UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/332,005	05/27/2021	Daniel DINES	1678.0043CON3	1220

94165 7590 05/16/2024

LeonardPatel PC
9891 Irvine Center Drive
Suite 100
Irvine, CA 92618

EXAMINER	
TRAN, TUYETLIEN T	

ART UNIT	PAPER NUMBER
2179	

NOTIFICATION DATE	DELIVERY MODE
05/16/2024	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

mleonard@leonardpatel.com
patents@leonardpatel.com
spatel@leonardpatel.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte DANIEL DINES

Appeal 2023-002881
Application 17/332,005
Technology Center 2100

Before JOHN A. JEFFERY, ERIC S. FRAHM, and JUSTIN BUSCH, *Administrative Patent Judges*.

JEFFERY, *Administrative Patent Judge*.

DECISION ON APPEAL

Under 35 U.S.C. § 134(a), Appellant[1] appeals from the Examiner's decision to reject claims 1–20. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] Appellant identifies the real party in interest as UiPath, Inc. Appeal Br. 3.

STATEMENT OF THE CASE

Appellant's invention executes multiple graphical element detection techniques sequentially to find matches between user interface (UI) element attributes and UI descriptor attributes for an activity of a robotic process automation (RPA) workflow. *See* Abstract. Claim 1 is illustrative:

> 1. A non-transitory computer-readable medium storing a computer program, the computer program configured to cause at least one processor to:
>
> analyze a user interface (UI) at runtime to identify UI element attributes for one or more UI elements;
>
> compare the UI element attributes to UI descriptor attributes for an activity of a robotic process automation (RPA) workflow using one or more initial graphical element detection techniques; and
>
> when a match is not found between the UI element attributes and the UI descriptor attributes of the activity using the one or more initial graphical element detection techniques during a first time period:
>
> execute one or more additional graphical element detection techniques that compare the UI element attributes to the UI descriptor attributes for the activity of the RPA workflow instead of the one or more initial graphical element detection techniques.

THE REJECTIONS

The Examiner provisionally rejected claims 8 and 9 on the ground of nonstatutory double patenting as unpatentable over claims 8 and 9 of copending U.S. Application 17/746,582. Final Act. 6–9.[2]

[2] Throughout this opinion, we refer to (1) the Final Rejection mailed November 30, 2022 ("Final Act."); (2) the Appeal Brief filed January 5,

The Examiner rejected claims 1–20 under 35 U.S.C. § 103 as unpatentable over Butin (US 2010/0205529 A1; published Aug. 12, 2010) and Colciago (US 2009/0167493 A1; published July 2, 2009). Final Act. 9–70.

THE PROVSIONAL DOUBLE PATENTING REJECTION

Because Appellant does not contest the Examiner's provisional nonstatutory double patenting rejection of claims 8 and 9 (*see* Final Act. 6–9), we summarily sustain this rejection. *See* Manual of Patent Examining Procedure (MPEP) § 1205.02 (9th ed. rev. 07.2022 Feb. 2023) ("If a ground of rejection stated by the examiner is not addressed in the appellant's brief, appellant has waived any challenge to that ground of rejection and the Board may summarily sustain it, unless the examiner subsequently withdrew the rejection in the examiner's answer.").

Although Appellant filed (1) a terminal disclaimer on December 20, 2021, referencing Application Number 17/016,251; and (2) a terminal disclaimer on June 24, 2022, referencing Application Numbers 17/014,171 and 17/331,558, on this record, there is no terminal disclaimer referencing Application Number 17/746,582 that the Examiner cites in the rejection.

THE OBVIOUSNESS REJECTION

The Examiner finds that Butin discloses every recited element of independent claim 1, but may not *explicitly* disclose not finding a match between UI element attributes and UI descriptor attributes using one or more

---

2023 ("Appeal Br."); (3) the Examiner's Answer mailed April 3, 2023 ("Ans."); and (4) the Reply Brief filed June 2, 2023 ("Reply Br.").

initial graphical element detection techniques during a first time period. *See* Final Act. 9–14. Although the Examiner finds that Butin suggests this time period, the Examiner nevertheless also cites Colciago as teaching this feature in concluding that the claim would have been obvious. *See id.* at 15–17.

