

IN THE HIGH COURT OF JUSTICE
CHANCERY DIVISION
PATENTS COURT

Rolls Building
7 Rolls Buildings
Fetter Lane
London EC4A 1NL

Date: 10/07/2013

Before:

MR JUSTICE BIRSS

Between:

HTC CORPORATION	<u>Claimant</u>
- and -	
GEMALTO S.A.	<u>Defendant</u>

and between:

HTC CORPORATION	<u>Claimant</u>
- and -	
GEMALTO N.V.	<u>Defendant</u>

Michael Tappin QC and Ben Longstaff (instructed by **Powell Gilbert LLP**) for the **Claimant**
James Mellor QC, Guy Burkill QC and Miles Copeland (instructed by **S. J. Berwin LLP**)
for the **Defendant**

Hearing dates: 24th, 25th, 26th, 29th, 30th April, 1st, 2nd, 3rd, 8th, 9th, 10th May

Judgment

Mr Justice Birss:

Topic	Paragraphs
Introduction	1
865: The issues	4
865: Background	7
865: Witnesses	25
865: Person skilled in the art	34
865: Common general knowledge of the skilled addressee	37
865: The patent	51
865: Claim construction	57
(i) microcontroller	60
(ii) having a set of resource constraints	104
(iii) loaded in memory	106
(iv) a converter	110
(v) claim 8	112
865: Priority	126
Priority – the law	127
Priority entitlement	130
Substantive priority	150
Priority - conclusions	195
865: Obviousness at the filing date	196
Skilled person and common general knowledge	206
Claim 1	207
Claim 8	214
Claims 15 and 18	221
865: Novelty at the priority date	223
The Caron article	227
Pasman	242
865: Obviousness at the priority date	246
Common general knowledge alone	266
The Caron article	283
Pasman	288
171 Application	289
865: Patentable subject matter	294
865: Infringement	303
Section 60(2) of the 1977 Act	334
<i>The 9062 patent</i>	344
9062: Background	347
9062: The witnesses	353
9062: The person skilled in the art	360
9062: Common general knowledge	361
9062: The patent	367
9062: Claim construction	376
9062: Novelty	390
9062: Obviousness	421
9062: Infringement	444
<i>Conclusion</i>	447

Introduction

1. This is a patent action concerning two patents, EP (UK) 0 932 865 (“Using a High Level Programming Language with a Microcontroller”) and EP (UK) 0 829 062 (“Smart Card Reader”). The patents belong to Gemalto (in fact the 865 patent belongs to Gemalto S.A. and the 9062 patent belongs to Gemalto N.V. but nothing turns on the distinction.) The proceedings began as actions for revocation of both patents by HTC. By counterclaim, Gemalto contends that the patents are infringed by certain HTC smart phones. The cases concerning the two patents are distinct but it was convenient to hear them together because they both involve the same general area of technology. The technology relates to what are sometimes called smart cards or chip cards.
2. For HTC the legal representation was the same throughout: Michael Tappin QC, leading Ben Longstaff instructed by Powell Gilbert. For Gemalto the legal team working on the ‘865 patent was James Mellor QC leading Miles Copeland instructed by S.J. Berwin whereas for the ‘9062 patent Gemalto were represented by Guy Burkill QC leading Miles Copeland instructed by S.J. Berwin.
3. This judgment concerns the issues of validity and infringement of both patents. HTC also relies on Gemalto’s declaration to the European Telecommunications Standards Institute (ETSI) in relation to the grant of a so-called FRAND licence (a licence on Fair Reasonable And Non-Discriminatory terms) under the 865 patent and the 9062 patent as limiting any relief to a monetary sum equivalent to a FRAND royalty. That question was stayed by paragraph 5 of the order of Arnold J dated 31st July 2012 pending judgment on the validity and infringement issues.

865 – The issues

4. The application for the 865 patent was filed on 22nd October 1997 claiming priority from an application filed on 25th October 1996. The claim to priority is disputed and is addressed below. The 865 patent was granted on 14th August 2002.
5. Gemalto contends that claims 1, 3, 8, 9, 15 and 18 of the 865 patent are independently valid. All these claims save for claim 9 are alleged to be infringed by HTC’s sales of the devices in issue. Part of the issue of infringement involves s60(2) of the 1977 Act. All the claims include the term “microcontroller” and a key argument is whether the HTC devices have microcontrollers at all.
6. In relation to validity, HTC contends that claims 1, 3, 8, 9, 15 and 18 are not entitled to the claimed priority date (October 1996) but only to the filing date (October 1997). Any claim which is only entitled to the filing date is alleged to be obvious over an intervening publication called Cyberflex. Any claim which is entitled to the priority date is said to lack novelty over two items of prior art, a paper by Caron and a paper by Pasman. Moreover any claim entitled to the priority date is said to be obvious. Four starting points for obviousness are relied on: common general knowledge alone, Caron, Pasman and a patent application called ‘171. HTC also contends that all claims are unpatentable subject matter contrary to s1 (2) of the 1977 Act (Art 52 EPC) since they are to computer programs as such. HTC had an insufficiency plea but it was not pursued.

865: Background

7. This action is concerned with programming computers. Computers have a number of elements: the central processing unit (CPU) which performs the computations, the memory and the input/output hardware to communicate with the outside world. Modern computers are implemented on silicon chips. A microprocessor chip is the chip which carries the CPU. In some systems the microprocessor chip may be connected by a system bus to other components such as memory chips and input/output devices to make a working computer. Other chips may have some memory and I/O devices on the same piece of silicon as the CPU. Inevitably, since space is limited, the amount of memory which can be fitted on the same chip as the CPU is limited. As technology has advanced, it has been possible to cram more and more circuitry onto a single piece of silicon. Thus the CPUs have become much more powerful over time and the amount of memory which can be made available has increased.
8. There are various different kinds of computer memory. Random Access Memory or RAM is a memory used by a computer to store intermediate results which can be accessed relatively quickly. The memory is volatile in that the contents are lost when power is removed. Read Only Memory or ROM is a type of memory which generally cannot be modified and is used to store programs or data what are not intended to change after manufacture. ROM is non-volatile, i.e. it retains data even when the power is off. Variants on ROM are PROM, EPROM and EEPROM. PROM is Programmable ROM, a form of non-volatile memory in which the contents are set by a user but once set cannot be changed. EPROM is Erasable Programmable ROM, one example of which was a form of PROM in which the data could all be erased in one go using ultraviolet light. EEPROM is Electrically Erasable PROM, a non-volatile memory which can be erased and reprogrammed repeatedly, albeit much more slowly than RAM. Although the letters in the term ROM originally meant “read only”, the term ROM as it is used in EEPROM is really referring to the non-volatile nature of the memory. FLASH memory is a kind of EEPROM.
9. Many processors also have registers. These are a limited number of volatile memory slots within the CPU itself offering very fast temporary storage for values being processed. They are really part of the CPU itself.
10. Another kind of memory arising in this case is cache memory. Cache is used in computer architecture to reduce the access time for reading and/or writing data. The principle works by placing into the cache memory a temporary copy of some of the data from the system’s main memory, that is the data which is currently being processed. The data can be the instructions to be performed or data to be manipulated. The data in cache is never unique because it is only ever a copy of data already in other memory in the system. Cache memories are faster and generally more costly than other kinds. They are located physically close to the CPU to speed up access times.
11. A distinction which arises in this case is between general purpose computers, such as a desktop or laptop computer, and embedded systems, such as the dedicated computer which controls a washing machine. Both contain computer chips. Often in an embedded system there will be a single silicon chip which carries the CPU and all the memory and other functional elements necessary. By contrast a desktop or laptop

computer will usually have a processor chip and separate memory chips and other things.

12. The claims refer to a “microcontroller”. There is no doubt a microcontroller is a kind of computer but beyond that the parties did not agree what the word meant and I will address that in the section on construction.
13. Computers are programmable. Each computer has an instruction set, a predefined set of instructions which the processor understands and can execute. Once given a program containing such instructions, the computer can perform the instructions sequentially. Typically the instructions are of one of three types. An instruction can be to perform an operation on a value, to access memory to store or retrieve information or instructions, or to jump to another location in the program if required to do so. The values the instruction is to operate upon are called operands. In the instruction “add A to B”, A and B are the operands.
14. The instruction set of any processor is defined in binary code, and programs expressed in this form are known as machine code (or native code). Instructions in machine code are represented by numbers or opcodes, with each instruction having a unique opcode. In order to make machine code more intelligible, one can replace opcodes with short mnemonic labels to stand in place of numbers, e.g. “ADD” instead of the numerical opcode for addition. These labels are also known as strings, since they are just strings of characters. Strings may also be used to stand for operands or other constant values that appear repeatedly in the program. The resulting mode of expression is called assembly code or assembly language. Assembly code is much easier to write and to understand than machine code. By its nature, any assembly language is specific to the particular processor on whose instruction set it is based.
15. In order to execute a program written in assembly code, the program must first be transformed into machine code. This is done by a tool called an assembler, which translates the mnemonic labels into binary instructions ready for execution by the processor. The operation of an assembler may include steps designed to optimise the output binary file, which includes eliminating unnecessary code and making what remains more concise and more efficient to execute. Such optimisation typically involves multiple passes over the program, since each pass may create fresh opportunities for optimisation.
16. Assembly language is an example of a “low-level” programming language. “Low-level” means that the language is close to the basic instruction set understood by the processor, and thus requires relatively little translation in order to be executable by the processor. As such, low-level programming languages force the human programmer to do the work of translating the conceptual functions the program is intended to perform into terms which the processor will be able to handle.
17. “High-level” programming languages are ones which are further from the processor’s basic instruction set, but which allow the programmer to employ more abstract concepts and express instructions more naturally (compared to low-level languages). The result is a language which lets the programmer write in terms of the problem to be solved, rather than having to think about how the program will run on the hardware. High-level languages thus move away from specific machine-focused instructions towards more powerful abstract concepts and commands, and so lend

themselves to more fluent and more sophisticated programming. As a rule of thumb, the more high-level the programming language, the closer its instructions and syntax will be to plain English and the further they will be from the limited set of opcodes understood by the processor. High level languages include BASIC, FORTRAN, C, PASCAL and JAVA (or Java).

18. There are different kinds of high-level programming languages (e.g. “procedural” and “object oriented”). At one stage in these proceedings the distinction between these types was important but by the closing, the relevant argument was dropped and the distinction does not require elaboration.
19. In order for a program written in a high level language to work on a computer, that program has to be translated into machine code. There are two different approaches to this, using a compiler or using an interpreter. A compiler is a program which takes a program written in the high level language (the source code) and creates an entire program written in another language (the object code). The object code can be machine code. In many cases a number of source files will cross refer to each other and also to other files containing libraries of code for standard functions. In order to create a single executable program, a tool called a linker or link editor is used to consolidate all of the files into a single file and resolve all the cross-references.
20. A different approach from using a compiler is to use an interpreter. An interpreter interprets the high level language at run time, effectively line by line. The software of the interpreter takes the original program as its input and works out what the result of executing that program should be and outputs the result. Generally using an interpreter is a less efficient way of executing a program than using a compiler but interpreters have other advantages which will emerge below.
21. During the transition from source code to executable program, the code can be optimised. This usually involves making the code smaller and more efficient to execute. The details of code optimisation are addressed below.
22. A virtual machine is a kind of interpreter program which operates as if it were a computer processor for the input program. In other words it is a virtualised implementation of a hardware architecture with its own virtual instruction set and opcodes and so on. The virtual machine may be designed to mimic a real computer or it may be a hypothetical machine. From the point of view of the input program a virtual machine behaves exactly like a real hardware processor, executing instructions and so on. Underpinning the performance of the virtual machine is a real machine on which the virtual machine is running. The virtual machine can only run its own instruction set by implementing each instruction in the instruction set of the underlying real machine. The correspondence between its instructions and those of the underlying hardware is often known as a dispatch table. Although running a program on a virtual machine is bound to be slower than running the same program in machine code on the actual underlying processor, there are advantages to using virtual machines. Also, since modern computers run so fast, a virtual machine may still run application programs at an acceptable speed. This allows the original executable code for arcade games to be run unchanged so that the original games can be played on modern computers. An advantage of virtual machines is that because they take charge of executing the program, they can be designed to enforce security checks on any

application being executed. This may include verifying the trustworthiness of an application, and ring-fencing access to data held by the computer.

23. Usually, for all but very small applications, the size of the code for an application when compiled to machine code is bigger than the total code size of the original application program plus a virtual machine to run that program.
24. Virtual machines can be set up to emulate a CPU with registers (a “register machine”) but even if the real CPU on which the virtual machine is running is a register machine, the virtual machine may be different. The virtual machine may be a “stack machine” in which the (virtual) processor, rather than storing values in registers, stores them in stacks. When a value is stored in a stack it is placed notionally on the top of the stack, above the previous entry. The previous entry still exists, now one place further down the stack. When data is retrieved from the stack, it is the value at the top of the stack which is read. A stack can be considered a “last in first out” system.

865: Witnesses

25. HTC called expert evidence from Dr David Greaves. He is currently a senior lecturer in computing science at the University of Cambridge, where he has taught courses on microcontrollers and embedded systems and on compilers. During his time at Cambridge he has had various part time roles in industry, including being chief scientist of the chipset company Virata and of the compiler company Tenison EDA. He has extensive experience in hardware, operating systems and compilers and is very familiar with interpreted and compiled high level languages.
26. Dr Greaves was a good witness seeking to help the court understand the issues in this case. An important submission about Dr Greaves’ evidence related to the way in which discussions with Dr Greaves had been initiated and approached. This related to obviousness and I will deal with it in context below.
27. Gemalto submitted that Dr Greaves had no or only peripheral experience of the relevant art and that by selecting him as an expert, HTC had side stepped the relevant mindset of the skilled person. I will address the relevant art below but in any event I do not accept this point in relation to Dr Greaves. Gemalto’s argument was that Dr Greaves’ particular expertise in compilers would never have been sought out by the skilled team. Even if that was correct, as an expert in computer science Dr Greaves assisted me in grappling with the issues arising in this case. His evidence was pertinent and helpful.
28. Gemalto called expert evidence from Professor Pierre Paradinas. Prof. Paradinas holds the Embedded Systems chair at the Conservatoire National des Arts et Métiers in Paris. From 1989 to 1995 he was co-director of a lab called “RD2P”. This was a research laboratory jointly run by Gemplus, a smart card manufacturer, and Lille University. Of the four items of prior art relied on by HTC which were published before the claimed priority date, three derive from the RD2P lab.
29. The 865 patent was granted to Schlumberger Systemes. That company changed its name to Axalto and then Axalto merged with Gemplus to form Gemalto. Since leaving RD2P in 1995, Prof. Paradinas has been independent of Gemplus.

30. Prof. Paradinas' mother tongue is French and an interpreter was sworn in to assist. However he gave his evidence almost entirely in English. During his cross-examination on 865 there were often lengthy pauses before the Professor answered counsel's questions. HTC submitted this could not simply be explained by the fact he was giving evidence in a second language and was indicative of a lack of credibility. I do not agree. On a number of occasions (although not all) the questions being asked were lengthy and would require careful thought by a native English speaker. The Professor also often used the laptop computer in court to read a live transcript of the questions, clearly as an aid to interpretation. This also took up time. In my judgment Prof. Paradinas was mindful of his duties as an expert and was seeking to assist.
31. HTC also argued that Prof. Paradinas's evidence fluctuated, particularly between what he said in cross-examination as compared to his report. I agree that a number of answers given in cross-examination were different from points made in his reports but again that did not strike me as a reason for doubting the Professor's credibility. I took this as an almost inevitable consequence of an expert having prepared over 200 pages of report on a single patent. I am not surprised his answers in cross-examination were not always the same as the contents of his reports.
32. Gemalto submitted that Prof. Paradinas was much closer to the relevant art than Dr Greaves. I accept that Prof. Paradinas was much more closely involved in smart cards than Dr Greaves and I will bear that in mind to the extent it is relevant.
33. Gemalto also relied on evidence of fact from Olivier Piou. M. Piou is the CEO of Gemalto's holding company, Gemalto N. V. He was not cross-examined.

865: Person skilled in the art

34. In *Schlumberger v EMGS* [2010] EWCA Civ 819 (Waller, Jacob and Sullivan LJ) Jacob LJ explained that the "person skilled in the art" is explicitly referred to three times in the European Patent Convention (paragraph 30 *et seq*). These three places are the Protocol to Art 69, Art 83 and Art 56. The first two are concerned with the position post-grant of the patent, determining the proper interpretation of the claims and the sufficiency of the disclosure. The third is concerned with the position pre-grant, the obviousness of the invention over the prior art. The conclusion in *Schlumberger* was that one can distinguish between the identity of the person skilled in the art from the point of view of considering obviousness and the identity of the person skilled in the art from the point of view of understanding and implementing the patent. The latter person is sometimes called the skilled addressee or skilled reader. No doubt in many cases the skilled person for obviousness will be the same as the skilled addressee but *Schlumberger* is authority for the proposition that it will not necessarily be the case. The correct identity of the skilled addressee obviously depends on the content of the patent specification, however the skilled person from the point of view of obviousness necessarily has not read the patent.
35. There can be no real doubt about the skilled addressee of the patent. It will be a team interested in programming a microcontroller using a high level language. In order to implement the patent this team would need people with expertise in the relevant hardware and in programming and software engineering including compilation and optimisation techniques.

36. Gemalto did not agree that this description of the skilled addressee applied to the skilled person from the point of view of obviousness. I will address that below.

865: Common general knowledge of the skilled addressee

37. I will not add to the words already written in the cases about the test for common general knowledge. At this stage I will consider only the common general knowledge of the skilled addressee in order to approach the task of construing the patent. The matters of background set out in the section above were part of the skilled addressee's common general knowledge. Although entitlement to priority is disputed, it is convenient at this stage to address the common general knowledge at the priority date (Oct 1995).

Smart cards and integrated circuit cards

38. The terms "smart card" and "integrated circuit card" are effectively interchangeable, smart cards being an example of an integrated circuit card. A smart card usually comprises a single chip which has a CPU, RAM, ROM and other non-volatile memory, usually EEPROM or FLASH. It is generally supplemented by security sensors. The card has a contact plate; the contacts put the card into contact with the outside environment. The ROM contains the card's program. The programs are stored during the manufacturing process. The EEPROM is used to store data. Security is provided in various ways. There may be sensors to detect attempts at reverse engineering. The fact that that elements of the card are on a single silicon substrate prevents an attacker from probing the card and reading the data stored in it. The card may execute cryptographic functions using keys present in the memory of the card which only the card and the transmitter know. Only the card and the transmitter will be able to understand enciphered messages they exchange.
39. At the relevant time smart card applications could be divided into three groups: payments, telecommunications and others. In payment applications the card represents a bank account holder. It can be used at an ATM (Automatic Teller Machine) or POS (Point of Sale) reader. In telecommunications the card will be used to represent the subscriber. Thus the Subscriber Identity Module (SIM) in a mobile phone is a smart card. In all these applications the card stores information about the card holder and the card issuer.
40. Professor Paradinas described the life cycle of a smart card in four phases: fabrication by the card manufacturer; personalisation when the card receives information about the card holder and card issuer; active phase when the card is in the real world, say being used as a payment card; and finally the card may be "blocked" or "deactivated".

Instruction sets

41. The instruction set for a computer will include numerous instructions. In some contexts it is convenient for a computer to have a large instruction set. These can be called CISC machines (CISC standing for Complex Instruction Set Computer). In other contexts it is convenient to have a much smaller instruction set. These are called RISC machines for "Reduced Instruction Set Computer". The difference in the size of the instruction set does not necessarily mean that a CISC machine can do things the RISC machine cannot. It is a reflection of different philosophies in the

design of the architecture of the computers. What would be a single instruction for a CISC machine can be represented in a program for a RISC machine by a number of instructions together.

42. This difference also reflects another concept relevant to this case, the idea of generic and specific instructions. Many instruction sets will include an instruction such as “ADD X,Y”. This means add X to Y or, to be more precise, it could mean take X, add Y to it, and make the new value of X the result of the sum. The same instruction set may also contain another instruction “INC X” which means “increment X”, in other words add 1 to X, in other words make the new value of X the result of the sum of X plus 1. Comparing ADD X,Y and INC X, it can be seen that ADD X,Y can be regarded as a generic instruction since any value Y can be added to X whereas INC X is a specific instruction since only the number 1 can be added to X. A human programmer or a compiler program has a choice. The single line of code “INC X” is identical to these two lines of code “MAKE Y = 1, ADD X,Y”.
43. Generally replacing generic instructions with specific equivalents will make the code smaller. An increment instruction (INC X) is a single instruction which might require perhaps two bytes, one for the byte code representing the instruction “increment” and the other for the operand (X); whereas the generic equivalent might need 6 bytes, three for the instruction to make Y equal the number 1 (one byte for make, one byte for the operand Y and one byte to represent the value 1) and three more for the instruction to add Y to X (one byte for the instruction add, one byte for X and one byte for Y).
44. Thus one form of code optimisation can involve identifying instances in which generic instructions can be replaced by specific ones, making the overall code smaller.
45. One common general knowledge technique was “subsetting”. If one is thinking about an approach which involves an interpreter for the code, it may be that the task in hand does not need to use all the features in the instruction set. For example some instruction sets include instructions for carrying out floating point arithmetic. These sorts of instructions may be useful for some applications but irrelevant for others. So the team could decide to subset the language. A subset of the instruction set is defined which the interpreter will be able to interpret (say every instruction except those dealing with floating point numbers). The interpreter is created on that basis. It is therefore a simpler interpreter to write since it does not have to handle floating point operations. All programs have to be written using the instructions in the subset. A program may be used to verify that the code does not contain instructions outside the subset since they will not be understood. Dr Greaves explained and I accept that subsetting was a technique in common use by programmers working on embedded systems. Even if the skilled team was not focussed on programming an embedded system, the existence of the technique and its utility in that context was part of the common general knowledge.

The compiler tool chain

46. A compiler is in fact a series of software components which are connected together and act in sequence, one after the other. Dr Greaves described this as the compiler “tool chain”. There are only two important elements to which I need to draw attention. As I have mentioned already when a compiler works on source code to create an

executable program it often seeks to optimise the code in order to make it smaller and more efficient to execute. There are numerous kinds of optimisation technique. One kind of optimisation which is relevant to this case is the replacement of generic instructions with specific ones since the specific instruction takes up fewer bytes than the generic one.

47. Another critical element in the compilation tool chain is linking. Programs often consist of a number of independent files of source code. Some standard code used in many applications may be in a code library. A linker is used to link together the compiled versions of all these source code files into a single executable file. Linking also provides another opportunity for code optimisation.

Java

48. There was no dispute that although Java was new at the relevant time, its existence was part of the common general knowledge. The Java package as a whole or Java “platform” consisted of a number of components. Java was a high level programming language. Once code had been written in Java, it was compiled by the Java compiler to produce an intermediate form of code called byte code. Once compiled to byte code, the computer files making up the program were called Class files. Byte code formed the code for a virtual machine called the Java Virtual Machine or JVM. The JVM interpreted the byte code. Thus the Java compiler was not machine specific, since it produced a program made up of byte code in the Class file format. On the other hand the JVM would be machine specific. The result was that as long as a computer had a JVM installed on it, that computer could run programs which had been written in Java because the programs were presented as Class files. This was very useful in the context of the internet. It meant that software in the form of Class files could be made available from a website. As long as the user’s computer had a JVM, it did not matter what sort of computer it was. The users could all obtain the same Class files from a website and run the same program. Java also had a number of security features. Some of them arose from the fact that the byte code was going to run on a virtual machine, which had security built in. All of this was part of the common general knowledge of the skilled addressee.
49. The JVM had a number of functional components. As well as an interpreter or “execution engine” which interpreted the byte code, the JVM also carried out linking, optimisation of the code and verification. The view of Prof. Paradinas was that the skilled team would regard the JVM as a single entity for running Class files and would have no particular reason to divide it up internally. Gemalto pointed out that Dr Greaves accepted that some people would just be users and would not care how the JVM works internally. Nevertheless I find that the idea that the JVM had consisted of components with different functions would be common general knowledge. The skilled team would know it was comparatively large and would know, without having to be told or reminded, that the JVM was not merely an execution engine for byte code but carried out other functions such as linking and security/verification tasks. However I think Prof. Paradinas’ evidence which I have referred to reflects something important about the skilled team in that, without any prompt to think more carefully about the JVM, the skilled team would not have more than a superficial interest in the contents of the JVM.

50. Another issue about Java and the common general knowledge was whether the skilled team regarded the Java Class file as “sacrosanct” (a word used by Prof. Paradinas). The advantage of portability between different computers was a key aspect of Java’s importance and a key reason it had become common general knowledge so quickly after it had been launched. That portability comes from the portability of Class files but I reject the argument that this had a relevant impact on the mindset of the skilled team in the sense that it might discourage them or act as a disincentive to alter Class files in some way. This portability was an important commercial matter but it was not a technical reason why one would not modify Class files. The skilled team would know very well that modifying the Class files would sacrifice portability but would also understand that this was a trade off. Whether it was worthwhile in a given situation would be a matter to take into account in deciding what to do.

865: The patent

51. The invention is said to relate in general to the field of programming and more particularly to using a high level programming language with a smart card or microcontroller. Paragraph [0003] explains that software applications written in the Java language have been so designed that an application in Java can run on many different computers. Class files, byte code and the JVM are referred to and in paragraph [0004] reference is made to certain texts which describe Java in more detail.

52. Paragraph [0005] points out that in order for a Java application to run on a specific platform, a Java virtual machine implementation must be written that will run within the constraints of the platform, and a mechanism must be provided for loading the desired Java application on the platform, again keeping within the constraints of this platform. Paragraph [0006] describes the problem as follows:

“Conventional platforms that support Java are typically microprocessor-based computers, with access to relatively large amounts of memory and hard disk storage space. Such microprocessor implementations frequently are used in desktop and personal computers. However, there are no conventional Java implementations on microcontrollers, as would typically be used in a smart card.”

53. From paragraphs [0007] to [0018] the background section discusses microcontrollers and other matters. Since this relates to a hotly contested issue of construction I will address these paragraphs below.

54. The “Summary of the Invention” section runs from paragraph [0019] to paragraph [0043]. This section rarely mentions microcontrollers despite the words of the claims. For example paragraph [0019] states:

“In general, in one aspect, the invention features an integrated circuit card for use with a terminal. The integrated circuit card includes a memory that stores an interpreter and an application that has a high level programming language format. A processor of the card is configured to use the interpreter to

interpret the application for execution and to use a communicator of the card to communicate with the terminal.”

55. From paragraph [0044] onwards the specification provides a detailed description of preferred embodiments, with drawings. The main preferred embodiment is a way of running Java programs on a smart card or integrated circuit card. The way this works is that the program is written in the Java high level language as normal and compiled to the normal Class file format using the normal Java compiler. Normally this produces multiple Class files. Next a converter is used to consolidate and compress the files to produce a single “card class file” and the card class file is then loaded onto the smart card using a conventional card loader [paragraph 0075]. The smart card contains a card Java virtual machine (Card JVM) which is used to interpret applications which are contained on the card [paragraph 0073]. In practice the Card JVM is a sort of cut down version of the normal JVM, having regard to the fact that the memory available on smart cards is much less than is usually available to computers which run a JVM. This way the problem posed in paragraph [0006] is solved. Programs written in a high level language (Java) can be run on a smart card.
56. This section of the specification is substantial, with details of the converter and the card JVM. It is followed by nine appendices which include further details. Nothing turns on the appendices.

865: Claim construction

57. The claims alleged to be independently valid are claims 1, 3, 8, 9, 15, and 18. They are set out below without their reference numerals.

