judge found claims 1 and 2 were invalid because they related to computer programs as such. He also found claim 1 (but not claim 2) invalid for obviousness in the light of the common general knowledge. Apple appeals against both of these findings.
3. The 022 patent relates to ways of unlocking computer devices with touch sensitive screens. Various claims were in issue and the judge found all of them invalid, some for lack of novelty and others for obviousness. Apple appeals only against the finding that claims 5 and 17 were invalid for obviousness in the light of an earlier device referred to as the “Neonode”.
4. Apple was also granted permission to appeal against the judge’s findings of non infringement of claims 1 and 2 of the 948 patent. However, Apple and HTC have agreed to settle their differences and, as a result, HTC has not appeared on this appeal and has no interest in its outcome. The appeal against the finding of non infringement is therefore not pursued. Nevertheless, Apple wishes to establish that the claims of its patents to which I have referred are valid and, in accordance with the guidance given by this court in *Halliburton Energy Services Inc v Smith International (North Sea) Ltd* [2006] EWCA Civ 185; [2006] RPC 26 and, more recently, in *Apimed Medical Honey Ltd v Brightwake Ltd* [2012] EWCA Civ 5, the Comptroller was invited to consider whether he wished to appear on the appeal on the basis that his costs would be paid by Apple. The Comptroller duly indicated that he did indeed wish to appear and has been represented by Mr Justin Turner QC. As explained in *Apimed*, the Comptroller’s role on such an appeal is to protect the public interest by intervening to the extent necessary to prevent invalid patents being restored to the register. With this in mind, Mr Turner has assisted us by meeting criticisms of the judge’s reasoning and drawing to our attention materials relevant to the judge’s conclusions, while maintaining a balanced view consistent with his position. I should also note that Apple’s solicitors have properly and helpfully assisted the Comptroller to perform this important task by providing to him copies of the relevant documents.
5. Accordingly, the issues which arise on the appeal are whether the judge fell into error in concluding that:
 - (i) claims 1 and 2 of the 948 patent are invalid because they relate to computer programs as such and so claim excluded subject matter;

- (ii) claim 1 of the 948 patent is invalid for obviousness in the light of the common general knowledge;
- (iii) claims 5 and 17 of the 022 patent are invalid for obviousness in the light of the Neonode.

The 948 patent

6. The 948 patent is entitled “Touch Event Model” and has a priority date of 4 March 2008. There was no dispute that it is addressed to a notional skilled but uninventive team working in industry in the development of system software of a graphical user interface (“GUI”) for a multi-touch device. As the judge held, the team would be concerned with the development of products rather than academic research and would include someone with expertise in software engineering and someone with experience of implementing GUIs.
7. Each side therefore called an expert witness to assist the judge as to the knowledge and attitudes of such a team. I must say a little about them because the judge’s findings on the issue of obviousness were founded on the evidence they gave. Apple called Dr Brad Karp, a Reader in Computer Systems and Networks and Head of the Networks Research Group in the Department of Computer Science at University College London. The judge rejected an attack on Dr Karp’s objectivity, noting that he was a cautious witness who chose his words with care. The judge also accepted that Dr Karp was a knowledgeable computer scientist. The judge considered, however, that Dr Karp’s expertise lay primarily in the field of computer networks; he had never been involved in writing system software for a GUI and his experience of GUIs and their toolkits was simply as a user. As a result, he was not well equipped to assist the court as to the thinking of a team concerned with writing system software for a GUI.
8. HTC called Dr Daniel Wigdor who is an assistant professor of computer science at the University of Toronto, and an affiliate of the School of Applied Science and Engineering at Harvard University. Between 2005 and 2010 he worked first at Mitsubishi Electric Research Laboratories (“MERL”) and later at Microsoft. His responsibilities at both companies included the development of multi-touch devices. Apple sought to characterise Dr Wigdor as being unduly creative and a member of the research community. They also suggested that the judge should approach Dr Wigdor’s evidence concerning the common general knowledge with caution. The judge accepted that he should approach all the evidence of common general knowledge, including that given by Dr Wigdor, with caution but nevertheless considered that Dr Wigdor had endeavoured to consider what would have been known to the uninventive skilled team and overall found him to be a frank and very helpful witness.

The technical background and common general knowledge

9. The judge set out the technical background and the common general knowledge at [22], [31]-[34] and Appendix 1. For the purposes of this appeal, the following matters are of particular relevance.

10. Computer software is commonly structured in layers. The lowest software layer is the operating system or OS which interacts directly with the hardware, for example, by reading external inputs and producing external outputs through hardware elements such as a touch pad or screen. Device drivers make up the bottom layer of the OS, closest to the hardware, and are responsible for directly reading and modifying the hardware's state.
11. Above the OS are the run-time libraries. These consist of reusable software routines that implement the functionality required by applications, for example by performing mathematical computations and setting timers. The OS and the run-time libraries are often called the system software.
12. Above the system software sit the applications which carry out tasks for the user, such as web browsing or reading and writing e-mails. Some application programs may be written by the manufacturer of the device, but often they are created by third parties.
13. There is a defined interface between the system software and the application software called the application programming interface or API. This enables the application software developers to assemble and use a set of user interface elements called UI elements, such as buttons, check boxes and scroll bars, which together form the user interface (UI) toolkit. These are of great importance because they ease the task of application software developers, as the judge explained at [32]:

“32. A general goal of operating system designers is to ease the task of application software developers. The success of an operating system is likely to be driven by the scale of its adoption by application developers as well as end users. This can be done by providing features within the system on which application developers can build, reducing the amount of software which they have to write. The decisions taken by system developers as to what facilities to include in the system software have an impact on the cost of development of the application software. Thus the provision of a “button”, a UI element, in the system software can allow the application developer to incorporate it by reference in the application, without the need to provide program code as to how it should look or how it should respond to input from the user.”

14. I must also say a word about inputs to the computer. These may be effected by a variety of means such as a keyboard or a mouse. The processing of an input into a signal or “event” begins in the OS, where a device driver is notified by the hardware that an input has occurred. The information processed by the driver is then passed either to a run-time library and from there to the application, or directly to the application. The judge explained the importance of UI elements in such a system and how their properties may be defined in the UI toolkit in these terms at [33]-[34]:

- “33. It was common to allow for the properties of a UI element to be defined by a software developer in the UI toolkit. Properties may be various. Where a property is capable of having only two possible values, it can be defined by setting the value of a “flag” attached to the UI element. The flag is stored as a single binary bit, and is either set (1) or not set (0). The property of a button whereby it is either enabled or not enabled could be indicated by a flag.
34. Dr Wigdor explained that it was well known to use the setting of a flag on a UI element to indicate whether or not particular events should be sent to that UI element. He also explained that the practice of limiting events sent to a particular UI element as a method of simplifying the development of software was also part of the common general knowledge. In each case he gave examples. Although Dr Karp quibbled with some of the examples in his written evidence, he accepted that it was common general knowledge to use a flag so that events *of a particular type* were not sent to the UI element. He also accepted that it was generally known that this could be beneficial for the software developer.”
15. In the case of a device with a touch screen, the GUI comprises a package of software, including applications and run-time libraries, which manages the content of the display and processes the touch screen inputs. Users can interact with the GUI by manipulating displayed content using the touch screen. Typically, the designer of a touch screen device has the responsibility of providing touch screen hardware, including electronics that detect the touches and, also system software that encodes such touches into events, and passes the touch events to the applications software. Any touch will be at a specific point or points on the display. Different parts of the display (called “views”) may be associated with different applications or with different items within a single application.
16. Devices with touch screens which responded to multiple touches were well known at the priority date. Prominent among them was the Apple iPhone 1, but its software was proprietary and not available to the public.

The specification

17. The 948 specification explains that the invention relates to multi-touch enabled devices and, more specifically, the recognition of single and multiple touch events in such devices.
18. The specification then provides the background to the invention, explaining that multi-touch enabled devices which could sense multiple touches at the same time were known in the art. So, for example, and adopting an illustration used by the parties at the hearing, a two fingered “pinch” and “de-pinch” gesture may be used to control zooming into and out of a screen image. But, as Apple emphasised, if a

user can touch multiple points at once, possibilities of conflict arise which have to be addressed by the programmer. So, for example, in the case of a music application, if the user presses two different buttons at once, such as “Play” and “Delete”, the question arises as to what should be done with the relevant music file. So also, in the case of a calculator application, if a user presses the same button twice, the device must determine whether this is to count as the entering of one or two separate digits. The need to assess and process multiple touch events therefore creates difficulties and introduces complexity, as the specification explains at [0006]:

“On the other hand, if a multi-touch interface is being used, two or more touch events can simultaneously occur at different portions of the display. This can make it difficult to split the display into different portions and have different independent software elements process interactions associated with each portion. Furthermore, even if the display is split up into different portions, multiple touch events can occur in a single portion. Therefore, a single application, process or other software element may need to process multiple simultaneous touch events. However, if each application, process or other software element needs to consider multiple touch interactions, then the overall cost and complexity of software running at the multi-touch enabled device may be undesirably high. More specifically, each application may need to process large amounts of incoming touch data. This can require high complexity in applications of seemingly simple functionality, and can make programming for a multi-touch enabled device generally difficult and expensive. Also, existing software that assumes a single pointing device can be very difficult to convert or port to a version that can operate on a multi-point or a multi-touch enabled device.”

19. The problem is also explained later in the specification at [0038] in these terms:

“The ability to handle multiple touches and multi-touch gestures can add complexity to the various software elements. In some cases, such additional complexity can be necessary to implement advanced and desirable interface features. For example, a game may require the ability to handle multiple simultaneous touches that occur in different views, as games often require the pressing of multiple buttons at the same time. However, some simpler applications and/or views (and their associated software elements) need not require advanced interface features. For example, a simple button (such as button 306) can be satisfactorily operable with single touches and need not require multi-touch functionality. In these cases, the underlying OS may send unnecessary or excessive touch data (e.g. multi-touch data) to a software element associated

with a view that is intended to be operable by single touches only (e.g. a button). Because the software element may need to process this data, it may need to feature all the complexity of a software element that handles multiple touches, even though it is associated with a view for which only single touches are relevant. This can increase the cost of development of software for the device, because software elements that have been traditionally very easy to program in a mouse interface environment (i.e. various buttons, etc) may be much more complex in a multi-touch environment.”

