



PATENTS ACT 1977

APPLICANT Testplant Europe Limited

ISSUE Whether applications GB 1602421.8, GB 1602422.6
 & GB 1602423.4 comply with Section 1(2) of the
 Patents Act 1977

HEARING OFFICER Stephen Brown

DECISION

Introduction

- 1 This decision concerns three Patent applications all filed on 10th February 2016 by Testplant Europe Limited. The applications were published as follows:

GB 1602421.8	-	GB 2547220
GB 1602422.6	-	GB 2547221
GB 1602423.4	-	GB 2547222
- 2 In each of these applications, the Examiner issued an Abbreviated Examination Report setting out an objection under Section 1(2)(c) of the Act. There have been several rounds of correspondence but the applicant has been unable to persuade the Examiner that the applications are allowable. The Applicant thus requested a hearing to resolve the matter.
- 3 The hearing was held on 21st June with the applicant being represented by Jonathan Markham of Beck Greener. The hearing was also attended by Anthony Edwards the COO of the applicant. I was assisted by Nigel Hanley. I am grateful to Mr Markham for his skeleton arguments. I am particularly grateful for the assistance of Mr Edwards for his explanations and demonstration of the system set out in the three applications.

The Applications

- 4 All three applications are concerned with performing an automated test on a system. In essence, the three inventions each provide a method for establishing a test program that is run on a first computing apparatus. The test program interacts with a

system under test by simulating a range of inputs and actions and capturing the systems responses to these. In practice, the system under test is another computer program running on a second computing apparatus. At the hearing, Mr Edwards helpfully demonstrated one example of these inventions. He used his laptop to run a test program which tested a banking application running on a mobile phone, situated back in his office.

- 5 The key to each patent application is the way in which a sequence of actions is generated by the test program and applied to the system under test. In GB'423.4 all possible actions are tested except for actions omitted by a precondition. In GB'422.6 the sequence of actions takes into account previous actions in the current, or earlier, tests. In GB'421.8 test description sections are associated with each action.
- 6 The claims on file for each application are those filed on 1st June 2018 for GB'421.8 and 14th May 2018 for applications GB'422.6 and GB'423.4. For ease of reading, copies of the independent claims are listed in Annex A of this decision. So called 'auxiliary' claims, received on 11th February 2019, can be found in Annex B.

The Law

- 7 The section of the Act concerning inventions excluded from patentability is Section 1(2). This reads:

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

...

(c) a scheme, rule or method for performing a mental act, playing a game or doing business or a program for a computer;

...

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

- 8 In order to decide whether an invention relates to subject matter excluded by Section 1(2), the Court of Appeal has said that the issue must be decided by answering the question of whether the invention reveals a technical contribution to the state of the art. The Court of Appeal in *Aerotel/Macrossan*¹ set out the following four-step approach to help decide the issue:

- 1) Properly construe the claim;
- 2) Identify the actual (or alleged) contribution;

¹ *Aerotel Ltd v Telco Holdings Ltd (and others) and Macrossan's Application* [2006] EWCA Civ 1371

- 3) Ask whether it falls solely within the excluded subject matter;
 - 4) Check whether the actual or alleged contribution is actually technical in nature.
- 9 The operation of the approach is explained at paragraphs 40-48 of the judgment. Paragraph 43 confirms that identification of the contribution is essentially a matter of determining what it is the inventor has really added to human knowledge, and involves looking at substance, not form. Paragraph 47 adds that a contribution which consists solely of excluded matter will not count as a technical contribution.
- 10 The case law on computer implemented inventions has been further elaborated in *AT&T/CVON*² which provided five helpful signposts to apply when considering whether a computer program makes a relevant technical contribution. In *HTC v Apple*³, Lewison LJ reconsidered the fourth of these signposts and felt that it had been expressed too restrictively. The revised signposts are:
- i) whether the claimed technical effect has a technical effect on a process which is carried on outside the computer;
 - ii) whether the claimed technical effect operates at the level of the architecture of the computer; that is to say whether the effect is produced irrespective of the data being processed or the applications being run;
 - iii) whether the claimed technical effect results in the computer being made to operate in a new way;
 - iv) whether the program make the computer a better computer in the sense of running more efficiently and effectively as a computer; and
 - v) whether the perceived problem is overcome by the claimed invention as opposed to merely being circumvented.