Claim 1

A microcontroller having a set of resource constraints and comprising:
a memory,

an interpreter loaded in memory and operating within the set of resource constraints, the microcontroller characterized by having:

at least one application loaded in the memory to be interpreted by the interpreter, wherein the at least one application is generated by a programming environment comprising:

- a) a compiler for compiling application source programs in high level language source code form into a compiled form,
- b) a converter for post processing the compiled form into a minimized form suitable for interpretation by the interpreter.

Claim 3

The microcontroller of Claims 1 or 2 wherein the compiled form is in a standard Java class file format and the converter accepts as input compiled form in the standard Java class file format and produces output in a form suitable for interpretation by the interpreter.

Claim 8

The microcontroller of Claims 4, 5, 6, or 7 wherein the compiled form is in a byte code format and the converter comprises means for translating from the byte codes in the compiled form to byte codes in a format suitable for interpretation by the interpreter using at least one step in a process including the steps:

- a) recording all jumps and their destinations in the original byte codes;
- b) converting specific byte codes into equivalent generic byte codes or vice-versa;
- c) modifying byte code operands from references using identifying strings to references using unique identifiers;
- d) renumbering byte codes in the compiled form to equivalent byte codes in the format suitable for interpretation; and
- e) relinking jumps for which destination address is effected by conversion step b, c, or d.

Claim 9

The microcontroller of any of the preceding claims wherein the application program is compiled into a compiled form for which the resources required to execute or interpret the compiled form exceed those available on the microcontroller.

Claim 15

A method of programming a microcontroller having a memory and a processor operating according to a set of resource constraints, the method comprising the steps of:

inputting an application program in a first programming language;

compiling the application program in the first programming language into a first intermediate code associated with the first programming language;

wherein the first intermediate code being interpretable by at least one first intermediate code virtual machine;

wherein the method of programming a microcontroller is characterized by:

converting the first intermediate code into a second intermediate code;

wherein the second intermediate code is interpretable by at least one second intermediate code virtual machine; and

loading the second intermediate code into the memory of the microcontroller (10).

Claim 18

The method of Claims 15, 16, or 17 wherein the step of converting comprises at least one of the steps of:

- a) recording all jumps and their destinations in the original byte codes;
- b) converting specific byte codes into equivalent generic byte codes or vice-versa;
- c) modifying byte code operands from references using identifying strings to references using unique identifiers;
- d) renumbering byte codes in the compiled format to equivalent byte codes in the format suitable for interpretation; and
- e) relinking jumps for which destination address is effected by conversion step a), b), c), d).

58. In *Samsung v Apple 726 and 675 patents* [2013] EWHC 467 (Pat), Floyd J (as he then was) summarised the law as follows:

“[66].In *Kirin Amgen v TKT* [2005] RPC 9 the House of Lords explained that the determination of the extent of protection only involves asking what a person skilled in the art would have understood the patentee to have used the language of the claim to mean. Guidelines to assist the court in construing the patent are summarised by the Court of Appeal in *Virgin Atlantic v Premium Aircraft* [2009] EWCA Civ 1062; [2010] FSR 10 at paragraph 5, approving the statement by Lewison J (as he then was) at first instance in the same case:

‘[5] One might have thought there was nothing more to say on this topic after *Kirin-Amgen v Hoechst Marion Roussel* [2005] RPC 9. The judge accurately set out the position, save that he used the old language of Art 69 EPC rather than that of the EPC 2000, a Convention now in force. The new language omits *the terms of* from Art. 69. No one suggested the amendment changes the meaning. We set out what the judge said, but using the language of the EPC 2000:

[182] The task for the court is to determine what the person skilled in the art would have understood the patentee to have been using the language of the claim to mean. The principles were summarised by Jacob LJ in *Mayne Pharma v Pharmacia Italia* [2005] EWCA Civ 137 and refined by Pumfrey J in *Halliburton v Smith International* [2005] EWHC 1623 (Pat) following their general approval by the House of Lords in *Kirin-Amgen v Hoechst Marion Roussel* [2005] RPC 9. An abbreviated version of them is as follows:

- (i) The first overarching principle is that contained in Article 69 of the European Patent Convention;

(ii) Article 69 says that the extent of protection is determined by the claims. It goes on to say that the description and drawings shall be used to interpret the claims. In short the claims are to be construed in context.

(iii) It follows that the claims are to be construed purposively—the inventor's purpose being ascertained from the description and drawings.

(iv) It further follows that the claims must not be construed as if they stood alone—the drawings and description only being used to resolve any ambiguity. Purpose is vital to the construction of claims.

(v) When ascertaining the inventor's purpose, it must be remembered that he may have several purposes depending on the level of generality of his invention. Typically, for instance, an inventor may have one, generally more than one, specific embodiment as well as a generalised concept. But there is no presumption that the patentee necessarily intended the widest possible meaning consistent with his purpose be given to the words that he used: purpose and meaning are different.

(vi) Thus purpose is not the be-all and end-all. One is still at the end of the day concerned with the meaning of the language used. Hence the other extreme of the Protocol—a mere guideline—is also ruled out by Article 69 itself. It is the terms of the claims which delineate the patentee's territory.

(vii) It follows that if the patentee has included what is obviously a deliberate limitation in his claims, it must have a meaning. One cannot disregard obviously intentional elements.

(vii) It also follows that where a patentee has used a word or phrase which, acontextually, might have a particular meaning (narrow or wide) it does not necessarily have that meaning in context.

(vii) It further follows that there is no general "doctrine of equivalents."

(viii) On the other hand purposive construction can lead to the conclusion that a technically trivial or minor difference between an element of a claim and the corresponding element of the alleged infringement nonetheless falls within the meaning of the element when read purposively. This is not because there is a doctrine of equivalents: it is because that is the fair way to read the claim in context.

(ix) Finally purposive construction leads one to eschew the kind of meticulous verbal analysis which lawyers are too often tempted by their training to indulge.'

67. I would stress only two points from this summary, given the importance of the issue of construction in this case. The first is that the

exercise is one of construing the language of the claims in the context of the specification. The meaning of that language is informed by the technical understanding gained from reading the specification. Thus the specification has an important role in understanding the meaning of the language used. It is not, however, a proper approach to construction to start with the specification and ask what a patentee who has made that disclosure might be intending to claim, and then to shoe-horn the meaning of the language of the claim to fit with that understanding, whatever language he has actually used. To do so would be to afford supremacy to the description over the claims, contrary to the guidance given by Article 69 EPC and its protocol.

68. The second point is this. The patentee may have described a number of embodiments or examples of increasing sophistication in the body of the specification. Having done so, the patentee has the freedom to set the generality of his claim at the level of his choosing. There is no presumption that he will have decided to pitch his claim at the level of the most sophisticated embodiment. It is the claims which will tell the skilled reader at what level the patentee has decided to stake his monopoly claim. The skilled reader would not be justified in assuming that the patentee has elected to claim the features of the most sophisticated embodiment, so as to compel the conclusion that those features are read into the claims. Equally, as sub-paragraph (v) in the above summary indicates, the skilled reader does not assume that the patentee is aiming at the widest possible construction consistent with his purpose.”

59. Some of the issues on construction are best dealt with in the context in which they arise. The points I need to address at this stage are set out below :

- i) “*microcontroller*”
- ii) “*having a set of resource constraints*”
- iii) “*loaded in memory*”
- iv) “*a converter*”
- v) Claim 8

(i) microcontroller

60. The parties’ rival constructions of the term microcontroller advanced in closing were far apart. HTC submitted that to the skilled addressee what the patent means by the term microcontroller is a reference to a single chip which contained a CPU and also had all its memory resources on the chip. In summary Mr Tappin’s argument was that this was the normal usage of the term microcontroller which the skilled addressee would have been aware of and when they read the patent, they would see that the patentee was using the word in that sense.

61. Gemalto’s case is summarised in paragraphs 99 – 100 of its closing argument:

“99. Microcontroller is an ordinary word in this technical field, with no special meaning to be derived exclusively from the description of the patent or externally. It means simply a controller which contains a microprocessor – hence microcontroller. The “controller” element connotes that it is not just a microprocessor, but also contains the elements necessary for it to exercise control, and so includes other functional elements such as memory, input/output etc.

100. In context, it is clear that the distinction being drawn in [0006] to [0009] of the background section of the patent and the purposes of the limitation to a “microcontroller” in the claim is between (a) microprocessor-based computers (i.e. general purpose computers) and (b) microcontrollers (in the sense of a dedicated system, as used in an embedded system). The claim is directed to dedicated systems – viz microcontrollers.”

62. Amongst other things therefore, Gemalto denies that the memory has to be on the same silicon substrate as the CPU.
63. The way to resolve this problem involves two steps. First I need to consider what if anything the expression meant to the skilled addressee without reading the patent. The word is not an expression used in ordinary English outside this technical field. It is a term of art and evidence is admissible to discern its meaning. This is in effect an exercise in considering the common general knowledge of a skilled addressee. Next I will consider how the skilled addressee would understand the patent. It may be that the skilled reader would understand the expression as having been used in its familiar sense or one of its familiar senses or perhaps in a different sense altogether. Although the final step is plainly vital, both steps are necessary.

How has the word “microcontroller” been used in the art?

64. A 1997 textbook “The Microcontroller Idea Book” by Jan Axelson was put to Dr Greaves in cross-examination. Its subtitle is “Circuits, Programs & Applications featuring the 8052-BASIC Microcontroller”. The 8052 microcontroller was a chip in the 8051 microcontroller family. On the chip was a CPU, 8 kilobytes of ROM, 256 bytes of RAM and some I/O ports. The addition of the tag “BASIC” referred to the fact that the 8052-BASIC was the same chip as the 8052 but the ROM contained a BASIC interpreter program.
65. On page 1 under the heading “What’s a Microcontroller?” the author explains that:
- “A microcontroller is a computer-on-a-chip or, if you prefer, a single-chip computer. *Micro* suggests that it is small, and *controller* tells you that the device might be used to control object, processes or events. Another term to describe a microcontroller is *embedded controller*, because the microcontroller and its support circuits are often built into, or embedded in, the devices they control.”

66. This passage illustrates the dispute. Each side's construction is effectively present. Does microcontroller mean a computer on a chip (with memory etc.) as HTC contends or does it mean a dedicated computer, a microprocessor based controller embedded into the device it controls?
67. The historical derivation of the term was not fully explained but it is clear that the term "microcontroller" emerged when microprocessor chips started to be used as controllers, i.e. in dedicated or embedded systems. The Axelson book explains some of the history at pages 2 to 3, parts of which were put to Dr Greaves. Along with the development of cheap powerful personal computers came an interest in small customised computers for specific uses such as automobile engine controllers. The small special-purpose computers are or were sometimes called "single-board computers" although Axelson explains that term can be misleading since the computer does not need to have a single circuit board and (by then) many other types of computers like laptops were now being manufactured on a single board as well.
68. Single chip computers were useful for embedded systems and no doubt that is why the term became associated with them. At some stage the expression "single chip microcontroller" was used. Plainly the point of the expression is to emphasise that this sort of microcontroller is one of the single chip kind, with the CPU, memory and other things on a single chip. Necessarily using the term "single chip" to qualify microcontroller shows that, at least at one stage, not all microcontrollers were single chips. Dr Greaves was asked about the phrase "single chip microcontroller" (Day 4/p458ln25 – p459 ln24). He said:

24 Q. So microcontrollers with their CPU and memory all on a single
25 chip, there were a large number of those in 1996?

...

2 A. Yes.

3 Q. And those were frequently referred to as single chip
4 microcontrollers?

5 A. I would have thought that by that date the single chip
6 qualification would largely be dropped.

7 Q. Are you aware that manufacturers still use the term single
8 chip microcontroller?

9 A. Yes, and no doubt -- in the general sense of what that phrase,
10 what that word means -- meant -- at that priority date, I do
11 not think it requires qualification. The patent does not talk
12 about a single chip microcontroller, it just gives a
13 definition in that paragraph of what it is talking about and
14 it does not qualify that the whole time.

15 Q. But the use of the term "single chip microcontroller"
16 distinguishes those from microcontrollers which do not have
17 all their functional blocks on a single chip. That is right,
18 is it not?

19 A. Prior art, prior times, previous decade, certainly.

20 Q. Also, if people are still using the term single chip
21 microcontroller, the term is ----

22 A. I do not think it is widely used like that any more. When we
23 say microcontroller, we mean single chip microcontroller, as
24 of roughly the priority -- well, start in 1990, say.

69. I accept Dr Greaves evidence in that while no doubt it is possible to find people in 1996 still using the term single chip microcontroller, the term single chip microcontroller had been used in the past but was not a widely used term by the priority date.
70. Another expression discussed in evidence was “ROMless microcontroller”. This was a phrase which was familiar to those in the field in 1996. An example of a ROMless microcontroller which Dr Greaves was aware of from the relevant time was a chip called the 8031 which was in the 8051 family but had no ROM on board. Part of the argument advanced by Gemalto was that the existence of the term ROMless microcontroller showed that a microcontroller did not as a matter of necessity have to have on-chip ROM. The argument may be flawless as a matter of logic but I do not accept it follows that just because a skilled person would understand what a ROMless microcontroller was, this displaces the point that what a skilled person would take as the typical meaning of the term microcontroller was a single chip with the CPU, memory (and the I/O interfaces) on that one chip.
71. Other examples put to Dr Greaves showed that manufacturers were selling chips called microcontrollers which accessed off-chip memory. One example was an SGS Thomson ST10R165. This chip could access 16Mbytes of external RAM or ROM. Another example put to Dr Greaves was the Zilog Z86C93. This was a ROMless microcontroller which could access 64 Kbytes of external memory. Dr Greaves’ position on the Z86C93 was the same as for all the examples put to him. Chips are made in families, in this case Z8 family of microcontrollers well known in the early 1990s. It is a ROMless version of a microcontroller and is part of the family. Other members of the family will have things on-chip. Dr Greaves was asked a number of times if it was a misuse of the term microcontroller to use it in the context of a ROMless device or a device which accessed external memory. In summary his view was that these were variants on the norm.
72. Despite all the examples of microcontroller put to Dr Greaves, it was clear that not all chips with CPUs on them were called microcontrollers. An example in evidence was the 8080 chip. This was an early microprocessor and no-one suggested that this item, as a chip, was ever called a microcontroller. It is not a single chip computer. The device needs extra memory and other things which are only provided off-chip in order to function.
73. There is one aspect of Dr Greaves’ evidence on this topic which I need to treat with care. He was approaching the issue as being one in which he was to explain what a skilled person would have said if asked to give a precise definition. It is no fault of Dr Greaves but it is not the question I have to decide. When experts assist the court in understanding what technical terms meant to a person skilled in the art, if a term has a loose meaning or is capable of multiple meanings, then so be it. The task is not to impose precision if it does not exist.
74. Dr Greaves’ evidence on the use of the term microcontroller in the art can be summarised as follows:
 - i) The skilled person would have understood the term “microcontroller” to mean that all the components (apart of course from power source) were on a single chip.

- ii) There were occasions where components could have access to off-chip memory as well as on-chip memory but were labelled as microcontrollers because they were variations on a family of microcontrollers.
 - iii) There were also sometimes ROMless microcontrollers in those families of microcontrollers, but they were described as such, rather than as microcontrollers without the qualifying adjective.
 - iv) When a microcontroller is connected to external memory and the CPU is executing code using that external memory, it is behaving as a microprocessor and not a microcontroller.
75. I note that point (iv) in this list was also something Prof. Paradinas accepted in cross-examination (Day 1 p169 19-23).
76. Prof. Paradinas's view was that the skilled person would have understood that while a microcontroller would have typically included both volatile and non-volatile memory on-chip, this did not preclude a microcontroller from accessing external off-chip memory. Mr Tappin did not submit that this was wrong but pointed out that all of the examples given by the Professor of microcontrollers which accessed off-chip memory also had on-chip memory.
77. Mr Tappin emphasised that Prof. Paradinas did not suggest that the skilled person would have understood that something could be a microcontroller and yet have neither RAM nor non-volatile memory on chip. I think Mr Tappin is right. Mr Mellor did rely in argument on an example of a microcontroller in which the RAM and ROM were off-chip, which he had put to Dr Greaves in cross-examination. It was a page from a magazine called Analog Dialogue from 1994 and described a chip set to implement the baseband portion of a GSM mobile phone handset. One of the chips in the set was an Application Specific Integrated Circuit (ASIC) which includes an embedded H8 16-bit microcontroller which the system diagram showed accessed off-chip RAM and ROM. Dr Greaves accepted this was like the other ROMless microcontrollers which had been put to him. For what it is worth it is not clear to me that the microcontroller itself had no on-chip memory.
78. Mr Mellor submitted that the evidence showed that there was no generally accepted universal definition of microcontroller in the common general knowledge. I agree in that the evidence is clear that there is no universal definition. The term is used in different senses in different contexts. Mr Mellor also submitted that the conclusion to be drawn from the evidence is that "microcontroller" did not have the precise definition put forward by Dr Greaves. It is quite clear the term was not one with a single well defined meaning, but that does not mean the expression was devoid of meaning nor does it mean that Gemalto's submission is right, that by 1996 a microcontroller was simply a controller which contains a microprocessor.
79. I find that the skilled addressee would be familiar with the word in 1996. It was used in a variety of senses. There are certain chips which in 1996 were called microcontrollers. That is one sense in which the word was used. If you told a skilled person that a chip was a microcontroller and told them nothing more about it, they would think they knew what you meant and would expect it to contain, on a single silicon substrate, a CPU, memory (RAM and ROM) and I/O interface devices. Thus

manufacturers could sell chips called microcontrollers irrespective of the use to which they were going to be put. No doubt in general they were best suited to be used in embedded systems and were in fact used for that application but whatever the chip was to be used for, the chip itself was a microcontroller. This meaning of the term was the normal usage by those skilled in the art in 1996.

80. An extended version of the normal usage covered variants on microcontroller chips, such as ROMless microcontrollers. These shared many of the characteristics of typical microcontrollers but in such a case the word was being used in an extended sense. Another extension of the normal usage related to microcontroller chips which could access off-chip memory. These clearly existed. However one thing about all microcontroller chips was that on a single silicon chip there was more than just a CPU. Normally the chip also had I/O interface devices and memory. I am not satisfied I was shown any example of a microcontroller chip with no on-chip memory (ignoring registers) and I reject the idea that a skilled person using the language in its normal or extended sense would be comfortable calling a chip with no on-chip memory a microcontroller.
81. A different sense in which the term was used was to refer to the computer in an embedded or dedicated system. Now if one pointed a skilled person to a washing machine and asked them what was inside controlling the machine, they might say a microcontroller. That would be true irrespective of the chip actually being used. It might be a standard 8051 microcontroller but there is no reason in principle why the microcontroller inside a washing machine might not be a board with an 8080 microprocessor on it as well as associated external RAM and ROM and I/O interfaces. That usage of the term microcontroller was not referring to a chip as such at all. This is the sense in which Gemalto contends the word should be interpreted, as a controller which contains a microprocessor. My impression from the weight of the evidence was that this usage was not common in 1996 but it was one of the senses in which the word had been used.
82. A point arose about whether a microcontroller could have cache. By 1996 the skilled person would know that some microprocessors had on-chip cache. Dr Greaves said that the skilled person would never describe something with only a CPU and cache on-chip as a microcontroller. Prof. Paradinas said that cache was more part of the microprocessor than the microcontroller, by which I understood him to mean that cache, like registers, are part of the architecture of the processor itself. A chip with a CPU which has only cache and no other on-chip memory (ignoring registers) is not within either the normal usage of the term microcontroller I have identified nor is it within the extended version. It is not a microcontroller chip. On the other hand the presence of cache on the chip has nothing to do with the wider definition of microcontroller which relates to a computer in an embedded system. If a computer exercising control in an embedded system had cache, that does not stop it being within the wider definition of microcontroller.

“Microcontroller” in the patent

83. I will start with the claims. The word microcontroller is used in all of them but there is nothing either side has been able to point to in the claim language itself which favours either side. The claims could be read with either Gemalto’s construction of the word or HTC’s.

84. What about the patentee's purpose? HTC submitted that this assisted their construction because the problem being described in the patent is that to run a Java application needs a JVM but the JVM, which is large, will not fit into the highly constrained memory on a microcontroller chip. Gemalto submitted that it was not correct to take the specific problem addressed in a patent specification and use that as a justification for reading limitations into a claim which are simply absent. I think there some force in Gemalto's point. Inventions often arise from a particular problem faced by an inventor which is described in detail in the specification. Once the solution has been found it is often possible to generalise and so to write a claim which is more broadly based than the particular problem addressed. In this case the fact that claim 1 is not limited to Java is an example. However HTC retorts that the reference to the problem as they describe it is in the background section of the patent, where one is entitled to expect more general ideas. Nevertheless, looking at the patent as a whole, I do not accept that discerning the inventor's purpose helps to resolve this dispute. The problem addressed is concerned with implementing high level languages when resources are limited. Microcontrollers can have limited resources whether one is using the term in HTC's sense or in Gemalto's sense. In this case purpose is neutral.
85. Another argument based on purpose was as follows: the specification refers to security; a key element of smart card security is that microcontroller chips have no external system bus because all their memory and I/O is on-chip; hence in order to achieve the patentee's objective of security, a microcontroller must have all its memory on-chip. I reject this argument. The argument would work to justify the claim *covering* the case of a microcontroller with all its memory on chip but in no sense does it justify reading the claim as *limited* to that case.
86. I will turn to consider the way the language is used in the specification.
87. I have set out paragraph [0006] of the specification above. The point being made is that, as the skilled addressee knows, the JVM is large and needs the larger memory and storage space available on typical desktop and personal computers. A contrast is being drawn between microprocessor based computers (desktops etc.) and microcontrollers typically used on smart cards. The point is that microcontrollers have access to very little memory. I think it is (just) possible to read this in both HTC's and Gemalto's sense of the term microcontroller but I believe it reads more naturally using HTC's sense of the word.
88. Paragraph [0007] continues the contrast between microprocessors and microcontrollers. What are being compared in paragraph [0007] are chips: microprocessors and microcontrollers. Mr Mellor submitted to the contrary; he said that "microprocessor" here is shorthand for "microprocessor based computer" mentioned in the previous paragraph. That is wrong. For example the paragraph states "*a microprocessor typically has a central processing unit that requires certain external components (e.g., memory, input controls and output controls) to function properly*". The term microprocessor here is being used to refer to a chip not a whole computer system. The contrast being drawn in paragraph [0007] is between the amount of memory which the chip can access. As it says "*A typical microcontroller can access one to sixty-four kilobytes of built-in memory, with sixteen kilobytes being very common.*" (my emphasis).

89. In paragraph [0009] the same sort of contrast is drawn but the comparison is between a microprocessor system and a microcontroller. This paragraph is neutral. In isolation the term microcontroller at line 54 could be being used in either HTC's or Gemalto's sense.
90. By stating "*due to the small number of external components required and their small size, microcontrollers frequently are used in integrated circuit cards, such as smart cards*" paragraph [0010] is clearly using microcontroller to refer to a chip.
91. Paragraphs [0014] and [0015] could be regarded as neutral although I believe they probably support HTC.
92. The paragraphs addressed so far are in the background section. That means they are not limited to the specific embodiments disclosed and are a legitimate place to for the skilled reader to look to understand the breadth of the disclosure. The summary of the invention section does not use the term microcontroller very much. It would be over-analytical to try and draw inferences from paragraphs which do not even use the word.
93. Paragraph [0020] mentions "embedded systems using microcontrollers can also gain ...", which I take as an example of usage in HTC's sense since the point being made is that in embedded systems which use microcontrollers (i.e. the chips with limited on chip memory) the invention may be useful. On Gemalto's approach to construction the term is redundant in the phrase.
94. Paragraph [0021] received a lot of attention. It includes the sentence "*at least a portion of the memory may be located in the processor*". Gemalto submitted this meant that the specification was acknowledging the possibility that the memory may not be located in the processor, i.e. may not be on the same chip as the processor. Thus Gemalto submitted this favoured their construction. I do not agree. It is preceded by the sentence, "*The processor may be a microcontroller.*" Like much of this section of the patent, this sentence appears to be based on the footing that the invention need not be a microcontroller. However the claimed invention is to a microcontroller and one is left with the impression that this paragraph is not well drafted. The skilled addressee would be wary of drawing conclusions and that is enough to undermine Gemalto's reliance on it. Moreover, if I was required to give a meaning to the relevant sentence, my view is the following. As I read these two sentences, they are in fact talking about a single case. The processor may be a microcontroller and if it is then that will have the consequence that at least a portion of the memory will be located in the processor, i.e. on the same chip. I do not accept this paragraph assists Gemalto. At best this is an indication that the patentee has contemplated what I have called the extended version of the normal usage of the word, referring to a microcontroller chip with on-chip memory albeit that it can access some off-chip memory too.
95. Paragraph [0031], still part of the Summary of the Invention section, provides:

"In general, in another aspect, the invention features a microcontroller that has a semiconductor substrate and a memory located in the substrate. A programming language interpreter is stored in the memory and is configured to

implement security checks. A central processing unit is located in the substrate and is coupled to the memory.”

96. This is clearly using the term in the HTC sense (the same can be said of paragraph [0131]). Of course it does not follow that one can or should merely read this paragraph into a claim.
97. Paragraph [0039] provides:

“In general, in another aspect, the invention features a microcontroller that includes a memory which stores an application and an interpreter. The application has a class file format. A processor of the microcontroller is coupled to the memory and is configured to use the interpreter to interpret the application for execution,”
98. This passage is probably best seen as neutral although I suspect a skilled addressee would actually think it used the term microcontroller to mean a chip.
99. In the Detailed Description section (including the figures) there are clear examples of the use of the term microcontroller to refer to a chip rather than to a dedicated system. An example is paragraph [0131] and another is figure 25, in which the microcontroller 210 is a chip on a printed circuit board. In paragraph [0069] the term appears to be used to refer to the CPU (8 bit microcontroller) as distinct from the memory, but while that does not assist HTC’s case neither does it advance Gemalto’s argument.
100. I find that the use of language in the specification strongly favours HTC. There are some neutral passages but there are clear examples in which the skilled reader would understand what the patentee had used the word to mean, the word has been used to mean a chip. There are no clear examples in which the word was used to mean a dedicated system.
101. On HTC’s construction, in order to find out whether something is a microcontroller, the skilled addressee has to look at the chip itself. Some chips are microcontrollers and some are not. On Gemalto’s construction to identify a microcontroller the skilled addressee looks at the system as a whole. A microprocessor based system dedicated to a particular task is a microcontroller regardless of the particular kind of chip inside or how it is configured.
102. In my judgment the word microcontroller in the claims is being used to refer to the chip. In its normal sense, it means a single chip which contains a CPU and has memory and in all likelihood other functional elements such as I/O interfaces on the chip. The skilled addressee would expect the claim to cover a microcontroller which was within the extended version of the normal usage but that still requires memory on-chip. A chip with a CPU which has registers and cache on-chip but no other on-chip memory is not a microcontroller. The claim is not using the word microcontroller to mean a dedicated system. I reject Gemalto’s construction.
103. I was referred to the judgment of the Mannheim Regional Court (2nd Civil Division) docket 2 O 117/12 between Gemalto and HTC regarding the German designation of the same patent. I believe I have reached essentially the same conclusion as the

Mannheim court. Although the argument presented by HTC in Mannheim was much the same as was presented here, the argument presented by Gemalto appears to have been different and so the two courts have not been faced with the same arguments.

(ii) “*having a set of resource constraints*”

104. Gemalto submitted the resource constraints could be anything: e.g. memory, power, processing speed etc. HTC submitted that the relevant resource constraints related to the fact that microcontrollers did not have sufficient non-volatile memory (ROM) to store the application and the interpreter nor did they have enough RAM to run the application using the interpreter. I reject HTC’s argument as a justification for reading this phrase in a limited sense. The reader would understand that the patentee has, for good of ill, chosen to claim broadly. The claim is not limited to constraints arising from a lack of memory.
105. A point arose on claim 9. Gemalto suggested that HTC’s interpretation of resource constraints in claim 1 required claim 9 to be read into it. Since I have rejected HTC’s interpretation on that point, I do not need to consider claim 9.