20. The nature of the invention is then outlined. The specification explains that, to simplify the recognition of single and multiple touch events, each view within a particular window can be configured as a multi-touch view or a single touch view. Further, each view can be configured as an exclusive or a non-exclusive view. Depending on the configuration of the view, touch events in that and other views can be ignored or recognised. Importantly, ignored touches need not be sent to the application.
21. As the judge explained, the patent proposes the use of particular flags associated with views on the screen. They are:
 - i) multi-touch flags which indicate whether a particular view is allowed to receive multiple simultaneous touches;
 - ii) exclusive touch flags which indicate whether a particular view allows other views to receive touches while the flagged view is receiving a touch.
22. Apple emphasised, fairly in my view, that these flags provide very specific and separate functionality. The multi-touch flag has no influence when separate touches are made to two different views, while the exclusive touch flag has no influence when simultaneous touches are made to the same view.
23. A flow chart showing the operation of the multi-touch flag according to one embodiment of the invention is shown in figure 4:

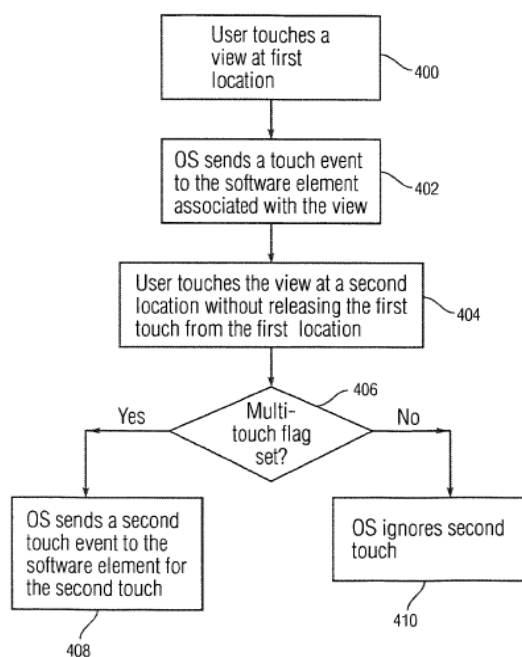
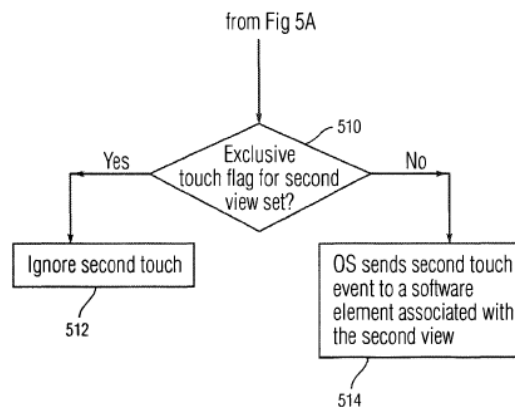
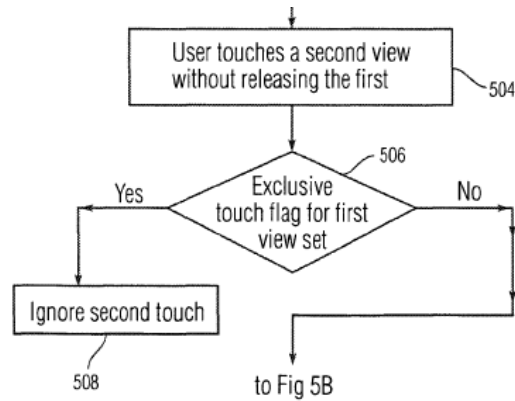


Fig. 4

24. In summary, the OS can determine whether the multi-touch flag for a particular view is set. If the flag is set, then the view can handle multiple contemporaneous touches. If, on the other hand, the multi-touch flag is not set, the OS can ignore or block the second touch with the result that no touch events associated with the second touch are sent to the software element associated with that view. The benefit is explained at [0045]:

“Thus, embodiments of the present invention can allow relatively simple software elements that are programmed to handle only a single touch at a time to keep their multi-touch flag unasserted, and thus ensure that touch events that are part of multiple contemporaneous touches will not be sent to them. Meanwhile, more complex software elements that can handle multiple contemporaneous touches can assert their multi-touch flag and receive touch events for all touches that occur at their associated views. Consequently, development costs for the simple software elements can be reduced while providing advanced multi-touch functionality for more complex elements.”

25. The method of operation of the exclusive touch flag is illustrated in figures 5A and 5B:



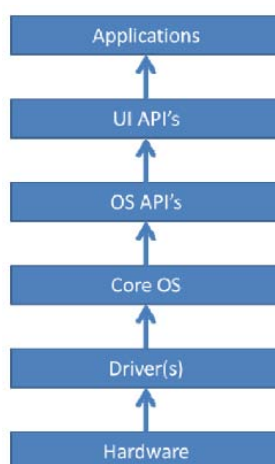
26. At the first step, the OS checks whether the exclusive touch flag for the first view is asserted. If it is, all touches at other views are ignored. If it is not, the OS can determine whether the exclusive view flag for the second view is asserted. If it is, the second touch at that other view is ignored. If it is not, the OS can send a touch event associated with the second touch to the software element associated with the second view.

27. The benefit of this aspect of the system is explained at [0049]:

“Thus, the exclusive touch flag can ensure that views flagged as exclusive only receive touch events when they are the only views on the display receiving touch events. The exclusive flag can be very useful in simplifying the software of applications running on a multi-touch enabled device. In certain situations, allowing multiple views to receive touches simultaneously can result in complex conflicts and errors. For example, if a button to delete a song and a button to play a song are simultaneously pressed, this may cause an error. Avoiding such conflicts may require complex and costly software. However, embodiments of the present invention can reduce the need for such software by providing an exclusive touch flag which can ensure that a view that has that flag set will receive touch events only when it is the

only view that is receiving a touch event. Alternatively, one or more views can have their exclusive touch flags unasserted, thus allowing multiple simultaneous touches at two or more of these views.”

28. The parties were agreed that the flags behave independently. Accordingly, one or more flags may be associated with one or more views. Some embodiments may use the multi-touch flag only and others, the exclusive touch flag only.
29. The following figure, taken from Dr Wigdor’s first report, illustrates the various system layers involved in the operation of a device using the claimed method:



30. The invention operates at all levels up to the user interface APIs (UI APIs) and the flags control events passed from the lower layers. If, in any layer, an event is ignored then it will not be passed up to the next layer. Thus, if the system software ignores a touch event it will never be delivered to the software element associated with that view. As a result, the invention facilitates the work of application programmers and reduces the need for them to design complex software.
31. Claim 1 (with reference numerals added by the judge) reads as follows:
- “(i) A method for handling touch events at a multi-touch device, comprising:
 - (ii) displaying one or more views;
 - (iii) executing one or more software elements, each software element being associated with a particular view;
 - (iv) associating a multi-touch flag or an exclusive touch flag with each view, said multi-touch flag indicating whether a particular view is allowed to receive multiple simultaneous touches and said exclusive touch flag indicating whether a particular view allows other views to receive touch events while the particular view is receiving a touch event;

(v) receiving one or more touches at the one or more views;
and

(vi) selectively sending one or more touch events, each touch event describing a received touch, to one or more of the software elements associated with the one or more views at which a touch was received based on the values of the multi-touch and exclusive touch flags.”

32. As the judge explained, it is common ground that although feature (iv) ends with the word “touch event” it should, for consistency, simply read “touch”. Claim 2 then adds the following feature:

“if a multi-touch flag is associated with a particular view, allowing other touch events contemporaneous with a touch event received at the particular view to be sent to the software element associated with the other views.”

Computer program as such

33. The exclusions from patentability are contained in s.1(2) of the Patents Act 1977. This implements Article 52 of the EPC which (as amended) reads:

- “(1) European patents shall be granted for any inventions, in all fields of technology, provided that they are new, involve an inventive step and are susceptible of industrial application.
- (2) The following in particular shall not be regarded as inventions within the meaning of paragraph 1:
- (a) discoveries, scientific theories and mathematical methods;
 - (b) aesthetic creations;
 - (c) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;
 - (d) presentations of information.
- (3) Paragraph 2 shall exclude the patentability of the subject-matter or activities referred to therein only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such.”

34. Upon this appeal we are concerned once again with the exclusion of computer programs contained in Art 52 (2)(c), subject to the qualification in Art 52(3) that it applies only to the extent to which the patent relates to such subject matter as such.