Application of the *Aerotel* test

Step 1: Properly construe the claims

- 11 Within these three applications there is a great degree of commonality that I believe can be construed relatively easily. As I understand it, each automated test is essentially a computerised method of generating a series of actions using a model of the 'system under test'. These actions are then automatically applied to the system and the results captured in some way. The term "system under test" is one commonly used in computing and I believe that its use here is completely standard.

² *AT&T Knowledge Ventures LP and CVON Innovations Limited v Comptroller General of Patents* [2009] EWHC 343

³ *HTC v Apple* [2013] EWCA Civ 451

Namely, it relates to the combination of specific software running on specific computer hardware that is to be tested.

- 12 The claims also refer to “control of a graphical user interface”. At the hearing Mr Edwards explained that this concerns the execution of the test via a GUI on the system under test. The actions generated by the test may be applied to the system either via its GUI or by inputting relevant signals to the system *as if* a user had performed the action on its GUI. Additionally, the system’s response to an action may be captured by monitoring the same GUI, as part of what the claims call the ‘associated validation steps’. I thus construe this term not to be “control”, as such, but rather ‘use’ of the GUI by the test program. At the hearing Mr Edwards admitted that this is a relatively common feature of many known test suites.
- 13 Next, I will construe what is meant by ‘defining/utilising a model of the system under test’. Each invention models the system under test as a collection of states and actions. Actions are associated with states and they transition the model system between states in known ways. For example, one state of the system could be the logon screen. The action ‘entered valid username and password’ is modelled to transit this state to a certain second state/screen. The automated test uses the model to choose a series of actions which are then applied to the system under test in a given sequence. In the ‘validation steps’ the automated test determines whether the actions have been successful – i.e. whether they have led to the correct new states as predicted by the model.
- 14 Finally, each independent claim finishes by specifying the step of ‘updating a log to enable debugging of the system under test’. I believe that this requires very little construction – it is well known to use the results of tests to ‘debug’ systems - i.e. to make them operate closer to their model behaviour. Such logs are typically lists of what went right and what went wrong and the circumstances around each instance.
- 15 I thus construe the independent claims of the applications to relate to automated tests that model a system under test, use the model to choose a series of actions which are applied to the system using its GUI, determine whether these actions have been successful, and then update a log with these results.
- 16 I believe this covers the common material across the three applications. The key difference between each application is in the details of the models and their actions used in the automated tests. I will address these in more detail under step two of the *Aerotel* test.

Step 2: Identify the actual (or alleged) contribution

- 17 The judgement in *Aerotel* makes it clear that this step is an exercise in identifying what has been added to the stock of human knowledge. Since it is known to run automated test programs which access the GUIs of systems under test and log the results, it follows that these features cannot form part of the contribution. Likewise, it is also known, in general, to model a system under test and decide how to test it based on such a model.

- 18 In short, I believe that what I have called the 'common material' between the three applications cannot form part of their contributions. It follows that for each application the alleged contribution is in the detail of their model, its model actions and how they are selected to form an automated test. Note, I say 'alleged' because no search has been performed by the examiner so I have no prior art to assess them against.
- 19 At the hearing, Mr Markham argued that the contributions were wider than this. Specifically, that the details of each test led directly to an improved 'system under test'. While this may, or may not, be true I am not sure that it forms part of the contribution. It is readily accepted that the running of a test program on a "system under test" is well known. Logging the results of such tests then debugging the system to improve it is also well known. These features are present when any competently constructed test is run on any system under test. I thus conclude that they are not what these applications have added to the stock of human knowledge.
- 20 This conclusion is reinforced when I consider the inventions from the point of view of the 'system under test' itself. The system itself is oblivious to the fact that it is being tested. It is merely running in its normal way. Indeed, that is a key feature of any good testing regime. Given this, I cannot say that the automated test is controlling the 'system under test', in the usual meaning of this term. Thus, I do not believe that the 'system under test' forms part of the contributions.
- 21 I am forced to conclude that what has been added to human knowledge in the current applications is *how* the test programs are generated. While I hesitate to use the term *assembled*, that does appear to be a good description of what is happening. Once a test program has been assembled it is then executed as would be the case with any other test program. I thus decide that the contributions lie in the generation of the test programs, not in running them and any results that may flow therefrom. I will now consider the contribution of each application in turn.