(iii) “*loaded in memory*”

106. This phrase appears twice in claim 1, the interpreter is “loaded in memory” and the application is “loaded in the memory”. HTC contended that these phrases mean storing the code in the non-volatile on-chip memory of a microcontroller. HTC argued that “loaded” is a reference to long term storage, where the code is kept in order for it to be executed later as opposed to the placing of code in a short term volatile memory as part of the execution process. Gemalto did not agree. It submitted there was no reason to read this limitation into the word “loaded” and the term “loaded in memory” was apt to cover a case in which code was put into some cache memory when the program was being run. Gemalto argued that “loaded” was different from “stored” and that loaded carried the connotation of “loaded for execution”.
107. The terms loaded and stored are both used in the specification but there is nothing in that usage which bears on this argument. HTC’s submission has some support in claim 15. That is to a method of programming and is a generalisation of the process described in the patent where an application written in Java (a first programming language) is compiled to byte code (a first intermediate code) which is interpretable by a first intermediate code virtual machine (the JVM) and then the byte code is converted to a second intermediate code (the Card JVM byte code) which is interpretable by a second intermediate code virtual machine (the Card JVM). In the patent it is clear that the steps so far do not take place on the microcontroller. Nevertheless claim 15 is to a method of programming a microcontroller and the last step of the claim is “*loading the second intermediate code into the memory of the microcontroller*”. In its context in claim 15 this is talking about storing the code in the memory of the microcontroller, it is not talking about the point at which the code is executed. After all the claim is to a method of programming, not to a method of running a program. I am sure that in claim 15 the term “loading ... into the memory of the microcontroller” is referring to putting the code in a place where it can be kept in the memory of the microcontroller so that it can be run when needed.

108. Claim 1 is an independent claim and so not necessarily using language in the same sense but in my judgment it does. Claim 1 is essentially a claim to the product produced at the end of claim 15. It is not precisely so given some limitations in claim 15 which are not in claim 1 but no matter. The reader would understand that the loading in memory referred to in claim 1 is the loading into the memory of the microcontroller which takes place at the end of the process of creating a suitable application. That is the same sense in which loading is used in claim 15.
109. The memory referred to is the memory of the microcontroller chip, i.e. the on-chip memory. (I suppose if the microcontroller in question was one of the kinds which could also access off-chip memory as well then there is no reason why that should be excluded.) Accordingly “loaded in memory” means put into the memory of the microcontroller. That could be ROM or FLASH but it could also be RAM (as long as the power was kept on). It is not talking about the mechanics of the execution process and has nothing to do with the process of using cache memory which some processors employ. The act of copying some data from the system’s main memory into cache memory is not what the skilled team would understand the inventor to be using this language to mean.

(iv) converter

110. Claim 1 calls for a converter for post processing the compiled form of the application into a minimized form suitable for interpretation by the interpreter. The parties are agreed that this requires a reduction in size of the application program but does not require the attainment of the minimum possible size.
111. An issue arose about whether the word converter itself had a meaning which added anything in the context in which it was used in sub-paragraph (b) of claim 1. Does “*a converter for post processing the compiled form into a minimized form suitable for interpretation by the interpreter*” cover anything for post processing the compiled form into a minimized form suitable for interpretation by the interpreter or does that fact that the thing is *a converter* impose an extra limitation? This issue is best seen in the context in which it arises in relation to novelty over the Caron paper and I deal with it there. The conclusion I have reached is that the word converter adds nothing. A converter is simply the thing which does the required post processing.

(v) Claim 8

112. Claim 8 relates to the properties of the converter referred to in claim 1. It must comprise means for translating from byte codes in the compiled form to byte codes in a form suitable for interpretation by the interpreter. The claim has five sub-paragraphs (a) to (e). These define a process including all five steps, with step (a) at the start, steps (b) to (d) in the middle and step (e) at the end.
113. Steps (a) and (e) are counterparts. The skilled addressee would be familiar with what these steps set out to achieve. They are not really concerned with translation of byte codes themselves but with ensuring that the translated code still works after translation. Step (a) takes place first. All the jumps and their destinations are identified. Next some “conversion” or translation process takes place. This is likely to change the size of the code. Carrying out this conversion means that the jump destinations as they were before conversion would probably be wrong. Once the

- conversion process is finished, step (e) is needed. This uses the list made in step (a) and corrects or “relinks” all the jump destinations to make sure they work.
114. Step (b) encompasses two possibilities. Specific byte-codes can be converted into generic byte codes or (“vice versa”) specific byte codes can be converted into generic byte codes. The idea of replacing generic byte-codes with specific ones to make code smaller is part of the common general knowledge.
 115. The reason for replacing specific instructions with generic ones is as follows. The task of an interpreter program is to read the source code, identify the instructions contained in it and translate them appropriately. In effect the interpreter has to look up each instruction in a dictionary. The larger the instruction set the interpreter has to be able to read, the larger the dictionary. An interpreter which only has to deal with a smaller instruction set will have a smaller dictionary. It is likely therefore to be smaller and to run faster than an interpreter dealing with a larger instruction set. Thus as long as the instruction set includes a suitable generic instruction, further specific instructions could be replaced by the generic one, reducing the size of the dictionary and so reducing the size of the interpreter program and speeding it up.
 116. Step (c) relates to a process of modifying byte code operands from references using strings to references using unique identifiers. In the byte code the operands may be identified by strings. So an instruction to add a value representing the Value Added Tax onto a value representing the price of an item can be written in a way which is understandable: “ADD VAT to PRICE” instead of writing it using incomprehensible references to the relevant memory locations for the operands. However strings like VAT and PRICE take up space, generally one byte per character. Step (c) replaces the strings with “unique identifiers”. Although the claim is not limited to this, the unique identifiers can be simple numbers to identify the value concerned. In the example, if the unique identifiers can be numbers in a single byte, the eight bytes needed for VAT and PRICE are reduced to two bytes.
 117. The patent gives an example of this step (c) in the context of Java class files at paragraphs [0077] and [0078]. In fact in the Java class file the identifying strings are not stored with the byte code, they form a separate list called the constant pool and the operands actually stored with the byte codes are integers which refer to the entries in the constant pool. Nevertheless the space saving is real enough because the class files contain all the strings. If the strings are replaced by smaller “unique identifiers” then space is saved.
 118. Step (d) is about renumbering byte codes in the compiled form to equivalent codes in a format suitable for interpretation. The claim requires the renumbering to be into “equivalent” codes and a point arises about that which it is convenient to address in the infringement section. One reason why renumbering of byte codes might be worthwhile is that it can allow the interpreter program to be more efficient by grouping instructions with similar operands together. The grouping means that fewer lines of code in the interpreter program are needed to process the instructions. This is described in the patent at paragraph [0087].
 119. The final step (e) in claim 8 was addressed with step (a) above. Steps (a) and (e) are clearly necessary in order to ensure that the effect on the size of the code of carrying out steps (b), (c) and/or (d) is handled appropriately.

120. Claim 8 requires the converter to perform at least one of the steps in this process.
121. There was a debate about the dependency of claim 8. Claim 8 is expressed to be dependent on claims 4, 5, 6 or 7. These claims ultimately all refer back to claim 1 and there is no doubt claim 8 itself therefore depends on claim 1. Gemalto submitted that claim 8 necessarily also depends on and therefore incorporates all the features of claim 4. This is an error. Although claims 5 and 6 refer back to claim 4, claim 7 does not. Hence claim 8 may incorporate claim 4 (if read through claims 4, 5, 6 or those parts of claim 7 reading through claims 4, 5, or 6) but it need not if it incorporates claim 7. For the purposes of this case the claim can be considered as two claims: claim 8 plus 4 plus 1; and claim 8 plus 7 plus 1.
122. Claim 4 calls for:
- The microcontroller (10) of any of the preceding claims wherein the compiled form (24) includes associating an identifying string for objects, classes, fields, or methods, and the converter comprises a means for (57) mapping such strings to unique identifiers (51 b).
123. This requires the compiled form of code to use identifying strings for things like classes, fields or methods and for the converter to map such strings to unique identifiers. This relates to the same idea as step (c) of claim 8 and, although the point probably does not matter, it is hard to believe that something performing step 8(c) would not also have to have the features of claim 4.
124. Claim 7 calls for:
- The microcontroller (10) of any of the preceding claims where in the high level language supports a first set of features and a first set of data types and the interpreter (16) supports a subset of the first set of features and a subset of the first set of data types, and wherein the converter (26) verifies (51c, 52) that the compiled form (24) only contains features in the subset of the first set of features and only contains data types in the subset of the first set of data types.
125. Neither side focussed on this claim at all. It requires sub-setting and also requires verification by the converter that the code does not include instructions outside the subset. Thus in claim 8 plus 7 plus 1, the converter has to perform the verification mentioned in claim 7 as well and the interpreter must only support a subset of the high level language features and data types.

865: Priority

126. HTC contends that the claims have lost priority on two grounds. The first relates to the derivation of the priority right. The second relates to the claims themselves and how they stand vis a vis the disclosure of the priority document. Gemalto denies both points. The significance of this argument is that in the priority period Schlumberger published the Cyberflex material which describes some details of what had been invented.

Priority – the law

127. Section 5(2)(a) of the 1977 Act provides that:

5(2). If in or in connection with an application for a patent (the application in suit) a declaration is made, whether by the applicant or any predecessor in title of his, complying with the relevant requirements of rules and specifying one or more earlier applications for the purposes of this section *made by the applicant or a predecessor in title of his* and the application in suit has a date of filing during the period allowed under subsection (2A) or (b) below, then –

(a) if an invention to which the application in suit relates is *supported by matter disclosed in the earlier relevant application or applications*, the priority date of that invention shall instead of being the date of filing the application in suit be the date of filing the relevant application in which the matter was disclosed or, if it was disclosed in more than one relevant application, the earliest of them;

(emphasis added)

128. The first italicised portion relates to HTC’s first point (“priority entitlement”), the second italicised portion relates to HTC’s second point (“substantive priority”).

129. Section 5 is one of the sections in the 1977 Act covered by s130(7) and so is meant to have the same effect as the corresponding provisions in the EPC and PCT. In this case these are Arts 87 – 89 EPC and Art 8 PCT. Art 8 PCT in turn refers to the Paris Convention. The Paris Convention (Art 4) provides that a person who files a patent application is entitled to a right of priority which they can use in any state of the Paris Union.

Priority entitlement

130. Kitchin J as he then was summarised the position in *Cook v Edwards* [2009] FSR 27 at paragraph 95:

“In my judgment the effect of Article 4 of the Paris Convention and section 5 of the Act is clear. A person who files a patent application for an invention is afforded the privilege of claiming priority only if he himself filed the earlier application from which priority is claimed or if he is the successor in title to the person who filed that earlier application. If he is neither the person who filed the earlier application nor his successor in title then he is denied the privilege. Moreover, his position is not improved if he subsequently acquires title to the invention. It remains the case that he was not entitled to the privilege when he filed the later application and made his claim. Any

other interpretation would introduce uncertainty and the risk of unfairness to third parties.”

131. At paragraph 99 of *Cook v Edwards* Kitchin J held that the right to claim priority belonged jointly to all those who had filed the priority document (or their successors in title) and that in a case in which the priority document was filed by more than one person, it was not sufficient that the applicant for the patent was only one of the original applicants who had filed the priority document (or its successor). The patent applicant has to derive title from all of them.
132. Gemalto did not challenge this and I will take it therefore that the applicant in the UK has to be a successor in title to the invention at the point in time when he files his UK application; a later acquisition of title to the invention is not enough. He must also be the successor in title of all the applicants.
133. Gemalto referred to the judgment of Arnold J in *KCI v Smith & Nephew* [2010] EWHC 1487 (Pat). In that case Mr Lina was the inventor and, in accordance with US law, he was the sole applicant of a US application which was the priority document in the case. On the international application under the PCT, KC Inc was named as the applicant. Thus the question was whether KC Inc had the right to claim priority. KC Inc was said to be Mr Lina’s successor in title by virtue of the terms of an agreement between KC Inc and Mr Lina whereby, in consideration of his employment with KC Inc, Mr Lina agreed to assign all (relevant) inventions to KC Inc. Although the proper law of the agreement was Texas law, no evidence of Texas law was filed and it was common ground English law should be assumed to be the same. Arnold J held that s7 of the 1977 Act proceeded on the basis that it was possible to assign the legal title in an invention before it was made and that as a result the agreement was effective to assign the legal title to the invention to KC Inc and thus KC Inc were, at the date of the PCT application, Mr Lina’s successor in title (paragraphs 55-68). Arnold J went on to say:

“69. I would add that, even if it was not effective to convey the legal title to the invention, para.3 of the Confidentiality Agreement was plainly effective to transfer the entire beneficial interest in the invention, including the right to file patent applications in respect of it, from Mr Lina to KC Inc. KC Inc would have been entitled to demand that Mr Lina convey the bare legal title to the invention to itself at any time, and to compel Mr Lina to do so if he failed or refused to do it. If necessary, I would hold that that was sufficient to make KC Inc Mr Lina's “successor in title” for the purposes of a claim to priority under art.87(1) of the EPC and art.4(A)(1) of the Paris Convention even if KC Inc had not acquired the bare legal title at the relevant date.

70 I am encouraged so to hold by the decision of the Legal Board of Appeal in Case J19/87 *Burr-Brown/Assignment* [1988] E.P.O.R. 350 that an assignment of an invention and a patent application from A to B with a covenant of further assurance was sufficient to entitle B to claim priority from an application filed by A even though the assignment of the patent

application was ineffective because it was not signed by B contrary to s.30(6) of the 1977 Act as it then stood. In holding that the priority claim was a good one, the Board (two of whose members were Peter Ford, later H.H. Judge Ford, and Gerald Paterson, later the author of The European Patent System) accepted an opinion from English counsel (Nicholas Pumfrey, later Pumfrey J.) stating that (i) the assignment of the invention (which post-dated the making of the invention) was effective in law even though the assignment of the patent application was not, and (ii) although the assignment was ineffective in law B had acquired an equitable interest in the patent application which was a proprietary interest. Although it could well be argued that point (i) was enough, the Board seems to have regarded point (ii) as significant as well.

71 To my mind, this makes sense. Article 4(A) of the Paris Convention and art.87(1) of the EPC are provisions in international treaties whose operation cannot depend upon the distinction drawn by English law, but not most other laws, between legal and equitable title. When determining whether a person is a "successor in title" for the purposes of the provisions, it must be the substantive rights of that person, and not his compliance with legal formalities, that matter".

134. Mr Mellor submitted that this showed that as long as an applicant had, at the relevant date, what English law would characterise as a beneficial title to the invention, even if the bare legal title had not been acquired, then the applicant was a successor in title in the relevant sense. I did not understand Mr Tappin to dispute that and I think he was right not to. In my judgment if the relevant person has acquired the entire beneficial interest in the invention at the relevant time then that should be enough to satisfy the law.
135. Mr Mellor did develop his argument by referring to constructive trusts, although he did not cite any authority relating to them. Mr Tappin did not accept that the English law of constructive trusts could make someone a successor in title within the meaning of the Paris Convention/EPC. I do not propose to resolve that question since it was not fully argued before me.
136. A second aspect of Arnold J's judgment in KCI v Smith & Nephew on which Gemalto relied was paragraph 98. This was the conclusion of a section in which Arnold J considered what would happen if another company, Mediscus, was a co-applicant with KC Inc. He said:

"98 Counsel for KCI accepted that he could not point to any written assignment, or even an oral agreement, but argued that the correct inference to be drawn from the circumstances surrounding the filing of the PCT Application was that KC Inc had agreed by conduct to transfer part of its interest in the invention to its subsidiary Mediscus. He submitted that this was sufficient to make Mediscus a successor in title for the purposes

of claiming priority, and that no greater degree of formality was required. I accept that submission.

99 For these reasons I conclude that it would not adversely affect the claim to priority if Mediscus was held to be a co-applicant”.

137. The only point I understood Mr Mellor was seeking to make here was to show that it was open to the court to infer from the circumstances that one party had agreed by conduct to transfer to another party part of its interest in the invention. That is unexceptional. Whether in a given case such an inference is to be drawn depends on the facts.

Priority entitlement – the facts

138. The priority document (US 60/029,057) was filed on 25 October 1996 in the names of Mr Wilkinson, Mr Krishna and Mr Guthery. The PCT application PCT/US97/18999 was filed on 22 October 1997 by Schlumberger Technologies Inc. (STI). For the priority entitlement to be established, STI has to have been the successor in title at the date the PCT application was filed. The circumstances in which the invention was made and the documents were filed are set out in M. Piou’s witness statement.
139. The parent company of the Schlumberger group is Schlumberger Ltd. Although the group began in the oil business, it diversified. One area into which it diversified was smart cards. In the early 1990s the Schlumberger group had two divisions: “Oilfield Services” and “Measurement and Systems”. A sub-division called Electronic Transactions covered smart cards. It was a sub-division of Measurement and Systems. A business plan for the project which led to the invention (called project OSMAN) was dated 7 July 1995. The document is in the name of “Schlumberger ET/Cards division”. It was decided to undertake the project in the USA. The group company which operated Schlumberger’s smart card business in the USA was STI. However STI lacked the infrastructure required for the project. The decision was made to use a facility in Austin, Texas. This was a facility owned by Schlumberger’s Oilfield Services division which operated through Schlumberger Technology Corporation (STC). The key work which led to the invention took place between spring and autumn 1996. Shortly after filing the US priority document Schlumberger announced its new smart card, Cyberflex.
140. HTC accepted that STI did acquire the rights from Mr Krishna before the PCT was filed and so no issue arises in relation to those. The question concerns Mr Guthery and Mr Wilkinson. Their position was the following. Scott Guthery was an existing employee of Schlumberger; he worked at the Austin facility and was employed by STC. Tim Wilkinson was retained on the Schlumberger smart card project as an independent contractor. He signed three agreements in the relevant period before Oct 1997. They were entered into with STC. The earliest was signed in April 1996. There is no evidence he did anything before that. It contains at clauses 5 and 6 an assignment of any inventions which arise from his work. The precise terms of this assignment do not matter.
141. There is an assignment, signed by both Mr Guthery and Mr Wilkinson of their rights in the invention to STI but it is dated 3rd November 1997 and is after the relevant date.

142. M. Piou says he does not understand why it would have been necessary for Mr Guthery and Mr Wilkinson to assign anything to STI prior to filing the PCT application because the smart card project was conceived, managed, budgeted, developed and exploited by Schlumberger's Electronic Transactions/ Cards division. M. Piou explains why this is so in detail. I should note that in making this point M. Piou is equating Schlumberger's Electronic ET/ Cards division with STI as far as the USA is concerned. It is not necessary to set out all the points M. Piou relies on since his evidence was not challenged. An example is that when Schlumberger needed a licence from Sun Microsystems, the originators of Java, to use the Java trade mark, the agreement was between Sun and STI. Finally M. Piou states that no additional payment was made to Mr Guthery or Mr Wilkinson for the November 1997 assignment.

Assessment

143. Working backwards, I am satisfied that as between STC and STI, STI was the beneficial owner of any inventions arising from the project. There is no formal agreement to that effect but I infer from the circumstances set out by M. Piou that STC must have agreed by conduct with STI to transfer any interest it had in the invention to STI. There is no credible reason why STC should have wanted to retain ownership of the invention at all. STI was entitled to call for the assignment of the bare legal title from STC and for that matter to compel STC to itself to exercise any rights it had to acquire that bare legal title if all it had was a beneficial title.
144. As regards Mr Wilkinson, I find that the effect of the consultancy agreements was that, as between himself and STC, STC was the legal owner of any relevant invention by the time the PCT was filed. That is the same conclusion, for the same reason, as reached by Arnold J in *KCI v Smith & Nephew* at paragraph 68. Equally, for the same reasons Arnold J gives in paragraphs 69 to 72, even if STC did not acquire the legal title from Mr Wilkinson as a result of the invention, they had the beneficial title. Putting this conclusion together with the previous one, I find that at the relevant time STI was the successor in title to Mr Wilkinson.
145. As regards Mr Guthery, no formal employment contract survives but Gemalto submitted that it was plain that an invention he was employed to devise must belong to his employer, STC in law. Thus, for the same reasons as above, the STI can derive title via STC and so STI were a successor in title to Mr Guthery as well.
146. However HTC contended that one could not assume that as between employer and employee, the invention must belong to the employer. After all in US practice, the priority document had to be filed in the name of the individual employee. HTC argued that although in English law section 39 of the 1977 Act provides that an invention made by an employee in this way would belong to his employer (save for certain exceptions which do not, or to be more exact do not appear to, apply), that section did not apply because, by s43(2) of the 1977 Act, sections 39 to 42 do not apply unless, at the time the invention was made, certain criteria are satisfied, both of which require a connection to the UK (employee or employer) and neither of which is satisfied. So, Mr Tappin argued, as a matter of English law s 7 of the 1977 Act applies but s39 does not. Thus the right to be granted a patent belongs primarily to the inventor (s7(2)(a)). He also said that HTC had sought to include US law materials

in the bundles for this case but Gemalto had refused on the ground that no US law was pleaded and so English law applied or should be presumed to apply.

147. This argument was developed in a reply to a reply during closing. I permitted Gemalto to submit a note on the point after the trial. Gemalto's counsel (Mr Burkill and Mr Copeland) submitted that this argument about s39, s43(2) and a lack of evidence of US law should not be accepted. They pointed out that neither side had led any evidence of US law and that the court should and can only proceed on the assumption that the applicable law is the same as English law. Section 43(2) contains limitations on the application of s39 to non-UK employees but that is on the obvious basis that foreign law, if proved, would apply instead. Absent any evidence that a different provision applies, the presumption means that a provision equivalent to s39 should be assumed.
148. Although I am not convinced either side has fully thought through its submissions on this issue I need to decide the issues on the material before me. In order to decide the relative rights to an invention as between the inventor and his employer when the person was working in Texas and was employed by an American company, I must apply US/Texan law. Gemalto's counsel are right that the reason s43(2) of the 1977 Act disapplies s39 cannot be to create a different English law of the ownership of foreign inventions, the section is disapplied because as a matter of English law, the correct law to apply to that question is the local foreign law. Since I have no evidence of the applicable US/Texan law the only basis on which I can decide this issue is to assume US/Texan law is the same as what English law would be applied to events which took place in England. On that basis section 39 would have the effect that the invention belonged to STC. In taking this approach, as Gemalto's counsel submitted, I am not applying s39 to an employee in the USA, I am presuming that the corresponding law is the same and applying that.
149. Accordingly I find that STI were a successor in title to all three of Mr Wilkinson, Mr Guthery and Mr Krishna when the PCT application was filed and thus were entitled to claim priority.

Substantive priority

150. HTC mounted a wholesale attack on the substantive priority of the claims in the 865 patent. Three arguments are advanced in relation to claim 1:

High level language

- i) The point pleaded is:

“The priority document does not clearly and unambiguously disclose (in an enabling manner or at all) the use of any high-level programming language (or any compiler, converter or interpreter therefore) other than Java. The priority document accordingly does not support the breadth of claim 1.”

Microcontroller

- ii) The point pleaded is:

“The priority document does not clearly and unambiguously disclose any device (or the programming of any device) other than a microcontroller, as defined at page 1 lines 22-28. If the defendants contend that claim 1 is to be construed as extending to a device other than a device of the type defined at page 1 lines 22-28 then the priority document does not support a claim of such breadth.”

Converter etc.

iii) The point pleaded is:

“The priority document does not clearly and unambiguously disclose “a converter for post processing the compiled form [of the application] into a minimised form”. In particular:

[...]

(b) The reordering of bytecodes described in section 2 on page 6 (and at page 5 lines 27-28) and shown in Figure 1 does not reduce the size of the compiled form of the application.

(c) Section 3 on page 7 (and page 5 lines 15-16) does not contain an unambiguous (or enabling) disclosure of any converter for post processing the compiled form of the application into a minimised form.

(d) Page 5 lines 35-38 refers to the minimisation of the size of the interpreter, not to the minimisation of the compiled form of the application.

Even if (which is denied) the priority document discloses any “converter for post processing the compiled form [of the application] into a minimised form”, it is only a converter which performs the processes referred to in sub paragraphs 1.3 (b)-(d) above, and the priority document does not support a claim of the breadth of the claim 1 which refers to any “converter for post processing the compiled form [of the application] into a minimised form”.

151. The microcontroller point (ii) and the converter point (iii) are both said to apply to all the allegedly independently valid claims. The high level language point (i) applies to claims 1, 8, 9, 15 and 18. Gemalto denies all these points.
152. A separate point was taken on claim 8. The pleaded case was that sub-paragraphs (a), (b) and (e) of claim 8 were not entitled to priority. This was accepted by Gemalto. Although the point did not emerge clearly until the end, by the close Gemalto’s case was that claim 8 was entitled to multiple priorities, with sub-paragraphs (c) and (d) being entitled to maintain the priority date even if (a), (b) and (e) were only entitled to the filing date. HTC did not agree that multiple priorities could be assigned in this way for this claim and did not accept in any event that sub-paragraphs (c) and (d)

were entitled to the priority date. The same arguments applied to claim 18, which has the same sub-paragraphs as claim 8.

153. A point was pleaded in relation to claim 12 (HTC Statement of case on Priority paragraph 5) but it was not mentioned at trial and I will not address it.

Substantive priority - the law

154. Kitchin LJ (with whom Lewison and Moore-Bick LJJ agreed) summarised the law of substantive priority in *MedImmune Ltd v Novartis Pharmaceuticals Ltd* [2012] EWCA Civ 1234 at paragraphs 151 – 154 as follows:

151. Section 5(2)-(a) of the Patents Act 1977 provides that an invention is entitled to priority if it is supported by matter disclosed in the priority document. By section 130(7) of the Act, section 5 is to be interpreted as having the same effect as the corresponding provisions of Article 87(1) of the European Patent Convention. Article 87(1) says that priority may be derived from an earlier application in respect of the “same invention.

152. The requirement that the earlier application must be in respect of the same invention was explained by the enlarged Board of Appeal of the EPO in G02/98 *Same Invention*, [2001] OJ EPO 413; [2002] EPOR 167:

"The requirement for claiming priority of 'the same invention', referred to in Article 87(1) EPC, means that priority of a previous application in respect of a claim in a European patent application in accordance with Article 88 EPC is to be acknowledged only if the skilled person can derive the subject-matter of the claim directly and unambiguously, using common general knowledge, from the previous application as a whole."

153. The approach to be adopted was elaborated by this court in *Unilin Beheer v Berry Floor* [2004] EWCA (Civ) 1021; [2005] FSR 6 at [48]:

"48.The approach is not formulaic: priority is a question about technical disclosure, explicit or implicit. Is there enough in the priority document to give the skilled man essentially the same information as forms the subject of the claim and enables him to work the invention in accordance with that claim."

154. In *Abbott Laboratories Ltd v Evysio Medical Devices plc* [2008] EWHC 800 (Pat), I added this:

"228. So the important thing is not the consistency clause or the claims of the priority document but whether the disclosure as a whole is enabling and effectively gives the skilled person what is in the claim whose priority is in question. I would add

that it must "give" it directly and unambiguously. It is not sufficient that it may be an obvious development of what is disclosed."