35. In *Aerotel Ltd v Telco Holdings Ltd; Macrossan's Patent Application* [2006] EWCA Civ 1371, [2007] RPC 7, this court reviewed various decisions of the EPO Boards of Appeal and earlier decisions in this jurisdiction. It was conscious of the need to place great weight on the decisions of the Boards of Appeal but given what it described as the state of conflict between them, it explained it would be premature to do so, noting that the matter might have to be reconsidered if and when the Enlarged Board ruled on the issue. In the meantime this court was bound by its own precedents and, in particular, the decisions in *Merrill Lynch's Application* [1989] RPC 561 (CA), *Gale's Application* [1991] RPC 305 (CA) and *Fujitsu Ltd's Application* [1997] RPC 608 (CA) to consider whether the invention made a technical contribution to the known art, with the rider that novel or inventive purely excluded subject matter does not count as a technical contribution.
36. The court also explained that the following four stage approach is consistent with its earlier decisions and the statutory test and provides a convenient way of addressing the exclusion:
- i) properly construe the claim;
 - ii) identify the actual contribution;
 - iii) ask whether it falls solely within the excluded subject matter;
 - iv) check whether the actual or alleged contribution is actually technical in nature.
37. The first step poses no difficulty for it simply involves a conventional exercise of interpretation. The second step is, the court noted, more problematical. How is the contribution to be assessed? In this regard, the court recorded with apparent approval the submission made by Mr Birss (as he then was) on behalf of the Comptroller that the exercise involves looking at substance not form and assessing what the inventor has added to human knowledge. The court continued at [44]:
- “Mr Birss added the words “or alleged contribution” in his formulation of the second step. That will do at the application stage – where the Office must generally perforce accept what the inventor says is his contribution. It cannot actually be conclusive, however. If an inventor claims a computer when programmed with his new program, it will not assist him if he alleges wrongly that he has invented the computer itself, even if he specifies all the detailed elements of a computer in his claim. In the end the test must be what contribution has actually been made, not what the inventor says he has made.”
38. The third step involves asking whether the contribution thus identified consists of excluded subject matter as such. The final step is then a check, which may not be necessary, and involves assessing whether the contribution is technical.
39. Some two years later this court again considered the exclusion in *Symbian v Comptroller-General of Patents* [2008] EWCA Civ 1066, [2009] RPC 1. In the

meantime, the Boards of Appeal had themselves considered *Aerotel* in decision T 0154/04 *Duns Licensing Associates*, describing it as “not consistent with a good-faith interpretation” of the EPC and, indeed, as “irreconcilable” with it. The Board in *Duns* explained that any reference to the prior art in considering Art 52 would lead to “insurmountable difficulties”, it being a concept “finely tuned” by a combination of Arts 54-56. It proceeded to endorse what this court had described as the “any hardware” approach, that is to say taking into account all the features of the claimed invention in considering Art 52 but only taking into account technical features in assessing inventive step; or, in other words, holding that the innovation must be on the technical side and not in a non-patentable field. A number of other decisions of the Boards of Appeal subsequent to *Aerotel* took broadly the same approach.

40. Despite the rather trenchant terms used by the Board in *Duns*, the court in *Symbian* explained that the approaches in *Aerotel* and *Duns* and in the great majority of other cases were, on analysis, capable of reconciliation. As Lord Neuberger of Abbotsbury said of the third step:

“So far as we can see, there is no reason, at least in principle, why that test should not amount to the same as that identified in *Duns*, namely whether the contribution cannot be characterised as ‘technical’.”

41. I respectfully agree in terms of result for it seems to me that whichever route is followed, one ought to end up at the same destination. On the *Aerotel* approach a claimed invention whose only contribution is not technical or lies in an excluded field falls to be rejected under Art 52 under steps (iii) and (iv), whereas on the *Duns* approach such an invention falls to be rejected under Art 56 because such a contribution must be cut out of the assessment of inventive step.
42. Nevertheless, conscious of the need for consistency, the court in *Symbian* considered whether it could be satisfied that the Boards of Appeal had formed a settled view on the point which differed from the conclusion expressed in its own previous decision in *Aerotel* and, if so, whether it should now follow that approach. It decided that it should not do so for various reasons. First, there had been no decision of the Enlarged Board; second, the approaches taken by the Boards in the various decisions since *Aerotel* were not identical; third, on least on one of those approaches, it seemed the computer program exclusion may have lost all its meaning; fourth, extra-curial remarks of Mellulis J of the Bundesgerichtshof suggested that the English courts were not alone in their concerns about the approach of the Boards; and finally, if this court was seen to depart too readily from its previous approach it would risk throwing the law into disarray.
43. A few days after this court had given its decision in *Symbian*, in a referral under Art 112(1)(b), the President of the EPO asked the Enlarged Board to consider a set of questions relating to the patentability of computer programs to which she considered the Boards of Appeal had given different decisions. In its decision G3/08 of 12 May 2010, the Enlarged Board ruled the reference was inadmissible on the basis that the notion of different decisions in Article 112(1)(b) had to be understood restrictively in the sense of conflicting decisions, and legal development could not on its own form the basis for a referral. It considered the

decisions identified by the President were not conflicting although, in one case, they did reveal a legitimate development of the case law.

44. In these circumstances neither Apple nor the Comptroller suggested it would be appropriate for this court to abandon the approach explained by this court in *Aerotel*. In my judgment they were right not to do so. For the reasons given in *Symbian*, I believe we must continue to consider whether the invention made a technical contribution to the known art, with the rider that novel or inventive purely excluded subject matter does not count as a technical contribution. Further, in addressing that issue I believe it remains appropriate (though not strictly necessary) to follow the four stage structured approach adopted in *Aerotel*.
45. How then is it to be determined whether an invention has made a technical contribution to the art? A number of points emerge from the decision in *Symbian* and the earlier authorities to which it refers. First, it is not possible to define a clear rule to determine whether or not a program is excluded, and each case must be determined on its own facts bearing in mind the guidance given by the Court of Appeal in *Merrill Lynch and Gale* and by the Boards of Appeal in Case T 0208/84 *Vicom Systems Inc* [1987] OJ EPO 14, [1987] 2 EPOR 74, Case T 06/83 *IBM Corporation/Data processing network* [1990] OJ EPO 5, [1990] EPOR 91 and Case T 115/85 *IBM Corporation/Computer-related invention* [1990] EPOR 107.
46. Second, the fact that improvements are made to the software programmed into the computer rather than hardware forming part of the computer does not make a difference. As I have said, the analysis must be carried out as a matter of substance not form.
47. Third, the exclusions operate cumulatively. So, for example, the invention in *Gale* related to a new way of calculating a square root of a number with the aid of a computer and Mr Gale sought to claim it as a ROM in which his program was stored. This was not permissible. The incorporation of the program in a ROM did not alter its nature: it was still a computer program (excluded matter) incorporating a mathematical method (also excluded matter). So also the invention in *Macrossan* related to a way of making company formation documents and Mr Macrossan sought to claim it as a method using a data processing system. This was not permissible either: it was a computer program (excluded matter) for carrying out a method for doing business (also excluded matter).
48. Fourth, it follows that it is helpful to ask: what does the invention contribute to the art as a matter of practical reality over and above the fact that it relates to a program for a computer? If the only contribution lies in excluded matter then it is not patentable.
49. Fifth, and conversely, it is also helpful to consider whether the invention may be regarded as solving a problem which is essentially technical, and that is so whether that problem lies inside or outside the computer. An invention which solves a technical problem within the computer will have a relevant technical effect in that it will make the computer, as a computer, an improved device, for example by increasing its speed. An invention which solves a technical problem outside the computer will also have a relevant technical effect, for example by

controlling an improved technical process. In either case it will not be excluded by Art 52 as relating to a computer program as such.

50. In *AT &T Knowledge Ventures LP's Patent Application* [2009] EWHC 343 (Pat), [2009] FSR 19 Lewison J (as he then was) reviewed many of the decisions referred to in *Aerotel* and *Symbian* and derived from them the following set of what he described as useful signposts:
- i) whether the claimed technical effect has a technical effect on a process which is carried on outside the computer;
 - ii) whether the claimed technical effect operates at the level of the architecture of the computer; that is to say whether the effect is produced irrespective of the data being processed or the applications being run;
 - iii) whether the claimed technical effect results in the computer being made to operate in a new way;
 - iv) whether there is an increase in the speed or reliability of the computer;
 - v) whether the perceived problem is overcome by the claimed invention as opposed to being merely circumvented.
51. I respectfully agree these are useful signposts, forming as they do part of the essential reasoning in many of the decisions to which we must look for guidance. But that does not mean to say they will be determinative in every case. I have also had the benefit of reading in draft Lewison LJ's judgment in this case. I respectfully agree with that too, including his observation that, in the light of Mann J's judgment in *Gemstar-TV Guide International Inc v Virgin Media Ltd* [2009] EWHC 3068 (Ch), [2010] RPC 10, he would adopt as his fourth signpost the less restrictive question whether a program makes a computer a better computer in the sense of running more efficiently and effectively as a computer. Indeed, this is, to my mind, another illustration of the still broader question whether the invention solves a technical problem within the computer.
52. The problem which the invention in this case seeks to address is how to deal with multiple touches in multi-touch enabled devices. Assessing and processing multi-touch events creates difficulties and complexities for designers. As the specification explains, if every application must consider all multi-touch events then they may need to process large amounts of information and this makes programming such applications difficult and expensive.
53. The invention of claim 1 deals with this problem by providing a method for handling touch events in multi-touch enabled devices in which particular flags are associated with views on the screen. These flags are of two kinds: multi-touch flags which indicate whether a particular view is allowed to receive multiple simultaneous touches; and exclusive touch flags which indicate whether a particular view allows other views to receive touches while it is receiving a touch. As Dr Wigdor explained, it involves first, the hardware and then each and every

layer of the system up to the cooperation between the UI APIs and the applications, and it has the effect of making it much simpler to write the programs for those applications.

