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.
14/478,420	09/05/2014	David John Babcock	84030806	5021

146568 7590 06/09/2020

MICRO FOCUS LLC
500 Westover Drive
#12603
Sanford, NC 27330

EXAMINER	
LEE, JENNIFER V	

ART UNIT	PAPER NUMBER
3625	

NOTIFICATION DATE	DELIVERY MODE
06/09/2020	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):

software.ip.mail@microfocus.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte DAVID JOHN BABCOCK,
SPENCER FIRESTONE, and JAGADEESH KANDASAMY

Appeal 2020-001281
Application 14/478,420
Technology Center 3600

BEFORE DONALD E. ADAMS, JEFFREY N. FREDMAN, and RACHEL H. TOWNSEND, *Administrative Patent Judges*.

ADAMS, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. § 134(a), Appellant[1] appeals from Examiner's decision to reject claims 1–5 and 7–21 (*see* Final Act.[2] 2). We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

[1] We use the word "Appellant" to refer to "applicant" as defined in 37 C.F.R. § 1.42. Appellant identifies the real party in interest as "EntIT Software LLC . . . a wholly-owned affiliate of Micro Focus International Plc" (Appellant's June 24, 2019 Appeal Brief (Appeal Br.) 1).

[2] Examiner's February 8, 2019 Final Office Action.

STATEMENT OF THE CASE

Appellant defines the term “demonstrate” as including “demonstrating, training, supporting, helping, and otherwise providing assistance according to the examples provided” in Appellant’s Specification (Spec. ¶ 8). In this regard, Appellant discloses that “[d]emonstrations and training for products, such as computer programs, can be challenging and complex. Varied demonstration environments, and the need to maintain demonstration environments in a ‘customer readiness’ state, make it difficult to demonstrate relevant features in an easily understood manner that allows hands-on interaction” (Spec.[3] ¶ 1). Appellant appreciates that “[p]roducts are available to help create product demonstrations,” but contends that “they . . . offer features for supporting static demonstrations replayed in various formats,” “may be limited to web-based products, and/or may be limited to demonstrating a single product” (*id.* ¶ 8). Appellant overcomes the limitations of prior products by providing

> guided demonstrations by engines that provide for interaction with one or more live products[4] to be demonstrated, creating a dynamic and flexible experience that is not limited to a mere viewing of a recorded replay, for a more gratifying, interactive, and educational demonstration experience based on content, scripts, and/or other materials. . . . For example, demonstrating a product may enable fully interactive participation by users, who may view and actively use the product as it is being demonstrated (e.g., by demonstrating it to him- or herself). In an example, a demonstration may be used as a “walk-through” or support tool to guide a user through installation of a product,

[3] Appellant’s September 5, 2014 Specification.

[4] Appellant discloses that “a computer application” is an example of a live product (Spec. ¶ 17).

> or through a support scenario to address a help call of a live product being used by the user.

(*Id.*)

Appellant discloses that its

> demonstration may be provided as a self-contained application, enabling focus to remain on demonstrating technical features/capabilities of the product to be demonstrated, without becoming distracted by a need to prepare the demonstration environment (computer system) itself. The demonstration may serve as a wrapper around one or more live products to be demonstrated, and guide the demonstration through the usage of those actual live products. Thus, examples described herein enable a customizable, multi-purpose, cross-platform framework for building guided live demonstrations. In an example, a demonstrator may look over a user's shoulder (literally and/or virtually) and point out features of a program. Examples provide unobstructed visual cues, automated and/or manual script triggers, and seamless process and/or window management with live products. Custom images and/or scripts may be incorporated easily, to augment the demonstration. Demonstrations may be packaged in a virtual machine (VM) package for ease of distribution, isolation from potential demonstration environment issues/conflicts, and simple restoration of modified data.

(*Id.* ¶ 9.)

Appellant's claims 1 and 12 are reproduced below:[5]

> 1. A computing system comprising:
>
> a processor;

[5] Examiner entered Appellant's September 20, 2018 claim amendments (*see* Final Act. 2). The Claims Appendix of Appellant's Brief reproduces Appellant's claims as amended September 20, 2018. Our review of this record is in the context of Appellant's claims as amended September 20, 2018.