GB'421.8

- 22 The GB'421.8 application defines a model of the system under test comprising states and associated actions. Its model actions have associated test description sections that cause the test program to execute specific operations on the system and then validate if they were successfully executed. I thus identify the contribution in this application to be 'generating a test program by selecting a sequence of actions from a model system comprising states and actions with associated test descriptions'.

GB'422.6

- 23 This application also defines a model of the system under test comprising states and associated actions. However, unlike GB'421.8, its actions do not have associated test descriptions and the test is assembled in a different way. In GB'422.6 the selection of actions is dependent upon the results of previous actions, either in the current test or in previous tests.
- 24 At the hearing, Mr Edwards provided a helpful example of how this might work in practice. In a case where several developers are responsible for the code in a system it may become apparent after several rounds of testing that the code of one

developer is more likely to contain errors than that of the others. Using the automated test of GB'422.6, tests can be assembled that focus on that developer's code. Once again, I am grateful to Mr Edwards for a very clear explanation.

- 25 Mr Markham argued that this ability to react to previous results provides tests with better coverage. This may be true but again I cannot see how this impacts on the contribution. I identify the contribution in this application to be 'generating a test program by selecting a sequence of actions from a model system comprising states and actions, where the selection depends upon previous selections of actions'.

GB'423.4

- 26 This application again defines a model system with states and actions. It differs in that preconditions are assigned to each model action. When the test program is generated all possible actions are selected unless the associated precondition is not satisfied. Page 28 of the description, lines 14 to 35, explains how a precondition relates to an action. It gives the example that the action of logging into a banking app will only be selected if the condition 'Username, password, and security character completed' is satisfied. I identify the contribution in this application to be 'generating a test program by selecting a sequence of actions from a model system comprising states and actions, where an action can only be selected if an associated precondition is met'.

Steps 3 and 4: Ask whether the contribution falls solely within excluded subject matter and whether it is technical in nature

- 27 It is clear to me that the contributions I have identified above all relate to computer programs. The question is thus: are they merely programs 'as such' and nothing more? To answer this, I will now turn to the modified AT&T signposts³.
- 28 At the hearing, Mr Markham highlighted the importance of the first signpost in all three applications, arguing that each contribution did have a technical effect on an external process. In particular, he stressed that in each application the GUI of the system under test was controlled by the test program. He then directed me to paragraph 12 of *Vicom*⁴ where the EPO Technical Board of Appeal stated that:

"The Board is of the opinion that a claim directed to a technical process which process is carried out under the control of a program (be this implemented in hardware or software), cannot be regarded as relating to a computer program as such within the meaning of Article 52(3) EPC, as it is the application of the program for determining in the sequence of steps in the process for which in effect protection is sought. Consequently, such a claim is allowable under Article 52((2)(c) and (3) EPC."

- 29 Mr Markham also directed me to a decision of this Office⁵ where, at paragraph 33, the hearing officer had noted "*from the case law before me, I take this (Vicom) to mean that the process must operate on something external to the computer on which*

⁴ *Vicom* (T208/04)

⁵ *Commonwealth Scientific and Industrial Research Organisation BL O/367/1*

the program is being run". Mr Markham argued that controlling the GUI of the system under test and actually testing that system met the Vicom criteria - i.e. it was an effect that was external to the computer upon which the test program was run.

