DESCRIPTION

TITLE OF INVENTION: INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING METHOD, AND COMPUTER READABLE NON-TRANSITORY RECORDING MEDIUM

TECHNICAL FIELD

[0001] The present invention relates to an information processing apparatus, an information processing method, and a computer readable non-transitory recording medium.

BACKGROUND ART

[0002] Recently, a GUI becomes enhanced, and a technique of outputting an execution result of a computer program by an image is also known. Further, in developing a system for a computer or the like, a test for checking whether or not the system normally operates is conventionally performed. Execution of the test has a problem of a burden put on a test manager such as preparation for the test, execution of the test, checking of results and so on. Hence, recently, a simple test by image comparison is increasingly performed. Patent Literature 1 discloses a technique of verifying an execution result of a computer program by image comparison. The technique in Patent Literature 1 performs image comparison after removing designated mask regions from two images created by the execution of the computer program respectively, and displays a comparison result.

CITATION LIST

PATENT LITERATURE

[0003] Patent Literature 1: Japanese Laid-open Patent Publication No. 2009-134406

SUMMARY OF INVENTION

TECHNICAL PROBLEM

[0004] However, there is also an anticipated case where though a partial image in an image displayed by a test object program and a partial image in an image displayed by a correct program accord, their display positions are different. In such a case, it is desired in some cases to determine that the program normally runs by regarding the partial images as according.

[0005] Embodiments of the present invention have been made in consideration of the above problem, and seek to appropriately execute verification of a program including a description relating to drawing of an image.

SOLUTION TO PROBLEM

[0006] A first aspect of the present invention provides an information processing apparatus comprising:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image

corresponding to a correct image, specify a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extract the object image at the specified position;

a determiner configured to determine whether or not the object image and the correct image accord; and

a display processor configured to display a result of determination by the determiner.

[0006a] A second aspect of the present invention provides an information processing apparatus comprising:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser positions of a plurality of object images in the first image, based on attribute information indicating a class common to the plurality of object images, the object images corresponding to a correct image, specify each position of a region where each of the plurality of object images is actually drawn by the browser, based on a response from the browser, the response including each position of the region, and extract the object image at specified each position,

a determiner configured to determine whether or not each of a plurality of the object images and each of a plurality of the correct images accord; and

a display processor configured to display a result of determination by the determiner.

[0006b] A third aspect of the present invention provides an information processing method executed by an information processing apparatus, the information processing method comprising:

a first acquisition step of acquiring a first image displayed on a display in response to execution of an object program being a test object;

an object image extraction step of inquiring of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image corresponding to a correct image, specifying a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extracting the object image at the specified position;

a determination step of determining whether or not the object image and the correct image accord; and

a display processing step of displaying a result of determination at the determination step.

[0006c] A fourth aspect of the present invention provides a computer readable non-transitory recording medium with a program causing a computer to execute:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image corresponding to a correct image, specify a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extract the object image at the specified position;

a determiner configured to determine whether or not the object image and the correct image accord; and

a display processor configured to display a result of determination by the determiner.

[0006d] There is also disclosed an information processing apparatus including: a first acquirer configured to acquire a first image displayed on a display in response to

execution of an object program being a test object; an object image extractor configured to extract a partial image as an object image from the first image, based on first specification information specifying the partial image, the partial image being included in the first image and corresponding to a correct image; a determiner configured to determine whether or not the

object image and the correct image accord; and a display processor configured to display a result of determination by the determiner.

ADVANTAGEOUS EFFECTS OF INVENTION

[0007] According to the present invention, it is possible to appropriately execute verification of a program including a description relating to drawing of an image.

BRIEF DESCRIPTION OF DRAWINGS

[0008] [Fig. 1] Fig. 1 is a diagram illustrating a hardware configuration of a test apparatus.

[Fig. 2] Fig. 2 is a flowchart illustrating verification processing.

[Fig. 3] Fig. 3 is an explanatory view of processing relating to execution of a correct program and an object program.

[Fig. 4] Fig. 4 is a view illustrating a display example of a test result.

[Fig. 5] Fig. 5 is a view illustrating another display example of the test result.

[Fig. 6A] Fig. 6A is a view illustrating a display example of the test result in the case of using edge detection.

[Fig. 6B] Fig. 6B is a view illustrating a display example of the test result in the case of using edge detection.

[Fig. 7] Fig. 7 is an explanatory view of a determination condition.

[Fig. 8A] Fig. 8A is an explanatory view of a

determination condition.

[Fig. 8B] Fig. 8B is an explanatory view of the determination condition.

