



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.
14/060,091	10/22/2013	Alan F. Babich	RSW920120144US2	1333
75532	7590	06/22/2018	EXAMINER	
LEE LAW, PLLC IBM SVL IP P.O. BOX 189 PITTSBORO, NC 27312			MCATEE, PATRICK	
			ART UNIT	PAPER NUMBER
			3689	
			NOTIFICATION DATE	DELIVERY MODE
			06/22/2018	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):

docteting@leelawpllc.com
docteting_archive@leelawpllc.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte ALAN F. BABICH, XIN FENG, MIKE A. MARIN,
CYNTHIA M. ROBERTS, MIMI P. VO, and ALAN T. YAUNG

Appeal 2016-006174
Application 14/060,091¹
Technology Center 3600

Before ANTON W. FETTING, MICHAEL C. ASTORINO, and
BRUCE T. WIEDER, *Administrative Patent Judges*.

WIEDER, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the Examiner's rejection of claims 1–24. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

¹ According to Appellants, the real party in interest is International Business Machines Corporation. (Appeal Br. 3.)

CLAIMED SUBJECT MATTER

Appellants’ “invention relates to case management. More particularly, the present invention relates to case management integration with external content repositories.” (Spec. ¶ 2.)

Claims 1, 9, and 17 are the independent claims on appeal. Claim 1 is illustrative. It recites:

1. A method, comprising:
 - detecting, by a programmed processor executing an inbound event handler of a distributed integration framework that integrates a case management system with an external content repository, a case creation event originating from the external content repository that invokes the inbound event handler of the distributed integration framework to cause creation of a case within the case management system;
 - instructing, via the inbound event handler, the case management system to create the case within the case management system; and
 - instructing, via an outbound event handler of the distributed integration framework, the external content repository to create a proxy case to remotely represent the case created within the case management system.

REJECTIONS

Claims 1–24 are provisionally rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1, 2, 9, 10, 17, and 18 of copending Application No. 13/655,004 (hereinafter “the ’004 application” or “the Copending Application”).

Claims 1–24 are rejected under 35 U.S.C. § 112(a) as failing to comply with the written description requirement.

Claims 1–24 are rejected under 35 U.S.C. § 112(b) as being indefinite for failing to point out and distinctly claim the subject matter that Appellants regard as the invention.

Claims 1–24 are rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter.

Claims 1, 2, 9, 10, 17, and 18 are rejected under 35 U.S.C. § 102(b) as anticipated by Johnson (US 2006/0085412 A1, pub. Apr. 20, 2006).

Claims 3–8, 11–16, and 19–24 are rejected under 35 U.S.C. § 103(a) as unpatentable in view of Johnson, Kandasamy (US 5,513,314, iss. Apr. 30, 1996) and Memorandum from the Network Working Group, NFS Version 3 Protocol Specification (June 1995) (hereinafter “Sun Microsystems”).

ANALYSIS

The provisional double patenting rejections of claims 1–24

Claims 1–6, 8–14, 16–22, and 24 are provisionally rejected on the ground of nonstatutory double patenting in view of the '004 application. (Answer 109–10.) In particular, independent claims 1, 9, and 17 of the present application are rejected in view of claims 1, 9, and 17 of the '004 application. (Non-Final Action 4–5.)

Appellants do not separately argue claims 1–6, 8–14, 16–22, and 24. Therefore, these claims will stand or fall together. *See* 37 C.F.R. § 41.37(c)(1)(iv).

Appellants argue that “there is no claimed integration of a case management system with any other component recited in the independent claims of the Copending Application. In contrast, the processing claimed

within the Copending Application is limited to content management systems” (Appeal Br. 32.)

Appellants have not persuasively argued that the “content management system” recited in the Copending Application is “patentably distinguishable from any generic computing device,” i.e., the case management system of the present application. (*See Answer 111.*) In particular, Appellants do not persuasively argue why claim 1 of the Copending Application, that recites “a first content management system to store proxy objects for root case instance folders” and “a second content management system to store real objects for the root case instance folders,” i.e., for the same folders, is not an integration of the first management system with the second management system.

Appellants also argue that the claims in the present application instruct

the respective components to each “create” the respective case and proxy case. In contrast, the independent claims of the Copending Application recite only “modify” actions - not create actions The modify actions claimed in the Copending Application are fundamentally different from the claimed instructions that cause creation of cases and proxy cases in the present application.

(Appeal Br. 33.)

We do not find this argument persuasive. The Copending Application broadly defines the term modify, i.e., “modify (e.g., create, update, delete etc.).” (’004 application Spec. ¶ 58.) Indeed, claim 1 of the ’004 application recites “a first operation to modify” and claim 2 recites “wherein the first operation [to modify] is one of creating, updating, deleting, filing, unfiling, and querying.”

Appellant also argues that “there is no claimed distributed integration framework that integrates a case management system with an external content repository, as is claimed to be performed within the present application, found to be recited within the claims of the Copending Application.” (Appeal Br. 33–34.)

The Examiner finds, and we agree, that

the “distributed integration framework” is merely a logical software interface, and a person having ordinary skill in the art would have understood that an equivalent software interface is present in the claims of [the ’004 application] as it would be required for a processor of a computer to perform the functions recited in [the ’004 application]. Again, the exact title given to the software does not patentably distinguish the present invention.

(Answer 112.)

Appellants other arguments were considered but were not found persuasive of error.