> *a non-transitory storage medium storing framework instructions executable on the processor to provide a framework that demonstrates a live product that is executable in a machine environment independently from the framework instructions*, wherein the framework includes a first graphical user interface (GUI) panel comprising a plurality of graphical elements representing respective steps of a demonstration of the live product, and *a second GUI panel to display content and at least a portion of the live product according to the steps*, the first GUI panel presenting the steps in a first order, wherein the steps are customizable for different audiences of the demonstration of the live product, the graphical elements in the first GUI panel selectable to navigate through the demonstration, *wherein the steps are performable responsive to selections of the graphical elements in an order different from the first order in which the steps are presented by the first GUI panel*; and
>
> a script executable in response to user selection of a graphical element representing a first step of the steps in the first GUI panel, where execution of the script causes manipulation of a feature of the live product.

(Appeal Br. i (emphasis added).)

> 12. A method, comprising:
>
> demonstrating, by framework instructions executed by a computer, a live product that is installed in a machine environment independently of the framework instructions, the demonstrating displaying a framework including a first panel comprising a plurality of graphical elements representing respective steps of a demonstration of the live product, and a second panel to display content and at least a portion of the live product according to the steps, the first panel presenting the steps in a first order, wherein the graphical elements in the first panel are selectable to navigate through the demonstration; and
>
> executing, by script instructions executed by the computer, a script in response to user selection of a graphical element representing a first step of the steps, wherein the steps are performable responsive to selections of the graphical

> elements in an order different from the first order of the steps presented by the first panel.

(*Id.* at iii.)

> 14. A non-transitory machine-readable storage media storing framework instructions executable by a computing system that, when executed, cause the computing system to:
>
> demonstrate a live product executable in a machine environment, the demonstrating based on a framework including a first panel comprising a plurality of graphical elements representing respective steps of a demonstration of the live product, and a second panel to display content and at least a portion of the live product according to the steps, the first panel presenting the steps in a first order, wherein the graphical elements in the first panel are selectable to navigate through the demonstration, wherein the live product is independent of the framework instructions; and
>
> execute a script in response to user selection of a graphical element representing a first step of the steps in the first panel, where execution of the script causes manipulation of a feature of the live product, wherein the steps are performable responsive to selections of the graphical elements in an order independent of the first order in which the steps are presented by the first panel.

(*Id.* at iii.)

Grounds of rejection before this Panel for review:[6]

Claims 1–5 and 7–21 stand rejected under 35 U.S.C. § 112, second paragraph.

[6] The statement of each of Examiner's rejections do not include Appellant's claim 21 (*see* Final Act. 9 and 15). We find the omission of Appellant's claim 21 from each ground of rejection to represent a harmless typographical error. Therefore, we include Appellant's claim 21 in our deliberations of each ground of rejection.

Claims 1–5 and 7–21 stand rejected under 35 U.S.C. § 102(a)(1)/(a)(2) as anticipated by Nicol.[7]

DEFINITENESS:

ISSUE

Does the preponderance of evidence support Examiner's conclusion that Appellant's claims are indefinite?

ANALYSIS

Claim 1:

The computing system of Appellant's claim 1 comprises, *inter alia*, "a non-transitory storage medium storing framework instructions executable on the processor to provide a framework that demonstrates a live product that is executable in a machine environment independently from the framework instructions" (Appeal Br. i). Examiner finds this phrase indefinite, because "it is unclear whether it is the *live product* or the *demonstration* of the live product that is executable independently" (Final Act. 9). Examiner further finds that "it is unclear[, from the foregoing phrase,] whether the live product, if executable independently from the framework instructions of the system, is . . . a component of the system or outside the system" (*id.*). Thus, for examination purposes, Examiner interpreted the foregoing phrase as "a non-transitory storage medium storing framework instructions executable on the processor to provide a framework to demonstrate a live product" (*id.*). We are not persuaded.