54. The judge came to the conclusion that the invention was not patentable because it related to a computer program as such. He held (at [94]) that one aspect of the contribution lay in the software that processed multi-touch inputs and this was plainly excluded subject matter; and the other aspect of the contribution was the advantage that it made it easier to write software for multi-touch devices, and this too lay wholly within excluded subject matter.
55. He continued (at [95] – [98]) that ease of writing application software was not a relevant technical effect; that although the claimed method applied at the OS level, not every method operating at this level was patentable; that the invention did not make multi-touch devices work in a new way in any relevant sense because it simply involved a redistribution of the necessary data processing; and there was no evidence the invention caused any increase in the speed or reliability of such devices.
56. I believe the judge fell into error in reaching this conclusion and reasoning as he did. The problem which the patent addresses, namely how to deal with multiple simultaneous touches on one of the new multi-touch devices, is essentially technical, just as were the problems of how to communicate more effectively between programs and files held in different processors within a known network which lay at the heart of the decision in *IBM Corporation/Data processing network*, and how to provide a visual indication about events occurring in the input/output device of a text processor which lay at the heart of the decision in *IBM Corporation/Computer related invention*.
57. Second, the solution to this problem lies in a method of dividing up the screen of such a device into views and configuring each view as a multi-touch view or a single-touch view using flags with a specific functionality in the manner I have described. This is a method which concerns the basic internal operation of the device and applies irrespective of the particular application for which the device is being used and the application software which it is running for that purpose. It causes the device to operate in a new and improved way and it presents an improved interface to application software writers. Now it is fair to say that this solution is embodied in software but, as I have explained, an invention which is patentable in accordance with conventional patentable criteria does not become unpatentable because a computer program is used to implement it. I believe the judge took his eye off the ball in focussing on the fact that the invention was implemented in software and in so doing failed to look at the issue before him as a matter of substance not form. Had he done so he would have found that the problem and its solution are essentially technical in nature and so not excluded from patentability.
58. Third, a practical benefit of the invention is that it presents a new and improved interface to application programmers, including third party programmers, and makes it easier for them to write application software for the multi-touch device. The device is, in a real practical sense, an improved device. This is not because it now runs different application programs but because it is, as a device, easier for

programmers to use. Once again, this emphasises the technical nature of the invention.

59. For all these reasons I consider the invention does make a technical contribution to the art and its contribution does not lie in excluded matter.

Obviousness

60. HTC contended that the 948 patent was invalid over the common general knowledge, and also over two prior art systems called Jazz Mutant Lemur and Zotov. But its primary attack was based upon the common general knowledge alone.
61. This attack was developed through the evidence of Dr Wigdor who explained that the patent is directed to two issues; namely how to enable developers to port what he described as legacy software for use with a multi-touch device, and how to ease the task of creating software which takes advantage of the functionality that multi-touch devices offer. He continued that the success of the new platform was known to be dependent in part upon the quality and quantity of content available for it. The creator of such a new platform would therefore consider those interested in porting legacy software and also the designers of new software.
62. As for legacy software, Dr Wigdor explained that these were programs which were designed to receive events associated with one touch at a time and so the skilled team would be faced immediately with the problem of how to handle concurrent touches. An extreme and simple solution would be not to allow concurrent touches at all so that the legacy application would consider it was running on a single-touch device. But this approach might, he thought, discourage the development of new multi-touch applications based upon the legacy programs. He continued that the skilled team would appreciate that an alternative would be to provide more fine-grained control and so would consider offering a way in which application developers could define whether particular UI elements, as the basic building blocks of applications, could opt in or out of receiving such multi-touch inputs.
63. Turning to new software, Dr Wigdor explained that it would be immediately apparent to the skilled team that many components of applications would not require multi-touch functionality, for example, individual buttons in a simple calculator application. He continued that it would be obvious to the team that requiring the authors of such software to write code to deal with multi-touch events would impose upon them an unnecessary extra burden. An obvious solution would, he thought, be to allow developers to opt-out of multi-touch input within and across individual UI elements. Dr Wigdor also thought it would be apparent that disabling the ability of the application to receive contemporaneous touches entirely would prevent developers of new application software from taking advantage of multi-touch inputs such as “pinch to zoom” gestures which were, by that time, well known from the iPhone 1. Thus, Dr Wigdor reasoned, it would occur to the skilled team, simply applying common sense, to allow some software elements to opt in or opt out of multi-touch functionality.

64. Having regard to the fact that most UI toolkits included the notion of a UI element having associated with it a flag which prevented some input events from being sent to that element, Dr Wigdor considered that an obvious solution to the challenge facing the skilled team would be to:

- i) define a flag to indicate the ability of a UI element to handle multiple touches or not, that is to say a multi-touch flag;
- ii) define a flag to indicate whether concurrent touches with other elements should be allowed, that is to say an exclusive touch flag.

65. In assessing an attack such as this, based as it is entirely on the common general knowledge, it is, I think, particularly important to have well in mind the structured approach to determining obviousness explained by this court in *Pozzoli v BDMO SA* [2007] FSR 37:

“(1)(a) Identify the notional ‘person skilled in the art’.

(b) Identify the relevant common general knowledge of that person.

(2) Identify the inventive concept of the claim in question or, if that cannot readily be done, construe it.

(3) Identify what, if any, differences exist between the matter cited as forming part of the "state of the art" and the inventive concept of the claim or the claim as construed.

(4) Ask whether, when viewed without any knowledge of the alleged invention as claimed: do those differences constitute steps which would have been obvious to the person skilled in the art or do they require any degree of invention?”

66. Step (1)(b), involves identifying the relevant common general knowledge relied upon and doing so with some precision. In the case of an attack founded only upon the common general knowledge, this is an essential preliminary step to identifying the differences between what was known and the inventive concept, and then ascertaining whether or not those differences constitute steps which required any degree of invention. Thus, as the judge himself explained in his earlier decision in *Ratiopharm v Napp* [2008] EWHC 3070 (Pat), [2009] RPC 11 at [156]:

“The second point is that it is important to be precise about what it is that is asserted to be common general knowledge. For example, in the present case it is admitted that “the existence of oxycodone” was common general knowledge. But the dispute here is not about whether a skilled person knew about oxycodone. The real dispute is about what oxycodone was used for. If the skilled person has not used oxycodone as an alternative to morphine for oral administration for moderate to severe pain, it becomes

difficult to argue that it would occur to him to use oxycodone in the course of deciding on a controlled release formulation for use in such circumstances.”

67. Then, in addressing the fourth crucial question, it is particularly important to be wary of hindsight. As I said in *Abbott Laboratories v Evysio* [2008] RPC 23 at [180]:
- “It is also particularly important to be wary of hindsight when considering an obviousness attack based upon the common general knowledge. The reason is straightforward. In attacking a patent, attention is focussed upon the particular development which is said to constitute the inventive step. With this development in mind it may be possible to mount an attack which is unencumbered by any detail which might point to non obviousness: *Coflexip v Stolt Connex Seaway* (CA) [2000] IP&T 1332 at [45]. It is all too easy after the event to identify aspects of the common general knowledge which can be combined together in such a way as to lead to the claimed invention. But once again this has the potential to lead the court astray. The question is whether it would have been obvious to the skilled but uninventive person to take those features, extract them from the context in which they appear and combine them together to produce the invention.”
68. Ultimately, as in all obviousness cases, the judge must assess all the relevant circumstances in answering the simple statutory question: was the invention obvious at the priority date? This has been described in many cases as a kind of jury question and the answer to it given by the judge should be treated with appropriate respect by an appellate court. Indeed, absent some error of principle, an appellate court should be very cautious in differing from a judge’s evaluation.
69. The judge began by directing himself as to the relevant legal principles at [4]-[8]. Then, at [76], he recorded Apple’s response to HTC’s case, explaining that it involved, on Apple’s case, a series of steps, each framed with the benefit of hindsight.
70. The judge then proceeded to analyse the parties’ respective positions, reasoning as follows. First, he recognised force in Apple’s submission that the primary focus of the skilled team would be new rather than legacy applications. Nevertheless, he considered that the skilled team would consider carefully how to design the interface to facilitate the writing of software which made use of the new multi-touch capability. He held, at [77], that the skilled team would see immediately that, while the multi-touch capability delivered desirable additional functionality, there would be situations where individual UI elements would not want to receive multi-touch events, and situations where concurrent touch events between UI elements should not be allowed.
71. This led the judge to identify, at [78], what he perceived to be the critical question:

“The critical question on obviousness is, as it seems to me, whether the skilled person would see that the way of dealing with the need identified in the previous paragraph would be at system level, or whether the skilled person would consider, as Dr Karp suggested, that the way to do it would be to send the events to the application software and “*consider that his work was done*”.”

72. At [79]-[80] the judge recorded Dr Karp’s acceptance during the course of his cross-examination that applications running on the well known iPhone 1 multi-touch device would contain four types of UI elements:

“(i) UI elements which will need the ability to receive multiple concurrent touches;

(ii) UI elements which require only a single touch, and multiple concurrent touches to that element will not be acted upon: e.g. keyboard buttons;

(iii) UI elements which need to be able to receive input which is concurrent with other input at other UI elements: eg holding down a shift key whilst pressing a letter;

(iv) UI elements whose functionality should not be invoked concurrently with that of other UI elements: for example operations which were in conflict with each other, such as “yes” and “no”.”

Further, the skilled team would want to make sure that applications running on any new multi-touch device would be able to exhibit similar behaviour.

73. That brought the judge to his answer to the critical question. Dr Karp said that the skilled team, faced with designing an OS to support applications of this sort, would take the path of least resistance and send all input events to the application software. As I have said, Dr Wigdor, on the other hand, maintained that an obvious solution was to use flags associated with each UI element, with those flags providing the necessary functionality. In evaluating these rival opinions, the judge was particularly influenced by first, Dr Karp’s acceptance that leaving the writing of the software to the application developer would be to impose a significant burden upon him. Second, the judge was not persuaded by Dr Karp’s evidence that the skilled team would not arrive at the use of flags as a way of introducing the necessary system filter in the light of his acceptance that flags were a known means of filtering types of event. In short, the judge preferred Dr Wigdor’s evidence on this issue.

74. Finally, the judge turned to consider the forensic question: if it was obvious, why was it not done before? In this regard, Apple relied upon an application called DTMouse used by Dr Wigdor’s team on a multi-touch device sold by MERL called the DiamondTouch. In this system flags were not used to control touch events because although multiple touches could be sensed by the hardware, they

were all stripped out by DTMouse so that only single touch events were ever sent to the application software. If the invention was obvious, why, Apple asked, was it not incorporated into DTMouse? The answer given by Dr Wigdor, which the judge accepted, was that DTMouse allowed the user of the DiamondTouch device to use Microsoft Office software by mouse emulation. The designers did not have access to the source code for Microsoft Office but wanted users to be able to use that software. The judge also considered that the forensic question had dangers in this case because it was based upon the implicit assumption that if the invention had been implemented before it would have been possible to find out about it. Most developers, however, treat their code as proprietary, just as Apple did with the iPhone 1.

