Intellectual Property Office

BL O/480/22

06 June 2022

PATENTS ACT 1977

APPLICANT	Worldpay (UK) Limited
ISSUE	Whether GB1507342.2 is excluded under Section 1(2) of the Patents Act 1977
HEARING OFFICER	Mr Peter Mason

DECISION

Introduction

1 Patent application GB1507342.2 was filed on 29th April 2015 and published as GB2537880A on 2nd November 2016.

2 The application relates to a computer implemented method for identifying units of redundant code in a system having a main code-base, which code-base include a set of deployable units each of which are loadable. The method comprises creating a list of required deployable units and comparing this to a list of available deployable units to identify potentially redundant deployable units.

3 The examiner considered that the invention relates to subject-matter excluded from patentability under section 1(2)(c) of the Patents Act 1977 ("the Act"), specifically to a program for a computer as such and has maintained an objection under section 1(2)(c) throughout the examination process. The applicant has attempted to overcome this objection through arguments contained in three responses (dated 28th May 2021, 2nd September 2021 and 11th February 2022) but has been unable to persuade the examiner that the invention has met the requirements of the Act. The examiner invited the applicant to request a hearing and, in the absence of such a request from the applicant, has asked that a Hearing Officer decide the matter based upon the papers on file.

4 The issue to be decided is whether the invention consists solely of a program for a computer, which the Act excludes from patentability under section 1(2)(c).

The invention

5 The invention is a computer-implemented method of identifying and optionally removing redundant code from the main code-base of a system. The main code-base of the system is described (in page 2 of the Attorney's letter dated 11th February 2022) as being a collection of source code used to build a program or application.

6 Page 1 line 21 to page 2 line 23 of the application as filed states:

"*It is often too costly in terms of time, money and/or resources to re-design the entire system from the ground up each time a new requirement is identified. Instead, additional functionality is often implemented as bolt-ons' to the code-base of the original system. This approach has the benefit of adding new functionality to the system relatively quickly and cheaply. However, it creates problems too: in adding the bolt-ons, existing portions of the system's code-base may become obsolete but are not recognised as such. These portions of the code-base may be referred to as 'unused' or 'dead' code. These obsolete functions remain unused by the system, bloating the code-base of the system and potentially consuming valuable resources. This state of affairs is common across most industries and is programming language agnostic.*

*If not actively managed, this problem of code bloat increases the level of complexity, obscurity and generally decreases the efficiency of both the development process and also the system during execution. This process has been known in the art generally as the accumulation of 'Design Debt'. The decrease in efficiency of the system during execution can be somewhat mitigated, with investment, by buying more hardware for the system. However, the problem of decreasing design efficiency is harder to resolve -even with hiring more developers as there are a couple of limiting factors: * There are a limited number of developers who understand the system and therefore there is a time lag before a newly hired developer is effective in the development process; and * As the number of developers increases, the propensity for developers to trip up on each other's heels grows, as the changes of one developer impact those made by the other. This problem will become more acute the less well-structured and separated the code-base is."*

7 The method of the invention identifies redundant code and allows for its removal from the main code-base of a system.

8 The latest claims were filed on 28th May 2021. There are two independent claims, a method claim 1 and a system claim 26. The claims differ in form but are substantially the same and will stand or fall together. Claim 1 is set out below:

1. A computer-implemented method for identifying redundant code in a system having a main code-base that is associated with an application component, wherein the main code-base includes a set of available deployable units that each have an associated unique identifier, wherein each deployable unit is loadable by a loading mechanism that is configured to load deployable units that are part of the set of available deployable units, and wherein each deployable unit has a corresponding source code deployable unit that is stored in a main source code code-base, the method comprising:

monitoring the system to create a list comprising at least one required deployable unit that is loaded by the loading mechanism during the monitoring;

comparing the list to the set of available deployable units to determine if at least one potentially redundant deployable unit that is part of the set of available deployable units but which was not loaded by the loading mechanism during the monitoring exists and, in the affirmative, the method further comprising:

modifying the set of available deployable units to identify the at least one potentially redundant deployable unit as potentially redundant; and

deleting the at least one potentially redundant deployable unit or moving the requested deployable unit back into the main code-base based on one or more activities of the loading mechanism.