Appellant argues that Butin fails to disclose RPA workflow or associated activities, much less comparing UI element attributes to UI descriptor attributes for an activity of an RPA workflow as claimed. *See* Appeal Br. 20–28; Reply Br. 2–7. According to Appellant, ordinarily skilled artisans would not conclude that Butin's interactive guidance scripts are RPA workflows that provide instructions to RPA robots. Appeal Br. 22–23; Reply Br. 5–7. Appellant also contends that Butin does not determine that a match is not found between UI element attributes and UI descriptor attributes for an activity of an RPA workflow as claimed, and Colciago does not cure that deficiency. *See* Appeal Br. 28–35; Reply Br. 7–13. Appellant adds that Colciago is non-analogous art with respect to the claimed invention and, therefore, not reasonably combinable. *See* Appeal Br. 35–40; Reply Br. 13–14. Appellant argues other recited limitations summarized below.

ISSUES

Under § 103, has the Examiner erred by finding that Butin would have taught or suggested:

(1) a computer program configured to cause a processor to (a) compare identified UI element attributes to UI descriptor attributes for an activity of an RPA workflow using one or more initial graphical element detection techniques; and (b) when a match is not found between those attributes using the initial detection technique(s) during a first time period,

execute one or more additional graphical element detection techniques that compare the attributes instead of the initial detection technique(s) as recited in claim 1?

(2) when a match is not found using the one or more additional graphical element detection techniques during a second time period, executing one or more supplemental graphical element detection techniques instead of the one or more initial and supplemental graphical element detection techniques as recited in claim 2?

ANALYSIS

Claims 1, 3–8, 10–14, and 16–20

We begin by noting that claim 1 recites a conditional limitation, namely *when* a match is not found between the UI element attributes and UI descriptor attributes using at least one initial graphical element detection technique during a first time period, executing at least one additional graphical element detection technique instead of the initial technique.

Our emphasis on the term "when" underscores claim 1's condition that is a key aspect of the claimed invention. As the Examiner correctly indicates (*see* Ans. 48–49), when a condition is part of a recited method, as in independent claim 8, the condition need not be satisfied to meet the claim. *See Ex parte Schulhauser*, Appeal No. 2013-007847, at 6–10 (PTAB Apr. 28, 2016) (precedential); *see also Cybersettle, Inc. v. Nat'l Arbitration Forum, Inc.*, 243 F. App'x 603, 607 (Fed. Cir. 2007) (unpublished) (noting the Federal Circuit has held that "[i]f the condition for performing a contingent step is not satisfied, the performance recited by the step need not be carried out in order for the claimed method to be performed"); *Applera*

Corp. – Applied Biosystems Grp. v. Illumina, Inc., 375 F. App'x 12, 21 (Fed. Cir. 2010) (unpublished) (affirming a method claim's interpretation as including a step that need not be practiced if the condition for practicing the step is not met); MPEP § 2111.04(II) (citing *Schulhauser*).[3]

Although conditional limitations need not be met to satisfy method claim 8, the same cannot be said for claim 1 that recites a computer-readable medium whose stored instructions perform functions that occur only if the recited conditions are satisfied. *See Schulhauser*, at 14–15. That is, claim 1—and apparatus claim 14—still require structure for performing the functions should the conditions occur. *See id.*

Despite this distinction, this appeal nevertheless turns on a key fundamental question: what is an "RPA workflow?" The Specification does not define the terms "RPA" or "RPA workflow" unlike other terms whose concrete definitions leave no doubt as to their meaning. *See, e.g.*, Spec. ¶¶ 25 (defining "screen"), 43 (equating the terms "user" and "developer"), 46 (defining "activities"), 109 (defining "certain embodiments").