[Fig. 9] Fig. 9 is an explanatory view of a determination condition.

[Fig. 10A] Fig. 10A is an explanatory view of a determination condition.

[Fig. 10B] Fig. 10B is an explanatory view of the determination condition.

[Fig. 11] Fig. 11 is an explanatory view of a determination condition.

[Fig. 12] Fig. 12 is a flowchart illustrating verification processing according to a second embodiment.

[Fig. 13] Fig. 13 is a flowchart illustrating verification processing according to a third embodiment.

[Fig. 14] Fig. 14 is a view illustrating examples of a correct entire image group and an object entire image group.

DESCRIPTION OF EMBODIMENTS

[0009] Embodiments of the present invention will be described below referring to the drawings.

[0010] A test apparatus 100 according to this embodiment is an apparatus that tests (verifies) whether or not a program being a test object normally runs. Hereinafter, a test object program will be called an object program. The test apparatus 100 executes a correct program prepared for the object

program and executes the object program, and compares images obtained from the respective programs with each other. In this event, the comparison is performed not for the entire images obtained from the respective programs, but the image comparison is performed in units of partial image. Note that, for example, when a program after version upgrade is regarded as the object program, the test apparatus 100 can use the program before the version upgrade as the correct program. Note that the object program that is regarded as the test object by the test apparatus 100 according to this embodiment and the correct program corresponding to the object program include processing of displaying images.

[0011] Fig. 1 is a diagram illustrating a hardware configuration of the test apparatus 100 as an information processing apparatus. The test apparatus 100 has a CPU 101, a ROM 102, a RAM 103, an HDD 104, a display unit 105, and an input unit 106. The CPU 101 reads a control program stored in the ROM 102 and executes various kinds of processing. The RAM 103 is used as a main memory and a temporary storage area of the CPU 101. The HDD 104 stores various kinds of information such as image data, various kinds of programs and so on. The display unit 105 displays various kinds of information. The input unit 106 has a keyboard and a mouse and accepts various kinds of operations by a user. Note that later-described functions and processing of the test apparatus 100

are realized by the CPU 101 reading the program stored in the ROM 102 or the HDD 104 and executing the program.

[0012] Fig. 2 is a flowchart illustrating verification processing by the test apparatus 100. The verification processing mentioned herein is processing realized by a computer of the test apparatus 100 executing a verification program. At S200, the CPU 101 starts execution of the object program and the correct program. Note that the CPU 101 starts execution of the object program and the correct program, for example, upon accepting a start instruction from the user via the input unit 106.

[0013] The object program and the correct program include descriptions relating to drawing of images. Execution of draw commands relating to the descriptions is performed by a browser application (hereinafter, referred to simply as a browser). More specifically, the CPU 101 executes the object program to thereby display the image on the display unit 105 by using the browser. The CPU 101 similarly executes the correct program to thereby display the image on the display unit 105 by using the browser. Hereinafter, the image drawn and displayed by execution of the object program is referred to as an object entire image. Besides, the image drawn and displayed by execution of the correct program is referred to as a correct entire image. Note that timings of executing the object program and the

correct program are not particularly limited, but one of the programs may be executed first or the programs may be executed at the same time.

[0014] Fig. 3 is an explanatory view of processing relating to execution of the correct program and the object program. Upon execution of a correct program 300, the browser interprets an HTML text 310 in the correct program 300 and sends a draw command to an OS (operation system). In response to the draw command, the OS draws an image corresponding to the HTML text 310, namely, a correct entire image 320. In the example in Fig. 3, the correct entire image 320 includes partial images 321, 322, 323. The partial images 321, 322, 323 mentioned here are images drawn corresponding to descriptions 311, 312, 313 associated with image IDs "001", "002", "003" in the HTML text 310.

[0015] Upon execution of an object program 330, the browser similarly interprets an HTML text 340 in the object program 330 and sends a draw command to the OS (operation system). In response to the draw command, the OS draws an image corresponding to the HTML text 340, namely, an object entire image 350. In the example in Fig. 3, the object entire image 350 includes partial images 351, 352, 353. The partial images 351, 352, 353 mentioned here are images drawn corresponding to descriptions 341, 342, 343 associated with image IDs "001", "002", "003" in the HTML text 340.