- 30 I am afraid that I remain unconvinced. Leaving aside that Vicom was primarily concerned with the presence of a mathematical method in a process, the process in question was about digitally enhancing a photograph. It had a clear input and a clear output. In that regard it did exercise control over something external to the system – namely a photograph.
- 31 Where I believe this differs from the current applications is that their contributions do not include controlling an external system. What the current inventions do is interface with the system under test in the way it is designed to be interfaced with – i.e. in a known way that does not form part of the contributions, as argued above. As identified above, the contribution of each application lies in how you generate the test program and that is entirely internal to the computer on which the generating takes place.
- 32 Mr Markham then referred me directly to *HTC*³ where, in paragraph 154, Lewison LJ stated:

"The claimed invention has a technical effect on the process carried on outside the computer. At one end, it makes the writing of the software easier. I do not agree with the judge that this is simply a redistribution of labour. It is a reduction of labour, because once the device is programmed with the claimed invention software writers can use it over and over again. ... At the other end, it interacts with end user who touches the screen, This too, is a process happening outside the computer."

And, in paragraph 58, Kitchen LJ stated:

"The device is, in a real practical sense, an improved device. This is not because it now runs different application programs but because it is, as a device easier for programmers to use. Once again, this emphasises the technical nature of the invention".

- 33 Again, I am unconvinced. Firstly, as Mr Markham conceded in his skeleton argument, filed immediately before the hearing, the multi-screen technology used in *HTC*³ is very different to that of the current applications. In that case an indicator was programmed to allow any software writer to use that feature "over and over again". While Lewison LJ talks about interacting with a screen being an external process, in *HTC*³ this was a new way of interacting and hence part of that case's contribution. In the current applications the interaction with the GUI is entirely standard and thus not part of their contributions.
- 34 Finally, Mr Markham turned to *Haliburton*⁶, quoting the conclusion that "*outputting to a resource the results of the method*" was sufficient to render a claim non-excluded.

⁶ *Haliburton [2011] EWHC 2508 (Pat)*

From this he argued that the production of a log to enable debugging was a sufficient external effect.

- 35 Yet again, I am not convinced. In *Haliburton*⁶ the contribution is a better design for a drill bit. Birss J decided that this did not fall solely within excluded territory. In the current applications I have identified the contributions to be better ways of generating test programs – that is very different and does not include, as argued above, the standard step of updating a log to enable debugging.
- 36 Thus, despite all of Mr Markham’s points, I decide that the contributions of the current applications do not include a technical effect on a process which is carried on outside the computer on which the test programs run. That is, they do not meet the first AT&T³ signpost.
- 37 Turning to the second signpost, Mr Markham argued that the test results were produced irrespective of the data being processed or the applications being run on the ‘system under test’. Thus, he argued, the applications meet the second signpost.
- 38 Again, I must disagree and respectfully point out that Mr Markham is focussing on the wrong computer. In the computer that is generating and running the test programs it is clear to me that the contribution is operating at the application level. As mentioned above, Mr Edwards was kind enough to show me an example of the software running on his laptop during the hearing. I saw nothing there and can see nothing in any of the applications that indicates that this was running at the level of the architecture of the computer. As for the ‘system under test’, that computer is operating in a completely normal way – that is the point of the test regime after all. It is certainly not altered at the level of architecture by the invention even were it part of the contributions. I thus conclude that the contributions do not meet the second AT&T³ signpost.
- 39 I will deal with the third and fourth signposts together. I can see no evidence that the computer generating and running the tests is operating in a new way, neither is it running more efficiently or effectively. It is acting as it would when running any program. Likewise, the computer in the ‘system under test’ is running in a completely standard way, though as argued above, it is not part of the contributions in any case. Thus, the contributions do not meet the third and fourth AT&T³ signposts.
- 40 The final signpost is, I believe, inconclusive. At the hearing, Mr Markham argued that the inventions overcame a “technical problem” in that they solved the problem of “generating sufficient, sufficiently diverse and comprehensive, pathways to ensure that the ‘system under test’ is tested accurately.”
- 41 While this may be a real technical problem, the contributions I have identified above are clearly programs running on a standard computer. The problem they solve is how to best test another program running on another standard computer. They have failed to meet the first four signposts and I cannot see how the fifth signpost can save them. I thus decide that the contributions fall entirely within the program for a computer exclusion.

