

5 The manner in which the objects interact with each other may change critically during the development process, and therefore it is important to monitor the development of each object and to retain copies of the objects (versions) at various stages of their development. This is particularly important when the development of the objects is pursued along more than one branch. For example, if a program is required to interface with a customer's existing database, and the developers experience problems accessing the database, they may generate a new "branch" of objects specifically to resolve this particular problem. Typically the programs generated in this new branch would not require the full functionality of the main branch, so objects relating to eg. user interface, printing functions etc., can conveniently be left out of the new branch altogether to simplify the problem-solving process.

6 According to the invention, objects stored in the data structure are correlated in two tables. The first table, called an "object branches table", contains records based upon the branch and object identity, and only has one row for each object in a branch. Figure 3 of the patent application (reproduced above right) illustrates an object branches table. I have circled the current version of each object in the table.

200	201	202	203	204	205	206				
Branch	Object Identity	Version Sequence	Object Data			Predecessor	...			
210	X	A	5	ΔXA1	ΔXA2	ΔXA3	ΔXA4	XA5
211	X
212	Y	A	3	ΔYA1	ΔYA2	YA3	...	XA2
213	Y	B	1	YB1
214	Y	C	4	ΔYC1	ΔYC2	ΔYC3	YC4
215	Y
216	Z	A	3	ΔZA1	ΔZA2	ZA3	...	XA3
217	Z

7 The current "working" version of each object is stored as a complete file with the current version number being recorded in a column in the object branches table whilst previous versions are stored as a compressed delta file. The compressed delta file is a compacted record of all the changes that would need to be made to the current (or previous) version to recover that version. Each later delta file must be used in sequence to recover an older version. Delta files are indicated in the above figure by the prefix Δ, eg. ΔXA1.

8 The second table is called an "object versions table". It includes one row corresponding to each version of each object in each branch. An object may have any number of versions in any number of development branches. In turn each of these branches might be associated with other objects and versions. Figure 4 of the patent application (see right) is an example of an object versions table.

300	301	302	303	304	305
Branch	Object Identity	Version	Working
310	X	A	1
311	X	A	2
312	X	A	3
313	X	A	4
314	X	A	5
315	X
316	Y	A	1
317	Y	A	2
318	Y	A	3
319	Y	B	1
320	Y	C	1
321	Y	C	2
322	Y	C	3
323	Y	C	4
324	Y
325	Z	A	1
326	Z	A	2
327	Z	A	3
328	Z

9 It is clear from the description in the application, that the tables shown here are very much simplified for the purpose of explaining how the invention works. In practice, there is likely to be a very large number of data objects that would need to be accessed. According to the application, the use of conventional data

structures means that many hours of system developer time may be lost because of the time it would take for a computer to access a large number of objects.

The claims

10 There is only one independent claim in the application, claim 1. Claim 11, although a dependent claim, is also worth repeating here for reasons that will become apparent later. These two claims read as follows:

1. A computer-implemented data storage system for storing a number of versions of a number of objects, the versions of the objects being arranged in branch groups, wherein at least one version of at least one object defines a branch group, and wherein the versions of the objects are stored in accordance with their branch group, the system comprising:

an object branches table including a branch column, an object identity column, a version sequence column and an object data column, wherein data identifying the branch group, the object and the number of versions of the object in the branch group, are stored in the branch column, object identity column and the version sequence column respectively, and the versions of the objects in each branch group are stored in the object data column, wherein the object branches table is arranged such that only one row of the object branches table is provided corresponding to each object in each branch group, and wherein a number of the object versions in the object data column are stored in a compressed format; and,

an object versions table having a branch column, an object identity column, a version column and a working column, wherein data identifying the branch group, the object and the object version in the branch group are stored in the branch column, object identity column and version column respectively, wherein in use the stored versions of the objects are accessed by users to generate subsequent versions of the objects in the working column, and wherein the object versions table is arranged such that one table row is provided corresponding to each version of each object in each branch group.

11. A computer program comprising computer program code means for implementing the system according to any of the preceding claims.