155. Mr Tappin referred to paragraph 106 of the judgment of Floyd J (as he then was) in *Samsung Electronics Co Ltd v Apple Retail UK Ltd* [2013] EWHC 467 in which the court's task was summarised as being:
- (a) to read and understand, through the eyes of the skilled person, the disclosure of the priority document as a whole;
 - (b) to determine the subject matter of the relevant claim;
 - (c) to decide whether, as a matter of substance not of form, the subject matter of the relevant claim can be derived directly and unambiguously from the disclosure of the priority document.
156. This formulation corresponds to the approach to deciding questions of added matter set out by Aldous J (as he then was) in *Bonzel v Intervention* [1991] RPC 553 at 574, albeit that instead of starting with the application as filed for added matter, for priority one starts with the priority document. Nevertheless I do not understand Floyd J to have been holding that the test for priority is the same as the test for added matter with the only difference being that different documents are being compared.
157. In some ways it is easy to say what the test for priority is not. It is not a novelty test or an infringement test. As Floyd J said in the next paragraph of his judgment in *Samsung*, the test for claiming priority in respect of the same invention has more substance, and is less formal, than that. It is certainly not an obviousness test (para 228 of *Abbott* above). One respect in which it is clear that the test for priority is not or at least not only an added matter test is in relation to enablement. Enablement is certainly one of the requirements for a successful claim to priority and this applies both in the sense that what is specifically disclosed must actually be enabled (*Asahi* [1991] RPC 485) and also in the sense described by Lord Hoffmann in *Biogen* [1997] RPC 1 when he held that the claim in question lost priority because it covered other ways of achieving the goal which owed nothing to the principle disclosed. Nevertheless as Kitchin J said at first instance in *Abbott v Evvysio* at paragraph 226, enablement is not the only requirement, the general and overriding requirement is that set out by the Enlarged Board in *G2/98* (cited above).
158. Arguments about added matter can have a tendency to descend into a formulaic comparative analysis of the claims of the granted patent as compared to the claims of the application. It is clear that the priority test is not formulaic (paragraph 48 of the Court of Appeal in *Unilin* quoted above). The Court in *Unilin* went on in paragraphs 49-50 to say:
- "49. Before going to the details of the priority document in this case I should deal with Mr Carr's submission about the main claim or consistory clause of the priority document, *i.e.* that although not determinative it is nearly so. That he could not get out of *02/98* or indeed any other authority. *02/98* refers to "the previous application as a whole," not the main claim nor the

“main statement of invention” nor the “consistory clause”. Likewise there is nothing in Art.87 which compels or suggests this conclusion. And Art.4H of the Paris Convention is positively against it. The claims (if any—there is no rule that there should be) of the priority document are not determinative. They are just part of its disclosure. For the purposes of priority one just looks at the disclosure as a whole.

50. If the rule were otherwise one of the main functions of a priority document would be lost. Inventors and their advisors would have to start worrying not only about the technical information disclosed in the document but how it was to be claimed: have I drafted my main claim or consistory clause broadly enough? That is not the purpose of the system: the purpose at this point is to get the information justifying the later claim into a patent office of a Union country. If you do that, you can have your priority, whether you express that in a proposed claim, consistory clause, statement of invention, other text or drawing or in any combination of these. Time is of the essence because the world-wide system (except for the Americans) works on the first to file basis. The detailed framing of a claim based on that information may then be done within the Convention year.”

159. The English authorities emphasise that the “same invention” test for priority is one of substance not form. They also show in my judgment that entitlement to priority is a different question from the other familiar elements of patent validity: sufficiency of disclosure, novelty, inventive step and added matter. It is its own test which arises in its own context and with its own policy considerations. It contains aspects which are common to other areas. The test for enablement must be the same as in other contexts and the test for disclosure (clearly and unambiguously derivable, explicitly or implicitly) must be the same whether one is looking at a priority document for priority or an application for added matter. However I agree with Mr Tappin’s submission that as between priority and added matter, one is looking at the disclosure for a different purpose.
160. A question arising in relation to claim 8 is about what are sometimes called partial or multiple priorities. The idea that one claim can be entitled to multiple priorities is enshrined in Art 88 (2) EPC. It is permitted “where appropriate”. In G2/98 the EPO Enlarged Board held that multiple priorities in a single claim were possible as long as the claim related to a limited number of clearly defined alternative subject-matters. Although one can sympathise with the desire for a limited number, I doubt there is any principled basis for such a requirement but I accept the need for clearly defined alternative subject-matters if a single claim is to be given multiple or partial priorities.
161. The relevant part of G2/98 was considered by Kitchin J in Novartis v Johnson & Johnson [2009] EWHC 1671 (Pat) at para 122:
- “I discern from this passage that the EPO considers it is permissible to afford different priority dates to different parts of a patent claim where those parts represent a limited number of

clearly defined alternative subject-matters and those alternative subject-matters have been disclosed (and are enabled) by different priority documents. Further, this principle applies even if the claim has adopted a generic term to describe and encompass those alternatives. I do not detect anything in the decisions of the Court of Appeal in *Pharmacia* and *Unilin Beheer* which is inconsistent with this approach and in my judgment is one which this court should adopt."

162. This has a bearing on another aspect of HTC's case. Part of HTC's case on priority included a submission that because a dependent claim (in fact claim 7) was not entitled to priority, but is encompassed within claim 1, it follows that claim 1 cannot be entitled to priority either. Gemalto submitted this was wrong in law because otherwise an opponent, by citing some inventive improvement not disclosed in the priority document but merely covered by a granted claim, would automatically knock out the priority of the claim, a proposition Gemalto argued only had to be stated to be rejected. I agree with Gemalto up to a point but care is needed not to state things too broadly. If a specific feature is not disclosed in the priority document then a claim that expressly refers to that feature must lose priority. However the fact that a broad claim will cover something (among other things) is irrelevant to priority as long as the priority document supports the broad claim at its level of generality. That conclusion is at the heart of the reason of the House of Lords in *Biogen*. If the broad claim covers things which owe nothing to the disclosure of the priority document then the priority document will not support the broad claim and priority will be lost, as happened in *Biogen* on its facts, but that is a different matter. I do not believe Kitchin J was seeking to say anything different. All the passage in paragraph 122 means is that if the claim is one for which partial or multiple priorities are to be ascribed, the use of generic wording does not prevent that from happening.
163. Mr Tappin referred me to the judgment of Arnold J in *Nestec v Dualit* [2013] EWHC 923 (Pat). At paragraphs 90 to 94 Arnold J dealt with the law on priority, citing *G2/98*, *Unilin*, *Abbott v Evysio* and in particular paragraph 122 from Kitchin J's judgment in *Novartis* I have set out. As I understood Mr Tappin's submission, the way this was applied in *Nestec* (particularly paragraph 96) showed that the analysis applicable to the assignment of multiple or partial priorities necessarily applied to any generic terms in a claim. Thus he submitted that unless you can have partial priority for different alternatives within the generic claim, the claim loses priority. I do not accept that submission as a proposition of law. It is inconsistent with *Biogen*. I also note that the factual situation dealt with by Arnold J in *Nestec* is very different from the one I am faced with here and that Arnold J did not there state the law in the way stated by Mr Tappin which I have rejected.
164. Finally I need to mention the role of experts on this issue. Once the court has been educated in order to assume the mantle of the skilled person, the role of experts in this area is a limited one. Expert evidence related to enablement is relevant but the experts' reports in this case went considerably further than that, no doubt because the legal teams asked their experts to do so. An awful lot of that material was not useful.

The priority document

165. The priority document is in a number of parts. Pages 1 to 17 are drafted like a conventional patent application. The paragraph above is headed “Technical Field” and is followed by a background section and a section which gives a description of a preferred embodiment. At page 18 a different part starts, headed “High Level Design” and “Functional Specifications”. This section appears to be a specification document such as might be used in a computer system development project. The text focuses on the Card Application Programming Interface. At the end (page 27) there are incomplete specifications for the Terminal Application Programming Interface, the Development Environment and the User Interface. The third part of the priority document starts on page 28. It is headed “Specification of the JavaCard Virtual Machine and Application Program Interface”. This part defines the JavaCard virtual machine in various ways (including a list of bytecodes) and discusses JavaCard security. At the end of the text there is a paragraph such as one commonly finds in patent specifications which states that what is disclosed is illustrative rather than restrictive but is so general as to be meaningless. Finally there is a set of five figures, none of which found their way into the granted patent specification. They are referred to in the first part. There are no claims in the priority document.
166. The priority document as a whole discloses the idea of executing programs written in the Java programming language on a microcontroller (pace the debate about the meaning of microcontroller) and discloses using a virtual machine on the microcontroller to do so. It explains that due to the severe resource constraints of “today’s smart card single-chip computers” it is impossible to fit the current smallest Java environment on a smart card and describes “a yet-smaller version of the Java virtual machine and runtime environment which is specifically tailored for use on smart cards”. In some places in the document this is called a JavaCard virtual machine. The extent to which the priority document discloses a converter is controversial and I will deal with it below.

(i) High level language

167. The problem here is a simple one. Claim 1 is explicit that the application source programs must have been “*in a high level language source code form*”. The claim is therefore not limited to application source programs written in Java, but expressly encompasses application source programs written in any high level language.
168. By contrast the priority document from beginning to end is concerned with Java. It is entitled “Implementation of the Java programming language for microcontrollers”. The first paragraph reads:
- This invention relates in general to the field of programming, and more particularly to a method of programming a microcontroller using a high level programming language called “Java,” and most particularly with programming a central processing unit of such a microcontroller installed in an integrated circuit card or “smart card”.
169. Throughout the document only the Java language is mentioned. The document occasionally makes the point that Java is a high level programming language (see e.g.

the first paragraph above) but that is not the same as a teaching that what is being described in the priority document is applicable to high level languages generally and not just Java.

170. In the course of the trial various sentences and half sentences were focussed upon to seek to extract a shade of meaning which might be said to disclose the idea of using any high level language. The high point of that argument related to a sentence on p19 of the priority document in a section 2.1.3 which was concerned with the “General Purpose Operating System” application architecture. The sentence reads “*The GPOS will be programmable in a high level language; specifically Java.*” and it was argued that this disclosed the idea of using a high level language in general and not only using Java. By closing however Gemalto sensibly abandoned trying to squeeze such a disclosure from this sentence. The sentence says no more than the first paragraph. Moreover this particular sentence could not help Gemalto anyway since it is not talking about the source code of the application, it is talking about how to program the operating system itself.
171. Mr Mellor submitted that the non-Java claims were still entitled to priority for the following reason. If the skilled person thinks about the disclosure, it is a method of programming. The skilled person would see that it is described in relation to Java but, it is submitted, he would understand the disclosure to apply to anything, i.e. any high level language. Mr Mellor also emphasised this argument in the context of claim 15, which is a claim to a method of programming. Like claim 1 it is not limited to Java but unlike claim 1 the words used are not “high level language” but simply “first programming language”.
172. Attractively put though it was, I am unable to accept Mr Mellor’s argument. In my judgment the disclosure of the priority document both expressly and by implication, is clear and is all one way: the invention involves and requires the use of Java. That is true whether one looks at the invention as a microcontroller set up in a particular way or as a method of programming a microcontroller. Nowhere do the inventors suggest or even hint that what is described in the priority document may be applicable more generally than to Java. An invention in which the application source code is not written in Java is not the same invention as the one disclosed in the priority document. I conclude that claim 1 is not entitled to priority from the priority document. Claim 3, which is limited to Java, is unaffected by this point. Claims 8 and 9 insofar as they depend on claim 3 are unaffected but otherwise they lose priority. Claim 15 is also not entitled to priority and neither is claim 18.
173. From now on I will consider the priority of claims limited to Java, i.e. to claims 1 and 3 combined and claim 8 insofar as it is dependent on claims 1 and 3.

(ii) microcontroller

174. The argument about microcontroller does not arise on my finding on construction of the term. In any case it is manifest that what the priority document is talking about is what I have called the normal meaning of the term. Page 1 In 22 of the priority document states unequivocally that:

A microcontroller comprises a central processing unit, memory and other functional elements on a single chip. Typical

microcontrollers have one to thirty-two kilobytes of memory. with four kilobytes being very common. Only 512 bytes of this memory is RAM memory; the remainder is ROM and EEPROM memory. The latter performs read functions easily but is much more difficult to write to than is RAM memory. One of the uses of microcontrollers is for integrated Circuit cards.

175. For what it may be worth, I note that the priority document does not refer in terms to a microcontroller of the kind which also accesses off-chip memory (the extended version of the normal meaning of microcontroller) but I believe nothing turns on this. The priority document is clear that a microcontroller is a chip with on-chip memory. Whether or not it can also access off-chip memory is not relevant. There is no difference in the disclosure between the priority document and the patent here. The invention is the same in both.

(iii) Converter

176. HTC's argument has a number of strands. HTC points out that the job of the converter is to produce a minimized form of the application. In other words it is not concerned with minimizing the size of the interpreter program, it is concerned with minimizing the size of the application program. HTC then argues first that the priority document discloses no such converter at all, or at least provides no unambiguous or enabling disclosure of any such thing. Second HTC focuses on a disclosure of a process of reordering byte codes in the priority document. HTC submits that this does not reduce the size of the application, it reduces the size of the interpreter and so is irrelevant. Third HTC submits that section 3 on page 7 of the priority document, which relates to a namespace map, is not enabling.
177. It is clear that a post processor or converter of some kind, which acts upon the bytecodes of the application program, is disclosed in the priority document albeit those words are not used. For example page 5 in 32-38 clearly describes retranslating the bytecodes of the application into a secondary form. It is true that the objective of the particular retranslation is so that the size of the interpreter needed to interpret this secondary form is minimized but that does not matter. The concept of something acting on the compiled form of a Java application program is disclosed here.
178. At page 5 a number of objects of the invention are set out. Line 13 refers to a program occupying a very small amount of space but it is clear that this is another reference to the interpreter program not the application. However at line 15-16 the priority document states:

It is also the object of this invention to compact an application by use of a pre-defined namespace map to refer to functions existing on the card.

179. This is a clear disclosure of the idea of compacting an application, i.e. making it smaller. Nothing turns on the fact that claim 1 refers to a "minimized" rather than a "compact" form and nothing turns on the absence of the word "converter". This is another example of the idea of a thing (call it a converter or post processor) acting on the compiled form of a Java application. Thus to say that the priority document does

not disclose the idea of something which makes the application smaller is wrong. Moreover for reasons which I will address below in relation to element (c) of claim 8, I find that the disclosure includes the idea that the converter or post processor can operate off-card, before the application is loaded into the card memory.

180. This leaves two points outstanding. If the disclosure is limited to achieving compaction by using the namespace map then that may not be enough to support a claim of the breadth of claim 1, which is not so limited. Furthermore HTC says the disclosure of the namespace map method is not enabling. If that is right then there is no enabling disclosure of any particular method of making the application smaller.
181. Sections 2 and 3 of the first part of the priority document relate to post processing the Java class files for the same purpose - time and space efficiency. Section 2 on page 6 describes "Reordering of Bytecodes for Time and Space Efficiency" and section 3 on page 7 describes "Mapping of Names for Time and Space Efficiency". The namespace map is addressed in section 3. I can set it out in full:

"The namespace is managed by use of a map. Java identifies all objects, classes, fields and methods by using textual strings. This takes far too much space and is unsuitable for the secondary form that will reside on the card. The preferred embodiment of the present invention therefore translates all textual references for these objects into unsigned integers (the size of which depends on the architecture of the microprocessor on the card). Each integer uniquely identifies a particular textual reference in the application. It is necessary to embed parts of the compacted system into the card when it is manufactured while allowing newly loaded applications to make calls to it."

182. Dr Greaves' opinion was that it was very hard to make sense of this paragraph. He thought that the reference to translating all textual references into unsigned integers would be understood as nothing more than the usual consolidation of string tables undertaken by a link editor. He thought that even if it might seem that the paragraph was teaching the skilled reader to leave the strings off the card, this was contradicted by the last sentence. He concluded that if this section was meant to disclose a detailed post processing step it was very far from doing so clearly or unambiguously.
183. In cross-examination Mr Mellor took Dr Greaves through the paragraph with care. As a result of the cross-examination I am not satisfied that Dr Greaves' approach to this paragraph reflects the approach a skilled reader of the priority document as a whole would take. For example when section 3 is read in the context of the priority document as a whole, it would be clear to the skilled reader that it is supposed to lead to compacting the application program specifically (see p5 line 15 of the priority document). That is important. Also Dr Greaves' approach did not take full account of the words "all" and "unique" in the paragraph. As Prof Paradinas explained, if *all* text strings are translated then this must include the text strings in the constant pool of the class file. Replacing these will undoubtedly reduce the space needed on the card for the secondary form of the program. Moreover there will usually be more than one class file for an application and this process will therefore necessarily eliminate duplicates of textual strings in class files since, as Prof Paradinas explained, the same

string may appear in each class file. When this is understood the last sentence makes sense. The predefined namespace map is embedded onto the card when manufactured thus allowing newly loaded programs to make calls to it. As Gemalto submitted, this confirms that the secondary form of the application program which resides on the card has been made smaller. In other words something (call it a converter) has converted the compiled form of a Java application into a secondary form and made the form which goes onto the card smaller. I reject HTC's case that this section of the priority document is non-enabling.

184. I now need to decide whether the general idea of a converter for post processing the compiled form of an application in standard Java class file format into a minimized form suitable for interpretation by the interpreter is entitled to the priority of the priority document. The issue turns on a fine but important distinction. Does the priority document disclose the idea of compacting in general with the namespace map as an example, in which case priority follows, or does it disclose only the specific idea of using a namespace map to compact the application, in which case the generalisation has no priority?
185. The skilled reader of this document is a person or team with skills in programming and software engineering including compilation and optimisation techniques. Those are the skills necessary to put the patent and for that matter the priority document into practice. This skilled person is very familiar, as a matter of common general knowledge, with the idea that there are ways of processing the code to change the size of a program. This aspect of the skilled reader's common general knowledge means that when they read the priority document, in my judgment they would understand the idea of the namespace map as a way of compacting the code but they would know there were other ways of doing it. The priority document does not simply teach that the namespace map could be used, it expressly draws attention to the fact that this is part of a general idea of achieving time and space efficiency. The teaching is explaining that space efficiency can be achieved by post processing the application program to allow the interpreter program to be smaller (section 2) and by post processing the application program to make that program smaller as well (section 3). A skilled team given the priority document who decided to reduce the size of the application for time and space efficiency using other techniques apart from the namespace map would not think they were doing anything different from the principles of general application the inventors of the priority document have taught. In my judgment the converter referred to in claim 1 is entitled to the priority of the priority document. The fact that the later patent specification includes express disclosures of other ways of reducing the size of the application does not mean that priority is lost.
186. I conclude that the combination of claim 1 and 3 is entitled to substantive priority from the priority document.

Claims 8 and 18

187. It is admitted what could be called element 8-(a), (b) and (e) are not entitled to priority. The issue is whether elements 8(c) and 8(d) are entitled to priority. Claim 18 has lost priority because it is not limited to Java.

188. Element (c) of claim 8 refers to modifying byte code operands from references using identifying strings to references using unique identifiers. Although I have found the namespace map in section 3 of the priority document, which replaces these strings with unique integers, is an enabling disclosure, it does not necessarily follow that element 8(c) must be entitled to priority. One point was that the processing referred to in section 3 may only relate to the constant pool in a Java class file and not directly to the byte code operands but I do not accept that this is a relevant distinction. In the context of a Java class file, a reference to byte code operands as “references using identifying strings” would be understood by a person skilled in the art as taking into account the fact that inside a class file the strings themselves are actually in the constant pool, separate from but cross-referenced to the byte code.
189. Another point was that, in his third report Prof Paradinas expressed the view that the process described in Section 3 necessitated some modification of byte code operands. I agree. HTC submitted that Prof Paradinas accepted in cross-examination that there was no disclosure that any such modification had to be done off-card. That is not how I understood the Professor’s evidence. He accepted that the document did not make it explicit that the modification takes place off-card but I understood his evidence to be that the skilled person would understand it was not possible to do this on-card. I find that the skilled person would understand section 3 of the priority document as a disclosure of a step which necessarily takes place off-card. Accordingly I find that element 8(c) is entitled to substantive priority.
190. That leaves element 8(d) concerning renumbering bytecodes. The argument here was that although this is disclosed in the priority document, the technique does not affect the size of the application program, it only affects the size and speed of the interpreter program. The premise is correct (renumbering does not change the size of the application) but the conclusion that priority is lost does not follow. Claim 8 is dependent on claim 1. All this combined claim requires is that the converter has to reduce the size of the application (claim 1) and also has to renumber the bytecodes (claim 8(d)). There is no requirement in the combined claim that renumbering has to cause a size reduction. Thus I find that elements 8(d) would be entitled to priority.
191. Next I need to consider the impact on the priority of claim 8 of the claims from which it depends. In the section on construction above I have identified that claim 8 can be notionally regarded as two claims: claim 8 plus 4 plus 1 and claim 8 plus 7 plus 1. The parties’ arguments did not chase every possible permutation of the arguments but at this point I need to decide if either of these notional claims is entitled to priority. Claim 4 is related to the same namespace map disclosure which supports 8(c) and so that claim has priority. However although sub-setting of the Java byte code is disclosed in the priority document (page 29) I do not believe the idea that the converter will verify the sub-setting is disclosed and so claim 7 is not entitled to priority. Thus claim 8 plus 7 plus 1 is only entitled to the filing date irrespective of elements 8(c) or (d). That leaves claim 8 plus 4 plus 1. Strictly I should also include claim 3 because unless the claim is limited to Java, it loses priority.
192. The remaining question therefore is whether a claim with the dependency claim 8 plus 4 plus 3 plus 1, is entitled to multiple priorities so that even though 8(a), (b) and (e) lose priority, 8(c) and 8(d) keep it.

193. Claim 8 is not like a claim to a chemical compound with five possible alternative substitutions. Elements (a) to (e) are not alternatives. As I addressed in the section on construction, the claim expressly relates to a process including all five steps (a), (b), (c), (d) and (e) and then requires the converter to perform at least one of those steps. In my judgment this claim is not entitled to multiple priorities because at its heart is a process consisting of all five steps. They are not alternatives. I can see that claim 18 with the same five sub-paragraphs is drafted in a different way, without reference to a process including all five steps, and that might raise different issues. However it cannot make any difference that claim 8 might have been drafted differently. I find that claim 8 is not entitled to multiple priorities.
194. As I understand the argument, the priority of claim 9 stands or falls with the priority of the claims from which it depends.

Priority - conclusions

195. I will summarise my conclusions on priority :
- i) Claim 1 is entitled to the filing date;
 - ii) Claim 3 (combined with claim 1) is entitled to the priority date;
 - iii) Claim 8 is not entitled to multiple or partial priorities and so is only entitled to the filing date;
 - iv) Claim 9 stands or falls with the claims from which it depends;
 - v) Claim 15 is entitled to the filing date;
 - vi) Claim 18 is entitled to the filing date.

865: Obviousness at the filing date

196. It is convenient at this stage to deal with the obviousness of any claims not entitled to priority in the light of intervening prior art. In closing HTC limited its case to obviousness over the Cyberflex materials.
197. HTC pleaded that the following was made available to the public before the filing date (22nd Oct. 1997):

“The development software for preparing applications for loading onto the Cyberflex smart card product made available to the public in about May 1997 together with the Cyberflex smart card product made available and intended to be used together with that development software from about May 1997 and together with the documentation relating to the said product and software made available at least on the website www.cyberflex.austin.et.slb.com as of 30 May 1997 (as evidenced by the state of the said website on that date shown by the Wayback Machine - extracts from which are annexed hereto). The Claimant also infers that the Defendant made available examples of the Cyberflex smart card product with at

least one application loaded in memory as samples from about May 1997.”

198. There was no dispute that all these things were made available at the relevant time. Although the case as pleaded refers to software and to actual products, in practice the case advanced at trial focussed on the documents on the Cyberflex website. These documents are:
- i) A title page headed “Presenting...” above a picture of the Cyberflex card (p.1);
 - ii) A contents page with various subheads including “What is Cyberflex?” and “Announcements” (p.2);
 - iii) Four pages headed “Cyberflex / What is it?” (pp.3-6);
 - iv) Four pages headed “Cyberflex / Announcements” (pp.7-10);
 - v) Two pages headed “What’s New” (pp.11-12);
 - vi) Three pages headed “The Universe of Smart Cards / Cyberflex / Frequently Asked Questions” (pp.13-15);
 - vii) Three pages headed “Development Toolkits” (pp.16-18);
 - viii) Four pages of a further FAQ under the headings “Documentation / Frequently Asked Questions / JavaCard: A Technical Briefing” (pp.19-22).
199. There was no dispute that it was legitimate to take these pages together as a single disclosure. The material discloses that Cyberflex is a smart card that runs programs written in Java. There is a CD-ROM with software called MakeSolo. Once the programmer has written a Java application, MakeSolo prepares the Java class file for loading onto the card. The MakeSolo software processes the Java byte code (from a standard Java compiler) and checks code for compliance with the Java Card API Specification. Certain Java byte codes are not supported: for example Cyberflex does not support floating point, exceptions and threads. The card contains an operating system and a virtual machine called Solo. The Solo Virtual Machine (SVM) is an interpreter which translates the byte code into instructions which the microprocessor on the card can understand. The virtual machine on the smart card takes up only 4KB of memory. In comparison the normal full JVM needs 150KB.
200. The structured approach to the assessment of obviousness was set out by the Court of Appeal *Pozzoli v BDMO* [2007] EWCA Civ 588, [2007] FSR 37. It is:
- (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) 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?

201. In *Conor v Angiotech* [2008] UKHL 49, [2008] RPC 28 the House of Lords considered the issue of obviousness. There Lord Hoffmann (with whom the others of their Lordships agreed) approved the following statement of Kitchin J made in *Generics v Lundbeck* [2007] RPC 32:

“The question of obviousness must be considered on the facts of each case. The court must consider the weight to be attached to any particular factor in the light of all the relevant circumstances. These may include such matters as the motive to find a solution to the problem the patent addresses, the number and extent of the possible avenues of research, the effort involved in pursuing them and the expectation of success.”

202. In *Medimmune v Novartis* [2012] EWCA Civ 1234 the Court of Appeal emphasised that the nature of the court’s task was ultimately to answer a single question of fact.

203. At paragraph 93 Kitchin LJ said:

“Ultimately the court has to evaluate all the relevant circumstances in order to answer a single and relatively simple question of fact: was it obvious to the skilled but unimaginative addressee to make a product or carry out a process falling within the claim. As Aldous LJ said in *Norton Healthcare v Beecham Group Plc* (unreported, 19 June 1997):

‘Each case depends upon the invention and the surrounding facts. No formula can be substituted for the words of the statute. In every case the Court has to weigh up the evidence and decide whether the invention was obvious. This is the statutory task.’”

204. Lewison LJ addressed the task in paragraphs 177 to 186. He also emphasised (paragraph 181) that the statutory question is: was *the invention* obvious? While in paragraphs 182 and 183 he said:

“182 [...] We should stick to the statutory question, which has to be applied in all sorts of circumstances and in all sorts of different fields of endeavour.

183 An invention is, at least usually, either a product or a process. So the statutory question is: was it obvious to make the product or to carry out the process? In order to answer the statutory question it is, of course, necessary to decide what the invention is. [...]”

205. Moore-Bick LJ agreed with both Kitchin LJ and Lewison LJ.

Skilled person and common general knowledge

206. Although I have identified the skilled addressee of the patent above I have not identified the notional person skilled in the art from the point of view of assessing obviousness. The debate between the parties about the skilled person was not really concerned with the argument over the Cyberflex prior art, it was concerned with the pre-priority prior art which I will consider later. Since Cyberflex expressly teaches that Java byte code is processed before being ready for the virtual machine, if the skilled team did not have the skills necessary to create such a processor, they would need to acquire them. That is the sort of task a programmer with specific knowledge of compilers would perform. Accordingly in my judgment the person skilled in the art, at least for the issue of inventive step over Cyberflex, can be regarded as the same team as the skilled addressee of the patent. I have identified the common general knowledge of this team above.

Claim 1

Inventive concept

207. I have construed claim 1 above. Since neither side advanced an inventive concept for the claim as such I will not do so either. I note that one of the key elements of the invention is a converter which takes code compiled from high level language source code and post processes that code into a smaller form suitable for interpretation by the interpreter on the microcontroller. Of course claim 1 covers a system based on Java.