42 The final step of the *Aerotel*¹ test is to check whether the contributions are technical in nature. Since I have decided that they do not have technical effects beyond those of programs running on a computer, they also fail this step of the test. I thus decide that the claims are excluded under section 1(2).

Auxiliary Claims

43 The auxiliary claims can be viewed at Annex B. In a brief summary they change the claims in two respects. In the first instance they alter the claims to “A method for performing an automated test, the automated test being configured, when executed **by a test program**, to ~~control~~ **cause a test computer system to communicate with** a graphical user interface of a system under test...”.

44 The key question is whether this alters the construction of the claims and my subsequent identification of their contributions. I do not believe that it does. If anything, it reinforces my conclusions by making it clear that the computer running the test *communicates* with the system under test and does not control it. In respect of this part of the amendments I believe that my previous analysis still applies.

45 The second instance uses slightly different wording in each application, However, the common thread is to clarify that the validation steps “*utilise image analysis to identify one or more image objects in the graphical user interface of the system under test by searching an expected image object in the graphical user interface using one or more reference images associated with the automated test*”.

46 Again, the question is whether this alters the contribution of the claims. I do not believe that it does. The use of image comparison when testing systems with GUIs is known. It thus does not form part of the contributions and my analysis remains unchanged. I thus conclude that the auxiliary claims are excluded under section 1(2).

Decision

47 I have decided that the inventions defined in the independent claims fall solely within matter excluded under Section 1(2) as programs for a computer as such. Having reviewed the applications, I do not consider that any saving amendments are possible. I therefore refuse all three applications under section 18(3).

Appeal

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

Stephen Brown

Deputy Director, acting for the Comptroller

GB 1602421.8**Filed: 1st June 2018****Claim 1**

A method for performing an automated test the automated test being configured, when executed, to control a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test having a plurality of operational states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the method comprising the steps of:

- a) Defining an executable model of the system under test comprising a plurality of model states, wherein at least some of the model states are representative of operational states of the system under test; and
- b) Defining one or more selectable model actions, wherein each model action being representative of one or more executable actions on the system under test, one or more model states and one or more model actions are operable, when selected, to transition the model between model states;
- c) Associating one or more test descriptions sections with the one or more model actions, wherein each test description section comprises one or more operation commands configured, when executed, to cause a test program to execute one or more operations on the system under test and one or more validation steps operable, when executed to identify whether the one or more operations have been executed successfully: and
- d) Executing using a test program, an automated test on the system under test by:
 - e) Selecting a sequence of model actions;
 - f) Utilising the test description sections associated with the selected sequence of model actions to generate a sequence of operation commands for execution on the system under test;
 - g) Executing, using the test program, the generated sequence of executable actions on the system under test in order to execute one or more executable actions on the system under test;
 - h) Determining, using the associated validation steps, whether the one or more executable actions, have been successfully executed: and
 - i) Based on step h), updating a log to enable the debugging of the system under test.

Claim 26

Computing apparatus operable to perform an automated test by controlling a graphical user interface of a system under test using a test program, the system under test comprising one or more computer programs being executed on one or more computer devices, the system under test having a plurality of operation states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the computer apparatus being configured to:

- a) Utilise a model of the system under test comprising a plurality of model states, at least some of the model states are representative of operational states of the system under test; and
- b) Select, using the model of the system under test, one or more selectable model actions, wherein each model action is representative of one or more executable actions of the system under test, one or more model actions are associated with one or more model states and one or more model actions are operable, when selected, to transition the model between model states;
- c) Utilise one or more test descriptions sections associated with the selected sequence of model actions to generate a sequence of operation commands for execution on the system under test, and one or more associated validation steps operable, when executed, to identify whether the one or more operations have been executed successfully: and
 - d) Execute using the test program the sequence of operation commands on the system under test in order to execute one or more executable actions on the system under test;
 - e) Determine, using the associated validation steps, whether the one or more executable actions, have been successfully executed: and
 - f) Based on step e), updating a log to enable the debugging of the system under test.