[7] Nicol et al., US 2007/0234196 A1, published Oct. 4, 2007.

"The definiteness inquiry focuses on whether those skilled in the art would understand the scope of the claim when the claim is read in light of the rest of the specification." *Union Pacific Resources Co. v. Chesapeake Energy Corp.*, 236 F.3d 684, 692 (Fed. Cir. 2001). As Appellant explains, on this record, when its claims are interpreted in light of Appellant's Specification, those of ordinary skill in this art would understand that it is "the live product that is executable independently from the framework instructions," i.e., "in a machine environment independently from the framework instructions," as required by Appellant's claimed invention and disclosed in Appellant's Specification (Appeal Br. 7 (citing Spec. ¶¶ 18 ("Systems based on the framework . . . may take advantage of standard, live products . . . installed on a given machine environment, that may be installed and/or launched separately from the framework system.") and 24 ("If the live product . . . is not present (e.g., not installed on the current computing environment), the system may display a message informing the user to install and/or execute the live product . . ., and/or may automatically install the live product."))).

The computing system of Appellant's claim 1 further requires "the framework includes a first graphical user interface (GUI) panel comprising a plurality of graphical elements representing respective steps of a demonstration of the live product, and a second GUI panel to display content and at least a portion of the live product according to the steps" (Appeal Br. i). Based on Examiner's interpretation of Appellant's claimed computing system as comprising "a non-transitory storage medium storing framework instructions executable on the processor to provide a framework to demonstrate a live product," Examiner further interprets, for examination purposes, Appellant's claim as comprising a framework that includes, *inter alia*, "a second GUI panel to display content and at least a portion of the at least one live product according to the steps" (Final Act. 9). We are not persuaded. As Appellant explains, Examiner's interpretation of the second GUI panel is incorrect because it is based on an incorrect interpretation of the phrase "a non-transitory storage medium storing framework instructions executable on the processor to provide a framework that demonstrates a live product that is executable in a machine environment independently from the framework instructions" (Appeal Br. 9; *see id.* at i).

Appellant's claim 1 further requires that "the steps are performable responsive to selections of the graphical elements in an order different from the first order in which the steps are presented by the first GUI panel" (Appeal Br. i). According to Examiner "the *steps* are not presented by the framework but rather the *graphical elements* representing the steps are the components included in the framework provided" (Final Act. 9). Thus, Examiner finds that "it is unclear [from this requirement] how the steps are performable in an order different from an order in which the steps are *presented* by the framework when the steps are never presented" (*id.* at 9–10). We are not persuaded.

As Appellant explains:

> Claim 1 specifically recites that the framework instructions are executable on the processor to provide a framework that includes a first GUI comprising a plurality of graphical elements representing respective steps of a demonstration of the live product. Thus, the foregoing language of claim 1 clearly establishes that the steps are in fact presented by the plurality of graphical elements in the first GUI panel of the framework that

> is provided by the framework instructions executable on the processor.

(Reply Br. 2.) In addition, Appellant explains, that

> Claim 1 further recites that the steps (that are presented by the plurality of graphical elements in the first GUI panel of the framework provided by the framework instructions) are performable responsive to selections of the graphical elements in an order different from the first order in which the steps are presented by the first GUI panel. What this language recites is that the graphical elements that are part of the first GUI panel of the framework provided by the framework instructions can be selected to cause performance of steps, and the order of the selections of the graphical elements can cause the steps to be performed in an order different from the first order in which the steps are presented by the first GUI panel.

(*Id.* at 3.) Thus, as Appellant explains, when interpreted in light of its Specification:

> A person of ordinary skill in the art would understand that graphical elements provide a visual representation of the steps. Therefore, since the graphical elements of claim 1 are part of the first GUI panel, the first GUI panel would in fact present the steps in the form of the graphical elements. Therefore, the language "the steps are presented by the first GUI panel" is not indefinite.

(Appeal Br. 10; *see generally* Reply Br. 3.)