Identify differences

208. There is no dispute that the Cyberflex materials disclose all the features of claim 1 save for those in sub-paragraph (b). The only difference between the claim and Cyberflex arises from whatever it is that the MakeSolo program does. It is clear that MakeSolo is a post processor of code. The code it starts with is a compiled form of a program written in a high level language. MakeSolo processes that code into a form suitable for interpretation by the interpreter. To that extent it is a converter as I have construed the claim. However, as Dr Greaves said in his first report, it is not absolutely clear that MakeSolo will reduce the size of the compiled code. I agree. The difference which exists between Cyberflex and claim 1 is whether MakeSolo reduces the size of the code. That is the only difference.

Are the steps obvious?

209. The Cyberflex materials are not published in an academic journal and no doubt are aimed primarily at announcing Cyberflex to the industry with a view to selling it. Many people reading Cyberflex would do so with a view merely to consider whether to buy and use it. Nevertheless the materials comprise a technical teaching which describes something. Although no doubt it would involve a lot of work, in my judgment no inventive step is required for anyone skilled in the art (regardless of the precise definition or composition of that team) to embark on developing their own product based on the ideas described in these materials. In other words the skilled team would not simply be buying the Cyberflex system as a user and speculating about how it might operate, the team is considering what is taught with a view to producing their own system.

210. Dr Greaves' view was that it would be plain to the skilled person that MakeSolo was doing linking, optimising and verifying so that those processes did not have to happen on the card, given the manifest size constraints. Gemalto submitted this was an exercise in hindsight. I do not agree. The Cyberflex materials explain that the virtual machine on the card is 4KB in size as compared to the 150KB size of the normal Java Virtual Machine. It is obvious that steps which are carried out by the JVM normally, such as linking, optimisation and verification, are being carried out off-card in the Cyberflex system. No hindsight is involved. I also do not accept the argument that the skilled person in this context does not know the details of what is contained inside a JVM. Since the Solo Virtual Machine is clearly a cut down version of the JVM, in order for the team to even begin to contemplate creating their own version of the Solo Virtual Machine, the team has to have an understanding of the functions carried out inside a JVM. This team will not treat the JVM as a monolithic block.
211. I should mention that the significance of linking and optimisation in this argument is that they are very likely to reduce the size of a block of compiled code. The point is not that the skilled team is necessarily setting out to reduce the size of the code but they are taking steps which would in fact have that effect.
212. In his reports Prof. Paradinas did not address the differences between claim 1 and Cyberflex, he focussed on claim 8 and the same can be said for Gemalto's opening skeleton. In closing Gemalto took a general point about Dr Greaves' approach which would apply to all claims. Gemalto submitted that in cross-examination Dr Greaves accepted that his approach to obviousness over Cyberflex utilised a number of building blocks and pieces of deduction that he had used in the section of his report on obviousness over common general knowledge alone at the priority date. This came down to two criticisms. The first was that there was no basis for the idea of using an interpreter on a smart card to run applications originally written in Java - but Cyberflex puts this very idea before the skilled team. The second was about whether all the details of the functions of the JVM were part of the common general knowledge. However starting from Cyberflex the team would find this out if they did not already have that information. Without these points the criticism has little force and I reject Gemalto's submission about Dr Greaves' approach in relation to Cyberflex.
213. Standing back, I think it would be obvious to the skilled team that MakeSolo reduced the size of the application and that the functions which brought this about (such as linking and inter-procedural optimisation) are functions which would be obvious to implement in an off-card post processor of the compiled code. I find that claim 1 is obvious over Cyberflex.

Claim 8

214. The person skilled in the art and the common general knowledge have been addressed above. The claim to be considered is claim 8, dependent on claim 1 but not on claim 3. As I have mentioned above, the claim can be considered notionally as two claims: claim 8 plus 4 plus 1; and claim 8 plus 7 plus 1. Furthermore claim 8 itself contains various sub-paragraphs. In order to avoid this issue becoming unwieldy I will address the various elements separately, always being in mind that in the end it is the various combinations as a whole which have to be considered.

215. I am sure that if claim 1 is obvious over Cyberflex then so too are notional claims 8(a), (b) and (e). Elements 8(a) and (e) are familiar aspects of code optimisation which would be entirely obvious given that I have found the post processor would make the code smaller. Any skilled person designing a post processor which made the code smaller would know that the jump destinations had to be handled and preserved appropriately in this way. Element 8(b) is also obvious. It covers the idea of replacing generic instructions with specific ones, which was part of the common general knowledge and would be one obvious way of reducing the size of the code.
216. Dr Greaves explained that element 8(c) is simply part of a conventional linking process. The skilled team would know that this was one of the functions of the JVM which, in Cyberflex, must be done off-card. Prof. Paradinas agreed that if linking was to be done off-card, this was an obvious step. I find that element 8(c) would be obvious over Cyberflex at the filing date.
217. Claim 4 is obvious for the same reason as element 8(c).
218. Dr Greaves' view was that element 8(d) was a common general knowledge approach and an obvious concomitant of subsetting the virtual machine. The point was that if you subset the instruction set, there would be gaps in the dispatch table and by renumbering the bytecodes the gaps can be filled making it more efficient, giving a performance gain. Prof. Paradinas accepted that the skilled person would know this. Cyberflex expressly teaches that certain Java bytecodes are not supported, i.e. that there has been subsetting. I find that element 8(d) would be obvious over Cyberflex at the filing date.
219. Insofar as claim 8 is dependent on claim 7, there is nothing in claim 7 which involves an inventive step. Cyberflex teaches that there has been subsetting. I have already decided it would be obvious to carry out verification off-card and that would naturally include checking that the code did not include instructions which were not present in the subset the interpreter could deal with.
220. Thus I find that if and to the extent that claim 8 is only entitled to the filing date, in all its various alternatives involving as they do claims 1, 4, 7 and the elements (a) to (e), it is obvious over Cyberflex. I have considered whether the various possible combinations add anything as far as inventive step is concerned. They do not. All of 8(a) to (e) together are obvious in combination over Cyberflex since they are obvious elements of a post processor. The same goes for the combinations with claim 4 and 7.

Claim 15 and 18

221. If, as I have found, claim 1 is obvious over Cyberflex, so too is claim 15. Claim 15 requires the first intermediate code to be interpretable by at least one first intermediate code virtual machine but that is already disclosed in the Cyberflex materials since the first intermediate code is simply Java byte code, which is interpretable by a normal JVM.
222. My conclusions in relation to all the elements (a) to (e) of claim 8 also apply to elements (a) to (e) of claim 18 and so this claim is obvious as well.

865: *Novelty – at the priority date*

223. There was no dispute of law in relation to novelty. To be invalid for lack of novelty, the matter relied on as prior art must disclose subject matter which, if performed, would necessarily result in infringement of the patent (per Lord Hoffmann *Synthon v SmithKline Beecham* [2005] UKHL 59)
224. By the closing, HTC's case on lack of novelty was limited to claims 1 and 15. They were said to lack novelty over:
- i) The paper "OCEAN: A C compiler for new smart card applications" by Oliver Caron and Georges Grimonprez of the RD2P-LIFL group in Lille, France published in 1994, ("the Caron article");
 - ii) The paper "Implementation of Pascal on an 8080 Microcomputer" by W.J.A. Pasman published in *Euromicro Journal* 5 (1979) 363-369 ("Pasman").
225. HTC did not maintain its pleaded anticipation case against any other claims and did not advance an anticipation case over a citation called 171. A document called the Caron thesis had been pleaded along-side the Caron article but by the end of the trial no separate point arose in relation to it.
226. Although I have found claims 1 and 15 have lost priority and lack inventive step over the intervening art, I will address novelty at the priority date in case this action goes further.

The Caron article

227. The Caron article relates to smart cards. The abstract states that smart cards are already emerging as a technology in banking, telecoms, and pay TV and suggests that more ambitious applications may arise in future. Topics to be considered include the security, the chip size and the memory limits of smart cards. The paper introduces something called OCEAN ("Outils de Conception et d'Evaluation d'Architectures Nouvelles"). OCEAN is a dedicated software tool for smart cards which helps the designer to produce suitable hardware and software for an application.
228. Part of Gemalto's case was that the focus of Caron and the OCEAN tool was about modelling the instruction set of a processor, and was only concerned with evaluating hardware. I do not accept that. Caron includes other ideas, as I shall explain.
229. Section 2 describes existing ways of developing applications for smart cards. Writing the code in the assembly language of the target machine (i.e. the microcontroller) is mentioned. The idea of writing the code in a high level language like C and compiling it into the machine code of the microprocessors on the smart card is also mentioned but the constraints of the microprocessors (such as the ROM memory size) are said to cause problems. One problem is that code generated by compilers is generally two to four times larger than code directly written in assembly language. Section 3 discusses the future. As I have said, one idea is that different hardware (i.e. not the existing microcontrollers identified in Caron) will have to be used. Section 4 describes the C_Card compiler, which is part of the OCEAN tool. Section 5 explains that the C_Card compiler compiles the C code into an intermediate code. The

intermediate code is the code for a virtual machine called QUAD. The QUAD language runs on a QUAD virtual machine.

230. Section 6 deals with the translation process. It describes two approaches. One is to translate the QUAD code to code for the target machine. It calls this approach “Direct Code”. The other approach is to translate the QUAD code to code for a virtual machine and store in the ROM memory of the chip both the interpreter and the translated code. This is called “Virtual Code” by Caron.
231. The paper compares this Virtual Code approach to Direct Code. The Direct Code method is 15 times faster, no doubt because the code is directly executable on the processor, However the Virtual Code takes up less space at least if the application is above a certain size. Caron shows this in figure 6 and states:

“If the size of the C_Card application is greater than about 500 lines, the size of code from the [*Virtual Code*] method is more compact.”

232. Thus Caron discloses an arrangement in which an application is written in C, a high level language, compiled off-card by the C-Card compiler to the QUAD language and then translated off-card into another code for a virtual machine to run on the card. The translated code and the interpreter are to be loaded onto the card (i.e. onto the microcontroller).

Comparing the Caron article to the claims

233. At one stage in the proceedings Gemalto contended that claim 1 was novel over the Caron article because C was not a high level language within the meaning of claim 1 but that argument was dropped and I need not consider it. At one stage it was suggested by Gemalto that Caron did not disclose loading the code into the memory of a microcontroller. I am not sure this was formally dropped as a point but it is a bad point in any event. As Prof. Paradinas agreed in cross-examination, the aim of the Caron article is to load applications onto smart cards, in other words onto the memory of the microcontroller (not named as such) on the smart card.
234. Prof. Paradinas’s evidence raised two points. One was whether the intermediate code could indeed be executed by a virtual machine. This is really a point directed to claim 15 not claim 1 but it is convenient to deal with it here. The other was whether Caron disclosed compilation followed by conversion. Prof. Paradinas said that the translator would be regarded as part of the compiler rather than as a separate converter. Both issues involve points on construction of the claims but it is convenient to address them here in the context in which they arise.
235. I reject the first point. The idea of interpreting the intermediate code on the QUAD virtual machine is part of the disclosure of the Caron paper. The intermediate code in the Caron article corresponds to the first intermediate code in claim 15. As a matter of construction, all claim 15 requires is that the first intermediate code be interpretable by a virtual machine. It cannot matter in principle whether a virtual machine for the code is actually described along with the code itself. The disclosure in Caron is that the code is code with the capacity to be interpreted by a virtual machine. That is all the claim requires. If there was some basis to say such a virtual machine could not be

made then things might be different, but nothing of that kind is said here. Furthermore, although it is not a necessary requirement to fall inside the claim, I accept Dr Greaves' evidence in cross-examination that anyone doing this would make (if they did not already have) an interpreter for the intermediate code.

236. The second point was based on paragraphs 11.79 - 11.80 of Prof. Paradinas' first report. He said:

“11.79 Secondly, looking at [*Caron's*] two different methods:

(a) If you look at the direct translation approach, there is no high level language (because C is not a typed object oriented language), no converter for post processing the compiled code into a minimized form and no interpreter.

(b) If you look at the virtual machine approach, there is still no high level language and there is no converter for post processing the compiled code: the translator described would be regarded as part of the compiler rather than as a separate converter.

11.80 Assembly for the QUAD machine is not the same as post processing by a converter because in a converter you start with compiled code and end up with compiled code, whereas in *Caron/Grimonprez* you start with compiled code and end up with assembled machine code.”

237. The high level language point in 11.79(a) and (b) has fallen away. The questions are whether the remaining point in 11.79(b) or the point made in 11.80 are relevant distinctions.
238. The process disclosed in *Caron* includes the idea that the source code written in C is compiled into an intermediate language and then translated into another code. Prof. Paradinas calls the translated code “assembled machine code”. Whether or not that is a fair description does not matter as long as it is remembered that in *Caron's* Virtual Code approach that “assembled machine code” is code for a virtual machine on the card. In claim 1, as a matter of construction, the word “converter” simply means the thing which post processes the compiled form into a minimized form suitable for interpretation by the interpreter. As I understand Prof. Paradinas' evidence here, it is based on a narrower construction of the term “converter”. Prof. Paradinas construction of “converter” covers a post processor which takes compiled code and ends up with “compiled code” but not a post processor which starts with compiled code and ends up with “assembled machine code”. I reject that distinction. There is no basis for it in the specification. All the claim requires for a post processor to be a converter is that the code produced by the converter satisfies the requirements in the claim (suitable for interpretation by the interpreter and in a minimized form). Prof. Paradinas accepted that the process of translation would involve optimisation which would reduce the size of the code.
239. I also do not accept that it matters whether the skilled person regards the process called translation in *Caron* as a process which is part of the compiler. Even if the

translator is to be regarded as part of the compiler, the Caron article expressly describes a two stage process in which the original C source code is compiled to the intermediate language (QUAD) and the intermediate language is translated into a code for a virtual machine. The virtual machine which runs the translated code will have to be different from the one which runs the QUAD code.

240. I find that all the requirements of claim 1 are satisfied by the Caron article.
241. As for claim 15, it requires the first intermediate code to be interpretable by a virtual machine, which I have considered already. Also claim 15 does not expressly require the converter to produce a minimized form. I find claim 15 also lacks novelty over the Caron article.

Pasman

242. Pasman was published in 1979. It relates to the use of a high-level language (Pascal) on what is now regarded as an early 8 bit microprocessor called the 8080. Also used in Pasman is a minicomputer called the DEC 10. At the time the term “minicomputer” was used to refer to larger, more powerful computers than microprocessor based microcomputers. The idea is to use a high level language to reduce the programming effort in writing applications for the microcomputer. The problem is that the resulting programs are too large for a microcomputer so Pasman wants an implementation which gives efficient memory usage. Pasman refers to a portable compiler for Pascal called the P-compiler which produces a code called P-symbolic code. This consists of assembler statements for a “hypothetical computer”. In other words, P-symbolic code is intermediate code which could be interpreted by an interpreter. Pasman also refers to an assembler (the P-assembler) which translates the P-symbolic code into P-machine code. A DEC 10 minicomputer was used to perform all compilation and assembly of the Pascal and P-symbolic code, prior to loading onto the 8080. The 8080 had limited amount of memory available. It could access 64Kbytes (all off-chip).
243. Pasman describes three possible approaches to generating code to run on the 8080. The first two involve creating machine code for the 8080 itself. The third is the idea of generating code for a “P-processor” and interpreting it on the 8080. Pasman notes that the first two approaches would produce two or three times as much code as the third, and proposes to write an interpreter for P-machine code. Optimisation of the code is described as part of the process of creating P-machine code from P-symbolic code. One such optimisation involves replacing generic opcodes with specific ones in order to save space. These optimisations are performed by the P-assembler. In other words, on the DEC 10 the compiled intermediate code produced by the P-compiler is being post processed by the P-assembler to produce optimised code suitable for interpretation by the interpreter on the 8080.

Comparing Pasman with the claims

244. The 8080 chip in Pasman has no on-chip memory, except for registers, which I have already said do not count. HTC’s case is that the 8080 is not a microcontroller but that Pasman discloses all aspects of claims 1 and 15 other than the use of a microcontroller. Gemalto took a number of points relating to Pasman on the issue of inventive step, in particular emphasising the age of the paper and the obsolescence of

the 8080 chip by the priority date. That may be so but it is not relevant to novelty. In his first report Prof. Paradinas gave three distinctions between Pasman and claim 1. One was that Pascal was not the right kind of high level language but, as with Caron, that point was abandoned by Gemalto. Another was a similar argument to the one I have rejected over Caron that the assembler in Pasman is part of the compilation stage and that the intermediate P symbolic code has no practical use. Both points are irrelevant or are based on a construction of claim 1 I do not accept. Finally Prof. Paradinas said the 8080 was not a microcontroller because all its memory was off-chip. While that accords with the construction I have arrived at, it is not the construction of microcontroller advanced by Gemalto in closing.

245. Since I have not accepted Gemalto's construction of claim 1 I do not have to make a finding about novelty on that construction. I will say only that I am not convinced claim 1 or claim 15 would be novel on Gemalto's construction of these claims; the Pasman disclosure applies to applications for the 8080 in general, whether they are for dedicated systems or general purpose computing.

865: Obviousness at the priority date

246. HTC contended that all the claims were obvious at the priority date in the light of:
- i) Common general knowledge alone.
 - ii) The Caron article.
 - iii) Pasman.
 - iv) French Patent Application 2 667 171 entitled "*Support portable à micro-circuit facilement programmable et procédé de programmation de ce micro-circuit*" published on 27th March 1992. Although published in French the parties used an agreed translation.
247. I will consider inventive step in relation to claim 3 (i.e. the combination of claim 1 and claim 3).
248. As before I will apply the *Pozzoli* approach to the assessment.
249. It is always important to be wary of hindsight and that is all the more so when, as here, the claim is said to be obvious based on common general knowledge alone. Gemalto reminded me of the various warnings about obviousness arguments based on common general knowledge alone. I will refer only to one. Recently in *HTC v Apple* [2013] EWCA Civ 451 Kitchin LJ referred at paragraph 67 to what he had said in *Abbott 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.”

Person skilled in the art

250. In *Schlumberger* when Jacob LJ was focussing on the identity of the person skilled in the art from the point of view of obviousness he said “*What the combined skills (and mind-sets) of real research teams in the art is what matters when one is constructing the notional research team to whom the invention must be obvious if the Patent is to be found invalid on this ground.*” (paragraph 42)
251. Gemalto contended that the person skilled in the art from the perspective of obviousness at the priority date was an individual or team developing code for a microcontroller, such as was used in a smart card. Such a team would consist of good practitioners of hardware specific assembly languages or of C. It submitted that it was in those languages, which produced compact programs, that the programming of microcontrollers typically happened. Typical examples of the person skilled in the art are those who were employed at a smart card manufacturer.
252. HTC submitted that the skilled team was someone with expertise in high level languages and in developing tool chain components for such languages. In particular HTC relied on Dr Greaves’ view about the person skilled in the art expressed in paragraphs 20-22 of his first report. He said:

“20. I consider that this would have included a team of people with expertise at the Priority Date in (i) microcontroller hardware; and (ii) programming and software engineering. I can envisage at least two straightforward circumstances in which such a team would be assembled, as follows:

21. In the first, a software developer (or team of such developers) would be very familiar with high-level languages generally and compiler tool chain techniques, and in particular the relatively new (as at the Priority Date) but highly publicised language of Java. They would be aware of the general interest surrounding Java and of the commercial and technical desirability of developing JVMs for many different hardware platforms, including microcontrollers. In order to explore the possibility of developing interpreters and compiler tool chains for microcontrollers, it would make sense for the software developer to approach others with knowledge of microcontroller hardware.

22. In the second scenario, the person or team skilled in microcontroller hardware (forming a development team at an enterprise working with microcontroller hardware, such as a consumer electronics or smart card company, for example) would be aware of the desirability of allowing programmers to write applications for microcontrollers using high-level languages, as opposed to hardware-specific assembly languages known only to a small minority of programmers. It is likely that that person or team would also have heard of Java specifically and be aware of its various benefits. Even if they had not, they would still have ample motivation to explore the possibilities of high-level language programming for microcontrollers. It would be natural to approach others with skill in programming and in developing interpreters and compilation tool chains for such languages. In this latter case, I believe it is also perfectly possible that such a development team might already include one or more persons with knowledge of developing interpreters and compiler tool chains (see below).”

253. Schlumberger makes clear that the way to resolve this problem is by considering what real research teams in the art were doing at the relevant time.
254. Prof. Paradinas said that the idea of using high level languages to develop microcontroller applications was not really accepted. He said it was not really done save for the CQL card with which he had been involved. In his oral evidence Prof. Paradinas said that the culture in the smart card industry was an “electronics” culture rather than a “computer science” culture, a point which was seeking to emphasise that the focus was on assembly language programming and hardware. Prof. Paradinas said the “first language” of such individuals was assembly language. This supports Gemalto’s view of the skilled person.
255. Mr Tappin took Prof. Paradinas to a number of documents including a 1991 article from the chip maker Motorola, a 1994 Smart Card News article about a new Hitachi chip and an article in 1996 about an Atmel microcontroller. The point of these references was to support Dr Greaves’ view that those skilled in art were aware of the desirability of programming microcontrollers in high level languages. The references showed microcontroller manufacturers producing microcontroller chips designed to run applications programmed in high level languages. The Atmel document stated that high level languages, with C being the most widely used, are rapidly becoming the standard programming methodology for embedded microcontrollers. In his answers Prof. Paradinas said that there was in fact some demand for programming smart cards in high level languages and said that high level languages were used for programming some microcontrollers but not all microcontrollers.
256. In the course of this cross examination the Professor appeared to suggest that his evidence that programming smart cards with high level languages was not really done referred to the position in 1993 as opposed to 1996. Mr Mellor in closing submitted that what the Professor meant was that the position at the priority date in 1996 had not moved on from the position in 1993. I accept Mr Mellor’s submission about the Professor’s evidence (otherwise his oral testimony would have been seriously at odds with the evidence in his reports).

257. The conclusions I draw from the evidence on this issue are as follows. First I believe the evidence was clear that those working in smart cards at all material times were well aware that programming applications in high level languages was a desirable thing to do in some circumstances. Two examples will suffice. The Caron Thesis, which reflects work done up to 1994, refers to the C_Card project begun in 1990 using a high level language to meet the need to program more complex applications. Slides from a CardTech /SecurTech '96 conference in May 1996 refer to a "Smart card management and application generator" which allows applications to be written in a high level language. Those working in this field knew it was desirable and knew that the major problem was that the size constraints made the process of programming in a high level language difficult.
258. Second, I was not persuaded that the position in 1996 is to be equated with the position in 1993. A number of the documents relied on by HTC dated from after 1993. I find that whatever the position was in 1993, by the priority date in October 1996, the idea of programming a smart card in a high level language was a very familiar idea to those working in that field.
259. Third, I believe care needs to be taken in necessarily equating the smart card industry with microcontrollers in general. Although microcontroller chips were used in smart cards, the use of microcontrollers was not limited to smart cards. Prof. Paradinas' experience was very much focussed on smart cards. I have no reason to doubt his characterisation of the people in the smart card industry in 1993 as people with an "electronics" culture focussed on programming in assembly language. However the patent claims are not limited to smart cards or to microcontrollers in smart cards. Even in 1993 there was a wider microcontroller industry, not focussed specifically on smart cards.
260. I find that by October 1996 there were skilled people considering the problem of how to program microcontrollers using high level languages. Necessarily such people were not focussed on programming in assembly language.
261. HTC also referred to a post-published book by Guthery & Cronin (2002). The author Mr Guthery is one of the inventors of the 865 patent. In the chapter "A short history of byte code interpreters on smart cards" the authors set out a table listing a number of systems and teams working on byte code interpreters for smart cards. There are twelve entries prior to 1996 and a number of high level languages are mentioned (C, Pascal, Java and Basic). There is no suggestion that the existence of all of those teams and the work they were doing was common general knowledge or even had all been published before the priority date. The primary evidence I have relied on is the evidence mentioned above but this book is supportive of the conclusion I reached.
262. I appreciate that Gemalto's definition of the skilled team includes a reference to the language C and seeks to take that into account but I reject Gemalto's definition of the person skilled in the art. It gives undue emphasis to hardware, C, and assembly language. In my judgment real research teams in 1996 were seeking to program microcontrollers using high level languages. Such a team would include members with knowledge of microcontroller hardware but it would also include members who were software developers with a computer science background. They would be very familiar with high-level languages generally and also would be familiar with compiler tool chain techniques. The latter is true because the task of the team was not simply to

write application code in a high level language, this team was dealing with the programming process itself.

Common general knowledge

263. Having identified the person skilled in the art I need to address their common general knowledge.
264. The common general knowledge of the team would include all the matters I set out in the section on common general knowledge.
265. I accept that the idea of programming microcontrollers in high level languages was common general knowledge and I also accept that Java was a high level language which was part of the common general knowledge but I do not accept that the idea of programming or trying to program a microcontroller in Java was common general knowledge. The fact that programming microcontrollers in high level languages in general was common general knowledge does not carry that idea with it. Whether it was obvious over common general knowledge alone to embark on programming a microcontroller in Java is a matter I will consider separately below.

Obviousness of claim 3 over common general knowledge alone

266. I will start with the submission that claim 3 is obvious over common general knowledge alone although I do not think it is HTC's strongest attack. HTC summarised the argument as follows:
 - i) It was known to be desirable to program microcontrollers using high level languages – for that purpose it was necessary to include in the skilled team someone with expertise in high level languages and developing interpreters and tool chain components for high level languages.
 - ii) Java was a good candidate high level language, not least because of its security features.
 - iii) The skilled team knew that if they used Java they would have to write a JVM for a microcontroller. They knew that a standard JVM was too large to fit on a microcontroller, but they also knew that it consisted of various functional components in addition to the execution engine itself, including a linker (which also optimised), a loader and a verifier. They would assess the size of the execution engine itself as being no more than 10KB.
 - iv) The common general knowledge approach to programming embedded systems such as microcontrollers was to link code before loading.
 - v) The skilled team knew that one way to reduce the size of a virtual machine was to subset its instruction set.
 - vi) In the light of that common general knowledge, it was obvious to use Java bytecode on a microcontroller and to achieve that by reducing the size of the virtual machine by subsetting its instruction set and doing the linking, optimising and static verification before loading, in accordance with the standard approach to programming embedded systems. This would have the

result that the skilled team created a converter/post-processor which satisfied claim 1.

267. HTC's case was supported by Dr Greaves' evidence and in cross-examination he maintained his opinion in relation to a number of the key elements in the argument: conceptually dividing up the JVM, appreciating that the execution engine in the JVM was no more than 10KB in size, subsetting the language being a point which might have commercial significance but was technically obvious, and the approach to programming embedded systems which involved linking before loading. HTC also referred to a statement by one of the inventors (Scott Guthery) that the idea of Java on a smart card was a "no-brainer".
268. Prof. Paradinas did not agree. His major point was that it was not obvious how to proceed because the scale of the reduction in size of the JVM which would be required was so great as to make it not seem feasible. HTC pointed out that this view was based on his experience at Gemplus and is likely to have been influenced by factors which are legally irrelevant such as the need to obtain EU funding on a project in this field at the time.