[0016] Further, in this embodiment, partial images being verification objects, namely, comparison objects in the correct entire image and the object entire image are defined in advance in the verification program. In this embodiment, image IDs are defined as information for specifying partial images being the verification objects, namely, specification information in the verification program. This means that a partial image drawn according to a description associated with an image ID is set as a verification object. For example, when the image IDs "001", "003" are defined in the verification program as illustrated in Fig. 3, the partial images 321, 323 in the correct entire image and the partial images 351, 353 in the object entire image are the partial images being the verification objects. Note that, hereinafter, the partial image being the verification object in the correct entire image is referred to as a correct partial image, and the partial image being the verification object in the object entire image is referred to as an object partial image.

[0017] Further, the correct partial image and the object partial image specified by the same image ID become comparison objects, namely, a set of objects to be determined whether or not both images accord. More specifically, in the example in Fig. 3, the correct partial image 321 and the object partial image 351 form a set, and the correct partial image 323 and the object partial image 353 form a set.

Note that the image units of the correct partial image and the object partial image are not limited to those in the embodiment. For example, an image made by combining the partial image 321 and the partial image 322 may be defined as one correct partial image in the verification program.

[0018] Though this embodiment will be described using, as an example, a case where the same image ID is assigned to a correct partial image and an object partial image that is supposed to be equal to the correct partial image in the correct program and the object program, there is a case where verification needs to be performed by comparing the correct partial image and the object partial image associated with different image IDs. In this case, a creator, a verifier or the like of the object program only needs to define different image IDs as the correct partial image and the object partial image in the verification program. Note that the object partial image and the correct partial image are images being the comparison objects in this embodiment, and are examples of an object image and a correct image, respectively. Further, the image ID is an example of an identifier that identifies an image.

[0019] Returning to Fig. 2, after the processing at S200, the CPU 101 advances the processing to S201. At S201, the CPU 101 acquires a screenshot of the correct entire image and a screenshot of the object entire image, which are displayed on the display unit

105, as the correct entire image and the object entire image, respectively.

[0020] Next, at S202, the CPU 101 extracts partial images specified by image IDs defined in the verification program as correct partial images from the correct entire image (correct image extraction processing). The CPU 101 similarly extracts partial images specified by image IDs defined in the verification program as object partial images from the object entire image (object image extraction processing).

[0021] Hereinafter, the processing of extracting the correct partial image from the correct entire image will be described in detail. The CPU 101 inquires of the browser a position and a size of a correct partial image while designating an image ID. In response to the inquiry, the browser executes again the draw command of the correct program, and specifies a position and a size of a region drawn based on the description of the HTML text associated with the designated image ID. Note that a coordinate system for the position and size specified here accords with a coordinate system of a position and a size on the screenshot acquired at S201. The browser further specifies a rectangular or square region as a region of the correct partial image. The browser then returns information indicating an upper left vertex and longitudinal and lateral lengths of the region of the correct partial image as information

indicating the specified position and size, to the CPU 101. The CPU 101 extracts the correct partial image from the correct entire image acquired at S201, based on the position and size received from the browser.

[0022] For example, in the example in Fig. 3, the CPU 101 makes an inquiry with specified image IDs "001", "003" to the browser. The browser specifies regions of the partial images 321, 323 drawn according to the descriptions 311, 313 of the correct program corresponding to the image IDs "001", "003", specifies positions and sizes of the regions of the partial images 321, 323, and returns the specified positions and sizes to the CPU 101. The CPU 101 then extracts the partial images 321, 323 from the correct entire image 320.

[0023] Note that the information indicating the position and size of the region of the correct partial image is not limited to that in the embodiment. Another example of the partial image information may be information indicating positions of opposite vertices (two points). Besides, the shape of the region specified by the browser as the region where the correct partial image is drawn is not limited to that in the embodiment. As another example of the shape, the browser may specify a circular or triangular region as the region of the correct partial image. When the browser specifies a circular region, the browser only needs to return

information indicating a center position and a radius of a circle as the information indicating the position and size thereof to the CPU 101. Besides, when the browser specifies a triangular region, the browser only needs to return information indicating positions of three points as the information indicating the position and size thereof to the CPU 101.

[0024] The processing of extracting the correct partial images from the correct entire image at S202 has been described in the above, and processing of extracting the object partial images from the object entire image is the same as the processing of extracting the correct partial image from the correct entire image.

[0025] The CPU 101 extracts partial images (regions) drawn according to the descriptions associated with the image IDs of the description of the HTML text of the correct program as the correct partial images through use of the image IDs defined in the verification program as search keys as described above. The CPU 101 similarly extracts partial images (regions) drawn according to the descriptions associated with the image IDs of the description of the HTML text of the object program as the object partial images through use of the image IDs defined in the verification program as search keys. Therefore, the test apparatus 100 can correctly extract the partial images corresponding to each

other even in the case where the positions of the object partial images in the object entire image differ from the positions of the correct partial images in the correct entire image.