75. In all the circumstances the judge concluded that the invention of claim 1 was obvious in the light of the common general knowledge. The skilled team, tasked with designing an OS for a multi-touch device, would arrive at the invention by the routine application of common general knowledge design principles. The attack was not, however directed at claim 2 which therefore survived.
76. Upon this appeal, Mr Burkill QC, who has again appeared on behalf of Apple with Mr Joe Delaney, submitted that the judge fell into error at every stage of his analysis. First, he argued that the judge ought to have considered a notional “base line” or starting point from which the skilled team would have been working at the date of the patent. Further, the judge should have held that this base line comprised multi-touch devices but with the qualification that they had been known in the research community for many years, and academic researchers had published details of systems running on such devices and how they had been implemented. Further, commercial devices, such as the iPhone 1, had also been marketed, but details of their internal implementation were neither public nor capable of being determined; and the only multi-touch implementation identified in evidence which supported legacy applications was the DTMouse system, which constituted a simple “all or nothing” arrangement.
77. I am unable to accept this criticism. At the outset, the judge directed himself, entirely correctly, that an attack over the common general knowledge must be treated with caution. Further, he reminded himself that it is necessary to focus not only on the aspects of common general knowledge relied upon to undermine the patent, but also all of the other aspects of the general knowledge which would have affected the thinking of the skilled team and which might tend to support the inventiveness of the patent.
78. As for those parts of the common general knowledge which formed the basis of the obviousness attack, the judge focused, entirely properly, upon the iPhone 1 which was, as he said, a very well known example of a multi-touch device at the priority date but the software for which was not in the public domain. The judge also had in mind that the skilled team would have been aware of the use of flags to filter events by type and so reduce the burden falling on software developers. I recognise that these flags were being used in single mouse or single touch environments but that does not mean to say they would have been considered irrelevant by the skilled team in deciding how to design the system software for a multi-touch device.

79. Further, I do not accept that the judge did not consider the existence of other multi-touch devices which had been developed and used for a number of years in the research community. Of these, the most significant was the DTMouse system which, as the judge recognised, implemented a legacy application by not allowing concurrent touches at all, and so effectively ran as a single touch device. The others used purpose-designed software but were plainly not as relevant as the iPhone 1, there being no evidence that they were developed with an eye to making the lives of application software developers any easier. Overall, therefore, the judge had well in mind the baseline upon which Mr Burkill relied.
80. Mr Burkill next criticised the judge for failing to identify the inventive concept of the claim. He argued that the judge wrongly concluded that the inventive concept was obvious in the light of the common general knowledge, without first having identified it.
81. I do not think there is anything in this submission. The judge explained at [37] that the patent proposes the use of flags associated with views on the screen and, in particular, a multi-touch flag which indicates whether a particular view is allowed to receive multiple simultaneous touches; and an exclusive touch flag, which indicates whether that view allows other views to receive touch events while it is receiving a touch. He then described the operation of the invention by reference to figures 4 and 5 and set out the inventive concept in the form of claim 1, broken down into six different features. He asked himself, entirely correctly, whether it was obvious to devise a method having all of these features.
82. Mr Burkill's third criticism had, in my judgment, more substance. He argued that the case of obviousness as developed by Dr Wigdor involved a series of steps and constituted a reconstruction with the benefit of hindsight. Thus, he submitted, in relation to multi-touch software, it involved:
- a) appreciating that not all components would necessarily need to act upon multiple touch inputs;
 - b) appreciating that the OS could assist in simplifying the handling of multiple touch inputs in cases where some components would not need to act upon them;
 - c) providing such assistance in the OS by means of flags used to control the passing of second or subsequent multiple touches to views;
 - d) appreciating that there were two separate situations which should be addressed: cases in which the same element received multiple simultaneous touches and cases in which different elements each received a simultaneous touch;
 - e) appreciating that each such situation could and should be addressed by a separate flag, which should address one situation but not the other.
83. Moreover, submitted Mr Burkill, Dr Wigdor's evidence was structured in a somewhat unusual two part way in that initially he was not shown the 948 patent and was asked to set out what he considered to be the common general knowledge

in the field. He was then shown the patent and the claims, and set out his views on obviousness. Mr Burkill continued that, at this point, Dr Wigdor even volunteered that he was asked to describe how *one could* arrive at the invention.

84. I recognise the force of these submissions and I have been particularly troubled by the way in which Dr Wigdor approached his consideration of what, if any, step it was obvious for the skilled team to take in the light of the common general knowledge. In particular, it seems to me to be undesirable that an expert is asked to describe how *one could* arrive at the invention. Nevertheless, after a careful review of all of the evidence I am satisfied that the judge was entitled to reach the conclusion that he did. His assessment was to a significant extent founded upon the evidence given by Dr Karp. The upshot was first, the iPhone 1 was the first multi-touch device aimed at the general consumer market and would therefore have been of great interest to the skilled team. It made extensive use of multi-touch functionality and that is something the skilled team would have wished to examine.
85. Second, the skilled team would have appreciated that applications running on the iPhone 1 contained the four different types of UI elements set out by the judge at [79] and referred to earlier in this judgment at [72]. This appreciation is of importance because it means the skilled team would have understood the need for a system which had the versatility called for by claim 1 and which lies at the heart of the invention.
86. Third, the judge was therefore entirely right to say as he did (at [78]) that the critical question was whether or not the skilled team would consider this functionality should be introduced at the system level or at the level of the application software.
87. Fourth, the skilled team would not have been focussing on legacy applications but rather on facilitating the writing of software which could take advantage of the new multi-touch capability. The skilled team therefore had a clear incentive to introduce the functionality at the system level and so make the platform more attractive to those developing application software and, ultimately, users. Despite the way in which Dr Wigdor was introduced to the issue, it seems to me that, having come this far, the judge was entitled to prefer Dr Wigdor's evidence to that of Dr Karp, particularly bearing in mind the acceptance by Dr Karp that flags were a known way of filtering types of event and that the alternative course of leaving it all to the application developers would have been to impose a significant burden upon them.
88. I therefore reject the submission that HTC's case depended upon the "step by step" course condemned by Lord Diplock in *Technograph Printed Circuits Ltd v Mills & Rockley (Electronics) Ltd* [1952] RPC 46. I am satisfied that the judge asked himself the correct question, namely whether it was obvious to carry out a method having all of the features of the claim.
89. Finally, Mr Burkill relied once again upon the question: why was it not done before? But this is another matter to which the judge properly had regard. He pointed out, entirely correctly, that most developers treat their code as proprietary, and that was certainly the case with Apple. As for those developing products for

use in more specialist applications, the judge focused fairly upon DTMouse which he addressed in the manner I have already described. This was not a matter which could begin to displace the conclusion to which the judge was minded to arrive upon the primary evidence of the experts.

90. For all of these reasons I would dismiss the appeal against the finding of obviousness of claim 1.

Conclusion in relation to the 948 patent

91. I would dismiss the appeal in relation to claim 1 on the basis that the judge was entitled to hold it was obvious. I would, however, allow the appeal in relation to claim 2. This was neither obvious nor invalid as relating to a computer program as such.

The 022 patent

92. The 022 patent is entitled “Unlocking a device by performing gestures on an unlock image” and has a priority date of 23 December 2005, rather earlier than that of the 948 patent. The parties were agreed that it is directed to a worker in the field of human computer interaction with a graduate degree in a subject in or concerned with the field of user interface design and about three years of industry experience.

93. At trial the following claims were in issue:

- i) Independent claims 1, 6 and 18. These are respectively a method claim, a device claim and claim to a computer program product. It was agreed their validity stood or fell together.
- ii) Dependent claims 5 and 17. These are respectively a method claim dependent on claim 1 and a device claim dependent on claim 6. Once again, it was agreed their validity stood or fell together.
- iii) Claim 9, another device claim dependent on claim 6.

94. The judge held that all these claims were invalid. Specifically, he held:

- i) Claims 1, 6 and 18 were anticipated by a piece of prior art called “Hyppönen”.
- ii) Claims 1, 6, 9 and 18 were invalid for obviousness in the light of a piece of prior art called “Plaisant”. Importantly, claims 5 and 17 were not obvious in the light of this citation.
- iii) All of the claims were obvious in the light of the Neonode.

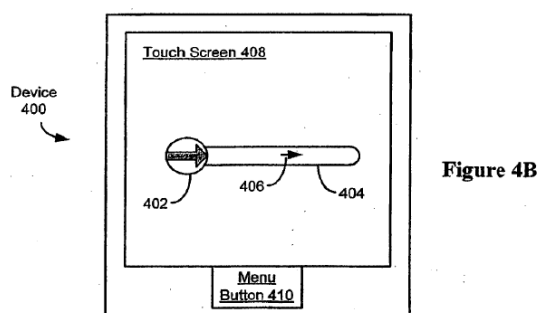
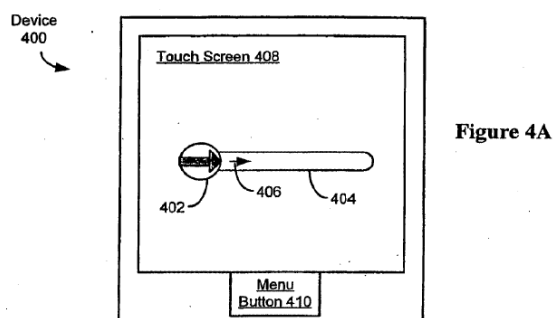
95. On this appeal Apple only challenges the finding that claims 5 and 17 were obvious in the light of the Neonode. It emphasises, entirely fairly, that the focus of the trial was on the infringement and validity of claim 1 and corresponding claims

6 and 18 and that in consequence only a relatively small part of the judgment is concerned with claims 5 and 17. It also says, and I accept, that HTC's case that claims 5 and 17 were obvious over the Neonode was not advanced in the written evidence of its expert, Professor Greenberg. His written evidence only suggested claim 5 was obvious in the light of Plaisant. The case of obviousness of claim 5 in the light of the Neonode was advanced for the first time in the cross examination of HTC's expert, Professor Keyson. This is a matter to which I must return later in this judgment.