The Specification's paragraphs 45 to 57 do, however, explain that "RPA" stands for "robotic process automation" and describe the RPA system 100 shown in Figure 1. As shown in that figure, the RPA system includes a "designer" 110 that facilitates developing and deploying workflows and robots. *Id.* ¶ 45. Some types of workflows may include, *but*

[3] Although the limitations at issue in *Schulhauser* were rendered conditional by reciting the term "if," we discern no meaningful distinction between the recitation of "if" and "when" in this context. One dictionary definition of "when" is "in the event that : IF." *When*, MERRIAM-WEBSTER.COM, https://www.merriam-webster.com/dictionary/when (2024).

are not limited to, sequences, flowcharts, finite state machines (FSMs), and/or global exception handlers. *Id.* ¶ 47.

As the Specification's paragraph 46 explains, the system enables automating rule-based processes by giving the developer control of execution order and the relationship between a custom set of steps developed in a workflow (i.e., "activities"). Each activity may include an action, such as clicking a button, reading a file, writing to a log panel, etc. *Id.* ¶ 46.

Once a workflow is developed in the designer, "conductor" 120 orchestrates business process execution by orchestrating one or more "robots" 130 that execute the workflows developed in the designer. *Id.* ¶ 48. These robots are "execution agents" that run workflows built in the designer and include, *but are not limited to*, attended robots 132, unattended robots 134, development robots, and nonproduction robots. *Id.* ¶ 49. Both attended and unattended robots may automate various systems and applications including, *but not limited to*, (1) mainframes; (2) web applications; (3) virtual machines (VMs); (4) enterprise applications; and (5) computing system applications, such as those for desktops, laptops, mobile devices, and wearable computers, etc. *Id.* ¶ 50. The conductor's capabilities include, *but are not limited to*, provisioning, deployment, versioning, configuration, queueing, monitoring, logging, and/or providing interconnectivity. *Id.* ¶ 51.

Although this description—replete with permissive and non-limiting terminology—informs our understanding of "RPA" and "RPA workflows" in the context of the invention, our interpretation of those terms is not limited to that particular description; nor will we import that description into the claim. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (en banc) ("[A]lthough the specification often describes very specific

embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments. . . . [C]laims may embrace different subject matter than is illustrated in the specific embodiments in the specification." (citations and internal quotation marks omitted)).

We, therefore, construe the term "RPA workflow" with its understood meaning in the art consistent with its usage in the Specification. According to the "Federal RPA Community of Practice," a collaboration among different U.S. Federal agencies to accelerate RPA adoption government-wide, "RPA" is a "low to no-code Commercial Off the Shelf (COTS) technology that can be used to automate repetitive, rules-based tasks." Federal RPA Community of Practice, *RPA Program Playbook*, Version 1.1 (Jan. 17, 2020) ("RPA Program Playbook"), at 2, 4. This government document further explains that "[l]ike an Excel macro operating within a spreadsheet, RPA can record actions performed across a personal computer, access systems, and perform delineated tasks for human users." *Id.* at 4. Although RPA products can vary in their exact capabilities, all RPA technologies emulate human actions, and popular RPA uses include data entry, data reconciliation, spreadsheet manipulation, systems integration, automated data reporting, analytics, and customer outreach and communications. *Id.*

The U.S. General Services Administration (GSA) also defines RPA as "a business process automation technology that automates manual tasks that are largely rules-based, structured and repetitive using software robots . . . RPA tools map a process for a robot to follow which allows the [ro]bot to

operate in place of a human." U.S. GSA, *Robotic Process Automation*, Tech at GSA, tech.gsa.gov (July 8, 2019).[4]

Another authority defines RPA similarly, namely "a productivity tool that allows a user to configure one or more scripts (which some vendors refer to as 'bots') to activate specific keystrokes in an automated fashion." *Robotic Process Automation (RPA)*, Gartner Information Technology Glossary, Gartner, Inc. (June 5, 2019) ("Gartner IT Glossary").[5] The Gartner glossary further explains that "[t]he result is that the bots can be used to mimic or emulate selected tasks (transaction steps) within an overall business or IT process. These may include manipulating data, passing data to and from different applications, triggering responses, or executing transactions." *Id.* Moreover, "[t]he scripts can overlay on one or more software applications." *Id.*

Based on these authorities, we construe the term "RPA" broadly, but reasonably, as technology that automates tasks that are largely rules-based, structured, and repetitive.