[0026] Returning to Fig. 2, after the processing at 202, the CPU 101 selects the object partial image and the correct partial image corresponding to the same ID which have been extracted at S202, as one set, at S203. In the example in Fig. 3, for example, the correct partial image 321 and the object partial image 351 are selected as a set. Then, at S204, the CPU 101 compares the object partial image and the correct partial image selected at S203 and determines whether or not both the images accord. When both the images accord (Yes at S204), the CPU 101 advances the processing to S206. When both the images do not accord (No at S204), the CPU 101 advances the processing to S205.

[0027] The CPU 101 compares pixel values of corresponding pixels of both the images, and determines whether or not both the images accord, based on a result of the comparison. In the comparison between the pixel values of the pixels, when a difference between the pixel values is less than a threshold value, the CPU 101 determines that both pixels accord, and when pixel values of all of corresponding pixels accord, the CPU 101 determines that both the images accord. There is a case where even if there is a certain degree of difference

between the pixel values, both the images may be regarded as according. In this case, it is only necessary to set a comparatively large value as the threshold value. The basis for determining that both the images accord as described above can be freely set according to a result required by the user.

[0028] As another example, the CPU 101 may determine that both the images accord when the number of pixels where both the pixels accord is a threshold value or more. As still another example, the CPU 101 may remove noise in the correct partial image and the object partial image before the comparison between the pixel values.

[0029] At S205, the CPU 101 performs emphasis processing on a disaccord portion of the object partial image which does not accord with the correct partial image. The CPU 101 similarly performs emphasis processing on a disaccord portion of the correct partial image which does not accord with the corresponding object partial image. In this embodiment, the CPU 101 displays an emphasis frame surrounding the disaccord portion in a superimposed manner. The emphasis processing, however, only needs to be processing of changing the disaccord portion into a display form discriminable from other portions so that the user can easily specify the disaccord portion, and concrete processing is not limited to that in the embodiment. Another example of the emphasis display can be processing of changing a

display color of the disaccord portion.

[0030] Next, at S206, the CPU 101 confirms whether or not the processing at S204, S205 has been completed for all of sets of the object partial images and the correct partial images which have been extracted at S202. When the processing has been completed for all of the sets (Yes at S206), the CPU 101 advances the processing to S207. When unprocessed sets exist (No at S206), the CPU 101 advances the processing to S203. In this case, at S203, the CPU 101 selects the unprocessed sets and continues subsequent processing.

[0031] At S207, the CPU 101 displays a determination result as a test result (display processing). In this embodiment, when a disaccord portion is detected, the CPU 101 displays the object entire image and correct entire image after the emphasis processing, as the test result. Besides, when any disaccord portion is not detected, the CPU 101 displays the object entire image and correct entire image with no emphasis processing performed thereon as the test result. Note that the CPU 101 only needs to be able to display an image where the use can identify the disaccord portion in the object partial image at S207, and may display, for example, only the object partial images and the correct partial images. As another example, the CPU 101 may display only the object partial images or only the object partial image indicating the disaccord portion. As still another

example, the CPU 101 may display both the object partial images and the correct partial images together with the object entire image and the correct entire image.

[0032] Fig. 4 is a view illustrating a display example of the test result. A test result screen 400 indicating the rest result is displayed on the display unit 105. A correct entire image 410 is displayed on the left side in the test result screen 400, and an object entire image 420 is displayed on the right side in the test result screen 400. In the example in Fig. 4, three disaccord portions are detected, and the emphasis frame is displayed in a superimposed manner at each of the disaccord portions. An emphasis frame 411 in the correct entire image 410 and an emphasis frame 421 in the object entire image 420 are emphasis frames superimposed on corresponding disaccord portions. In regions corresponding to the emphasis frames 411, 421, fonts of </button> are different between the images. Other emphasis frames 412, 413 are displayed in the correct entire image 410, and emphasis frames 422, 423 corresponding to the emphasis frames 412, 413 respectively are displayed in the object entire image 420. In regions corresponding to the emphasis frames 412, 422, positions of "button in content" are different. In addition, both the images are different also in that "button outside content" in the region corresponding to the emphasis frame 412 does not exist in the

region corresponding to the emphasis frame 422. Further, in regions corresponding to the emphasis frames 413, 423, both the images are different in that "displayed here" is displayed in a region corresponding to the emphasis frame 413, whereas " ddisplayed here" is displayed in a region corresponding to the emphasis frame 423. Since the emphasis frames are displayed as described above, the user can easily specify the disaccord portions.