Claim 1

A method for performing an automated test configured, when executed to control a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test comprising a plurality of operational states and one or more executable actions, the method comprising the steps of:

- a) Utilising a model of the system under test, the model comprising a plurality of model states and one or more selectable model actions, wherein one or more of the model states are representative of operational states of the system under test and each model action is associated with one or more executable actions on the system under test;
- b) Generating an automated test by:
- c) Selecting a sequence of model actions, wherein the selection of one or more model actions is dependent upon the previous selection of model actions in the current and/or one or more previous automated tests; and
- d) Defining, based on the selected sequence of model actions, an automated test configured, in use, to perform associated validation steps to identify whether the one or more executable actions have been executed successfully:
- e) Executing, using a test program, the generated sequence of executable actions on the system under test as an automated test;
- f) Determining, using the associated validation steps, whether the one or more executable actions have been executed successfully: and
- g) Based on step f), updating a log to enable the debugging of the system under test.

Claim 19

Computing apparatus operable to perform an automated test by controlling a graphical user interface of a system under test using a test program, the system under test comprising one or more computer programs being executed on one or more computer devices, and comprising a plurality of operational states and one or more executable actions, the test program being configured to utilise a model of the system under test, the model comprising a plurality of model states and one or more selectable model actions, wherein one or more of the model states are representative of operational states of the system under test and each model action is associated with one or more executable actions on the system under test, wherein the computing apparatus is configured to generate an automated test by at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the computer apparatus being configured to:

- a) Selecting a sequence of model actions;
- b) Defining based on said selected sequence of model actions, an automated test configured, in use, to execute the sequence of executable actions on the system under test and to perform associated validation steps to identify whether the one or more executable actions have been executed successfully; wherein the selection of one or more model actions in step a) is dependent upon the previous selection of model actions in the current and/or one or more previous automated tests; and wherein the computing apparatus is further configured to:
 - c) Execute using the test program, the generated sequence of executable actions on the system under test as the automated test;
 - d) determining using the associated validation steps of the automated test, whether the one or more executable actions comprising the automated test have been executed successfully;
 - e) Based on step d), updating a log to enable the debugging of the system under test.

Claim 1

A method for performing an automated test configured, when executed to control a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test comprising a plurality of operational states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the method comprising the steps of:

- a) Defining an executable model of the system under test comprising a plurality of model states, wherein at least some of the model states are representative of operational states of the system under test; and
- b) Defining one or more selectable model actions, each model action being representative of one or more executable actions on the system under test and configured such that one or more preconditions are assignable thereto, wherein each model action is configured to be available to be selected unless any preconditions assigned to the respective model action are not satisfied:
- c) Executing using a test program, an automated test on the system under test by:
- d) Selecting, using a test program, a sequence of executable actions on the system under test:
- e) Executing, using the test program, the sequence of executable actions on the system under test:
- f) Determining, using one or more validation steps associated with the sequence of executable actions, whether the one or more executable actions have been executed successfully; and
- g) Based on step f), updating a log to enable the debugging of the system under test.

Claim 27

Computing apparatus operable to perform an automated test by controlling a graphical user interface of a system under test using a test program, the system under test comprising one or more computer programs being executed on one or more computer devices, and comprising a plurality of operational states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the computer apparatus being configured to:

- a) Utilise an executable model of the system under test comprising a plurality of model states, wherein at least some of the model states are representative of operational states of the system under test; and
- b) Select a sequence of selectable model actions representative of a sequence of executable actions on the system under test to define an automated test, each model action being representative of one or more executable actions on the system under test and configured such that one or more preconditions are assignable thereto, wherein each model action is configured to be available to be selected unless any preconditions assigned to the respective model action are not satisfied: and
- c) Execute using a test program, the sequence of executable actions of the automated test on the system under test:
- d) determine using one or more validation steps associated with the sequence of executable actions, whether the one or more executable actions have been executed successfully: and
- e) Based on step d), updating a log to enable the debugging of the system under test.