Claims 2–5, 7–11, and 16–18:

Examiner finds that because Appellant's claims 2–5, 7–11, and 16–18 depend from Appellant's claim 1 they are indefinite for the same reasons as Appellant's claim 1 discussed above (Final Act. 10). Having found Appellant's claim 1 definite, we are not persuaded by Examiner's indefiniteness rejection of Appellant's claims 2–5, 7–11, and 16–18.

Claim 5:

Appellant's claim 5 depends from and further limits the computing system of Appellant's claim 1 to require that "the framework instructions are responsive to the script to perform a task to change the content displayed in the second GUI panel" (Appeal Br. ii).

Examiner reasons that, "in claim 1, in order for the script to be executed the framework instructions must have been executed in order to provide the framework including the first GUI panel with its graphical elements, as it is one of those graphical elements that must be selected to execute the script" (Final Act. 10). "Thus, [Examiner reasons,] it is unclear how the framework instructions are responsive to the script in claim 5, when in claim 1, the script is responsive to the framework instructions" (*id.* at 10–11). Stated differently, Examiner finds that it is unclear from Appellant's claim 5:

> [W]hether the Applicant intended for 1) the framework instructions, which are responsive to the script, to perform a task to change the content displayed in the second GUI panel, or intended that 2) the framework instructions are responsive to the script, the script to perform a task to change the content displayed in the second GUI panel, or had another intended meaning.

(*Id.* at 10.) Therefore, for examination purposes, Examiner interpreted Appellant's claim 5 as "changing the content displayed in the second GUI panel" (*id.* at 11). We are not persuaded.

As Appellant explains, "the term 'framework instructions' refer to 'instructions' in the plural sense. A person of ordinary skill in the art would clearly understand that framework instructions can include multiple instructions that perform respective different tasks" (Reply Br. 5). Thus, Appellant explains:

> The action of the framework instructions performed in claim 5, task . . ., is separate from the actions of the framework instructions performed in base claim 1 . . . . There is no requirement that framework instructions can perform just the actions recited in the claim 1. It is possible for the framework instructions to first provide the framework that includes the first GUI panel comprising graphical elements representing respective steps, then the script is executable in response to user selection of a graphical element representing a first step in the first GUI panel, and then the framework instructions are responsive to the script to perform another task, such as the task to change the content displayed in the second GUI panel recited in claim 5.

(*Id.* at 4; *see also id.* at 5 (Appellant contends that "[t]he framework instructions recited in claim 5 are capable of performing multiple different tasks in response to different events or inputs, and claim 5 has clearly set forth what these tasks are").) For the foregoing reasons, we agree with Appellant's contention that "[f]rom the context of claim 5, it is clear that it is the framework instructions that are to perform a task to change the content displayed in the second GUI panel" (Appeal Br. 11).

Claim 7:

Appellant's claim 7 depends from and further limits the computing system of Appellant's claim 1 to require that "responsive to the execution of the script, the framework instructions are to perform a task to automatically test for a presence of a window of the live product, and if the window of the live product is not present, cause the live product to launch the window" (Appeal Br. ii).

Examiner reasons that, "in claim 1, in order for the script to be executed the framework instructions must have been executed in order to provide the framework including the first GUI panel with its graphical elements, as it is one of those graphical elements that must be selected to execute the script (Final Act. 11). "Thus, [Examiner reasons,] it is unclear how the framework instructions are responsive to the execution of the script in claim 7, when in claim 1, the script is responsive to the framework instructions" (*id.*). Stated differently, Examiner finds that it is unclear from Appellant's claim 7:

> [W]hether the Applicant intended for 1) the framework instructions, which are responsive to the execution of the script, to perform a task to automatically test for a presence of a window of the live product, or 2) the framework instructions are responsive to the execution of the script, the script to perform a task to automatically test for a presence of a window of the live product, or had another intended meaning.