The specification

96. The specification begins by explaining that the invention relates to the unlocking of user-interfaces in portable electronic devices. It continues with a description of the background art and says that a problem associated with touch screens on portable devices is the activation or deactivation of functions due to unintentional contact with the screen. Thus, it continues, portable devices, and their touch screens and applications running on them, may be locked in various circumstances, such as upon entering an active call, after a pre-determined time of idleness or upon manual locking by a user.
97. The specification then acknowledges several well known procedures for unlocking touch screen devices, such as pressing a defined set of buttons or entering a code or password. But these procedures are said to suffer from the drawbacks that they may be hard to perform or, in the case of passwords and codes, difficult to memorise.
98. The specification explains at [0005] that there is therefore a need for more efficient user-friendly procedures for unlocking touch screen devices and their screens and applications. It continues:

“More generally, there is a need for more efficient, user-friendly procedures for transitioning such devices, touch screens, and/or applications between user interface states (e.g., from a user interface state for a first application to a user interface state in the same application, or between locked and unlocked states). In addition, there is a need for sensory feedback to the user regarding progress towards satisfaction of a user input condition that is required for the transition to occur.”
99. The solution provided by the patent involves, in its broadest form, a method of unlocking a screen which involves contacting the screen with a finger or an object such as a stylus and moving an unlock image along a pre-defined displayed path across the screen in accordance with a pre-defined gesture. Thus the unlock image and pre-defined display path along which the unlock image must be moved are both displayed to the user and the user is provided with feedback as to the progress of the unlock action.
100. An embodiment of the invention is illustrated in figures 4A and 4B:



101. In figure 4A, the device 400 includes a touch screen 408 and a menu button 410. The device 400 is locked and the touch screen displays an unlock image 402 and a channel 404 indicating the path of movement along which the unlock image 402 must be dragged; and one or more arrows 406 indicating the direction of movement. The end of the channel 404 serves as a pre-defined location to which the unlock image 402 must be dragged. The unlock image 402 may also include an arrow to remind the user of the direction of movement. As shown in figure 4B, the arrow 406 may move along the channel 404 and disappear when it reaches the end of the channel.
102. Claim 1 (with reference numerals added by the judge) reads as follows:
- “(i) A computer implemented method of controlling a portable electronic device
 - (ii) comprising a touch-sensitive display,
 - (iii) comprising detecting contact with the touch-sensitive display while the device is in a user interface lock state;
 - (iv) transitioning the device to a user-interface unlock state if the detected contact corresponds to a predefined gesture;
 - (v) and maintaining the device in a user-interface lock state if the detected contact does not correspond to the predefined gesture;

- (vi) characterised by moving an unlock image along a predefined displayed path on the touch sensitive display in accordance with the contact,
- (vii) wherein the unlock image is a graphical, interactive user-interface object with which a user interacts in order to unlock the device.”

103. Claim 5 is directed to the method of claim 1 with the additional feature that the device displays two unlock images, each corresponding, for example, to a different application. Performing the unlock action using one of the images unlocks the device and displays the application corresponding to that image. It reads as follows (again, with reference numerals added by the judge):

- “(i) The computer-implemented method of claim 1, further comprising:
- (ii) displaying a first unlock image and a second unlock image on the touch-sensitive display while the device is in a user-interface lock state; and
- (iii) wherein transitioning the device to a user interface unlock state comprises: transitioning the device to a first active state corresponding to the first unlock image if the detected contact corresponds to a predefined gesture with respect to the first unlock image; and
- (iv) transitioning the device to a second active state distinct from the first active state if the detected contact corresponds to a predefined gesture with respect to the second unlock image.”

104. For reasons which will become apparent, Apple emphasises that the claimed method starts from a point at which the device is in a “user-interface lock state” and requires the device to be transitioned to a “user-interface unlock state” state corresponding, in the case of claim 5, to the particular unlock image that the user chooses to manipulate.

105. The meaning of the phrase “user-interface lock state” is explained at [0045] as being a state in which the device is operational but ignores most, if not all, user inputs:

“In the user-interface lock state (hereinafter the “lock state”), the device 100 is powered on and operational but ignores most, if not all, user input. That is, the device 100 takes no action in response to user input and/or the device 100 is prevented from performing a predefined set of operations in response to the user input. The predefined set of operations may include navigation between user interfaces and activation or deactivation of a predefined set of functions. The lock state

may be used to prevent unintentional or unauthorized use of the device 100 or activation or deactivation of functions on the device 100. When the device 100 is in the lock state, the device 100 may be said to be locked.”

106. By contrast, in the “user-interface unlock state” the device detects and responds to user inputs corresponding to the user-interface, as the specification explains at [0047]:

“In the user-interface unlock state (hereinafter the “unlock state”), the device 100 is in its normal operating state, detecting and responding to user input corresponding to interaction with the user-interface. A device 100 that is in the unlock state may be described as an unlocked device 100. An unlocked device 100 detects and responds to user input for navigating between user interfaces, entry of data and activation or deactivation of functions. In embodiments where the device 100 includes the touch screen 126, the unlocked device 100 detects and responds to contact corresponding to navigation between user interfaces, entry of data and activation or deactivation of functions through the touch screen 126.”

107. It is important to note, however, that the lock state does not necessarily apply to the device as a whole and may restrict access to an application while the device is running, as the specification makes clear at [0075]:

“In some embodiments, the lock/unlock feature may apply to specific applications that are executing on the device 400 as opposed to the device 400 as a whole. In some embodiments, an unlock gesture transitions from one application to another, for example, from a telephone application to a music player or vice versa. The lock/unlock feature may include a hold or pause feature.”

108. Further, when running, the device may display the multiple unlock images the subject of claim 5, as the specification elaborates at [0095]:

“In some embodiments, the device may have one or more active applications running when the device becomes locked. Additionally, while locked, the device may continue to receive events, such as incoming calls, messages, voicemail notifications, and so forth. The device may display multiple unlock images on the touch screen, each unlock image corresponding to an active application or incoming event. Performing the unlock action using one of the multiple unlock images unlocks the device and displays the application and/or event corresponding to the unlock image. The user interface active state, as used herein, means that the device is unlocked and a corresponding application or event is displayed on the touch screen to the user. While the process flow 900 described below includes a number of operations that appear to occur in a

specific order, it should be apparent that these processes can include more or fewer operations, which can be executed serially or in parallel (e.g., using parallel processors or a multi-threading environment).”

109. It follows the device is in a lock state if it ignores certain inputs or is prevented from performing a pre-defined set of operations in response to those inputs. By contrast, it will be in an unlock state if it responds to user inputs corresponding to interactions with the user interface.
110. This does need a little elaboration, however. First, it is important to note that, as all parties accepted, a device may be in a lock state within the meaning of the claims even if the only imagery appearing on the screen is the unlock image and the path along which it must move. The teaching in the specification that the lock state requires the device to ignore certain inputs or prevents it from performing certain operations must be seen in that light.
111. Second, the method of the claim is not limited to unlocking a device after it has been switched on, or after a period of non-use. Nor is it limited to unlocking the device as a whole. As emerges clearly from [0075] and [0095], the device may have applications running when it becomes locked. Moreover, the lock/unlock feature may apply only to specific applications.
112. This aspect of the teaching poses a conceptual difficulty, as may be illustrated by the following example. Suppose a device as a whole is locked after a period of non-use and a method falling clearly within claim 1 is required to unlock it. Once the method has been performed the device is now in a user-interface unlock state. Now further suppose that, on navigating one of the menus, the user is presented with a lock feature which he must open in order to use that application. The specification plainly contemplates that the claimed method may be used in relation to such a lock feature so as to transition the device into a state in which the application may be used. But, it may be said, the device is already in a user-interface unlock state as a result of the first operation. How then can the method be used, requiring as it does a transition from a lock state to an unlock state?
113. The answer must be that, read purposively, the specification contemplates that the device may be in an unlock state with respect to one operation but in a lock state with respect to another. As will be seen, this is of particular importance in the context of the present case.
114. Third, I sought to explore in the course of the appeal hearing the distinction between a device or application which is in a lock state and one which has simply not been activated. In either case the device will ignore certain inputs. How may these two conditions be distinguished? I believe the answer to this question is also provided by a purposive interpretation of the specification as a whole. The problem to which the specification is directed is the unintentional activation (or deactivation) of a device as a result of unintentional contact with its touch screen. The solution is to lock the device (or the application) so that it will only respond to a touch in the form of a predefined gesture. So I believe it is the lack of response to any touch save the particular predefined gesture which effectively locks the device. The claimed method then involves assisting the user to perform

the necessary gesture by providing the unlock image and predefined displayed path along which it must be moved.

Neonode

115. The Neonode is a mobile telephone which was launched in Sweden in July 2004. It was supplied in various versions, two of which were available before the priority date.
116. Both versions incorporated a touch screen which was locked when the telephone was not in use. The user had first to press a physical key to the left hand side of the front of the hand set. Following this, the display lit up, revealing what was referred to as a lockscreen. This had an image of a padlock and, in the first version, the words “Right sweep to unlock” displayed below it. In the second version these words were replaced by an arrow. To unlock the device, the user had to swipe his finger across the touch screen from left to right, following the written instruction (in the case of the first version), and in the direction of the arrow (in the case of the second version).
117. Neither Professor Greenberg nor Professor Keyson proceeded to describe the operation of the Neonode beyond the initial lockscreen in their written reports. Fortunately, however, three exhibits to Professor Greenberg’s report, a Neonode start guide and two videos, reveal it worked in the following way.
118. First, after the user had made the appropriate sweeping gesture on the lockscreen, the device displayed a second screen called the “status screen”. This screen displayed various features showing the battery life, the reception status of the telephone and the network it was connected to, the time and the date, and an abstract logo in the background. There were also three icons at the bottom of the screen which corresponded to the three basic menus of the Neonode, the “start menu”, the “keyboard menu” and the “tools menu”. In order to activate each of these menus, the user was required to make another sweep gesture, but this time upwards from the icon to a point about half way up the screen. In making the upwards sweep gesture, the user’s finger had to pass over a number of the features shown on the screen. The relevant menu then appeared and responded to a tap on the desired symbol or icon.