[0033] Fig. 5 is a view illustrating another display example of the test result. A test result screen 500 indicating the rest result is displayed on the display unit 105. A bar 501 movable right and left in response to a user operation is displayed in a superimposed manner on the test result screen 500. An object entire image 520 is displayed on the right side of the bar 501, and a correct entire image 510 is displayed on the left side of the bar 501. In the display example in Fig. 5, contents disaccording between the correct entire image 510 and the object entire image 520 can be confirmed by moving the bar 501 without moving a line of sight. Note that emphasis frames are displayed in a superimposed manner at disaccord portions also in the test result screen 500.

[0034] As described above, the test apparatus 100 according to this embodiment can compare the object image obtained from the object program and the correct image in units of partial image. Accordingly,

even if the position of the correct partial image in the correct entire image and the position of the object partial image in the object entire image are different, both the images can be determined to accord when the correct partial image and the object partial image accord. Depending on a test case, only when images of the corresponding correct partial image and object partial image accord, it is desired to determine that the images accord in some cases even if their positions are deviated from each other. In this embodiment, determination of accord or disaccord can be made by comparison between the images regardless of their positions in such a case.

[0035] As a first modification example of the test apparatus 100 according to this embodiment, at S204, the CPU 101 may perform comparison between edges detected from the correct partial image and the object partial image respectively, in place of the comparison between the pixel values. More specifically, the CPU 101 performs edge detection in each of the object partial image and the correct partial image to obtain two edge images. The edge image mentioned here is an image indicating an edge detected from an image. The CPU 101 then compares the two edge images, and only needs to determine that the object partial image and the correct partial image accord when both the images accord.

[0036] Fig. 6A, Fig. 6B are views illustrating display examples of the test result in the case of

using the edge detection. As illustrated in Fig. 6A, Fig. 6B, two edge images are displayed in a superimposed manner. Thus, in the two edge images, one edge image is displayed for a portion where edges accord, and respective edge images are displayed for a portion where edges do not accord. Therefore, the user can easily specify the disaccord portion. The CPU 101 may further display an edge at an accord portion in black, an edge of the correct entire image of a disaccord portion in blue, and an edge of the object entire image of the disaccord portion in red. This makes it possible to more clearly indicate the accord portion and the disaccord portion.

[0037] As a third modification example, a position in a structure of the HTML text may be defined, in place of the image ID, as the specification information on the correct partial image and the object partial image which are the verification objects in the verification program. The HTML text mentioned here is one example of a structured document. In this case, the CPU 101 inquires positions and sizes of the correct partial image and the object partial image while designating the positions in the structures of the HTML texts. In response to the inquiry, the browser specifies the position and the size of a region drawn based on the description at the designated position. The browser only needs to return information indicating the specified position and size to the CPU 101. This

also applies to the object partial image.

[0038] As a fourth modification example, the CPU 101 does not have to execute, every time when executing an object program, a correct program corresponding thereto. For example, when the CPU 101 has already executed the correct program and acquired the correct entire image and extracted the correct partial image, the CPU 101 only needs to execute only the object program to acquire the object entire image and extract the object partial image in the verification processing.

[0039] As a fifth modification example, the image to be compared with the object partial image is not limited to the partial image. For example, in the case where the object program is executed to verify whether or not a specific image can be displayed, it is only necessary to compare a specific image prepared in advance with the object partial image extracted from the object entire image. In this case, the specific image (entire image) is the correct image.

[0040] (Second Embodiment)

Next, a test apparatus 100 according to a second embodiment will be described. The test apparatus 100 according to the second embodiment has a plurality of determination conditions for determining that both images accord in comparison between an object partial image and a correct partial image, and can determine accord or disaccord between both the images according

to a determination condition different for each object partial image. Hereinafter, the test apparatus 100 according to the second embodiment will be described about points different from the test apparatus 100 according to the first embodiment.

[0041] First of all, determination conditions that the test apparatus 100 according to the second embodiment can apply will be described. The test apparatus 100 according to this embodiment can allocate the following determination conditions to object partial images.

[0042] A first determination condition is for determining that both the images accord when the object partial image and the correct partial image, which form a set, accord as a whole as described in the first embodiment. Note that the test apparatus 100 can set also a determination condition in which the condition for determining accord (for example, the threshold value for the difference between pixel values for determining accord or the like) is different as described in the first embodiment.