Dr Greaves' reports

269. At this stage I need to deal with an issue arising from Dr Greaves' reports. He explained in his first expert's report that before reading the 865 patent he was asked to consider the disciplines which would have been included in a team tasked with developing a microcontroller (such as a smart card) which could be programmed in a high level language. In cross-examination he explained that he approached this as if he had been asked to consider how to solve a problem. Gemalto submitted that teams actually tasked with this problem did not exist in fact and submitted that this meant Dr Greaves' instructions necessarily entailed pointers to the idea and objective of using high level languages on microcontrollers. His second report indicated that these early discussions included compilers and interpreters. Gemalto submitted this also indicated that impermissible hindsight was involved. Another submission of Gemalto's was that a reference to memory constraints before Dr Greaves had read the patent meant he had approached the patent with a mindset exclusively about microcontrollers with memory constraints. Finally, arising from a muddle which Dr Greaves got into in cross-examination, Gemalto submitted that it was not clear whether the section of Dr Greaves' report which related to obviousness over common general knowledge alone had been prepared, as Gemalto contended it should have been, "free from the influence of the 865 patent".
270. All of these points are derived from a single source. Dr Greaves was instructed in a particular way by HTC's solicitors. He summarised it in paragraph 2 of his first report:

"2. When I was first contacted by PG, I was initially asked to provide an overview of my qualifications and expertise. Following this, I discussed with PG background information on the technology relating to high-level programming languages generally, and the operation and design of compilers and interpreters specifically. I also discussed the disciplines which would have been involved in programming hardware with

memory constraints (such as microcontrollers for embedded systems) and what would have been generally known to people skilled in this field in October 1996. This discussion took place before I had seen the '865 Patent or any of the prior art upon which I now understand HTC intends to rely, and without knowledge of the subject matter of the dispute¹. My views, as explained to PG at that time, are set out in Sections C and D of this First Report. Further details of my instructions are set out below.”

271. I mentioned a muddle which arose in cross-examination. I am quite sure the muddle arose, as did a number of answers from Dr Greaves on this topic, because he felt from the questions he was being asked that he had done something wrong and inappropriate or had in some way behaved improperly. Dr Greaves was clearly worried about this and he was unsettled by it. Mr Mellor’s cross-examination was entirely legitimate given Dr Greaves’ report. The problem is that Dr Greaves had been given to understand that there was a rule about the way his report had to be prepared. Sections of it had to be “free from the influence of the 865 patent”. He thought he had followed the rule but the questions suggested he had somehow breached it.
272. Patent cases are necessarily conducted *ex post facto*. It is not possible to do anything else. The courts dealing with patents are alert to the problem of hindsight and the need to guard against it. That is one reason why the emphasis in relation to obviousness is on the reasons for an expert’s opinion and not the opinion itself. The expert is not standing as a representative of the person skilled in the art whereby their report is a kind of test to see what the skilled person would do before being physically handed the patent. If the report is advanced as the result of a test then the party perhaps should have to disclose the results of all the other similar tests it has carried out (in other words all the other experts it has interviewed). That would add an enormously unwelcome cost and complexity to patent cases.
273. In *Medimmune v Novartis* [2011] EWHC 1669 (Pat) Arnold J addresses the preparation of expert witnesses in patent cases at paragraphs 99-114. I agree with what was said in those paragraphs. In paragraph 118 of the same judgment Arnold J indicated that the way in which an expert had been instructed in that case, by providing him with the prior art, then the priority documents and then patents, was useful in the context of that case. Plainly experts need to be aware of the risk of hindsight reasoning and of allowing ideas in the patent to influence their views about what the skilled person would do but I do not read Arnold J’s judgment as seeking to lay down a rule that this is how experts must be instructed in all patent cases.
274. Having seen Dr Greaves in this case, I believe it is important to emphasise that approaches like this are servants and should not become elevated into masters. The approach can risk creating a number of potential difficulties. First the expert naturally may think there is a legal requirement that this be done and that somehow a breach of this protocol is wrong or is his or her fault. There is no legal requirement that expert evidence in patent cases must be given this way. It will not always be fair on the expert to place this burden on them. Second, many patent lawyers are highly skilled at presenting information to the expert which, consciously or subconsciously, has the effect of leading the expert to the conclusion sought. Neither the expert nor the legal team may be aware of it at all. The expert may genuinely think they have formed

views without knowledge of the contents of the patent when in fact they may have been led there. Thus the approach may not in fact eliminate hindsight and is capable of giving a false impression.

275. The exercise in this case caused much more trouble than it could ever have been worth.
276. Since I have found that the common general knowledge did include the idea of programming a microcontroller in a high level language, one of Gemalto's main points about this part of Dr Greaves' evidence falls away. It was not hindsight to pose the question in that way. Also given the way in which I have identified the skilled team, the submission about compilers falls away as well. As for the point about how Dr Greaves read the patent. It does not arise in this part of the case but it is convenient to address it here. I reject it. I do not think Dr Greaves gave undue weight to memory constraints in reading the patent. I also note the argument is in danger of getting circular. Dr Greaves is being criticised both for considering common general knowledge before reading the patent and at the same time for not preparing his views on common general knowledge before seeing the patent.

Assessment

277. Considering the matter purely from common general knowledge alone, I think it would not involve an act of invention on the part of the skilled team to conceive of the idea of trying to run an application written in Java on a microcontroller. It would be manifest to that team that in order for this to work, they would need to create their own JVM to run on the microcontroller. The critical point is that at the priority date microcontrollers had very limited memory capacity. This limitation was one of the defining characteristics of microcontrollers. The team would think (rightly) that a normal JVM would just not fit in the memory of a microcontroller. The problem is not a minor one. The JVM is far too big. The team would know that the JVM is made up of components but to move forward the team has to decide how it will deal with them. Can functions be performed off-card? What effect does that have? What functions can/should be performed off-card and what functions might be dispensed with altogether? What functions must be kept on the card? Why not dispense with a virtual machine altogether and just compile to the machine code of the microcontroller?
278. Looking back with knowledge of the invention one can see that there are a number of choices which have to be made in a particular way if the end result is something within the claim. However starting from common general knowledge alone the skilled team has no framework in which to operate and no reason to make the choices which in fact lead to the claim.
279. It was put to Dr Greaves that the reasoning depended on a number of steps but Mr Tappin submitted that Dr Greaves' answer was that there was really only one step – to take Java byte code and put it in ROM with an interpreter. I do not accept it is that simple. This takes for granted that the skilled team has decided it can jettison all other elements of the JVM but does not grapple with the question of whether the skilled team would accept that was worthwhile. Moreover it also disregards the issue of linking code before loading onto the microcontroller. That is a critical element in HTC's case because in order to arrive within the claim, there needs to be a converter

which post processes the compiled form into a minimized form. To decide to create such a converter is another key step.

280. Inventors often think that what they did or a step in their reasoning was obvious. No doubt it was obvious to them but what Mr Guthery said is not a matter to which I will give much weight in this case.
281. Dr Greaves' view that the claim is obvious was persuasive but in the end I think it does not give sufficient weight to the severe memory constraint on microcontrollers at the priority date. HTC submitted that Prof. Paradinas's personal experience was a hindrance rather than a help in addressing this issue - I do not agree. In my judgment the skilled team would think in the way Prof. Paradinas described. His concern applies not just in the smart card sector of the microcontroller field but applies to all microcontrollers at that time. The team would think the memory available on a microcontroller was so limited that it was unlikely to be realistic to run Java applications on a microcontroller. I find that claim 3 was not obvious at the priority date because of the severe memory constraints on microcontrollers.
282. I have not considered the obviousness of claim 3 on Gemalto's construction of the term microcontroller in any depth. On Gemalto's construction the severe memory constraint does not exist. That would have a major impact on the balance of the obviousness argument.

Obviousness of claim 3 over the Caron article

283. I can deal with the obviousness argument over Caron briefly because although it is stronger than the argument based on common general knowledge alone, in my judgment it still founders on the same key difficulty.
284. A skilled team in 1996 presented with Caron would see a concrete proposal for writing applications in a high level language (C) to run on a smart card. The team would see that the idea involves compiling the application code into an intermediate code (QUAD) which is code for a virtual machine and then describes options for how to generate code which will actually be loaded onto the smart card. These are the Direct Code and Virtual Code options. The Virtual Code option is to translate the QUAD code to code for a virtual machine and store in the ROM memory of the chip both the interpreter and the translated code. The relative advantages and disadvantages of the two options are explained.
285. Dr Greaves' view was that it was obvious to apply Caron to Java in the same way he had described in his report dealing with obviousness over common general knowledge alone. It was put to him that all he was doing was using Caron as a "jumping off point" for the common general knowledge argument and the manner in which the point was put in cross-examination seemed to Dr Greaves as if he had approached his evidence in an inappropriate or wrong way. I reject that. Dr Greaves' report recognised that the various obviousness arguments being advanced had common elements. There is nothing wrong with that. Dr Greaves was not ignoring the prior art. Moreover, as Mr Tappin submitted, when one is dealing with obviousness there is nothing inappropriate in principle in a skilled person adding to or departing from the teaching of the prior art based on their common general knowledge. Whether such a step was or was not obvious still has to be determined.

286. I accept that a skilled team presented with Caron would think of applying its approach to Java. I understood an answer given by Prof. Paradinas in cross-examination (p313 line 15) to accept that the teaching of Caron was applicable to a language like Java with Class files. I think it would be obvious to the skilled team how Caron's teaching might be applied in the case of Java, in that Caron's intermediate QUAD code produced by compilation equates by analogy to the Class files with bytecode producing by the Java compiler. The skilled team would also understand that to apply Caron's Virtual Code approach in the context of Java would involve creating an interpreter to be loaded onto the smart card. Caron shows that skilled team that something along these lines can be done and provides a framework to the skilled team which can be applied to Java. This is why the case over Caron is stronger than the argument over common general knowledge alone. It involves much less hindsight.
287. However I still think it suffers from the same key problem. Caron teaches that it is feasible to use the interpreter approach within the memory constraint of a microcontroller. The size of the interpreter mentioned in Caron is 1.5 Kbytes. It would be obvious to the skilled team that the JVM was much larger than that. The existing JVM simply would not fit. So again the skilled team is faced with a dilemma about what to do about all the other functions of the JVM. They cannot all be retained because they will not fit in the memory but these functions are some of the important features which makes Java a desirable language to use in the first place. I do not think it was an obvious thing to do given the severe memory constraints on microcontrollers.

Obviousness of claim 3 over Pasman

288. The obviousness argument over Pasman is in essence the same as the argument over Caron. If claim 3 is not obvious over Caron I fail to see how it would be obvious over Pasman.

Obviousness of claim 3 over the '171 application

289. The '171 application is written in French but there was an agreed English translation. It was published in 1992. The document is concerned with programming microcontrollers and smart cards. It refers in terms to the desirability of programming these things using high level languages instead of machine code and mentions by name a number of high level languages (not Java). The severe memory constraints which arise in this context are referred to:

“The main difficulty encountered with microcircuit cards stems from the fact that the microprocessor with which they are outfitted is associated with a working memory (of RAM type, static or dynamic) of low capacity, oftentimes only 128 bytes, and with program memories (ROM: most of the time EPROM or even EEPROM type), also being of low capacity, generally limited to several dozen kilobytes, or even only a few kilobytes.”

290. The proposal in the 171 application is to use an interpreter on the smart card. The application program is written in a high level language and compiled to code which

the interpreter can interpret. The compiled code and the interpreter are loaded onto the smart card.

291. In addition to the point that the 171 application is not concerned with Java, the other key difference between the disclosure and claim 3 is that the 171 application describes a one step process, compiling the high level source code directly to code for the interpreter on the card. There is no converter for post processing compiled code at all.
292. If claim 3 is not obvious over Caron, it is not obvious over the 171 application.

Obviousness - other claims

293. Since I have rejected the obviousness attack on claim 3, it follows that any claim dependent on claim 3 which retains priority is also not obvious. That is claim 9.

865: Patentable subject matter

294. HTC contends that the claims of the 865 patent do not relate to a patentable invention in that their subject-matter consists of a program for a computer as such or otherwise does not make a technical contribution.
295. 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.

296. The law on this issue has very recently been considered by the Court of Appeal (Richards, Lewison and Kitchin LJJ) in *HTC v Apple* [2013] EWCA Civ 451. Kitchin LJ reviewed the English authorities on the question of computer programs as such, including the approach set out in *Aerotel Ltd v Telco Holdings Ltd; Macrossan's Patent Application* [2006] EWCA Civ 1371, the judgment in *Symbian v Comptroller-General of Patents* [2008] EWCA Civ 1066, [2009] RPC 1 and the

position in the European Patent Office. At paragraph 44 Kitchin LJ held that it would not be appropriate to abandon the approach explained in Aerotel. He said:

“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*.”

297. The four stage approach is :

- 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.

298. How this approach is to be applied, the problems which can arise and some useful signposts (first set out by Lewison J as he then was in AT&T Knowledge Ventures' Application [2009] EWHC 343 Pat) are addressed by Kitchin LJ in paragraphs 37 to 51. Lewison LJ also dealt with the law on this topic. He emphasised that the cases decided in both the EPO and the English courts had moved a long way from the language used in the legislation but accepted the approach in the cases, albeit regretfully. Kitchin LJ agreed with Lewison LJ. A specific point arising in the judgment of Lewison LJ was a modification to one of the AT&T signposts (paragraphs 150-151). Richards LJ agreed with Kitchin LJ.

299. I will consider the issue focussing on claim 3 since that claim would otherwise be valid. I have construed the claim already. For the purposes of this argument the claim can be seen as a claim to a computer programmed in a particular way. The computer is a microcontroller. It must have an interpreter and code suitable for interpretation loaded in memory. The code must have been created in a particular way.

300. What if anything is the technical contribution to the art made by claim 3? This needs to be considered at the priority date. In this case, whether one is considering the common general knowledge or the individual cited items of prior art the contribution made by claim 3 is the same. It is a microcontroller which can run applications originally written in Java. To achieve this, an interpreter has to be loaded onto the microcontroller as well as application code and the application program itself has to be processed in a new way before being placed on the card.

301. The only relevant exclusion is Art 52(2)(c) (programs for computers as such). Although software is at the heart of this invention I do not accept the contribution is excluded, for three reasons. First, the invention is a solution to a concrete and technical problem relating to a particular kind of computer chip (microcontrollers).

Second, the solution is generally applicable. This is not merely a new program which happens to run on a microcontroller, it is a new way of programming microcontrollers. It is applicable to microcontrollers in general and to Java applications in general. Third the subject matter of the claim, a microcontroller, has a capacity which known microcontrollers did not have. It can run applications written in Java which could not be run on microcontrollers before. In that sense it could fairly be called a “better” microcontroller. “Better” does not necessarily mean inventive. The points I have relied on would not make sense if the claim lacked novelty but they do not depend on the finding of inventive step.

302. In my judgment this contribution does not fall within the exclusion. It is not a computer program as such. It has a clear technical character. Claim 3 does not fall foul of Art 52(2)(c). This conclusion applies to all claims entitled to the priority date.

865: Infringement

303. The HTC devices alleged to infringe are: the HTC Sensation, the HTC Sensation XL with Beats Audio, the HTC Sensation XE with Beats Audio, the HTC Explorer, the HTC Rhyme, the HTC EVO 3D, the HTC Cha Cha, the HTC Salsa, the HTC Incredible S, the HTC Desire S, the HTC Desire Z, the HTC Desire HD, the HTC Wildfire S, the HTC Gratia and the HTC Flyer. These are referred to as the HTC Devices. There is no need to distinguish between them.
304. The HTC Devices each contain a Qualcomm chipset (“QC”). In some HTC devices the application processor is designed by ARM and in others it is a Qualcomm implementation of an ARM-compatible processor. The distinction does not matter in this case. The chipset contains the application processor and cache memory. There is between 320KB and 576KB of total cache memory depending on the HTC Device, divided into L1 and L2 cache. All the memory (save for registers and cache) is off-chip. It is RAM and FLASH. Depending on the HTC device there are between 384MB and 1GB of RAM and between 512MB and 32GB of Flash memory.
305. All the program code and data on the HTC Devices are stored in the FLASH memory. When the HTC Devices are turned off, the RAM, the cache and the application processor registers are empty.
306. The HTC devices use the Android system. The Android system has a number of layers. It has a low level operating system called a Linux kernel. This runs the very basic core services of the computer system, managing the resources and running device drivers. In order to run applications the Android system has a virtual machine called the Dalvik Virtual Machine or DVM. The applications which are run on a DVM include but are not limited to downloadable programs referred to today as “apps”. For every application which is run, a separate instance of the DVM is created and run. Part of this exercise involves a process called Zygote loading various code libraries, the application code, and an instance of the DVM, from FLASH into RAM.
307. Many Android apps are written at least in part in Java. The process of creating the application code to run on a DVM, starting from a developer writing in Java, is as follows. The Java code is compiled to byte code in .class files using the normal Java compiler. (In this infringement section of this judgment I will refer to these as “.class” files instead of “Class” files.) If the app was not to be run on an Android

system then this would be the end of the process. The app could be run on a computer with a JVM. To run the app on an Android system requires a further conversion process. This is carried out using a development environment called the Android SDK (software development kit). This creates .dex files from .class files by means of a tool called dx. After the .dex files have been produced, the .dex file and any resources the application needs are packaged together into an .apk file by a component called ApkBuilder. Android .apk files must be “signed” in order to be installed on an Android device. This refers to a cryptographic signature which identifies the author who created the application package. The .dex is optimised into an optimised dex format (.odex). Amongst other things this process verifies that the byte code does not perform illegal operations. When the application is run, the byte code used is that in the .odex file.

308. Applications may be pre-installed, in which case they are placed onto the FLASH of the HTC device at the same time as all the other software is installed, as part of an image of the entire system. Users can also download apps. When this occurs the Android Application Manager on the device unzips the .apk file.
309. There is no dispute that HTC are responsible for the sale of the HTC devices in the United Kingdom. The real debate as regards the product claims is whether the devices are products within those claims, at least when switched on. In relation to the method claims 15 and 18, there is a serious issue under s60(2) of the 1977 Act. In fact there is a point on s60(2) in relation to the product claims but I will mention it along with the other s60(2) issues. There was also reference in the pleadings to a point under s60(1)(c) (product obtained directly by the process) in relation to claims 15 and 18 but no argument was addressed to it by Gemalto and I will not address it.

Claim 1

310. On claim 1 the arguments are about the terms “microcontroller” “resource constraints” and “loaded into memory”. There is clearly an interpreter in the HTC device (the Dalvik Virtual Machine). There are clearly applications with source code in a high level language (Java) which is compiled into a compiled form (Java byte code). There might have been a point about whether the Android SDK is a converter if I had accepted Prof. Paradinas’ view about the meaning of converter (addressed in relation to the Caron paper). However I rejected that narrow interpretation. I find that the Android SDK is or includes a converter for post processing the Java bytecode into a minimized form suitable for interpretation by the DVM. Thus the process by which the apps are made and processed before being put into the FLASH memory satisfies all the requirements of claim 1.
311. On the “microcontroller” issue Gemalto’s infringement case stands or falls with the question of construction I have already decided. In my judgment the chip or chipset in the HTC devices is not a microcontroller. The CPU’s memory is not on the chip. The registers and cache are on-chip but as I have found, that does not turn a chip into a microcontroller. I find that the HTC devices do not infringe any of the claims of the patent since there is no microcontroller inside them.
312. Gemalto were able to point to one place in which the word microcontroller appeared in a relevant context. In the “ARM Systems Developers Guide” by Sloss, Symes & Wright (2004) Chapter 1 discusses ARM embedded systems and the RISC philosophy

which has made ARM processors so successful. Figure 1.2 is a diagram of a device. The device depicted has memory (ROM, SRAM FLASHROM and DRAM) which must be off-chip. The figure legend reads “An example of an ARM-based embedded device, a microcontroller”. I do not think this helps Gemalto. At best it is an example of the use of Gemalto’s construction of the term microcontroller, which I have accepted is used in the art but is not how I have construed the patent.

313. If I had accepted Gemalto’s construction of microcontroller then I think a difficult question would arise as to whether the HTC devices were truly dedicated systems. I accept, as Mr Mellor argued, that when GSM mobile phones first started, it would be fair to call the computers in them dedicated systems (c.f. the Analog Dialogue document) but the HTC devices, like many modern “smart” phones are quite different. From the point of view of a user, an Android phone is just as programmable as a laptop computer. It is really a hand held computer which has the ability to be a telephone as one of its capacities. Happily I do not have to resolve that question.
314. If the HTC devices were microcontrollers as Gemalto contends then I do not see why the application code is not “loaded in memory” as required by claim 1 by being put into the FLASH memory although Gemalto did not argue that it was. Gemalto’s argument focussed on the cache. It said that the relevant element of claim 1 was satisfied when the phone was turned on and the application code, along with the code for the DVM, was loaded into cache. This led to a lengthy argument about how cache works in the HTC devices, some of which was confidential. I will deal with that argument in case this action goes further but I will say at the start that on my construction of “loaded in memory”, the operation Gemalto relies on whereby blocks of code are put into cache for execution does not satisfy this part of the claim.
315. The following is not confidential. In order to run an application, blocks of data are copied from RAM into cache. This data will be the executable code to run the DVM and the executable code for the application itself. In this context the phrase “executable code” includes both instructions and information to be acted upon. HTC emphasised that these blocks are smaller than normal size of the application or the DVM. So at any one time, the cache will only contain part of the program(s) to be run. When the processor wishes to access an item of data in RAM, the memory controller first looks for it in the cache. If it is in the cache, the data is passed to the processor. Because there is a good chance the next piece of data a processor will want to access will be located close to the previous data, by putting data into cache in blocks, there is a good chance that the processor will run for quite a few instruction cycles just accessing the cache. Since cache can be accessed very quickly, the processor runs fast. When the data required is not in cache this state is called a “cache miss”. The data needs to be retrieved from the RAM. The confidential element of the argument was about whether it was possible for an item of data to get to the processor from the RAM directly in any circumstances or whether it had to go through the cache. In every case the data has to pass through what I will call the cache system. In fact the relevant processors have two caches, L1 and L2 but that extra layer of complexity seems to me to be irrelevant.
316. If I had accepted Gemalto’s case that the cache mattered and that putting data in the cache was an act of “loading in memory” within any of the claims, both of which I have rejected, then:

- i) I would have accepted Gemalto's case that when the processor is running from data in the cache, it would have satisfied the claim. The fact that cache misses take place very frequently and are inevitable could not mean the claim was not being infringed when the processor was using the cache.
 - ii) I would have accepted Gemalto's argument that in fact even whether there is a cache miss, the device would have satisfied the claim. The data always has to pass through the cache system.
317. Again, if I had accepted Gemalto's argument that putting blocks of code into cache is relevant then a further point arises. The DVM is too big to be loaded into the cache. What happens in practice is that only the parts of the DVM and application actually being executed are put into cache and the rest resides off chip in the RAM until needed. Gemalto submitted this still satisfied the claim but I am not so sure since claim 1 required "the interpreter" to be loaded in memory. The interpreter in the HTC devices is the DVM as a whole, not parts of it. However I do not have to resolve this point.
318. Finally on claim 1 I need to address resource constraints. Dr Greaves explained that as a matter of fact the HTC devices have sufficient resources to run the standard form of Java. I accept that. HTC submitted that this meant the devices did not "have a set of resource constraints" as required by claim 1. I do not agree that this conclusion follows from that premise. Plainly the available resources on these modern HTC devices are much greater than the ones available in the mid 1990s and referred to in the patent but I do not accept that a comparison with the particular values in the patent is relevant.
319. Prof. Paradinas exhibited a presentation about the DVM by Dan Bornstein of Google. This promotes Dalvik as a virtual machine to "run on a slow CPU, with relatively little RAM, on an OS [*operating system*] without swap space, while powered by a battery". In other words the DVM is promoted as a tool to be used when resources are limited. Gemalto also pointed out that the Android SDK converts the apps into a form in which they are smaller than they would otherwise be and results in a very significant saving of resources. This enables the HTC devices to run many more applications that they would be able to if they did not use the DVM and apps converted by the Android SDK.
320. In my judgment the chip in the HTC devices does have a set of resource constraints. The memory available, both in terms of RAM and longer term storage (FLASH or hard drive) is much less than a contemporary desktop system. Battery life is also limited. This element of claim 1 is satisfied.

Claim 3

321. No issue arises in relation to claim 3. If (which I have rejected) claim 1 was infringed, claim 3 would be infringed too.

Claim 8

322. Owing to the claim dependency of claim 8, I need to mention claims 4 and 7. Prof. Paradinas addressed claim 4 in his second report. There was no dispute about it. I

conclude claim 4 would be infringed. Neither side addressed claim 7 and I make no finding on the point.

323. To decide whether claim 8 would be infringed depends on the details of the conversion process carried out by the Android SDK. It is not disputed that the Android SDK comprises means for translating from byte codes in the compiled form (Java byte codes) to byte codes in a format suitable for interpretation by the interpreter (the DVM). There is no dispute that elements 8(a), (c) and (e) are satisfied. The dx tool in the Android SDK deals with the jumps as required by elements 8(a) and (e) and deals with byte code operands in a manner within element 8(c). The issues related to elements 8(b) and (d).
324. The main point is about the word “equivalent” in elements 8(b) and 8(d). HTC submits this means “semantically equivalent” that is to say that the instructions do the same thing. So far so superficially attractive. Then HTC argues that the Java byte codes are not semantically equivalent to the DVM byte codes because the Java virtual machine is a stack machine while the Dalvik virtual machine is a register machine. HTC argues that the semantics are close but they are different because of the different architecture of the virtual machines. HTC points out that in cross-examination Prof. Paradinas accepted that as a result of the difference in machine architecture the byte codes for the DVM are not semantically equivalent to byte codes for the DVM. In essence the point is that an instruction “take the first number off the stack and the second number off the stack, add them together and store the answer on the stack” has a different meaning from an instruction “add the contents of register 1 to the contents of register 2 and store the result in register 1”.
325. I reject HTC’s argument on this because it has lost sight of the patent specification. In a sense of course “equivalent” means semantically equivalent since the point of this process is to end up with a program which does the same thing as it did before, albeit using different byte codes. However as Gemalto pointed out the patent expressly states in paragraph [0088]:
- “In some embodiments, the card class file converter modifies the original byte codes 60 into a different set of byte codes designed for a different virtual machine architecture, as shown in Fig. 11”.
326. The example given in figure 11 is between two different kinds of stack architecture but that does not matter. It would be clear to the skilled reader that the patent recognised that the conversion process may involve replacing byte codes for one kind of machine architecture with byte codes for another. Plainly these would still have to be semantically equivalent so that the overall program did the same thing. In the example I set out above about adding to numbers, both instructions have the same essential meaning, they are just implemented in different ways to reflect the different machine architecture.
327. Moreover the fact that it is possible to find some individual byte codes in one instruction set which have no meaning in the other instruction set because of the difference in architecture does not advance this argument either. Dr Greaves pointed out that Java includes a byte code instruction “*iload_0*”. It means “push” (i.e. put) a value in local variable “0” onto the stack. He said the DVM cannot perform that

operation (because it does not have a stack). That is true but as Dr Greaves pointed out the DVM would perform a different operation, moving a value between a pair of specific registers. In my judgment this goes to show that although at a certain level of detail the instructions will differ because the machine architecture differs, in fact at a higher level byte codes can be regarded as equivalent when viewed in the context of the program as a whole. I reject HTC's meaning of "equivalent". It is too narrow.

328. Finally in relation to element 8(b) HTC argued that an example given by Prof. Paradinas of the conversion by the Android SDK of specific Java byte codes into generic DVM byte codes was not within the claim. Prof. Paradinas referred to some specific byte codes (such as *iconst_m1*, *iconst_0* and *iconst_1*). These all perform the same operation but using a different number (-1, 0 or 1 respectively (m1 referring to -1)) which is built into the opcode itself. He said they were converted by the dx tool into a DVM operation "*const/4 vA #+B*" or "*const/16 vAA, #+BBBB*". Leaving aside the detailed meanings of the codes, there is no doubt this operation performs the same thing as Java "*iconst*" opcodes but now the operand B must be supplied. In the conversion process B would be set at -1, 0 or 1 as appropriate. Dr Greaves did not accept this was an example of the conversion required by claim 8(b) because "*const/4*" was not the generic byte code. That is because there is a form of the operation which is more general again.
329. All this demonstrates is that there are levels of generality from very specific to very general. I reject the argument implicit in this approach that the skilled reader would interpret claim 8(b) as being limited to a case in which a specific byte code was converted to the most generic possible equivalent byte code. I find that the Android SDK does perform operations which satisfy claim 8(b) and the example referred to above which was given by Prof. Paradinas is an example of that.
330. The argument on element 8(d) also involved a point of construction. Prof. Paradinas gave examples in which equivalent opcodes (for "goto" instructions) were renumbered during the conversion process. HTC argued that one cannot sensibly speak of "renumbering" byte codes when moving from Java byte codes to byte codes for the DVM. The byte codes may have different numbers but that is because they are byte codes for different machines with different instruction sets. What claim 8(d) is talking about (in the context of Java) is giving a Java byte code a different number. Essentially HTC's argument is that this element is only applicable in the case of a conversion which is less drastic than a conversion to an instruction set for a machine with a different architecture. Although I agree that HTC's argument correctly characterises the renumbering described in the specification at paragraph [0087], I agree with Gemalto that the argument is really seeking to read words into the language of the claim which are not present. There is no reason to read limitations into the feature which are absent. On that basis the conversion carried out by the Android SDK falls within claim 8(d).