The Specification also does not define the term "workflow," but does indicate that workflows are (1) developed using "designer" 310; (2) executed by robots; and (3) *may* include user-defined and UI automated activities. *See* Spec. ¶ 67; Fig. 3. The Specification adds that workflows

[4] Although the GSA's web page is undated, the Google search result identifying and linking this page indicates that its date is July 8, 2019. We, therefore, presume that this document was publicly available on that date.
[5] Although the Gartner Information Technology Glossary web page indicates a 2024 copyright date, the Google search result identifying and linking this page nevertheless indicates that its date is June 5, 2019. We, therefore, presume that this document was publicly available on that date.

may be nested or embedded, and *some* types of workflows *may* include, *but are not limited to*, sequences, flowcharts, FSMs, and/or global exception handlers. Spec. ¶¶ 46–47.

Although this description, with its emphasized permissive and non-limiting terminology, informs our understanding of the term "workflow" in the context of the invention, our interpretation of the term is not limited to that particular description; nor will we import that description into the claim. *See Phillips*, 415 F.3d at 1323.

We, therefore, construe the term "workflow" with its commonly understood meaning in the art, namely "multiple tasks/steps/activities, of which there are two types: simple, representing indivisible activities, and compound, representing those which can be decomposed into subactivities." DICTIONARY OF COMPUTER SCIENCE, ENGINEERING, AND TECHNOLOGY 532 (Phillip A. Laplante ed. 2001).

Given these authorities, we construe the term "RPA workflow" broadly, but reasonably, as automated tasks, steps, or activities that are largely rules-based, structured, and repetitive.

With this construction, we see no error in the Examiner's reliance on Butin for suggesting not only an RPA workflow and associated activities, but also for suggesting the recited attribute comparison for an activity of an RPA workflow. As explained in the Abstract, Butin's system creates interactive guidance to present to a user of a computerized application by (1) receiving a user action entered through an input device with respect to the application; (2) adding an indication of the user action to an interactive guidance script associated with the application; and (3) storing the script in an interactive guidance script repository.

As shown in Figure 1, Butin's system includes device 110, which can be a personal computer, that includes "guided application" 150, such as a word processing or spreadsheet application, for which a user seeks guidance. Butin ¶ 63. Butin's device also includes "helper" 140 that may be used as an interactive tutorial and navigation tool for the guided application. *Id.* ¶¶ 64, 77. For example, after the user enters a question about using or operating the guided application or an associated feature or function, the helper (1) obtains interactive guidance and navigation; (2) presents that guidance and navigation; and (3) executes required operations automatically based on pre-recorded and pre-stored guidance scripts 151 that are played back to the user. *Id.* ¶ 77.

For example, the helper can present interactive help by automatically (1) displaying and moving a cursor over the guided application's windows; and (2) entering mouse clicks, keyboard presses, and keystrokes, thus executing the guided application's actual commands and operations based on the relevant predefined guidance script. *See id.*

These guidance scripts, which are stored locally and/or remotely, are created in advance by a "guidance professional," such as a trainer, teacher, instructor, support specialist, or administrator, using script recorder 153 on guidance station 190. *Id.* ¶ 78, Fig. 1. To this end, the script recorder tracks and records mouse movement, mouse clicks, and/or keyboard presses. *Id.* ¶ 115.

The import of this functionality is that the functions, commands, and operations that are executed automatically by Butin's guidance script are an "RPA workflow" under the term's broadest reasonable interpretation, namely automated tasks, steps, or activities that are largely rules-based,

structured, and repetitive. Notably, these activities comport fully with Appellant's definition of the term "activities" in paragraph 46 of the Specification, for they are a custom set of steps developed in a workflow. That the developer, namely Butin's guidance professional, can control not only the execution order, but also the relationship between those steps in the recording process underscores this point. Appellant's contentions to the contrary, including the contention that Butin's guidance professional ostensibly does not create a custom set of ordered steps (*see* Reply Br. 5–6) is unavailing, for the guidance professional does just that by selecting and recording particular steps in a particular sequence, where these recorded sequential steps execute particular functions in that order. *See* Butin ¶¶ 77–78, 115.