[0043] A second determination condition is for determining that both the images accord by comparing the whole object partial image realized by a scroll display and the correct partial image in the case where one object partial image is scroll-displayed in the object entire image. Fig. 7 is an explanatory view of the second determination condition. When the object partial image cannot be obtained from one

object entire image as in Fig. 7, a CPU 101 extracts a "c1" portion from an object entire image 700 where "c1" being a part of an object partial image "c" is displayed. The CPU 101 further extracts a "c2" portion from an object entire image 710 where "c2" being a portion other than the "c1" of the object partial image "c" is displayed. The CPU 101 then combines the extracted portions to thereby obtain the object partial image "c".

[0044] Note that in the case where one correct partial image is scroll-displayed in the correct entire image, the test apparatus 100 can set also a determination condition for comparing the whole correct image realized by scroll display and the object partial image. Further, in the case where the correct entire image and the object entire image are scroll-displayed, the test apparatus 100 can set also a determination condition of comparing the whole correct partial image and the whole object partial image which are realized by scroll display.

[0045] A third determination condition is for determining that both the images accord when a part of the object partial image and the correct partial image accord. One example of the third determination condition is for determining that both the images accord when a region except a lower side region of the object partial image and a corresponding correct partial image accord. Fig. 8A and Fig. 8B are explanatory views of this determination condition.

It is assumed that an object partial image "d" is expanded accompanying expansion of another partial image e in the longitudinal direction as in Fig. 8A and Fig. 8B. In this case, the object partial image "d" is drawn in a region 820 and a region 821 under the region 820 may become a blank space. In this case, the CPU 101 compares pixels of a correct partial image "D" and the object partial image "d" in order from a top, and determines whether or not both the images accord at a point in time when the comparison with all of the pixels of the correct partial image "D" ends. This makes it possible to determine whether or not both the images accord by the comparison between the object partial image "d" and the correct partial image "D" even if a blank space with unknown size is generated on the lower side.

[0046] Note that even when blank spaces are generated on the upper portion, the right side and the left side of the object partial image, the test apparatus 100 can set also a determination condition for determining whether or not both the images accord regardless of the sizes of the blank spaces.

[0047] Contrarily, there is a case where the correct partial image includes a blank space, whereas the blank space disappears in a corresponding object partial image. For coping with such a case, even when blank spaces are generated on the upper portion, the right side and the left side of the correct

partial image, the CPU 101 can set also a determination condition for determining whether or not both the images accord regardless of the sizes of the blank spaces.

[0048] A fourth determination condition is for determining whether or not images accord, when the size of a blank space between two object partial images changes, by comparing the two object partial images after removing the blank space with a corresponding correct partial image. Fig. 9 is an explanatory view of the fourth determination condition. It is assumed that a blank space 901 between correct partial images "F" and "G" in a correct entire image 900 is expanded to a blank space 911 in an object entire image 910 accompanying expansion of a correct partial image "H" in the correct entire image 900 to an object partial image "h" in the object entire image 910. Also in this case, the test apparatus 100 compares the images of the correct partial images "F" and "G" with object partial image "f" and "g" after removing (trimming) the blank space. This enables an accord determination ignoring the blank space.

[0049] A fifth determination condition is for determining that both the images accord even if partial regions of the correct partial image and the object partial image do not accord. Fig. 10A and Fig. 10B are explanatory views of the fifth determination condition. It is assumed that information different

for every execution of the program such as each log-in date and time is displayed in a region "J" of a correct partial image 1000 and a region "j" of an object partial image 1010 corresponding thereto. In this case, the CPU 101 determines, using the region "J" and the region "j" as mask regions, that both the images accord when a correct partial image "I" and an object partial image "i" other than the mask regions accord. Note that the mask regions are associated with the image ID in the program as with the object partial image and the like. Accordingly, the CPU 101 can specify the mask regions by inquiring of the browser. Note that, as another example, the CPU 101 may determine that the correct partial image "I" and the object partial image "i" including the mask regions accord, after making all of pixels in the mask regions "J", "j" into equal pixels such as into black pixels.

[0050] A sixth determination condition is for comparing the object partial image and the correct partial image after removing the mask regions also in a case where the size of the mask region changes. Fig. 11 is an explanatory view of the sixth determination condition. It is assumed that a mask region "l" of an object entire image 1110 corresponding to a mask region "L" of a correct partial image "K" in a correct entire image 1100 is expanded accompanying expansion of a partial image "m" in the longitudinal direction. Also in this case,

the test apparatus 100 compares a portion obtained by removing the mask region "L" from the correct partial image "K" and a portion obtained by removing the mask region "l" from an object partial image "k". This enables an accord determination ignoring the mask regions. The test apparatus 100 specifies also the mask regions using the image ID as a key as described above, so that even when the size of the mask region is changed, the test apparatus 100 can correctly specify the mask region after the change of the size.