Claim 15

331. Infringement of claim 15 raises a number of issues in common with infringement of claim 1. They are "microcontroller", "resource constraints", "loaded in memory". The same conclusions follow for those points which means claim 15 cannot be infringed on any view. A question specific to claim 15 is whether the first intermediate code and the second intermediate code are each interpretable by a virtual

machine. In this case they clearly are since the first code is interpretable by a JVM and the second by a DVM.

332. This leaves the question of whether selling the HTC devices can infringe claim 15 under s60(2) of the 1977 Act. I will address that below.

Claim 18

333. No separate issues arise under claim 18. The conclusions as regards sub-paragraphs 18(a) to (e) follow from the corresponding conclusions relating to claim 8(a) to (e). The same issue under s60(2) as for claim 15 also arises.

Section 60(2) of the 1977 Act

334. Section 60(2) and (3) provides:

“(2) Subject to the following provisions of this section, a person (other than the proprietor of the patent) also infringes a patent for an invention if, while the patent is in force and without the consent of the proprietor, he supplies or offers to supply in the United Kingdom a person other than a licensee or other person entitled to work the invention with any of the means, relating to an essential element of the invention, for putting the invention into effect when he knows, or it is obvious to a reasonable person in the circumstances, that those means are suitable for putting, and are intended to put, the invention into effect in the United Kingdom.

(3) Subsection (2) above shall not apply to the supply or offer of a staple commercial product unless the supply or the offer is made for the purpose of inducing the person supplied or, as the case may be, the person to whom the offer is made to do an act which constitutes an infringement of the patent by virtue of subsection (1) above”.

335. Jacob LJ considered aspects of the law on this in *Grimme Landmaschinenfabrik GmbH v Scott* [2010] EWCA Civ 1110. Arnold J considered further aspects in *Nestec* at paragraphs 154 *et seq.* Given that I am only considering s60(2) in deference to the arguments and in case this action goes further I will not analyse the law any further.

Assessment

336. There was argument about the pleadings. HTC said that for Gemalto’s case to succeed Gemalto had to establish that the steps of compilation and conversion of the relevant code took place in the UK but had not done so. The significance of this argument varies between the product claims and the process claims and I will address it in context.
337. Claims 1 and 8 are product claims. Gemalto’s infringement case is that a product within those claims does not exist when the power is switched off because the cache is empty. The relevant product is created when a HTC device is switched on and an

application is run on a DVM. At that point a product within the claim is created because the relevant code is “loaded in memory” by being placed into cache. I have rejected this argument (above) but if I accepted it (and the rest of Gemalto’s case on infringement of claim 1) then in my judgment an HTC device, switched off, as supplied by HTC to a customer in the UK is a means relating to an essential element of the invention for putting the invention into effect in the UK. It comprises a microcontroller and a memory (on Gemalto’s case) as well as an interpreter ready in the FLASH to be used when required. The device is set up to allow users to download apps for themselves. Once a user has downloaded an app, the code stored in the FLASH is code in a minimized form which has been prepared in the manner required by claim 1. HTC also knows this and intends that their customers should download these apps in the UK and use them in the UK. That use is what puts the invention into effect in the UK.

338. For the point of view of asking whether the device, when switched on and used to run an app in the UK, is a product within claim 1 on Gemalto’s infringement case, it does not matter where the application code was compiled nor where it was converted nor where the app was downloaded. Since these things do not matter, HTC’s state of knowledge about where those things might have happened does not matter either. Thus if, which I reject, Gemalto’s infringement case on claim 1 was sound, HTC would infringe under s60(2). The same goes for claim 8.
339. Claims 15 and 18 are method claims. The method of claim 15 has four steps; “inputting”, “compiling”, “converting” and “loading”. The first three steps do not take place on the microcontroller at all and that is so despite the reference to a microcontroller in the first words of the claim. In the claim there are also two clauses which define the first and second intermediate code as being interpretable by a first and second virtual machine but these are not method steps, they are definitions which define characteristics of the code.
340. The last step in claim 15 is loading the second intermediate code into the memory of the microcontroller. On Gemalto’s infringement case that is satisfied by putting a block of code into cache during the execution of an app. It happens inside an HTC device when an app is executed. Gemalto contends that the HTC device as sold is a means relating to an essential element of the invention for putting the invention into effect in the UK because it will permit users to take that last step of the method in the UK. All the other steps will have been performed by someone else, such as the developer of the app. The best way of putting Gemalto’s case here is to consider an app developer in the UK, performing all the steps of compilation and conversion in the UK. The HTC device allows the user to complete the claimed process and as a result the whole process will now be or have been performed in the UK. In effect Gemalto says that HTC is selling the physical items referred to in the claim (a microcontroller and memory) which are crucial elements necessary for the process to be completed and are configured to facilitate that process.
341. I do not have to decide the s60(2) question. I will only go as far as to say that I think Gemalto may be right that on its best case, the HTC devices are a means relating to an essential element of the invention in claim 15 for putting it into effect in the UK.
342. HTC denies that Gemalto has pleaded or established that the compilation and conversion steps for any apps have taken place in the UK or that HTC know or intend

anything relevant in that respect. It is plain that some apps will have been created in the UK and compiled, converted, downloaded and used within the jurisdiction and that this is all obvious to a reasonable person in the circumstances of HTC. It is true that the jurisdictional element in relation to the s60(2) issue was not developed in the pleadings and should have been but this applies to both sides. Gemalto ought to have pleaded the point properly but HTC ought to have covered it properly in their product and process description since otherwise they ought to have given disclosure on the issue of knowledge. If I had to decide the point, I would find that there are some Android apps downloaded in the UK for which all the steps of inputting, compiling and converting in claim 15 have also been carried out in the UK and that HTC know this or it is obvious to a reasonable person in HTC's position such that s60(2) would be satisfied.

343. Claim 18 stands or falls on this with claims 15 and 8.

The 9062 patent

344. The 9062 patent, entitled "Smart Card Reader", claims a priority date of 30 May 1995. Priority is not challenged. HTC contends that the patent is invalid in that it lacks novelty and/or is obvious over two citations:

- i) 'European digital cellular telecommunications system (Phase 2); Specification of Subscriber Identity Module – Mobile Equipment (SIM – ME) Interface for SIM Application Toolkit (GSM 11.PQ)'; version 0.0.2, dated April 1995; and
- ii) European Patent Application number EP 0 490 445 A2 published on 17th June 1992.

345. The first citation is known as "GSM 11.PQ". Gemalto does not accept that this formed part of the state of the art at the priority date. The second citation is known as Diehl. It is agreed that Diehl formed part of the state of the art. Gemalto denies that the patent lacks novelty or is obvious over GSM 11.PQ or Diehl. By the closing Diehl was advanced for obviousness only.

346. Gemalto contends that the following HTC mobile communication devices infringe the patent: the HTC Sensation, the HTC Sensation XL with Beats Audio, the HTC Sensation XE with Beats Audio, the HTC Explorer, the HTC Rhyme, the HTC EVO 3D, the HTC Cha Cha, the HTC Salsa, the HTC Incredible S, the HTC Desire S, the HTC Desire Z, the HTC Desire HD, the HTC Wildfire S and the HTC Gratia. This list of devices is not the same as the list in relation to the 865 patent. Collectively these have been referred to as "the HTC Smartphones". Nothing turns on any differences between these devices. It is common ground that these devices are capable of operating in accordance with a standard issued by the European Telecommunications Standards Institute (ETSI) number TS 102 223. This standard is known as the 233 standard. The issue of infringement is to some extent a squeeze argument in that HTC says that if operating in accordance with the 223 standard infringes then the patent must lack novelty over GSM 11.PQ because they are the same in all material respects. Gemalto does not accept that GSM 11.PQ is the same as the 233 standard. No point on s60(2) arises on the 9062 case. The issue is whether one of the HTC smartphones, with its SIM card installed, falls within the 9062 patent. In the 233 standard a SIM card is called a UICC. Nothing turns on the difference.

9062: Background

347. By 1995 smart cards were used in a number of spheres including in bank cards and in mobile telephones. At that time the GSM mobile telephone system was beginning to become successful and popular. The SIM card used in a GSM mobile phone is a smart card. A smart card always needs to have a reader. In banking the reader could be the ATM machine which would dispense cash when a bank card was inserted and the PIN transaction completed. In a mobile phone the reader is the phone itself.
348. In order for the reader to communicate with the smart card, a protocol governing that communication is needed. As at 1995 ISO 7816 was an international standard relating to smart cards. Part 3 (i.e. 7816-3) concerned the electrical interface and transmission protocols. The most commonly used protocol at that time was the one defined in the 7816-3 standard called T=0. There was another protocol in the standard, called T=1, but it was much less prominent.
349. In the T=0 protocol the reader always initiates the interaction with the card. The card cannot send a message to the reader other than by way of a response to a message sent by the reader to the card. In this field a relationship of this kind can be called a master and slave relationship. In T=0 the reader is the master.
350. In T=0, data are sent between the card and the reader in application protocol data units (“APDUs”). These have different structures depending on whether they are sent by the reader (“Command APDUs”) or by the card (“Response APDUs”). Command APDUs and Response APDUs form command-response pairs.
351. The ISO 7816 standard applied to smart cards in general. In the context of the GSM system, ETSI was the source of the standards. The ETSI GSM standard 11.11 contains much the same information as ISO 7816 but is specific to mobile phones.
352. The invention claimed in 9062 relates to the communications between a reader and a smart card.

9062: Witnesses

353. HTC called expert evidence from Dr Kristian Woodsend. He is currently a research associate in mathematics at Edinburgh University, but between 1993 and 2004 he worked in smart cards and mobile telecommunications at Panasonic, Aspects, NEC and Cellnet. In 1995 he was working initially for Cellnet as a GSM Development Engineer, before moving during that year to work for NEC Technologies as a Senior Systems Engineer. Between 1994 and 2000 he attended meetings of Special Mobile Group 9 (“SMG9”) of ETSI on behalf of his employers. He drafted most of GSM 11.PQ and presented it to SMG9 at a meeting in Nynashamn in May 1995. The GSM 11.PQ document is one of the items of prior art HTC rely on.
354. Gemalto submitted that Dr Woodsend was a careful witness and was “on the whole” fair. I thought Dr Woodsend was a careful and authoritative witness and was entirely fair.
355. Another matter arises from the manner in which Dr Woodsend was instructed and how that was described in his report. As with Dr Greaves in the case about the 865

patent, HTC's legal team went to some lengths only to show Dr Woodsend the patent in suit after they had asked for his comments on the prior art. This then led to trouble in cross-examination because a part of his report which he had said reflected his views expressed without seeing the patent included a claim chart. Self evidently the text must have been written after he saw the patent and Dr Woodsend was clearly puzzled about how he could have appeared to have made a mistake like this when it was put to him in cross-examination. In my judgment there was no mistake at all. I am sure Dr Woodsend did indeed formulate the basis for his views before seeing the patent and, when the text was prepared, felt that the text expressed views he had already formulated. The text put them into a proper context to show how they related to the patent. Dr Woodsend was clearly concerned he had done something wrong.

356. I refer to what I said about this exercise when it was undertaken with Dr Greaves at paragraphs [269-276] above. I will add the following. The approach of withholding the patent from an expert might be thought to imply that an item of prior art which is only identified after the expert has already seen the patent could not be successfully deployed in an obviousness attack. But that is incorrect. A related problem was present in this case. Dr Woodsend was the author of GSM 11.PQ and worked closely on it. To isolate Mr Woodsend from the ideas found in the patent when he was first considering Diehl was an impossible task. At least one of the key ideas absent from Diehl but present in the patent (implementing a proactive SIM arrangement using the T=0 protocol) is present in GSM 11.PQ. The witness has to rely on the legal team to tell him or her what is important and what is not in the context of the preparation of their reports. He or she can properly be questioned about it in the witness box. This may be entirely legitimate but if the problem has arisen as a result of the way the legal team has instructed the expert it can be unfair on the witness. That is what happened in this case.
357. HTC called factual evidence from Mr Michael Clayton. His evidence related to the terms on which the GSM 11.PQ document was distributed. I will deal with his evidence in context.
358. Gemalto relied on Professor Paradinas. My views on Professor Paradinas as a witness in the 865 case have been expressed already above. After giving evidence on 865, the Professor was released from his oath. He returned to the witness box a few days later to give evidence about 9062. On this second occasion he was a different person. I regret to say that I did not find the Professor's evidence on 9062 to be helpful. The views expressed by Prof. Paradinas in the course of cross-examination appeared to be ideas which were simply occurring to him as he sat in the witness box and were not the product of careful thought about the issues in the case. He appeared to have no memory of or even, I regret to say, understanding of the evidence in his reports. He raised new ideas which were not foreshadowed before. I suspect that there is some force in Mr Burkill's submission that Prof. Paradinas was very tired after dealing with the 865 case both as a witness and sitting in court to assist in the cross-examination of Dr Greaves. That may be but I do not believe it is a complete explanation. Moreover even if it is an explanation, it does not allow me to place any real weight on the Professor's evidence. During his evidence I thought Prof. Paradinas might find it easier if he allowed the interpreter to interpret the questions into his mother tongue and answered in the same language but he did not take up my suggestion. I suspect, although Prof Paradinas did not say this himself, that the nature of his expertise meant

that he was much more comfortable technically in relation to the issues arising on 865 than on 9062. I will be wary of placing any reliance on Prof Paradinas's opinions on 9062.

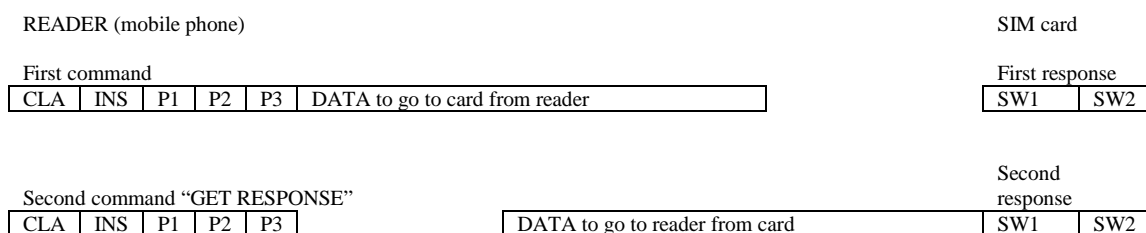
359. Mr Tappin submitted that what happened in relation to 9062 shed light on the Professor's evidence on 865. I have given this careful thought but I do not accept the submission. In formulating my view of Prof. Paradinas as a witness I have considered all his evidence as a whole but as I have said he behaved differently in relation to 865 from 9062. My views about the Professor in relation to the 865 patent are expressed above.

9062: Person skilled in the art

360. The skilled person will be someone working in the field of smart cards and in particular would be concerned with smart card – reader communications. They will have a degree in electronics or computing.

9062: Common general knowledge

361. The matters of background set out in the section above (save for the last paragraph) were part of the common general knowledge. A number of further points arise.
362. GSM 11.11 incorporated the T=0 protocol from ISO 7816-3 and applied it to the communications between the SIM and the telephone hand set. It also defined certain functions which cards had to support. These including functions like READ RECORD and UPDATE RECORD which allowed the card to be used as a form of data store. One command which was mandatory was GET RESPONSE. In fact the GET RESPONSE command was part of a draft standard called 7816-4 but whether 7816-4 was or was not common general knowledge is in dispute and I will address it below. What is not in dispute is that GSM 11.11 was common general knowledge and it defines GET RESPONSE. The GET RESPONSE command is used when it is needed to transfer data to the card and also get data from the card. The need arises because in the T=0 protocol it is not possible to do that in one command-response pair. The diagram below is based on one in GSM 11.11. The commands from the reader are on the left and the responses from the card are on the right. Commands are always initiated by the reader. They tell the card what to do in a five byte header consisting of CLA (class byte), INS (instruction byte) and P1, P2 and P3. P3 defines the length of the data field. The response always contains two status words (SW1 and SW2) which encode the response to the command (e.g. 90 00 means successful execution). The way in which GET RESPONSE works is depicted below:



363. HTC contended that the draft standard 7816-4 was common general knowledge by the priority date. The point of the argument about 7816-4 is that it defines a way for formatting the data field in an APDU using an approach called TLV for tag, length, value. TLV comes up later in relation to the obviousness attack based on Diehl. At this stage the issue is whether 7816-4 was common general knowledge.
364. 7816-4 had been going through the ISO standardisation process for some years. In 1993 a vote was taken to elevate it to a “Draft International Standard” (DIS) but the DIS status was rejected because of inconsistencies in the document at that time. By the priority date (30th May 1995) it had achieved DIS status but it only became a full standard in September 1995, after the priority date. HTC pointed to some pre-priority date literature including Smart Card News in which a number of manufacturers refer to 7816-4 in the context of new products they are introducing.
365. Gemalto accepted that 7816-4 was well known to the skilled person at the priority date but did not accept that it was regarded as a good basis for further action. The reference to “good basis for further action” is a reference to the well known formulation of what represents common general knowledge in General Tire v Firestone [1972] RPC 457 in the passage starting at p482. Gemalto argued that because the standard was still a draft and given its history, the skilled person could not know whether it would ultimately be adopted unchanged from the DIS form. Dr Woodsend said that everyone in the industry would know by May 1995 that the vote to accept 7816-4 had been passed. Prof Paradinas said that all the skilled persons knew there was a battle around the standard and knew that if they decided to implement it they were taking some risk.
366. Since smart card manufacturers were producing cards which were 7816-4 compliant, they clearly regarded it as a good enough basis on which to take further action. Whatever risk that might have involved was not high enough to deter those skilled in this art from acting. I find that 7816-4 was part of the common general knowledge. In any event even if (which I do not accept) 7816-4 as a whole was not common general knowledge, I find that the aspect relating to the TLV data field was sufficiently crystallised to be part of the common general knowledge.

9062: The patent

367. The language of the patent is French. Subject to one point, no issues of translation arose and the proceedings have been conducted using an English translation. All the references are to the translation unless otherwise stated.
368. The 9062 patent starts by describing known readers for integrated circuit (IC) cards, which may be self-contained or transparent (i.e. act as a port to a computer system). It notes that they often use the communication protocol defined in ISO 7816-3. It goes on to say that there are drawbacks to the intelligence lying in the reader and that while it has been suggested to transfer intelligence (i.e. transaction management) to the card, that raises the issue of compatibility of such modified cards with existing readers.
369. At p.3 lines 1-16, 9062 explains that:

“The present invention...relates to a general-purpose reader for various types of smart IC cards, using a single data exchange protocol which is compatible with the one used by unintelligent IC cards for exchanging data with their specialized readers.

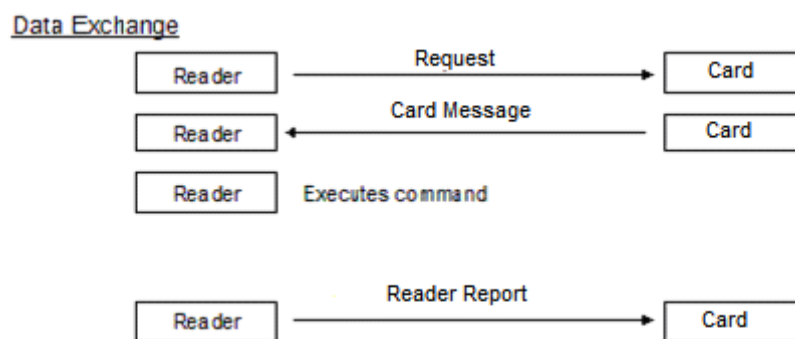
It relates to a smart IC card reader which is noteworthy in that it remains in control of the exchanges of information with a connected smart IC card, which take place on its own initiative, while simply having the function of execution in the performance of the transaction, which takes place on the initiative of the IC card.”

370. It goes on to say that advantageously the request by the reader for a card message is a command of the GET RESPONSE type, while the report by the reader consists of a command of the ENVELOPE or EXECUTE type. These are described in more detail at p.4 line 33 – p.6 line 10.

371. At p.6 line 11 – p.7 line 12 the 9062 patent describes the sequence of events which take place once the card is inserted in the reader. Dr Woodsend accurately summarised them in paragraphs 70-71 of his first report as follows:

“Briefly, the reader sends the card a reset command which initialises the card. The card then sends an answer to reset to the reader and begins the transaction management program programmed on the card.

There follows two steps: (A) a processing cycle by the smart card (in which the card is running its transaction management program), and (B) a data exchange cycle between the card and the reader. The processing cycle does not entail any commands being sent between the reader and card. The data exchange part of the process can be illustrated as follows:



372. At p.6 lines 17-24 the 9062 patent explains that the first processing cycle on the smart card leads to the preparation of the first card message:

“which it will be possible to communicate to the reader as soon as the latter has asked for it by means of a message provision request in the form of a “get response” command.”

373. Further processing and data exchange cycles can then follow – see p.7 lines 13-31. At p.7 lines 32-36 the 9062 patent explains that:

“The processing cycles, instigated by the smart IC card, and exchange cycles, instigated by the reader, thus succeed each other according to the transaction management program stored in the smart IC card.”

374. Finally, at p.7 line 37 – p.8 line 2, the 9062 patent states that:

“According to standard ISO 7816-3, the reader is in control of the exchanges in electrical terms, but the transaction runs at the instigation of the IC card which is a smart card.”

375. The essence of this is an adjustment to the known master-slave relationship between the reader and the card. Although the reader remains in control of the data exchanges and their instigation, the card is telling the reader what to do.

9062: Claim construction

376. As granted the ‘9062 patent had five claims. Gemalto applied to amend and since there was no opposition to the amendment, on 21st January 2013 the patent was amended. In their form as amended, claims 1 and 2 were said to have independent validity but in the end there was no need to consider claim 2 separately. The only claim which falls to be considered is claim 1. It provides:

Reader (1) for a smart IC card (2), characterised in that it includes:

- (a) means for managing, on its initiative, the exchanges of information with a connected smart IC card (2),
- (b) means for receiving and processing instructions and data received from the connected smart IC card (2), which manages a transaction on its initiative, and
- (c) means for developing and transmitting report messages to the connected smart IC card (2) on execution of its instructions by the said reader (1)
- (d) and further characterized in that the said means for managing the exchanges of information with the connected smart IC card (2) alternately and repetitively generate, for the purpose of being sent to the connected smart IC card (2),
 - (i) on the one hand a request for provision of a packet of instructions and data developed in the said smart IC card (2), this being referred to as the "card message" and,

- (ii) on the other hand, a report declaration associated with a report message regarding the execution of instructions previously received in card messages from the said connected smart IC card (2), the said report declaration and the report message being referred to as the "reader report".

377. The principles relating to claim construction were stated above in relation to 865.
378. It should be noted that the claim is to a reader with various characteristics.
379. Paragraph (a) would be understood to mean that the reader sends a message first to which the card responds.
380. Paragraph (b) refers to a transaction. Gemalto submitted that a transaction had to involve two cycles of instructions from the card to the reader. I doubt "transaction" is a term of art but for what it is worth Dr Woodsend said a transaction was a single unit of work that involves sending commands/responses between the reader and the card. Prof Paradinas said that something could not be a transaction if it involves only one command from the card. I do not accept that. The expression "transaction" is not being used in claim 1 in any sort of limiting sense. It is being used simply to refer to whatever task is to be performed. I reject the idea that it is limited to more than one cycle.
381. The reference in paragraph (b) to the card managing the transaction on its own initiative means that the card decides what messages to send to the reader in order to progress the transaction being run on the card. There was a suggestion that these words mean that the card had to be able to take decisions based on information present on the card itself and then take appropriate actions. Although I doubt it matters, I do not accept that this slightly limited view of paragraph (b) is correct. I expect in many cases the card will indeed take decisions based on information present on the card and take appropriate actions but there is no reason to limit the claim in that way.
382. Paragraph (d) requires that the reader (or to be exact the "means for managing the exchanges of information" which is an aspect of the reader) must "alternately and repetitively" generate the request and report specified in (d) (i) and (d) (ii) respectively. The word *alternately* defines the order in which they come: request then report; and the word *repetitively* means that one must have at least two cycles to satisfy the claim.
383. It was common ground that the claim did not exclude other different messages being sent interspersed amongst the relevant messages. The debate on construction was whether the claims were limited to a reader which can only generate requests and reports alternately and repetitively. In other words if a reader had the capacity to generate requests and reports alternately and repetitively in some circumstances but could also do other things, for example generate two reports back to back without a request in between, or a single cycle of a request and a report, then would it fall within the claim? Gemalto submitted that the claim was limited to a reader which could only generate requests and reports alternately and repetitively. HTC did not agree.

384. Mr Burkill drew an analogy with a random coin tossing machine, which might occasionally produce a pattern of alternate heads and tails. He submitted this sort of thing was outside the claim. In my judgment this analogy does not help. I doubt a random machine would be within the claim but that is because it is a random device. The issue is whether the claim covers a reader which is capable of generating the sequence referred to in the claim in some circumstances but is also capable of generating other sequences outside the claim in other (entirely deterministic) circumstances.
385. There is nothing in the claim language itself which supports Gemalto's interpretation. The word "only" is absent. Nor is there any passage in the specification which sets out to draw the distinction Gemalto seeks to rely on and explain that one is within the ambit of the invention disclosed and the other not.
386. I do not believe there is any support for this construction by considering the inventor's purpose. The purpose of 9062 is to provide a reader which has means for managing the exchanges of information with the card which allow the reader to remain in control of the exchanges of information (and so permit continued use of the T==0 protocol) and which also has means for receiving and processing instructions received and data from the card to allow the card to manage a transaction. So long as the reader can do what is called for in the claim I do not see why the skilled person would think the reader might not be permitted in some other circumstances to do other things.
387. For example perhaps in some circumstances the reader could also send a holding report to say that the execution of the request it has received is in progress but has not been completed, before sending a final report on the successful execution of the instructions. This cycle would involve a sequence of request, report, report. I can think of no reason why the skilled person would think the patentee wished to limit the ambit of the claim to exclude devices which operate in exactly the manner referred to with alternative and repetitive requests and reports simply because the device also has another distinct mode of operation: request, report, report.
388. I reject Gemalto's construction of claim 1 on this point.
389. In reaching this conclusion I have ignored the argument about the last paragraph of the specification. HTC said that properly understood this paragraph supported its position on construction because it would involve the handset stacking or queuing commands from other "passive" smart cards while it handled the instructions from the "active" smart card. That would not be possible on Gemalto's construction. Gemalto contended that the translation of this paragraph was wrong and asked the interpreter assisting Prof Paradinas to provide an alternative. I am not convinced the distinction between the two rival interpretations actually assisted Gemalto's argument but it is not necessary to resolve this dispute since even if Gemalto are right about the last paragraph it would not advance their interpretation, it would simply neutralise the point being put by HTC. Moreover I am not convinced HTC's interpretation of this paragraph helps anyway because in order for it to have a bearing on construction one needs to construct an elaborate analysis of how what is described in a rather delphic single sentence must work and then draw inferences from that about what the claim must mean. This approach may not be impermissible meticulous *verbal* analysis but it seems to me it is a similar impermissible kind of analysis. For the argument to work

at all the skilled person would have to think it was worth subjecting the relevant sentence to the sort of scrutiny and extrapolation required for HTC's argument in the first place and also to think that concrete conclusions could be drawn from the resulting inferences, none of which are expressed in the document. I reject that. A skilled person who got as far as HTC in the analysis of the last paragraph had no way of knowing if the inventor had thought it through to the same extent and could simply conclude that the inventor had not.