GB 1602421.8

Claim 1

A method for performing an automated test the automated test being configured, when executed **by a test program**, to ~~control~~ **cause a test computer system to communicate with** a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test having a plurality of operational states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the method comprising the steps of:

- a) Defining an executable model of the system under test comprising a plurality of model states, wherein at least some of the model states are representative of operational states of the system under test; and
- b) Defining one or more selectable model actions, wherein each model action is representative of one or more executable actions of the system under test, one or more model actions are associated with one or more model states and one or more model actions are operable, when selected, to transition the model between model states;
- c) Associating one or more test description sections with the one or more model actions, wherein each test description section comprises one or more operation commands configured, when executed, to cause a test program to execute one or more operations on the system under test and one or more validation steps operable, when executed to identify whether the one or more operations have been executed successfully: **wherein one or more operation commands and/or validation steps utilises image analysis to identify one or more image objects in the graphical user interface of the system under test by searching for an expected image object in the graphical user interface using one or more reference images associated with the automated test:** and
- d) Executing using **a the** test program, an automated test on the system under test by:
 - e) Selecting a sequence of model actions;
 - f) Utilising the test description sections associated with the selected sequence of model actions to generate a sequence of operation commands for execution on the system under test;
 - g) Executing, using the test program, the generated sequence of executable actions on the system under test in order to execute one or more executable actions on the system under test;
 - h) Determining, using the associated validation steps, whether the one or more executable actions, have been successfully executed: and
 - i) Based on step h), updating a log to enable the debugging of the system under test.

GB 1602422.6

Claim 1

A method for performing an automated test configured, when executed **by a test program to control cause a test computer system to communicate with** a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test comprising a plurality of operational states and one or more executable actions, the method comprising the steps of:

- a) Utilising a model of the system under test, the model comprising a plurality of model states and one or more selectable model actions, wherein one or more of the model states are representative of operational states of the system under test and each model action is associated with one or more executable actions on the system under test;
- b) Generating an automated test by:
- c) Selecting a sequence of model actions, wherein the selection of one or more model actions is dependent upon the previous selection of model actions in the current and/or one or more previous automated tests; and
- d) Defining, based on the selected sequence of model actions, an automated test configured, in use, to perform associated validation steps to identify whether the one or more executable actions have been executed successfully: **wherein one or more executable actions and/or performing one or more validation steps utilises image analysis to identify one or more image objects in the graphical user interface of the system under test by searching for an expected image object in the graphical user interface using one or more reference images associated with the automated test:**
- e) Executing, using ~~a~~ **the** test program, the generated sequence of executable actions on the system under test as an automated test;
- f) Determining, using the associated validation steps, whether the one or more executable actions have been executed successfully: and

Based on step f), updating a log to enable the debugging of the system under test.

Claim 1

A method for performing an automated test configured, when executed **by a test program**, to ~~control~~ **cause a test computer system to communicate with** a graphical user interface of a system under test comprising one or more computer programs being executed on one or more computer devices, the system under test comprising a plurality of operational states, at least one operational state having one or more executable actions associated therewith operable to execute predetermined operations and/or transition the system under test between operational states, the method comprising the steps of:

- a) Defining an executable model of the system under test comprising a plurality of model states, wherein at least some of the model states are representative of operational states of the system under test; and
- b) Defining one or more selectable model actions, each model action being representative of one or more executable actions on the system under test and configured such that one or more preconditions are assignable
- c) thereto, wherein each model action is configured to be available to be selected unless any preconditions assigned to the respective model action are not satisfied:
- d) Executing using **a the** test program, an automated test on the system under test by:
- e) Selecting, using **a the** test program, a sequence of executable actions on the system under test:
- f) Executing, using the test program, the sequence of executable actions on the system under test:
- g) Determining, using one or more validation steps associated with the sequence of executable actions, whether the one or more executable actions have been executed successfully; and
- h) Based on step f), updating a log to enable the debugging of the system under test, **wherein steps e) and/or f) utilises image analysis to identify one or more image objects in the graphical user interface of the system under test by searching for an expected image object in the graphical user interface using one or more reference images associated with the automated test.**