In short, Appellant's contention that Butin does not disclose an "RPA workflow" is unavailing and not commensurate with the scope of the limitation. That recognized authorities in the art have defined RPA in a way that is strikingly similar to Butin's scripting technique underscores this point. *See, e.g.*, Gartner IT Glossary (defining "RPA" as "a productivity tool that allows a user to configure one or more *scripts* (which some vendors refer to as 'bots') to activate specific keystrokes in an automated fashion" (emphasis added)).

Nor do we find error in the Examiner's reliance on Butin for at least suggesting comparing identified UI element attributes to UI descriptor attributes for[6] an activity of an RPA workflow using one or more initial

[6] For clarity, we note the limitation recites attributes *for* an RPA workflow activity—not *of* an RPA workflow activity as Appellant inartfully indicates. *See, e.g.*, Appeal Br. 30, 32, 34–35.

graphical element detection techniques as claimed. *See* Final Act. 10–12; Ans. 36.

As Butin's paragraph 148 explains, Optical Character Recognition (OCR) algorithms are used to identify the appropriate place to record and/or play the scripts' actions. To this end, in the recording phase, OCR engine 172, which is in guidance station 190, (1) identifies a text item that appears under or near the mouse pointer; and (2) assists in identifying an image located under or near the mouse pointer by, for example, determining boundaries. *See* Butin ¶ 149, Fig. 1. In the playback phase, OCR engine 171, which is in device 110, is used to find on the screen the location of a particular textual item or image appearing in the recorded guidance script. *See* Butin ¶ 149, Fig. 1. As noted in paragraph 150, Butin's system also bridges possible gaps between data recorded by script recorder 153 and data required by script player 152 by determining partial matches or sufficiently close matches in recognized text strings.

The import of this functionality is that to locate the appropriate place of a UI element on a screen that is used to execute a recorded script, Butin's system compares "UI element attributes" (e.g., images, text, and their associated qualities, properties, or characteristics which are consistent with the non-limiting examples of UI element attributes listed in the Specification's paragraphs 25, 97, and 104) to "UI descriptor attributes" (i.e., attributes that at least pertain to a set of instructions for finding a UI element, which are consistent with the Specification's description of UI descriptors in paragraph 31).

We reach this conclusion noting that the Specification does not define the term "attributes" which, under its plain meaning, is quite broad, namely

"quality, property or characteristic of anything." *See* HARPERCOLLINS COMPACT DICTIONARY & THESAURUS 49 (2003). We, therefore, construe the term "attributes" similarly.

In short, identifying and locating a particular element on a screen, such as an image or a text item that appears under or near a mouse pointer, as taught by Butin, compares attributes (i.e., qualities, properties, or characteristics) of a UI element (e.g., images, text, and/or the mouse pointer) with other attributes (e.g., data associated with those elements' boundaries, locations, or other indicia) that at least pertain to properties or characteristics used to find the UI element. We reach this conclusion even assuming, without deciding, that Butin's OCR functionality is not part of the script itself as Appellant contends (*see* Appeal Br. 27), for the OCR's identifying, locating, and recognizing functions are nevertheless performed automatically in connection with script recording and playback. Indeed, these automatic functions are key initial steps in Butin's script-based workflow.

Simply put, automatically recognizing particular images or text on a screen to find the appropriate location at which to play a script, as taught by Butin, involves, among other things, comparing and matching UI element attributes and UI descriptor attributes for the activity associated with that script which, as noted previously, is an activity of an RPA workflow, namely automated tasks, steps, or activities that are largely rules-based, structured, and repetitive. To the extent Appellant contends otherwise, such arguments are unavailing and not commensurate with the scope of the limitation. That the Specification's paragraph 24 states, quite broadly, that *any* suitable

attributes and graphical element detection techniques can be used without deviating from the scope of the invention underscores this point.

Nor do we find error in the Examiner's reliance on Butin for at least suggesting that when a match is not found between the UI element attributes and UI descriptor attributes using the initial detection technique during a first time period, executing one or more additional graphical element detection techniques that compare those attributes instead of the initial graphical element detection technique. *See* Final Act. 12–14; Ans. 18–19.