9062: *Novelty*

390. To be valid an invention must be new, which means it must not form part of the state of the art (s1(1)(a) and s2(1) of the 1977 Act). The state of the art includes all matter made available to the public before the priority date (s2(2) of the 1977 Act).
391. There was no dispute about the law of novelty. I referred to *Synthon BV v SmithKline Beecham plc* in relation to the novelty of the 865 patent.
392. HTC contends claim 1 of the '9062 patent lacks novelty over GSM 11.PQ. Gemalto does not accept that GSM 11.PQ forms part of the state of the art and, even if it does, Gemalto argues that claim 1 is novel over the document.

GSM 11.PQ - made available to the public?

393. As a matter of law, it is sufficient to make a document available to the public if it is communicated to a single person who is free in law and equity to make use of it for himself or herself. If the communication is encumbered with an obligation of confidence, expressed or implied, the communication has no invalidating effect (see paragraph 111 of the judgment of Floyd J, as he then was, in *Qualcomm v Nokia* [2008] EWHC 329 (Pat)). The burden of proving that matter was made available to the public lies with the party asserting it (Floyd J, paragraph 113). In this case that is HTC.
394. The factual background starts in 1982. At that time an organisation called CEPT (the European Conference of Postal and Telecommunication Organisations) set up a group to define a standard for digital mobile telephones. The group was called Groupe Spécial Mobile or GSM. In the late 1980s (1988 or 1989) GSM became an ETSI technical committee (although the precise relationship between the GSM and ETSI is in issue – see below). By the early 1990s the GSM mobile phone network was up and running. The term “GSM” now means Global System for Mobile Communications). The technical committee was renamed SMG (Special Mobile Groups) at some point. SMG had a number of sub-technical committees, SMG 1, SMG 2 etc. These had the task of developing technical recommendations about various aspects of mobile telephone technology such as radio and network aspects (SMG 2 and SMG 3), data services (SMG4) and so on. One of the SMG sub-technical committees was SMG9. SMG9 was concerned with integrated circuit cards (i.e. SIM cards). The SMGs were responsible for drafting and developing standards.
395. The individual sub-technical committees had meetings known as Working Group Meetings. There were also plenary meetings of all the Special Mobile Groups together. During 1995 SMG9 was responsible for developing the GSM 11.14 standard which defines the interface between the SIM and the mobile handset for something

called the “SIM Application Toolkit”. There were four working group meetings in 1995. The meeting relevant to this case was the second one, on 16-19 May 1995 at Nynashamn in Sweden.

396. In 1995 Dr Woodsend was a young engineer working at Cellnet for Colin Hamling. Mr Hamling was vice-chairman of SMG9 and Dr Woodsend worked with him and was also a member of SMG9. Dr Woodsend drafted the document GSM 11.PQ prior to its submission in advance of the Nynashamn meeting. The contents of the document incorporated his own thoughts and the thoughts of Colin Hamling and those expressed by other SMG9 members at earlier meetings.
397. A small point arose on the dates shown in the GSM 11.PQ document itself. Some pages are marked April 1995 and others March 1995 but Dr Woodsend explained this was an artefact of the way in which he prepared it and had no other significance. I accept that.
398. Dr Woodsend did not mark GSM 11.PQ as confidential. Dr Woodsend remembered discussing GSM 11.PQ at the Nynashamn meeting including within a splinter group meeting. He said those discussions were not designated confidential either. He said that documents discussed by SMG9, including GSM 11.PQ were not treated as confidential by any members of SMG9. The approach Cellnet took was that everything submitted to any ETSI group including SMG9 was public. In Dr Woodsend’s experience documents concerning the SIM Application Toolkit were distributed widely outside SMG9 and he has no reason to think GSM 11.PQ was treated any differently.
399. In cross-examination Dr Woodsend was barely challenged on this. It was suggested to him that he had prepared the document on behalf of ETSI but he explained that was not correct. It was also put to him that Cellnet did not submit any confidential material to SMG meetings out of an abundance of caution. Dr Woodsend said:
- “anything confidential we would have kept within the company and only discussed with other companies under a non-disclosure agreement and so on. We would never mention that within a group like SMG9. It was very much we had to be conscious, and I think all companies were like this, that we were in a public, effectively a public meeting with all our competitors, loads of companies from throughout the industry and no one was under any obligation not to disclose.”
400. The secretary of SMG9 from 1992 to 1997 was Michael Clayton. His role involved liaising with the Chairman, formulating meeting agendas, receiving and disseminating documents and drafting the minutes of meetings. From the records of the meeting, Mr Clayton confirmed that GSM 11.PQ was a document referred to by number in the meeting Agenda. He would have distributed copies of it at or before the meeting to the SMG9 distribution list. The list is attached to the minutes and includes over 30 names including many from well known telecommunication companies.
401. Mr Clayton explained that part of his role was to impose confidentiality restrictions on documents received by him for dissemination if they were designated confidential. If GSM 11.PQ was to be treated as confidential Mr Clayton would have expected to

see the document marked as such (it was not) and he would have recorded that fact in the minutes (he did not). Mr Clayton said that confidentiality was raised infrequently during his time as secretary of SMG9 and in practice he recommended that truly confidential documents should not be submitted to the applicable SMG at all. That was because it was common practice for ETSI documents to be freely distributed amongst the companies to which the SMG members were connected and to other entities such as national standardisation bodies.

402. Mr Clayton's witness statement describes the Special Mobile Groups as Sub-technical Committees of ETSI. His evidence referred to and quoted paragraph 10 of an ETSI policy governing the confidentiality of ETSI meetings and exhibits "a copy of the ETSI Policy that was in force in May 1995". In his evidence about common practice I have summarised above, Mr Clayton refers to ETSI documents.

403. Paragraph 10 of the ETSI policy Mr Clayton quoted states:

"The proceedings of a COMMITTEE shall be regarded as non-confidential except as expressly provided below and all information submitted to a COMMITTEE shall be treated as if non-confidential and shall be available for public inspection unless: - the information is in written or other tangible form; and - the information is **identified in writing, when submitted, as confidential**; and - the information is first submitted to, and accepted by, the chairman of the COMMITTEE as confidential."

[Mr Clayton's emphasis]

404. In his witness statement Mr Clayton emphasised this as being supportive of his evidence. He said that this version of the ETSI Policy applied to SMG9's Working Group meetings.

405. In cross-examination however Mr Clayton said something rather different. His oral evidence was that although ETSI had assumed the role of looking after GSM/SMG in the late 1980s, in fact SMG was largely autonomous. Although it operated under some ETSI rules, it did not operate under all of them. He drew a distinction between ETSI standards and SMG standards. For standards which started in SMG, the sub-groups and working groups would do their work and the standard would be approved at SMG (the Technical Committee) and then implemented. After that the standard would go to ETSI and ETSI "did what they did". The point was that once a standard was approved by SMG, it was put into practice by the industry.

406. The distinction Mr Clayton was drawing emerged at the outset of his cross-examination, while Mr Burkill was covering some basic background matters. However the significance of the distinction emerged when counsel put Gemalto's case to Mr Clayton. Gemalto's case is that, understood as a whole, the ETSI rules and procedures treated documents like GSM 11.PQ as confidential. They were available only to members of ETSI. However Mr Clayton's position was that irrespective of whether it is a correct statement vis a vis ETSI, it did not apply to SMG.

407. Given that Mr Clayton's witness statement drew no distinction at all between ETSI and SMG, this came as an understandable surprise to Gemalto. As Mr Burkill pointed out, the line being taken by Mr Clayton involved him contradicting even his own C.V. since that characterised him as an ETSI person at the relevant time when in fact, based on his testimony, he was not.
408. Mr Burkill submitted that Mr Clayton was an argumentative witness. I think that is a fair characterisation but it does not mean that Mr Clayton was unreliable and it does not mean he was wrong, mistaken or lying. It became clear that Mr Clayton was someone who had resisted or at least resented transfer of the GSM group into ETSI and he was at pains to emphasise the independent attitude of SMG. It is a pity that Mr Clayton's witness statement did not explain the point and in retrospect the witness statement could be regarded as positively misleading but I do not believe Mr Clayton set out to conceal anything.
409. Mr Clayton's oral evidence about SMG was clear and forthright. He was there at the time and was the secretary of two SMG working groups (SMG1 as well as SMG9). He will have known and understood the rules and procedures applicable to SMG at the time. I accept his testimony about the position of SMG in relation to ETSI. It follows from this that one cannot assume that even if, read as a whole, the ETSI rules and procedures would mean that a document submitted to an ETSI sub-technical committee remained confidential, this applies to GSM 11.PQ submitted as it was to SMG9.
410. Gemalto took a fair point based on the front page of GSM 11.PQ. It has an ETSI front sheet which purports to show ETSI asserting their rights over the document. However Mr Clayton explained that this was an error (by Dr Woodsend) and Dr Woodsend agreed he had used the wrong style sheet for the document.
411. Gemalto took another fair point that ETSI was a "members only" club and so documents were only distributed to members. Whether or not that limited circulation might have a bearing on the issues does not matter because Mr Clayton explained that in SMG he would circulate SMG documents very widely and not just to members but to non-members as well. Mr Burkill pointed out that this evidence was general and not specific to the GSM 11.PQ document. That is true but Mr Clayton was making the point to emphasise his position about SMG's independence from ETSI. It supports HTC's case that the participants in SMG meetings were not obliged to keep such documents confidential.
412. Since I have accepted Mr Clayton's testimony about SMG, Gemalto argument that GSM 11.PQ was not made available to the public by being distributed to SMG9 at the Nynashamn meeting must fail. It is plain, taking the evidence of Dr Woodsend and Mr Clayton together, that the document formed part of the state of the art.

GSM 11.PQ – disclosure

413. GSM 11.PQ defines the interface between the SIM and what is called the ME. ME stands for Mobile Equipment, i.e. the phone. The document defines mandatory ME procedures for the "SIM Application Toolkit". This is a set of commands and procedures for use during the network operation phase of GSM. One of the elements of the SIM Application Toolkit is a mechanism called Proactive SIM. Proactive SIM

provides a mechanism whereby the SIM can initiate actions to be taken by the ME. The mechanism stays within the T=0 protocol. The mechanism involves the SIM using the status words 91 XX to indicate that it has a message to send to the ME; the ME then uses GET RESPONSE to obtain that message. The ME then deals with the message from the SIM and reports back using the COMMAND RESULT procedure. This procedure involves using a T=0 protocol command called ENVELOPE to send a message to the card described as ENVELOPE (COMMAND RESULT). It is then up to the SIM to decide what to do next, including issuing another command using the same mechanism.

414. In Annex B GSM 11.PQ describes a number of examples of command sequences for Proactive SIM. Case 4 relates to a case in which the SIM has asked the ME to do something, the ME has encountered a problem and the SIM then asks the ME to try again. Case 4 is:

ME	SIM
Normal Command	Normal Data, if any 91 lgth
GET RESPONSE	Proactive SIM command 90 00
[ME performs command]	
COMMAND RESULT (temporary problem)	91 Lgth
GET RESPONSE	Repeat of proactive SIM command 90 00
[ME performs command]	
COMMAND RESULT (OK)	90 00

415. In this example the dialogue starts with a normal command from the ME to the SIM. The SIM responds in the normal way as in the T=0 protocol but uses the status words to signal to the ME that the SIM has something it wishes to say to the ME. In the example this is signalled by the number 91 in the first status word (see above). The ME then sends the GET RESPONSE message to the SIM and the SIM responds with a message which contains the command the SIM wishes to give to the ME. The command is called the Proactive SIM command. In GSM 11.PQ the possible commands are specified. They include commands like PLAY TONE which asks the ME to play a tone in its earpiece; SET UP CALL which asks the ME to set up a phone call; and GET INPUT which asks the ME to display a message on the screen and request a response, allowing a dialogue between the SIM and the user.

416. In Case 4 the ME then tries to perform the command but for some reason there is a temporary problem. The ME therefore sends a COMMAND RESULT message to the SIM which indicates that a temporary problem has occurred. The SIM responds (as it always will in the T=0 protocol) but the response status word contains the signal (91) which indicates that the SIM has something to say to the ME. Thus in effect the cycle is repeated. The ME knows to send a GET RESPONSE message to the SIM and does so. The SIM responds with a proactive SIM command, the ME performs the task and this time succeeds, the ME reports to the SIM with COMMAND RESULT and the

SIM's response this time uses the status words which indicate successful execution (90 00).

417. There is much more in GSM 11.PQ than the example in case 4 of Proactive SIM. The document proposes that an option would be to allow commands to be stacked in the ME and queued in the SIM. In order to facilitate stacking and to allow the system to keep track of commands, each Proactive SIM command must be given a number. One of the specified commands which the SIM can give to the ME is called ASK RESULT. This requests the result of a previous proactive command. GSM 11.PQ also specifies the kinds of result which the ME can pass back to the SIM. There are four: OK, result pending, temporary problem and permanent problem. The distinction between temporary problem and permanent problem is that if the problem is temporary, as illustrated in Case 4, the SIM understands that it might be worth trying again. The result "result pending" tells the SIM that the proactive command has not yet been completed.

Novelty of claim 1 of 9062 - assessment

418. The issue of novelty comes down to a single point. Clearly the ME is a reader and the SIM is a smart IC card. The exchanges of information between the reader and the SIM are managed by the reader at the reader's initiative. The reader can receive and process instructions and data received from the SIM and the SIM manages a transaction at its, i.e. the SIM's initiative. The reader has means for developing and transmitting report messages to the SIM on execution of its instructions by the reader. The reader can generate requests for instructions and reports, as called for by claim 1 and there is no dispute that, for example in Case 4, a sequence of requests and reports which is alternate and repetitive as required by the claim is disclosed.
419. Gemalto submitted that what is disclosed in GSM 11.PQ did not fall within claim 1 because although the equipment described in the document was capable of generating different sequences of requests and reports, that did not always happen. Thus on Gemalto's construction of claim 1, GSM 11.PQ did not anticipate. However I have rejected Gemalto's construction of the claim. In my judgment claim 1 covers what is disclosed in GSM 11.PQ. The reader (ME) described in GSM 11.PQ is a reader which has a means for managing the exchanges of information with the SIM which alternately and repetitively generates the request and response messages called for. Case 4 proves that. The fact that in other circumstances it does other things is irrelevant.
420. Accordingly I find claim 1 of 9062 lacks novelty over GSM 11.PQ.

9062: Obviousness

421. HTC contends claim 1 of the '9062 patent is obvious over GSM 11.PQ and also over Diehl. Since I have found that the claim is anticipated by GSM 11.PQ I will address obviousness on the hypothesis that claim 1 is to be construed as Gemalto submit. That is not relevant to the argument over Diehl anyway.
422. I will follow the *Pozzoli* approach to the assessment of obviousness referred to above in relation to the 865 patent.

423. I have identified the skilled person and the common general knowledge and construed the claim above. In terms of an inventive concept I will approach it, contrary to my decision above, on the basis that the invention is a reader which in the relevant circumstances cannot do other than perform the alternate and repetitive sequence of requests and reports.

GSM 11.PQ

424. I have dealt with the disclosure of GSM 11.PQ above. The difference between what is disclosed and the claim interpreted as Gemalto contends is that although the equipment described in GSM 11.PQ can operate by generating requests and reports alternatively and repetitively, in other circumstances it acts differently, for example by acting in the sequence “request, report, report”.

425. The argument about obviousness which came into focus during the trial was not one which had been canvassed in the experts’ reports, at least in part because Gemalto’s point on construction of claim 1 had not crystallised until trial. Nevertheless the parties’ rival positions and the views of the experts were clear enough by the closing. Gemalto’s position was that the GSM 11.PQ proposal was confusing and ill thought through in a number of respects. In summary the difficulties said to exist in GSM 11.PQ are:

- i) At some points the reader can decline to respond to commands which the SIM has issued;
- ii) The reader can stack more than one command. This has a number of difficulties. The card has no way of determining how many commands can be stacked. No matter how big the stack, it is always possible that it will be full and if so then ASK RESULT will not be able to be handled using the GET RESPONSE mechanism; and
- iii) The “result pending” message from the ME is useless. If there is no stacking it is pointless. Even if there is stacking, there is no provision for the card to decide to abort or continue after “result pending”.

426. Mr Burkill put to Dr Woodsend that the obvious way forward over GSM 11.PQ was to try and fix the problems mentioned while maintaining the ability to stack commands. The point of this argument is that if stacking is maintained then on Gemalto’s construction the result is still one which is outside claim 1. Dr Woodsend did not agree with the proposition put to him. He accepted that stacking/queuing would lead to problems but his view was that the skilled person would appreciate that and would implement the system without queuing / stacking. He pointed out that GSM 11.PQ itself states that stacking is optional. The thrust of his evidence can be seen from the following answer (at day 8 p969-970):

“A. A lot of these problems are related to queuing, or handling more than one command. It is clearly written that that is an option and I think a skilled person that has some knowledge of readers and how constrained they were in terms of memory and processing power and so on, they would think seriously about which option they want to go down, the sort of minimal option

of just doing one command at a time or possibly doing several commands at a time. I think it would cause them to think about the possibilities and consequences.”

427. Another element in the argument was that although, as Dr Woodsend said in the passage quoted above, a lot of the problems related to queuing, even if the skilled person decided not to implement queuing/stacking of commands, the problematic ASK RESULT command was not said in the document to be something which could be dispensed with if there was no queuing. The same point could be made about the problematic “result pending” response.
428. I should say that I did not find Prof Paradinas’ views on this element of the case to be of assistance.
429. The obviousness case over GSM 11.PQ involves two general considerations. First, as Mr Burkill said, it has always been recognised that invention can lie in simplification (e.g. Lord Herschell in Siddell v Vickers (1890) 7 RPC 292 at 304). Mr Burkill submitted that GSM 11.PQ was an over engineered proposal which stipulated mandatory features which were useless and were inconsistent with the simple alternation of requests and reports. It missed what he called the simplicity of the invention in 9062. That is a fair point and I will take this submission into account.
430. The second issue is the position of Dr Woodsend. He drafted GSM 11.PQ and after the Nynashamn meeting he continued to work in SMG9. After the priority date he was rapporteur to SMG9. The work led to the GSM standard which was the forerunner of the 223 standard in place today. Stacking is not permitted, ASK RESULT has gone and there is no result pending message. The steps which are said to be obvious have been taken. They were taken inside SMG9 with Dr Woodsend’s full involvement. I will bear this in mind as well.
431. An uninventive skilled person in 1995 who read GSM 11.PQ would see that stacking was described but could not help but see that it was optional. It would plainly be more complicated to implement such a system than it would to build a system with no stacking. I think it did not require an inventive step by this person to decide to implement the proactive SIM system described in GSM 11.PQ in a mobile telephone without queuing / stacking.
432. The only remaining question is whether such a person would also realise that without stacking there was no need for the ASK RESULT command or the “result pending” message. I think that is obvious too. A system which functioned without these elements would obviously function perfectly well. Once one has decided not to implement stacking they are obviously superfluous. I find that claim 1 of the 9062 patent, even if construed as Gemalto submits, was obvious.

Diehl

433. Diehl is an application by Thomson Consumer Electronics SA which concerns a method and apparatus for interfacing smart cards with terminals. It mentions PAY TV as a context in which the invention may be used. Diehl explains that the memory capacity and features of smart cards are increasing exponentially. The idea of Diehl is

to allow a smart card to interface with any terminal. In particular it allows the user to access new features in the smart card using the terminal.

434. Diehl sets out various commands which can be issued by the smart card and by the terminal. The list of commands in the section of Diehl headed “Invention” includes four issued by the smart card and four issued by the terminal. The commands listed obviously are designed to be used in a dialogue between the user and the card. Dr Woodsend produced a diagram which showed how some of the commands disclosed fitted together:

	User	Terminal	Smart card
(a)		ACTION LIST REQUEST	→
(b)			← SEND ACTION LIST
(c)		← Displays list of actions to the user	
(d)	Selects an action	→	
(e)		ACTION SELECTED	→

435. The terminal can issue the command ACTION LIST REQUEST whereby the terminal asks the card to send a list of actions. The card can issue SEND ACTION LIST which sends the terminal a list of actions which the terminal can present to a user on the screen of the terminal. Once the user has selected an action, the terminal can send the card ACTION SELECTED which tells the card which action the user has selected. Although Dr Woodsend’s diagram is not in Diehl itself, a skilled person reading the Diehl would understand it in this way.
436. Diehl is concerned with the idea of the card issuing commands to the reader (and vice versa). It is not concerned with the mechanics of how the two entities communicate. There is no reference for example to the T=0 protocol or any other equivalent transmission protocol. Although Diehl uses the term “communications protocol”, that expression is referring to the exchange of the commands described in Diehl, it is not referring to a transmission protocol governing communications between the terminal and the card.
437. A point was taken that the lists of commands in Diehl varies. I was not impressed with the argument. The point is that the list of commands under the heading Invention is shorter than the list of commands under the heading “Preferred embodiments” and there is a third list of commands in claim 1 of Diehl. However even to articulate the argument is to exaggerate its significance. The list of commands under “Preferred embodiments” is the same list of commands under “Invention” with two extras: CLEAR SCREEN and END OF SESSION. In fact CLEAR SCREEN is referred to in the Invention section but not given a place in the list. A skilled person would not be put off by this and would not think anything less of the disclosure. The list in the claim does not use the command names which means matching the lists takes a little time but it is not difficult. None of this makes the disclosure obscure or worthy of scepticism. In the cross-examination of Prof Paradinas the point about lists of commands in Diehl was one of the matters about which his evidence was confused.

438. Diehl clearly does not disclose a reader as claimed in claim 1 of 9062 since it does not address how the messages are to be exchanged between the reader and the card. HTC contend that claim 1 is obvious over Diehl because it would be obvious to a skilled person to implement Diehl using the T=0 protocol which was the predominant protocol at the priority date and if one did so then one would arrive at a scheme in which the reader behaved in the manner required by claim 1.
439. Prof Paradinas said the reader would approach Diehl with some scepticism, relying at least in part on the point about the lists of commands. I do not accept that. I think a skilled person would be interested in Diehl since it describes a change to the usual master/slave relationship between reader and card.
440. Dr Woodsend provided an example of implementing Diehl to set up a parental lock feature for PAY TV. He drew up the command sequence required. He thought it was an obvious implementation of Diehl. He thought the skilled person would do this using the T=0 protocol and using the TLV data field formatting approach set out in ISO 7816-4. It is clear that this command sequence is one which falls within claim 1 of 9062. The issue is whether it represents an obvious step for the skilled person to take.
441. Prof Paradinas did not think it was obvious but I am reluctant to place any weight on his opinions on this part of the case. However it does not follow that simply because I am not prepared to place weight on Prof Paradinas' opinion, I am bound to accept Dr Woodsend's view and bound to find that the claim is obvious over Diehl.
442. One of the points made in the evidence and argument is about the choice by Dr Woodsend of the command to use to display messages on the terminal screen. Diehl has two commands DISPLAY MESSAGE and DISPLAY ERASABLE MESSAGE. They are sent by the card to the terminal. The first one requires the terminal to display a message and waits for the next action performed by the smart card. The second one is identical to DISPLAY MESSAGE but the user will initiate message clearing. Dr Woodsend's protocol uses DISPLAY ERASABLE MESSAGE. Prof Paradinas pointed out that a command sequence using the other command, DISPLAY MESSAGE could not be implemented using the T=0 protocol because the screen would freeze. Dr Woodsend did not disagree but said in cross-examination that the answer was to use DISPLAY ERASABLE MESSAGE as he had done or include an extra message so that DISPLAY MESSAGE resembled all the other commands. Dr Woodsend thought that he would fix DISPLAY MESSAGE so that it looked like all the others; he said it was easier to do that than to invent a whole new protocol. However it seems to me that it is wrong to ignore the fact that, as disclosed in Diehl, the DISPLAY MESSAGE command indicates that it is the card which will take the initiative so far as both the next exchange of information and the next command are concerned. That is contrary to the invention claimed in 9062.
443. I am not persuaded that claim 1 is obvious over Diehl. My reasons are these. Diehl does not refer to a transmission protocol at all and there is nothing to indicate that compatibility with an existing transmission protocol like T=0 was an objective. Although it was the well known protocol at the relevant time it was not the only one and the skilled person did not have to use it. There is no master/slave relationship in Diehl so why would the skilled person assume one as far as the transmission protocol is concerned? A key element in Dr Woodsend's protocol is the TLV format from ISO

7816-4. However TLV fields were used to format data whereas, when pressed into service here the TLV mechanism would be used to send commands. Although the TLV format is common general knowledge, the way it would be used in Dr Woodsend's protocol was new. I have also mentioned the DISPLAY MESSAGE point. Although I recognise that it was not intended to be by either HTC or Dr Woodsend, I think the argument put forward over Diehl is based on looking back from claim 1 of 9062 and seeing how to arrive at the invention. At various stages in the reasoning a choice is made: adopting T=0, using TLV, and not using DISPLAY MESSAGE or altering it. At each point I am not convinced there is a reason to take the step in the manner which leads to the claim other than impermissible knowledge of the invention. I reject the argument that these various steps represent what a skilled person would do without knowledge of the invention.

9062: Infringement

444. The HTC smartphones implement the ETSI TS 102 223 standard. The standard uses different language from GSM 11.PQ and so for example GET RESPONSE has been replaced with FETCH and ENVELOPE has been replaced with TERMINAL RESPONSE. On my construction of claim 1, there is no dispute that the HTC smartphones all fall within the claim.
445. HTC had contended there was a squeeze in the sense that if a phone in accordance with the 223 standard infringed then the claim must cover GSM 11.PQ. That is not necessarily so because, as Gemalto pointed out, there are differences between the 223 standard and GSM 11.PQ. Under the 223 standard commands cannot be stacked, there is no equivalent in the 223 standard of the ASK RESULT command in GSM 11.PQ and the RESULT PENDING response in GSM 11.PQ has been dropped. So at least in theory it could be the case that a claim which covered the 223 standard may or may not also cover GSM 11.PQ.
446. Gemalto submitted that on their construction of claim 1 the claim was novel over GSM 11.PQ but the HTC smartphones were covered. Although it does not arise on my findings in this case I will state that I was not convinced that the HTC smartphones would infringe on Gemalto's construction. The HTC smartphones under the 223 standard are capable of operating in a way which has only a single cycle of a request and a report. In that case the reader would have sent *alternate* requests and reports but they would not be *repetitive*. Although Mr Burkill said that was within his construction of claim 1 I do not see how it can be. If the claim means that the reader must only operate by sending requests and reports alternately and repetitively then it seems to me the HTC smartphones do not satisfy that requirement.

Conclusions overall

447. In summary my findings are:

The 865 patent

- i) The 865 patent is not infringed by any HTC devices.
- ii) Claims 1, 8, 15 and 18 are not entitled to priority.

- iii) All claims which lose priority and therefore take the filing date are obvious over the intervening Cyberflex materials.
- iv) Claims 1 and 15 lack novelty over the Caron paper available at the priority date.
- v) Claim 3 is entitled to priority and is not obvious over the cited prior art available at the priority date.
- vi) None of the claims are unpatentable computer programs as such.

The 9062 patent

- vii) If the 9062 patent were valid, the HTC smartphones would infringe.
- viii) The 9062 patent is invalid in that it lacks novelty over GSM 11.PQ.
- ix) On Gemalto's construction it would be obvious over GSM 11.PQ.