The law

9 The examiner has raised an objection that the invention is not patentable because it relates to one or more of the categories of subject-matter which are not considered to be inventions under the Act. This 'excluded matter' is set out in section 1(2) of the Act:

> *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. [my emphasis]*

The Court of Appeal's judgement in Symbian[1] tells us that in order to determine whether an invention falls solely within the any of the exclusions listed in section 1(2), the four-step test set out in its earlier judgement in Aerotel[2] must be used. The four steps are:

(1) properly construe the claim(s);

(2) identify the actual (or alleged) contribution;

(3) ask whether it falls solely within the excluded subject-matter;

(4) check whether the actual or alleged contribution is actually technical in nature.

The fourth step of the test is to check whether the contribution is technical in nature. In paragraph 46 of Aerotel it is stated that applying this fourth step may not be necessary because the third step should have covered the question. I shall consider whether the contribution is excluded alongside the question of whether the contribution is technical in nature, meaning I will consider the third and fourth steps of Aerotel together.

Argument and analysis

Step 1 - Properly construe the claim

10 The examiner and attorney have each stated there is no difficulty in construing the claim. However, during the rounds of correspondence there has been discussion and disagreement as to what the "main code-base" of the system is, especially in relation to whether it may be considered to be at the level of the architecture of the system.

11 The examiner considers the code-base to be a program[3] while the applicant states that it is a collection of source code used to build a program or application[4].

12 Source code is widely understood to be the version of software as it is originally written (i.e., typed into a computer) by a human in plain text (i.e., human readable alphanumeric characters)[5]. Once created by a programmer the source code is often transformed by an assembler or compiler into binary machine code that can be executed by a computer processor. Therefore, I construe the code-base of this application to be a collection of program units available to programmers. The code-base as construed is not at the level of the architecture of the system.

Step 2 – Identify the actual (or alleged) contribution

13 Paragraph 43 of Aerotel suggests that the contribution can be assessed from the point of view of the problem to be solved, how the invention works and what the advantages are, stating "What has the inventor really added to human knowledge perhaps sums up the exercise". The examiner succinctly summarised these

[1] Symbian Ltd. v Comptroller-General of Patents [2008] EWCA Civ 1066
[2] Aerotel Ltd v Telco Holdings Ltd and Macrossan's Application [2006] EWCA Civ 1371
[3] paragraph 8 of the examination report dated 6th July 2021
[4] page 2 of the Attorney's letter dated 2nd September 2021
[5] http://www.linfo.org/source_code.htmlThe Linux Information Project

considerations as: “The addition to human knowledge is an improved method of identifying and removing redundant code, providing the advantage of simplifying the overall code used and making code designing and running easier and more efficient, solving the problem of having to manually find and remove redundant code”

14 There has been a full discussion of the contribution in the correspondence between the examiner and the applicant; this is publicly available at Intellectual Property Office - Patent document and information service (Ipsum) (ipo.gov.uk) . The discussion resulted in the examiner and applicant agreeing that the contribution is:

“ A computer implemented redundant code identification and removal system which allows for the efficient identification and removal of redundant code from the main code-base of the system”

15 However, this contribution is unclear in that there are two distinct systems present:

i) the first system that identifies and allows the removal of redundant code units, and

ii) the second system that is built using the code units of the code-base.

Also, it needs to be clear that the invention is applied to the main code base of the system but does not include it. Therefore, I will restate the contribution as:

“ A computer implemented method for identifying redundant code units in a main code-base of a system, which method allows for the efficient identification and removal of redundant code units from the main code-base of the system”.

Steps 3 & 4 - Ask whether it falls solely within the excluded subject-matter and check whether it is actually technical

16 It is clear that the contribution is put into effect by a computer program running on conventional data processing hardware.

17 To assist in determining whether the contribution relates solely to a program for a computer, we use the signposts to technical contribution set out in AT&T/CVON[6] and by the Court of Appeal in HTC v Apple[7]. These are:

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

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

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

[6] AT&T Knowledge Venture/CVON Innovations v Comptroller General of Patents [2009] EWHC 343 (Pat)
[7] HTC Europe Co Ltd v Apple Inc [2013] EWCA Civ 451

iv) whether the program makes the computer a better computer in the sense of running more efficiently and effectively as a computer;

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

18 These signposts are useful guidelines only, providing a list of some of the factors that can assist in determining whether a contribution may be technical.