Butin's Figure 5 shows the steps involved when a mouse click is replayed in a guidance script. Butin ¶ 164. As shown in that figure, once the system finds a relevant window in which the user action is to be performed in step 517, image recognition is then performed in step 525 to find the image on which the action is performed. *See id.* ¶¶ 164–169. If image recognition is successful, the user action imitation is then executed at the exact relevant screen position or a position closest to the original screen position. *See id.* ¶ 172, Fig. 5 (steps 532–541).

But if image recognition is unsuccessful, OCR is performed to search in the window for relevant text on which the action is to be performed. *See id.* ¶¶ 171, 173, Fig. 5 (steps 531, 545). If OCR is successful, the user action imitation is then executed at the exact relevant text item or a text item closest to the original text item. *See id.* ¶ 176, Fig. 5 (steps 552–576).

The import of this functionality is that the initial graphical element detection technique in Butin's Figure 5, namely image recognition, may not find an image on which an automatically-imitated user action is performed. *See* Butin ¶¶ 164–169. That the system cannot find the image on the screen associated with that automated activity using this initial detection technique

at least suggests that the associated UI element and descriptor attributes, namely at least some of their qualities, properties, or characteristics, do not match.

Notably, this initial graphical element detection technique, namely image recognition, is used during at least *some* time period even assuming, without deciding, that Butin does not specify its duration. That is, Butin uses this initial graphical element detection technique during a "first time period" as claimed which, as the Examiner indicates, is *some* time period. *See* Final Act. 14 (finding that, in Butin, *some* period of time must elapse before determining whether image recognition was successful); Ans. 37 (same).

This finding renders the Examiner's reliance on Colciago superfluous to Butin in this regard, for the Examiner cites Colciago merely to teach what Butin *may* not disclose *explicitly*, namely the phrase "during a first time period." *See* Final Act. 14; Ans. 37–38. In short, Colciago is unnecessary to the Examiner's rejection, which finds that Butin at least suggests *every* recited limitation—including the first time period limitation. *See* Final Act. 14; Ans. 37–38.

As noted above, when image recognition is unsuccessful in step 531 of Butin's Figure 5, an additional graphical element detection technique, namely OCR, is executed to search in the window for relevant text on which the action is to be performed. *See id.* ¶¶ 171, 173, Fig. 5 (steps 531, 545). As with the initial graphical element detection technique that recognizes images, Butin's additional OCR technique likewise compares the associated UI element and descriptor attributes, namely at least some of their qualities, properties, or characteristics, to recognize and find the associated textual

elements for script playback. That is, like image recognition, automatically recognizing particular text on a screen to find the appropriate location at which to play a script as in Butin involves, among other things, comparing and matching UI element and descriptor attributes for the activity associated with that script which, as noted previously, is an activity of an RPA workflow, namely automated tasks, steps, or activities that are largely rules-based, structured, and repetitive. Appellant's arguments to the contrary are unavailing and not commensurate with the scope of the claim.

In short, we find—as does the Examiner—that Butin alone at least suggests every recited limitation—including the first time period limitation. *See* Final Act. 14; Ans. 37–38.

We, therefore, see no harmful error in the Examiner's rejection despite Colciago being cumulative to Butin. Although Butin may not state *explicitly* that the initial graphical element detection technique is used during a first time period, Butin nonetheless suggests as much.

Accordingly, we sustain the Examiner's obviousness rejection based solely on Butin, for we may rely on fewer references than the Examiner in affirming a multiple-reference obviousness rejection where, as here, we rely on the same teachings relied upon by the Examiner. *See In re Kronig*, 539 F.2d 1300, 1302 (CCPA 1976); *see also In re Bush*, 296 F.2d 491, 496 (CCPA 1961); *accord* MPEP § 1207.03(a)(II)(3) (discussing *Kronig* and *Bush*).

Appellant's arguments, then, regarding Colciago's alleged shortcomings, including the reference ostensibly constituting non-analogous art, are moot given our sole reliance on Butin.