The case at trial

119. As I have mentioned, the only case of obviousness of the patent in the light of the Neonode developed by Professor Greenberg in his report was directed at claim 1 of the patent and was based upon the initial lockscreen. The arrow was, he thought, a graphical user-interface object, but he accepted that the user did not interact with it and it did not move in accordance with the user’s contact. Nevertheless, he thought it would have been routine, and part of the standard design process, to provide some form of object on the screen with which the user could interact, in conjunction with feedback, so that the user knew what progress he was making. He considered the skilled person would have viewed it as a straightforward improvement to place a feature on the screen, and it would have been obvious to design the feature in such a way that the user would have known what he had to do to unlock the screen and upon which specific part of the screen

he had to make the required input. One obvious addition would have been to add some sort of slider, and for the user to drag an object or handle along the slider.

120. The only case of obviousness developed by Professor Greenberg in his report against claim 5 was based upon Plaisant. At trial, however, a further case was put to Professor Keyson in the course of cross-examination that claim 5 was not inventive in the light of the Neonode because it would have been obvious to add multiple unlock images above each of the symbols for the three basic menus displayed on the status screen once the user had moved beyond the lockscreen.

121. A key part of his evidence ran as follows:

“Q. This suffers from the same lack of feedback as the unlock screen, does it not?

A. If I use my understanding of feedback as I now understand it based upon the knowledge we have today in terms of gestural interfaces, if we look at the literature now we have available to us as designers, then I would say yes, this lacks feedback for the gestures.

Q. Right. It has the same problem as the unlock screen. The unlock screen had a lateral movement for which no feedback was provided and this has got three upward movements for which no feedback is provided. It is the same problem.

A. I think it is a slightly different problem because, as a designer, you have a bit more room for the slide to unlock, so to speak, because it is the first screen, there is nothing there but the unlock screen, so you can make a nice beautiful elegant slide to unlock image or whatever to unlock it. Once you are in the application itself it is a bit more complex because you already have icons on the screen so you do not have as much freedom to put all kinds of animations, moving with your finger, because it would then confuse the screen. It is already about the small display to begin with, the Neonode, as you know. It is not the same design challenge, you have a different problem here.”

122. Then, a little later, the following exchange took place:

“Q. I understand your point, you may have to think about just exactly how you lay out the screen, but these present the same issue as the unlock screen basically, which is that there is no feedback.

A. Yes, you would want to have feedback for the – I am not sure exactly how you would do that honestly. I cannot come up with a design solution off the top of my – because it is a different design problem here than the initial unlock screen.

So you can see how difficult it is to design these things. Even now, as you are asking me on the spot, I cannot come up with a solution here. But you would somehow want to cue the user to provide some kind of visual cues to indicate the path of movement which they could follow. It may not require – it could be a rather simple cue and you would want to somehow have an interactive cue, ideally, some kind of animation or feedback, which would indicate the direction of movement, so animating – an animated arrow pointing out changing in brightness or whatever kind of cue you would want to show to show the direction of movement you should take with that object. So it could be a rather simple animated display of something that would indicate the movement you should follow and the direction you should follow so you would understand how to use that gestural feedback. I would not suggest in such a display, for example, using the channel as a visual cue because it would clog up the whole display. That is what I was trying to point out. You would probably want to use a different kind of visual cue to show the natural path of movement you would want to take.”

123. Professor Greenberg was cross-examined after Professor Keyson and he took the opportunity to raise the same issue. When asked about the Neonode and whether it would have been advantageous to incorporate a second unlock image, he mentioned the status screen and that three different swipe gestures each led to a different type of application and added that there would have been a motivation to add an unlock image above each of the basic menu symbols. The relevant part of his evidence reads:

“Q. Now, we have just looked at paragraph 97, when you said it was highly desirable to have an unlock image. Even with your knowledge of the iPhone in 2011, did it ever cross your mind that it would be advantageous to incorporate a second unlock image?

A. Are you talking about in relationship to the Neonode?

Q. Yes.

A. So, if I may, I would like to bring your attention to the fact that the Neonode actually has, as one of its user interface states, a series of three chevrons going across from left to right of the screen, which acts in that purpose, although there is no path indicator problem with the Neonode, but there are three different ways. Sorry, it is vertical. There are three different swipe gestures that you can actually do, each leading to a different type of application. So from looking at that screen, there it would be motivation to add an image on top of that as well, which would track from that particular state to a particular other state.

Q. That would make the screen extremely complex?

A. It depends how you design. It is a matter of graphical design. I was just going to say that I actually experimented with that screen and it is very resistant to accidental activation. I just moved my finger all around it and unless you get exactly the right upward gesture on the background it does not activate to other applications.

Q. So they have built in a security feature?

A. I am just saying it is resistant to accidental activation.”

The judgment

124. The judge began with a description of the Neonode disclosure, focusing on the lockscreen. He then turned to consider the allegation of obviousness, adopting the structured approach described by this court in *Pozzoli*. He noted, entirely correctly, that HTC relied principally upon the arrow unlock feature of the Neonode and observed that the only difference between the Neonode disclosure of the arrow unlock feature and the inventive concept of claim 1 was the absence of an unlock image with which the user interacted and which was moved along the pre-defined display path. The sole question was, therefore, whether it was obvious to add these features to the Neonode, without knowledge of the invention.

125. The judge was satisfied on the evidence that he heard that the skilled person would have appreciated that the arrow unlock feature suffered from a lack of feedback. He then proceeded to consider whether it would have been obvious to improve the user-interface by providing an unlock image in the form of a cursor which the user could drag along the unlock arrow. He recited the relevant evidence of Professor Greenberg and Professor Keyson and then expressed his conclusion in these terms:

“229. I consider that it would be obvious to the skilled team, faced with the lateral-swipe arrow unlock of Neonode, that it could be improved by the provision of feedback. The skilled team would be aware that visual feedback for a lateral gesture could be provided by the extremely familiar sliders from his common general knowledge, such as the Windows CE slider.

230. It is true that this simple improvement was not done by Neonode. This is a secondary consideration which may in some circumstances support a case of inventiveness. On its own, which it would be in this case, it is of little weight.”

126. The judge then turned to consider claim 5 and his whole analysis is contained in one paragraph:

“234. Claim 5 adds the feature that the device can be moved to a variety of different unlocked states from the locked state. Beyond the initial unlock screen, Neonode disclosed the use of three swipe gestures in order to unlock different applications: the start menu, the keyboard menu and the tools menu. Once it is accepted that claim 1 is obvious in the light of Neonode, it seems to me that claim 5 is obvious as well. The skilled person would readily see the applicability of the swipe-with-feedback to unlocking a plurality of applications.”

The appeal

127. Mr Burkill submitted the judge made two distinct errors of principle in reaching his conclusion that claim 5 was obvious in the light of the Neonode. His first and fundamental error was to find (as he must implicitly have done) that once the Neonode was in a state beyond the initial unlock screen it was nonetheless in a user-interface lock state within the meaning of the claims.
128. Mr Burkill developed this submission as follows. He argued that a device in a user-interface lock state must be in a state in which certain inputs are ignored, as opposed to its normal operating state in which user inputs are acted on. However, once the Neonode displayed the status screen, it was in its normal operating state, and that the upwards gestures needed to open the basic menus were basic moves used in every application, as the manual explained.
129. Therefore, Mr Burkill continued, irrespective of what (if anything) the skilled person might have thought it was obvious to add to the Neonode status screen, it would never have resulted in a method or device within claims 5 or 17 because the transition between the status screen and the menus was not a transition from a user-interface lock state to a user-interface unlock state.
130. This is a powerful submission but I believe that the answer to it lies in the aspects of the interpretation of claim 1 to which I have referred. As I have explained, the specification contemplates a device which is in a lock state with respect to one operation but an unlock state with respect to another. More specifically, it contemplates a device being transitioned from an initial overall lock state to an unlock state, and also a device in which an unlock gesture is required to allow the user to navigate between functions or move from one application to another.
131. Turning to the Neonode, there can be no doubt that transitioning the device from the initial lockscreen to the status screen involved unlocking the device, and doing so by detecting a contact which corresponded to a predefined gesture. So the only difference between this method and that of claim 1 is the absence of features (vi) and (vii), which the judge found to be obvious. The device as a whole was now unlocked and the user had the opportunity access each of the three basic applications or menus. But, to my mind, each of these applications nevertheless remained locked at this stage because none could be accessed unless the device detected another predefined gesture, namely, for each application, a sweep up from the relevant icon to a point about half way up the screen. As a practical

matter this prevented unintentional activation of the function, as emerged from Professor Greenberg's cross examination set out at [123] above.