[0051] In the test apparatus 100 according to this embodiment, one of the above-described determination conditions can be set in units of object entire image for the object partial image. The test apparatus 100 can further set a different determination condition for each of a plurality of object partial images contained in the object entire image. Note that setting of the determination condition for each of the object partial images has been already completed before execution of the verification processing. Further, setting of the determination condition for each of the object partial images is performed, for example, by the CPU 101 recording the determination condition, for example, in the ROM 102 while associating the determination condition with the image ID according to a user operation. As another example, the determination condition may be defined in association with the image ID as the specification information on the verification object in a

verification program.

[0052] Fig. 12 is a flowchart illustrating the verification processing by the test apparatus 100 according to the second embodiment. Note that the same processing as each processing of the verification processing (Fig. 2) according to the first embodiment is denoted by the same numeral. The CPU 101 selects a set of the partial images at S203, and then advances the processing to S1200. At S1200, the CPU 101 specifies the determination condition set for the selected object partial image. The CPU 101 then advances the processing to S204. At S204, the CPU 101 determines whether or not both the images accord according to the determination condition specified at S1200. Note that the configuration and processing other than the above of the test apparatus 100 according to the second embodiment are the same as the configuration and processing of the test apparatus 100 according to the first embodiment. The CPU 101 according to the second embodiment can determine whether or not each of the plurality of object partial images with the correct partial images according to different determination conditions as described above.

[0053] (Third Embodiment)

Next, a test apparatus 100 according to a third embodiment will be described. In a verification program executed in the test apparatus 100 according to the third embodiment, a class is defined as

specification information on a verification object. A CPU 101 extracts a correct partial image group composed of a plurality of correct partial images drawn according to a description associated with a class defined in the verification program, of descriptions of an HTML text of an object program. The CPU 101 similarly extracts an object partial image group composed of a plurality of object partial images drawn according to a description associated with a class defined in the verification program, of the descriptions of the HTML text of the object program. The class mentioned here is one example of an attribute. Hereinafter, the test apparatus 100 according to the third embodiment will be described about points different from the test apparatuses 100 according to other embodiments.

[0054] Fig. 13 is a flowchart illustrating verification processing by the test apparatus 100 according to the third embodiment. Note that the same processing as each processing of the verification processing (Fig. 2) according to the first embodiment is denoted by the same numeral. After the processing at S201, the CPU 101 advances the processing to S1300. At S1300, the CPU 101 extracts partial images specified based on a predetermined class as the correct partial images from a correct entire image. The CPU 101 further extracts partial images specified based on the same class as the class referred to when extracting the

correct partial images, as the object partial images from an object entire image.

[0055] Next, at S1301, the CPU 101 compares each of the object partial images extracted at S1300 with each of the correct partial images extracted at S1300 to thereby specify the correct partial image to be paired up with each of the object partial images. The CPU 101 confirms whether or not all of object partial images and correct partial images extracted at S1300 have been paired up with each other. The CPU 101 performs full search for the correct partial images and thereby detect the correct partial image to be paired up with each of the object partial images.

[0056] For example, it is assumed that a correct partial image group including correct partial images "N", "O", "P" is extracted from a correct entire image 1400 and an object partial image group including object partial images "n", "p", "o" is extracted from an object entire image 1410 as illustrated in Fig. 14. In this case, the CPU 101 retrieves a correct partial image to be paired up with the object partial image "p" first. More specifically, the CPU 101 selects one partial image among three correct partial images. For example, when selecting the correct partial image "N", the CPU 101 compares the object partial image "p" and the correct partial image "N". When both the images do not accord, the CPU 101 selects the correct partial

image "O", and when both the images do not accord again, the CPU 101 selects the correct partial image "P". Then, when the object partial image "p" and the correct partial image "P" accord, the CPU 101 determines that both the images pair up with each other. Through the same processing, the CPU 101 retrieves correct partial images to be paired up with the object partial images "o", "n".

[0057] When all of the object partial images and correct partial images extracted at S1300 are paired up with each other (Yes, at S1301), the CPU 101 advances the processing to S1303. When at least one of an unpaired object partial image and an unpaired correct partial image exists (No at S1301), the CPU 101 advances the processing to S1302. Examples of the unpaired case mentioned here include a case where some or all of the object partial images and correct partial images do not accord and a case where the number of object partial images and the number of correct partial images do not accord so that corresponding partial images (object partial images or correct partial images) do not exist.