(*Id.* at 11.) Examiner's further finding that Appellant's claim 7 requires that "the framework instructions play[] a role in causing the live product to launch the window" and, [c]onsequently, it is unclear how the live product – whose window, if not previously presented, is subsequently launched by the framework instructions – can be 'executable independently from the framework instructions . . .' as recited in [Appellant's] claim 1" (*id.* at 11–12 (emphasis omitted)). Therefore, for examination purposes, Examiner interpreted Appellant's claim 7 as "displaying a window of the live product" (*id.* at 12). We are not persuaded.

As Appellant explains:

> Claim 7 recites that the framework instructions are to perform a task to automatically test for presence of a window of the live product, and if the window of the live product is not present, to

> cause the live product to launch the window. This merely refers to an interaction between the framework instructions and the live product. However, that does not preclude the ability for the live product to be executable independently from the framework instructions. Note that outside of the interaction between the framework instructions and the live product recited in claim 7, the live product and the framework instructions can execute independently of one another.

(Appeal Br. 14.) In addition, claim 7 requires that "[t]he foregoing action of the framework instructions are 'responsive to the execution of the script'" (*id.* at 13). For the foregoing reasons, we agree with Appellant that Examiner's interpretation of claim 7 "'as merely displaying a window of the live product' . . . ignores [the] express words of the claim, and is clearly in error" (*id.* at 14).

Claim 8:

Appellant's claim 8 depends from and further limits the computing system of Appellant's claim 1 to require that "responsive to the execution of the script, the framework instructions are to perform a task to automatically test for an attribute of a window of the live product, and if the attribute is not consistent with a second step of the steps, the framework instructions are to manipulate the window of the live product to be consistent with the second step" (Appeal Br. ii).

Examiner reasons that, "in claim 1, in order for the script to be executed the framework instructions must have been executed in order to provide the framework including the first GUI panel with its graphical elements, as it is one of those graphical elements that must be selected to execute the script" (Final Act. 12). "Thus, [Examiner reasons,] it is unclear how the framework instructions are responsive to the script in claim 8, when in claim 1, the script is responsive to the framework instructions" (*id.* at 12–13). Stated differently, Examiner finds that it is unclear from Appellant's claim 8:

> [W]hether the Applicant intended for 1) the framework instructions, which are responsive to the script, to perform a task to automatically test for an attribute of a window of the live product, or 2) the framework instructions are responsive to the script, the script to perform a task to automatically test for an attribute of a window of the live product, or had another intended meaning.

(*id.* at 12.) Therefore, for examination purposes, Examiner interpreted Appellant's claim 8 as "manipulating the window of the live product" (*id.* at 13). We are not persuaded.

As Appellant explains, its "claim 8 expressly recites 'the framework instructions are to perform a task to automatically test for an attribute of a window of the live product.' Therefore, there is no ambiguity. The framework instructions perform this task responsive to the execution of the script" (Appeal Br. 15). For the foregoing reasons, we agree with Appellant that Examiner's interpretation of claim 8 "'as merely manipulating the window of the live product' . . . ignores [the] express words of claim 8, and is clearly erroneous" (*id.*).

Claim 10:

Appellant's claim 10 depends from and further limits the computing system of Appellant's claim 1 to require that "the framework instructions are executable to display the first GUI panel on a first display screen, and the second GUI panel on a second GUI display screen separate from the first display screen" (Appeal Br. ii). Examiner finds that Appellant's "[c]laim 10 is directed to a system and recites ‘a first display screen’ and ‘a second GUI display screen’” and “[a]s recited, it is unclear whether the screens are intended to be components of the recited system” (Appeal Br. 13 (emphasis omitted)). We are not persuaded.

As Appellant explains:

> The first and second display screens can either be part of the computing system or can be separate from the computing system. The framework instructions can present the first and second GUI panels either on display screens that are part of the computing system, or on display screens that are remote from the computing system but which are connected by a network to the computing system. There is no requirement that Applicant must limit claim 10 to one of these different possible embodiments to satisfy § l 12(b).

(Appeal Br. 16–17.)