The Law

11 The examiner has reported that the application relates to a program for a computer as such. This objection is based on section 1(2) of the Act, the essential parts of which are shown in bold below:

1(2) It is hereby declared that the following (among other things) are not inventions for the purposes of this Act, that is to say, anything which consists of -

- (a) a discovery, scientific theory or mathematical method;
- (b) a literary, dramatic, musical or artistic work or any other aesthetic creation whatsoever;

- (c) a scheme, rule or method for performing a mental act, playing a game or doing business, or **a program for a computer**;
- (d) the presentation of information;

but the foregoing provision shall prevent anything from being treated as an invention for the purposes of this Act only to the extent that a patent or application for a patent relates to that thing as such.

12 Mr Lord recognised that the Patents Court has recently provided some helpful guidance explaining how this section of the Act should be interpreted in the *CFPH* case², and he agreed that the correct test is the two stage test set out by the Deputy Judge in *CFPH*. That is:

- (1) Identify what is the advance in the art that is said to be new and not obvious (and susceptible of industrial application).
- (2) Determine whether it is both new and not obvious (and susceptible of industrial application) under the description “an invention” in the sense of Article 52 of the European Patent Convention (EPC) — broadly corresponding to section 1 of the Patents Act 1977.

Mr Lord’s submissions

13 In the course of his submissions, Mr Lord made several references to the *CFPH* judgment. In particular he drew my attention to the “little man” test in paragraph 104, and emphasised that the deputy judge clearly recognised that an invention is not excluded from patent protection merely because it uses a computer program. In order to appreciate the “little man” test in its proper context, it is necessary to consider paragraph 104 in its entirety. It reads as follows:

104. But the mere fact that a claimed artefact includes a computer program, or that a claimed process uses a computer program, does not establish, in and of itself, that the patent would foreclose the use of a computer program. There are many artefacts that operate under computer control (e.g. the automatic pilot of an aircraft) and there are many industrial processes that operate under computer control (e.g. making canned soup). A better way of doing those things ought, in principle, to be patentable. The question to ask should be: is it (the artefact or process) new and non-obvious merely because there is a computer program? Or **would it still be new and non-obvious in principle even if the same decisions and commands could somehow be taken and issued by a little man at a control panel, operating under the same rules? For if the answer to the latter question is ‘Yes’ it becomes apparent that the computer program is merely a tool, and the invention is not about computer programming at all.** It is about better rules for governing an automatic pilot or better rules for conducting the manufacture of canned soup.

(My emphasis)

14 According to Mr Lord, the “little man” test replaces the former requirement for technical contribution. He said it was a deliberate move on the part of the court,

²*CFPH LLC’s Application* [2005] EWHC 1589 Pat.

choosing a test which is easier to apply; primarily because it does not use the word “technical”, that so many have struggled with in the past. He also submitted that the “little man” test moves the UK position closer to the European Patent Office’s current practice and would result in more applications being granted by the UK Patent Office.

- 15 I think Mr Lord was putting too much emphasis on the little man test — at least, in the particular context of this application. Firstly, the ‘test’ was put forward as a means of assessing the function of a computer program where there is either an artefact or an industrial process that is operated under computer control (eg. an automatic pilot or making canned soup). In these situations, the little man test can be used effectively to show that the presence of a computer program is merely incidental to the real invention or, to use the words of *CFPH*, “that the computer program is merely a tool”. Secondly, it seems to me that the ‘test’ must take some account of processing speed since, as Mr Lord said at the hearing, in principle there is nothing that a computer can do that a human (regardless of gender and stature) cannot. Computers merely do most things much faster.
- 16 Thus it seems to me that the “little man” test is not really an appropriate question to ask in such a case as this. For not only is there no artefact or industrial process being controlled, but the application makes it clear that the purpose of the invention is to create a new data structure that can store objects more efficiently than conventional data structures — ie. so that the appropriate object versions can be accessed using less processor time than before. Therefore, as I understand the invention, a little man could not replace the computer in this system without defeating one of the fundamental purposes of the invention.
- 17 Leaving aside the little man test, I turn instead to the two stage test indicated in paragraph 12 above, because it seems to me that this is the fundamental test for patentability that is taught in *CFPH*. Firstly, the advance that is said to be new and not obvious is the use of an “object branches table” and an “object versions table” to manage the storage of data objects in a data storage system. The second step requires me to determine whether such an advance (assuming it is new and not obvious) is both new and not obvious under the description “an invention” in the sense of Article 52. I admit that I have not found it easy to address this step of the test in this case. To reach a clear decision, I have had to look further into the *CFPH* judgment, and consider some of the reasoning behind it.
- 18 Amongst the passages in *CFPH* to which Mr Lord referred me was paragraph 98 which reads:
98. ... Article 52 contains a series of exclusions. It is necessary to bear in mind the *reasons* for those exclusions, and in my judgment they are not uniform and the same. I have discussed them in paragraphs 34 to 41 above.
- 19 The deputy judge made similar remarks in paragraphs 27 and 31 where he indicates clearly that the various exclusions of Article 52 should be given a purposive or teleological interpretation. With regard to computer programs, the deputy judge states at paragraph 35 that the reason why they are not patentable is because:

“... at the time the EPC was under consideration it was felt in the computer industry that such patents were not really needed, were too cumbersome (it was felt that searching the prior art would be a big problem), and would do more harm than good. ... Copyright law protects computer programs against copying.”

20 A footnote to paragraph 35 further explains that:

“As late as 1971 the industry declared that it was content to be protected by the law of contract and trade secrets alone.”

21 The invention described (and claimed) in the specification of this patent application is not protected against copying by the law of copyright. A person who reads the patent specification and then writes a program that manages the storage of data objects in a data structure using the very ideas that he or she has learnt from the specification (ie. an object branches table and an object versions table) would not be infringing the applicant’s copyright. (On the contrary, the program that he or she wrote would attract copyright protection in its own right.) So perhaps Mr Lord would have been justified in claiming that because the invention disclosed in his application is not protected by copyright law, then it cannot be a computer program as such — because if it were a computer program as such, then it would be protected against copying under copyright law. This is an attractive argument, so far as it goes.

22 But *CFPH* goes further, and paragraph 103 states that:

103. **It was the policy of the “computer program” exclusion that computer programs, as such, could not be foreclosed to the public under patent law.** (Copyright law is another matter.) They would be foreclosed if it was possible to patent a computer when running under the instructions of the program, for example, or magnetic disk when storing the program. (My emphasis)

23 If computer programs are not to be foreclosed to the public, then it is clear to me that I cannot allow this application to proceed to grant. Not only would the present claims (if granted) foreclose computer programs to the public but, on my reading of the claims, there is little or nothing else that would be foreclosed by them. There can be no doubt of this with claim 11 because of the form in which it is drafted, but I think that in practice it is equally true of the other claims. I therefore conclude that the advance in the art that is said to be new and not obvious, is not “new-and-not-obvious” under the description “an invention” in the sense of Article 52.

24 Mr Lord also argued that the invention could not be a computer program as such because it did not merely automate a process that had been done before. Although I do not accept the logic of this argument, I did agree with Mr Lord at the hearing that the invention does not just automate something that has been done before; in that respect I acknowledge that the advance is new and inventive. But even at the hearing, I could not get away from the fact that it looks like a new and inventive data structure; and a data structure is essentially nothing more than a computer programming technique. More specifically, it looks to me like an advance in the field of computer programming that nobody outside the field of computer programming would understand or appreciate.

25 Nevertheless, I cannot say that I am entirely comfortable with the process by which I have arrived at my decision in this case. For one thing, it is possible that the new data structure at the heart of this invention could be said to represent a technical contribution. If that is correct, then this application would not have been refused in accordance with the practice of the UK Patent Office prior to *CFPH*. However, I am as confident as I can be that the decision I have reached is consistent with the ratio decidendi of the Patents Court in *CFPH*.

Conclusion

26 I have decided that the advance in the art that is said in this application to be new and not obvious (and susceptible of industrial application) does not satisfy these criteria under the description “an invention”. I have read the whole application carefully, and I cannot see any amendment that would overcome this deficiency. Consequently I refuse this application under section 18 on the grounds that the advance it describes and claims as an invention does not satisfy the requirements of section 1.

Appeal

27 Under the Practice Direction to Part 52 of the Civil Procedure Rules, any appeal must be lodged within 28 days of the receipt of this decision.

S J PROBERT

Deputy Director acting for the Comptroller