Therefore, we are not persuaded that the Examiner erred in rejecting claim 1 and claims 3–8, 10–14, and 16–20, not argued separately with particularity.[7]

Claims 2, 9, and 15

We also sustain the Examiner's rejection of claim 2 reciting, in pertinent part, when a match is not found using the one or more *additional* graphical element detection techniques during a second time period, executing one or more *supplemental* graphical element detection techniques instead of the one or more initial and supplemental graphical element detection techniques.

Our emphasis underscores that, when reading claim 2 in light of claim 1 from which it depends, the supplemental graphical element detection technique is a tertiary technique that is used when the first two graphical element detection techniques, namely the initial and additional graphical element detection techniques, do not find a match. In other words, claim 2 requires a specific sequence of graphical element detection techniques that are used when no match is found: (1) the initial technique; (2) the additional technique; and (3) the supplemental technique.

Although Butin's Figure 5 shows two graphical element detection techniques, namely image recognition and OCR, we nevertheless see no

[7] Although claims 8 and 14 are nominally argued separately (*see* Appeal Br. 41–80; Reply Br. 16), the arguments are similar to those made for claim 1. We, therefore, group these claims together. Moreover, as noted previously and as the Examiner correctly indicates (*see* Ans. 48–49), claim 8 is a method claim whose conditional limitations need not be satisfied to meet the claim.

error in the Examiner's reliance on the functionality in Butin's paragraphs 148, 152 to 154, and 179 for at least suggesting using a supplemental graphical element detection technique when the additional technique, namely OCR, does not find a match. *See* Final Act. 17–19; Ans. 44–46.

As Butin's paragraph 179 explains, some operations can be *repeated* until one or more conditions are met—operations that can be performed sequentially. Considering these sequential operations in light of Butin's teaching in paragraph 153 that one *or more* OCR or image recognition algorithms may be used *in addition* to those described, using a supplemental graphical element detection technique, such as one that employs a *different OCR or image recognition algorithm*, to find a match when previous techniques were unsuccessful, would have been at least an obvious variation. That Butin's paragraph 152 notes that OCR can be performed on a *batch-by-batch basis* further underscores the system's use of sequential graphical element detection techniques to find a match. Appellant's arguments to the contrary (Appeal Br. 40–41; Reply Br. 15–16) are unavailing and not commensurate with the scope of the claim.

In short, we find—as does the Examiner—that Butin alone at least suggests every recited limitation, including the second time period limitation. *See* Final Act. 17–19; Ans. 44–46. As the Examiner explains, because *some* period of time must elapse before determining whether OCR was unsuccessful, Butin at least suggests the recited second time period. *See* Final Act. 19.

We, therefore, see no harmful error in the Examiner's rejection despite Colciago being cumulative to Butin. Although Butin may not state

explicitly that the additional graphical element detection technique is used during a second time period, Butin nonetheless suggests as much.

Accordingly, we sustain the Examiner's obviousness rejection based solely on Butin, for we may rely on fewer references than the Examiner in affirming a multiple-reference obviousness rejection where, as here, we rely on the same teachings relied upon by the Examiner. *See Kronig*, 539 F.2d at 1302; *see also Bush*, 296 F.2d at 496; *accord* MPEP § 1207.03(a)(II)(3) (discussing *Kronig* and *Bush*).

Appellant's arguments, then, regarding Colciago's alleged shortcomings, including the reference ostensibly constituting non-analogous art, are moot given our sole reliance on Butin.

Therefore, we are not persuaded that the Examiner erred in rejecting claim 2 and claims 9 and 15 not argued separately with particularity.[8]

CONCLUSION

In summary:

Claims Rejected	35 U.S.C. §	Reference(s)/ Basis	Affirmed	Reversed
1–20	103	Butin, Colciago	1–20	
8, 9		Provisional Nonstatutory Double Patenting, Application No. 17/746,582	8, 9	
Overall Outcome			1–20	

[8] Although claims 9 and 15 are nominally argued separately (*see* Appeal Br. 60–61, 80–81; Reply Br. 16), the arguments are similar to those made for claim 2. We, therefore, group these claims together.

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv). *See* 37 C.F.R. § 41.50(f).

AFFIRMED