Claim 11:

Appellant’s claim 11 depends from and further limits the computing system of Appellant’s claim 1 to require that “the framework instructions are executable to display a third GUI panel associated with a feature of the demonstration, wherein the framework instructions are executable to highlight the feature in response to a step corresponding to the feature being active” (Appeal Br. ii). Examiner finds that Appellant’s claim 11 “recites ‘wherein the framework instructions are executable to highlight the feature in response to a step corresponding to the feature being active’” and “[a]s recited, it is unclear whether the Applicant intended for 1) a step, corresponding to the feature, being active (i.e. the step to be active) or 2) the step corresponding to the feature, the feature being active (i.e. the feature to be active) or had some other intended meaning” (Final Act. 13 (emphasis omitted)). We are not persuaded. As Appellant explains, its "[c]laim 11 is unambiguous in stating that it is the step being active, where the step corresponds to the feature. Therefore, claim 11 is not indefinite" (Appeal Br. 17).

Claims 12 and 15:

Examiner asserts that Appellant's "[c]laims 12 and 15 are rejected under substantially similar grounds as claims 1 and 11 [above]" (Final Act. 13 (emphasis omitted)). Having found Appellant's claims 1 and 11 definite, we are not persuaded by Examiner's indefiniteness rejection of Appellant's claims 12 and 15 (*see generally* Appeal Br. 17).

Claim 19:

Examiner finds that because Appellant's claim 19 depends from Appellant's claim 12 it is indefinite for the same reasons as Appellant's claim 12 discussed above (Final Act. 13). Having found Appellant's claim 12 definite, we are not persuaded by Examiner's indefiniteness rejection of Appellant's claim 19 (*see generally* Appeal Br. 17).

Claims 20 and 21:[8]

Examiner finds that because Appellant's claims 20 and 21 depend from Appellant's claim 14 it is indefinite for the same reasons as Appellant's claim 14 discussed above (Final Act. 13). Having found Appellant's claim

[8] As discussed above, although Examiner does not address Appellant's claim 21, we find the omission of Appellant's claim 21 to represent a typographical error and include it in our deliberation of this rejection.

14 definite, we are not persuaded by Examiner's indefiniteness rejection of Appellant's claims 20 and 21 (*see generally* Appeal Br. 17).

CONCLUSION

The preponderance of evidence fails to support Examiner's conclusion that Appellant's claims are indefinite. The rejection of claims 1–5 and 7–21 under 35 U.S.C. § 112, second paragraph, is reversed.

ANTICIPATION:

ISSUE

Does the preponderance of evidence on this record support Examiner's finding that Nicol teaches Appellant's claimed invention?

ANALYSIS

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987). Analysis of whether a claim is patentable over the prior art under 35 U.S.C. § 102 begins with a determination of the scope of the claim. We determine the scope of the claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004). The properly interpreted claim must then be compared with the prior art.

On this record, Examiner explains that the anticipation rejection is based on Examiner's interpretation of Appellant's claimed invention discussed above with respect to the definiteness rejection (*see* Ans. 10–11).

As discussed above, however, Examiner incorrectly interpreted Appellant's claimed invention. Thus, Examiner's anticipation rejection is based on an incorrect interpretation of Appellant's claimed invention.

As Appellant explains, when its claims are correctly interpreted, Examiner failed to establish an evidentiary basis on this record to support a finding that Nicol anticipates Appellant's claimed invention (*see* Reply Br. 5–6; *see also* Appeal Br. 18–24 (Appellant explains specifically why Nicol fails to anticipate its claims)). We agree with Appellant.

CONCLUSION

The preponderance of evidence on this record fails to support Examiner's finding that Nicol teaches Appellant's claimed invention. The rejection of claims 1–5 and 7–21 under 35 U.S.C. § 102(a)(1)/(a)(2) as being anticipated by Nicol is reversed.

DECISION SUMMARY

In summary:

Claims Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1–5, 7–21	112(b)	Indefiniteness		1–5, 7–21
1–5, 7–21	102(a)(1)/(a)(2)	Nicol		1–5, 7–21
Overall Outcome				1–5, 7–21

<u>REVERSED</u>