19 The invention is carried out within a computer system and has no technical effect on any process outside the computer, so signpost (i) does not assist in identifying a technical effect. As discussed above, the code-base is a collection of available programming units which do not operate at the architectural level of the computer, and so signpost (ii) does not assist. The computer is not operating as a computer in a new way except in so far as any computer running a new program operates in a new way, therefore signpost (iii) does not assist.

20 In relation to signpost (iv), the applicant argued[8] that a program for a computer is not the same as the main code-base of a system, stating that:

> "*a program….is developed to be installed and to run on the system when the program is initiated, whereas a main code base of a system comprises a collection of source code used to build a program or application. While it may be correct to say that the program running on a computer may, as a result of the present invention, run more efficiently, it is incorrect to say that the computer per se is not running more efficiently as well*".

Page 3 of the same letter also includes the argument that:

> "*the skilled person will appreciate that incorrect refactoring of a system's main code-base can lead to entire system malfunctions, compared to removal of essential code which would only affect the program itself*"

further:

> "*identifying and removing this redundant code has a technical effect of increasing the efficiency and reliability of the system as, if gone undetected, the redundant code will lead to design debt which creates inefficiencies within the system*"

21 The examiner disagrees with these arguments, stating[9]:

> "*I disagree with this reasoning. The computer per se is not running better or more efficiently as a result of your invention, it is simply processing less data (or even the same amount of data, as the removed code was redundant anyway) and therefore what is produced is not a better computer but a better program.*"

[8] page 2 of the Attorney's letter dated 11th February 2022
[9] in paragraph 17 of the examination report dated 4th March 2022

22 I agree with the examiner: the program itself may be improved but programming a computer to perform the method does not provide any inherent improvement in the computer per se.

23 I agree with the applicant's argument that identifying and/or removing unused programming units from the code-base results in clearer programs, and clearer programs are easier to develop and modify. The removal of "code bloat" reduces the number of available programming units a developer has to understand and consider and so also simplifies the system for developers. It is stated in page 8 lines 9-13 of the application as filed and as published:

> *"Speed is increased because time is not wasted loading or otherwise compiling and deploying code that is dead. Developer efficiency is increased, and hence development cost is decreased, as developers will only learn, maintain or extend parts of the code base that is actively part of the current solution"*

The applicant argues that this is similar to the invention in *HTC v Apple*[10] which was considered not to be excluded. In *HTC v Apple* a new and improved device interface provided to application programmers was considered patentable. The advance related to the recognition of single touch and multiple touches in a touch screen device and how this resulted in an improved device. The current invention relates to the automated identification of redundant code which may be either automatically separated from the code base or automatically deleted. The improvements from the invention relate to processor time saved from loading or compiling dead code or developer time saved in not learning or extending dead code; these are both improvements in the program. The comparison to HTC v Apple does not lead us to a technical effect.

24 Finally, signpost (v) relates to the problem to be solved. The problem relates to identifying redundant programming units of a code-base to reduce code bloat. This is the automation of a manual editing process and so this problem is considered to be administrative and not technical. The solution of an administrative problem does not assist in identifying a technical effect.

25 The comparisons with *Kapur v Comptroller General of Patents*[11] and *Shopolotto.com Ltd's Application*[12] are tangential and do not assist in identifying a technical effect.

26 None of the signposts point to a technical contribution. The other arguments considered do not lead us to a technical contribution. I therefore consider that the invention is excluded as a program for a computer as such.

27 For completeness. I confirm that I have also considered the dependent claims, numbered 2—25 and 27-49, and the rest of the specification. I have been unable to identify anything which would move the contribution beyond a computer program as such.

---

[10] HTC v Apple [2013] EWCA Civ 451
[11] Kapur v Comptroller General of Patents, Designs & Trade Marks [2008], EWHC 649 (Patents)
[12] SHOPALOTTO.COM LTD'S APPLICATION, Reports of Patent, Design and Trade Mark Cases, Volume 123, Issue 9, 2006, Pages 293–298

Conclusion

28 Having considered all of the arguments and papers on file, I am of the view that the contribution made by the invention falls solely within the computer program exclusion.

29 I therefore find that the invention claimed in GB1507342.2 is excluded by section 1(2)(c) as a program for a computer as such. I therefore refuse the application under section 18(3).

Appeal

30 Any appeal must be lodged within 28 days after the date of this decision.

Mr PETER MASON

Deputy Director, acting for the Comptroller