132. In my judgment it follows that when the Neonode device displayed the status screen it was in a user-interface lock state with respect to each of the three basic applications. It required a predefined gesture to open each application and so transition the device to a user-interface unlock state with respect to that application. I would therefore reject Mr Burkill's first submission.
133. Mr Burkill submitted that the second error of principle the judge made was to ignore the limited evidence there was from Professor Keyson and Professor Greenberg on this issue. The key question here is whether it was obvious to provide unlock images on the status screen above the three basic menu symbols, and for those images to have the functionality of features (vi) and (vii) of claim 1. In other words, whether it was obvious to provide them in such a way that the user could interact with them, in conjunction with feedback, so that he knew what progress he was making. Mr Burkill argued that what evidence there was suggested that the addition of such unlock images would not have been practical (let alone obvious) due to the presence of the other features on the screen.
134. This issue was never properly developed in the expert report of Professor Greenberg, as it plainly should have been. The purpose of exchanging expert reports in advance of the trial is to ensure that each side is well aware of the opinions of those experts and so also the case they have to meet. This enables the court to deal with the case expeditiously and fairly. Nevertheless, it was not suggested to the judge that he should decline to deal with the point and so he addressed it in the light of the evidence that he had.
135. I have set out the key parts of that evidence above. It is not extensive. But I believe Professor Keyson accepted the skilled person would have appreciated the desirability of having what the judge described as "swipe-with-feedback" to unlock each of the three applications. He went on to say that provision of a substantial cue such as a channel would have been difficult because of the other features displayed on the status screen. As I understand his evidence, he thought a simple cue would, however, have been possible. In any event, it is, in my judgment, clear that he thought the concept of claim 5 was obvious in the light of the Neonode despite the practical design issue of how to implement it in this particular device.
136. Professor Greenberg's evidence was to much the same effect. He thought that it would have been obvious to add an image which would track the swipe gestures required to unlock each of the basic menus shown on the status screen. When it was suggested to him that this would make the screen extremely complex, he said that would be a matter of graphical design.
137. I therefore believe the judge did have an evidential basis for making his findings in relation to claim 5. He was entitled to find it was obvious to apply what he called swipe-with-feedback to the unlocking of a plurality of applications.
138. It follows that I would dismiss the appeal in relation to claim 5.

Overall conclusion

139. For all the reasons I have given I would:

- i) dismiss the appeal in relation to claim 1 of the 948 patent;
- ii) allow the appeal in relation to claim 2 of the 948 patent;
- iii) dismiss the appeal in relation to claims 5 and 17 of the 022 patent.

Lord Justice Lewison:

140. This appeal requires us, once again, to venture into the minefield of the exclusion from patentability of computer programs “as such”. I want to say something about that aspect of the case.

141. Article 52 of the EPC provides:

“(1) European patents shall be granted for any inventions, in all fields of technology, provided that they are new, involve an inventive step and are susceptible of industrial application.

(2) The following in particular shall not be regarded as inventions within the meaning of paragraph 1:

...

(c)... programs for computers;

(3) Paragraph 2 shall exclude the patentability of the subject-matter or activities referred to therein only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such.”

142. Article 52 is transposed into domestic legislation by section 1 (2) of the Patents Act 1977 which provides:

“(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 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.”

143. It is, to me at least, regrettable that because these apparently simple words have no clear meaning both our courts and the Technical Boards of Appeal at the EPO have

stopped even trying to understand them. However we are so far down that road that “returning were as tedious as go o’er”. Instead we are now engaged on a search for a “technical contribution” or a “technical effect”. Instead of arguing about what the legislation means, we argue about what the gloss means. We do not even know whether these substitute phrases mean the same thing (see Terrell on Patents 17th ed para 13-15). In *Aerotel Ltd v Telco Holdings Ltd* [2006] EWCA Civ 1371 [2007] RPC 7 the preferred test for patentability was what Jacob LJ described as “the technical effect approach with the rider”. But when he described this test earlier in his judgment it appeared thus:

“(2) *The technical effect approach*

Ask whether the invention as defined in the claim makes a technical contribution to the known art—if no, Art.52(2) applies. A possible clarification (at least by way of exclusion) of this approach is to add the rider that novel or inventive purely excluded matter does not count as a “technical contribution”.

144. “Technical effect” (in the heading) and “technical contribution” (in the text) appear to be synonymous. In Case T 0154/04 *Duns Licensing Associates* the Board referred to yet another phrase: viz. that an invention must have “technical character”; and went on to say:

“For examining patentability of an invention in respect of a claim, the claim must be construed to determine the technical features of the invention, i.e. the features which contribute to the technical character of the invention.”

145. In *Symbian Ltd v Comptroller General* [2008] EWCA Vic 1066 [2009] RPC 1 this court in attempting to reconcile *Aerotel* and *Duns Licensing* described the test at [11] as:

“whether the contribution cannot be characterised as “technical”.”

146. At [15] the court went on to say:

“It plainly requires one to identify “the contribution” (which equates to stage 2 in *Aerotel*) in order to decide whether that contribution is solely “the excluded subject-matter itself (equating to stage 3 in *Aerotel*), while emphasising that the contribution must be “technical” (effectively stage 4 in *Aerotel*). The order in which the stages are dealt with is different, but that should affect neither the applicable principles nor the outcome in any particular case.”

147. So the upshot is that we now ignore the words “computer program ... as such” and instead concentrate on whether there is a technical contribution. It is, if I may say so, a singularly unhelpful test because the interaction between hardware and software in a computer is inherently “technical” in the ordinary sense of the word. If I buy a

software package that malfunctions the software house will often offer me “technical support”. But that is clearly not enough for the software to qualify as making a “technical contribution”. In *Symbian* this court declined to provide a definition of the right kind of technical effect; but instead provided a recommended reading list.

148. I studied the reading list in *AT & T Knowledge Ventures v Comptroller General* [2009] EWHC 343 (Pat) [2009] FSR 19 and tried to distil the essence of what they revealed. I said:

“... it seems to me that useful signposts to a relevant technical effect are:

i) whether the claimed technical effect has a technical effect on a process which is carried on outside the computer;

ii) whether the claimed technical effect operates at the level of the architecture of the computer; that is to say whether the effect is produced irrespective of the data being processed or the applications being run;

iii) whether the claimed technical effect results in the computer being made to operate in a new way;

iv) whether there is an increase in the speed or reliability of the computer;

v) whether the perceived problem is overcome by the claimed invention as opposed to merely being circumvented.”

149. These signposts have been found to be helpful in later cases (e.g. *Gemstar-TV Guide International Inc v Virgin Media Ltd* [2009] EWHC 3068 (Ch) [2010] RPC 10 at [34]; *Protecting Kids the World Over (PKTWO) Ltd's Patent Application* [2011] EWHC 2720 (Pat) [2012] RPC 13). Neither party to the appeal suggested otherwise. I should perhaps emphasise that these signposts were not intended to be prescriptive conditions; nor did I intend to suggest that if only one of the signposts was found to exist an invention would automatically be patentable.

150. On reflection the fourth of these signposts may have been expressed too restrictively. In *Gemstar* Mann J said at [42]:

“It would be a relevant technical effect if the program made the computer a better computer in the sense of running more efficiently and effectively as a computer.”

151. I think that this is a better signpost than an improvement confined to the speed or reliability of the computer. As HHJ Birss QC pointed out in *Halliburton Energy Services Inc's Patent Application* [2011] EWHC 2508 (Pat) [2012] RPC 12 at [37]:

“The “better computer” cases—of which *Symbian* is paradigm example—have always been tricky however one approaches this area. The task the program is performing is defined in such a way that everything is going on inside the computer. The task

being carried out does not represent something specific and external to the computer and so in a sense there is nothing else going on than the running of a computer program. But when the program solves a technical problem relating to the running of computers generally, one can see that there is scope for a patent. Making computers work better is not excluded by s 1(2).”

152. Since each case must be determined by reference to its particular facts and features (*Symbian* at [52]) we need to begin by asking: what does the computer program in this case actually contribute? In a nutshell the answer is that it makes it easier to write programs for applications to be run on the device that contains it. In Mr Burkill’s phrase, it makes the device “more programmable”. Mr Burkill submitted, and I agree, that *writing* software is not a “computer program as such”, although the product may be. I do not therefore agree with the judge that the contribution lies solely in excluded matter.
153. Next, I think it is helpful to consider the facts of some of the cases in which a computer program was held to be patentable. In *IBM CORP/Data processor network* (T6/83) [1990] OJ EPO 5; [1990] EPOR 91 the invention consisted of an improved method of communication between programs and files held at different processors within a known network. It was held to be patentable. In *Symbian* itself the patentable computer program was a new means of accessing dynamic link libraries which had potential application in a variety of devices such as cameras and mobile phones. If those inventions were patentable, why is the invention in the present case not?
154. Let me turn to consider whether the signposts can be found on the facts of this case:
 - i) The claimed invention has a technical effect on a process carried out outside the computer. At one end, it makes the writing of software easier. I do not agree with the judge that this is simply a redistribution of labour. It is a reduction of labour, because once the device is programmed with the claimed invention software writers can use it over and over again. To borrow Mr Burkill’s analogy, a multi-touch device which sends all touches to the applications is like a water system with an uncontrolled flow of mains water. The invention consists of the provision of taps. The application designer does not have to design and fit his own taps in order to use the device. All he has to do is to decide whether he wants the taps turned on or off. At the other end, it interacts with the end user who touches the screen. This, too, is a process happening outside the computer.
 - ii) The claimed invention operates at the level of the operating system of the device. It works with any application that it programmed to run on it irrespective of the data processed by the application. It will work just as well with a game as with a currency converter.
 - iii) I think it doubtful whether the device is made to operate in a new way.
 - iv) The invention makes the device a better device in the sense of running more efficiently and effectively as a device. It runs more effectively because it is easier to program.

v) The problem identified by the patent is difficulty of programming; and the claimed invention overcomes that perceived problem.

155. In my judgment, in respectful disagreement with the judge, there are sufficient signposts to justify the patentability of the invention. Claim 1 of the 948 patent succumbed to an obviousness attack; but claim 2 survived. I would therefore hold that claim 2 is valid. On all other aspects of the appeal there is nothing I wish to add to the comprehensive judgment of Kitchin LJ.

Lord Justice Richards:

156. I agree that the appeals should be disposed of in the manner proposed by Kitchin LJ and for the reasons he gives.