[0058] At S1302, the CPU 101 executes emphasis processing on a disaccord portion. For example, in the case where some or all of the object partial images and correct partial images do not accord, the CPU 101 regards portions not according or the whole images as disaccord portions, and executes emphasis processing on the disaccord portions. Besides, in

the case where the number of object partial images and the number of correct partial images do not accord, the CPU 101 regards partial images (object partial images or correct partial images) for which partial images to be paired up therewith have not been searched as disaccord portions, and executes emphasis processing on the disaccord portions. Note that the configuration and processing other than the above of the test apparatus 100 according to the third embodiment are the same as the configurations and processing of the test apparatuses 100 according to other embodiments.

[0059] Note that the test apparatus 100 described in this embodiment may be realized by a system composed of a plurality of apparatuses connected over a network. In other words, execution of the verification program, correct program, and object program by the test apparatus 100 may be individually performed by different apparatuses. Besides, execution of programs, image display, storage of image file may be individually performed by different apparatuses.

[0060] According to the above-described embodiments, it is possible to appropriately execute verification of a program including a description relating to drawing of an image.

[0061] Preferred embodiments of the present invention have been described above, but the present invention is not limited to those specific

embodiments and can be variously modified and changed within a scope of the present invention described in claims.

CLAIMS

[Claim 1] An information processing apparatus comprising:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image corresponding to a correct image, specify a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extract the object image at the specified position;

a determiner configured to determine whether or not the object image and the correct image accord; and

a display processor configured to display a result of determination by the determiner.

[Claim 2] The information processing apparatus according to claim 1, further comprising:

a second acquirer configured to acquire a second image displayed on the display in response to execution of a correct program; and

a correct image extractor configured to extract a partial image as a correct image from the second image, based on specification information specifying the partial image of the second image.

[Claim 3] The information processing apparatus according to claim 1 or 2, wherein the object image extractor extracts a partial image drawn according to a description as the object image, the description being associated with the first image identifier in the object program.

[Claim 4] The information processing apparatus according to claim 2, wherein:

the specification information is a second image identifier being information identifying the correct image included in the second image; and

the correct image extractor extracts a partial image drawn according to a description as the correct image, the description being associated with the second image identifier in the correct program.

[Claim 5] An information processing apparatus comprising:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser positions of a plurality of object images in the first image, based on attribute information indicating a class common to the plurality of object images, the object images corresponding to a correct image, specify each position of a region where each of the plurality of object images is actually drawn by the browser,

based on a response from the browser, the response including each position of the region, and extract the object image at specified each position,

a determiner configured to determine whether or not each of a plurality of the object images and each of a plurality of the correct images accord; and

a display processor configured to display a result of determination by the determiner.

[Claim 6] The information processing apparatus according to any one of claims 1 to 5, wherein the determiner determines whether or not the object image and the correct image accord, based on a result of comparison between pixels of the object image and the correct image.

[Claim 7] The information processing apparatus according to any one of claims 1 to 5, wherein the determiner determines whether or not the object image and the correct image accord, based on a result of comparison between an edge detected from the object image and an edge detected from the correct image.

[Claim 8] The information processing apparatus according to any one of claims 1 to 7, wherein the determiner determines that the object image and the correct image accord when a part of the object image and the correct image accord.

[Claim 9] The information processing apparatus according to any one of claims 1 to 7, wherein the determiner determines that the object image and the correct image accord when a region except a part of the object image and a region except a part of the correct image accord.

[Claim 10] The information processing apparatus according to any one of claims 1 to 5, wherein the determiner determines whether or not each of a plurality of object images and a correct image accord, according to a different determination condition, the different determination condition being set in advance for each of the plurality of object images.

[Claim 11] An information processing method executed by an information processing apparatus, the information processing method comprising:

a first acquisition step of acquiring a first image displayed on a display in response to execution of an object program being a test object;

an object image extraction step of inquiring of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image corresponding to a correct image, specifying a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extracting the object image at the specified position;

a determination step of determining whether or not the object image and the correct image accord; and

a display processing step of displaying a result of determination at the determination step.

[Claim 12] A computer readable non-transitory recording medium with a program causing a computer to execute:

a first acquirer configured to acquire a first image displayed on a display in response to execution of an object program being a test object;

an object image extractor configured to inquire of a browser a position of an object image in the first image, based on a first image identifier identifying the object image, the object image corresponding to a correct image, specify a position of a region where the object image is actually drawn by the browser, based on a response from the browser, the response including the position of the region, and extract the object image at the specified position;

a determiner configured to determine whether or not the object image and the correct image accord; and

a display processor configured to display a result of determination by the determiner.