In view of the above, we are not persuaded that the Examiner erred in provisionally rejecting claims 1–6, 8–14, 16–22, and 24 for double patenting.

Dependent claims 7, 15, and 23 are provisionally rejected for obviousness-type double patenting in view of Kandasamy and Sun Microsystems. (Non-Final Action 8–11.) However, Appellants do not provide separate arguments regarding these claims. (*See* Appeal Br. 35.) Therefore, for the reasons discussed above, we are not persuaded that the Examiner erred in provisionally rejecting dependent claims 7, 15, and 23 for obviousness-type double patenting.

The § 112(a) rejection of claims 1–24

Claim 1 recites “detecting, by a . . . processor . . . , a case creation event originating from the external content repository that *invokes* the inbound event handler . . . to cause creation of a case within the case management system.” (Emphasis added.) Claims 9 and 17 contain similar language.

“[T]he Examiner finds that the specification describes event handler invoking . . . API calls [0055, 0063], and that events themselves may be invoked [0079], but does not find support for invoking an event handler (either inbound or outbound).” (Non-Final Action 12.) Specifically, the Examiner finds that “every use of the term ‘invoke’ in Appellant’s [sic] specification refers to an action **by** an event handler **on** an event or API rather than an action **on** an event handler.” (Answer 115.) Therefore, the Examiner finds, “one of skill in the art would have understood from the original disclosure that ‘invoke’ refers to action **by** the event handler to cause an event or API call, not that ‘invoke’ refers to an action **on** the event handler as ‘invoke’ is used in the claim amendment.” (*Id.* at 116.)

Appellants disagree and cite paragraph 79 of the Specification. (Appeal Br. 39–40.) Paragraph 79 discloses that “process 700 creates a document in a content repository at block 712. . . . At block 714, the process 700 *triggers* the distributed inbound event handler to notify the case manager to create a case folder and to create a proxy document.” (Spec. ¶ 79, emphasis added.) Appellants argue that, in view of the teachings in the Specification, “[a] person of ordinary skill . . . would recognize that Appellant was in full possession of the subject matter as claimed.” (Appeal Br. 40.)

If . . . the specification contains a description of the claimed invention, albeit not in *ipsis verbis* (in the identical words), then the examiner or Board, in order to meet the burden of proof, must provide reasons why one of ordinary skill in the art would not consider the description sufficient. Once the examiner or Board carries the burden of making out a prima facie case of unpatentability, “the burden of coming forward with evidence or argument shifts to the applicant.” To overcome a prima facie case, an applicant must show that the invention as claimed is adequately described to one skilled in the art. “After evidence or argument is submitted by the applicant in response, patentability is determined on the totality of the record, by a preponderance of the evidence with due consideration to persuasiveness of argument.”

In re Alton, 76 F.3d 1168, 1175 (Fed. Cir. 1996) (citations omitted).

“[T]he applicant must . . . convey with reasonable clarity to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention.” *Vas-Cath, Inc. v. Mahurkar*, 935 F.2d 1555, 1563–64 (Fed. Cir. 1991). Here, the question is whether the disclosure of “trigger[ing] the distributed inbound event handler” would convey to one skilled in the art possession of the invention as claimed, i.e., “invokes the inbound event handler.”

The Examiner recognizes that “invoke” refers to an action that causes an event. (*See* Non-Final Action 116.) This is in accord with the dictionary definition. The Cambridge Dictionary defines “invoke” as “to cause something to be used; bring into effect.” (<https://dictionary.cambridge.org/us/dictionary/english/invoke>, last visited June 19, 2018.) The Cambridge Dictionary defines “trigger,” the term used in paragraph 79, as “to cause something to start.” (<https://dictionary.cambridge.org/us/dictionary/english/trigger>, last visited June 19, 2018.)

In this context, we agree with Appellants that one of ordinary skill in the art would understand from the Specification, including at least paragraph 79, that Appellants were in possession of the claimed subject matter at issue.

The § 112(b) rejections of claims 1–24

Claim 1 recites, in relevant part,

detecting, by a programmed processor executing an inbound event handler of a distributed integration framework that integrates a case management system with an external content repository, a case creation event originating from the external content repository that invokes the inbound event handler of the distributed integration framework to cause creation of a case within the case management system.

Independent claims 9 and 17 contain similar language. The Examiner concludes that

[t]he limitation is indefinite because it is not clear whether [1] the distributed integration framework, [2] the external content repository, or [3] the case management system “invokes” the inbound event handler, and subsequently it is not clear whether the distributed integration framework, the external content repository, the case management system, or the inbound event handler “causes” creation of a case within the case management system.

(Non-Final Action 13–14.)

Appellants disagree and argue “that the detected case creation event invokes (causes/triggers) the inbound event handler,” i.e., [4] the detected *case creation event invokes* the inbound event handler. (Appeal Br. 42.)

Appellants also argue that “it is the detection of the case creation event that

invokes (causes/triggers) the inbound event handler,” i.e., [5] *the detection* of the case creation event *invokes* the inbound handler. (*Id.*)

The Federal Circuit has stated that

when the USPTO has initially issued a well-grounded rejection that identifies ways in which language in a claim is ambiguous, vague, incoherent, opaque, or otherwise unclear in describing and defining the claimed invention, and thereafter the applicant fails to provide a satisfactory response, the USPTO can properly reject the claim as failing to meet the statutory requirements of § 112(b).

In re Packard, 751 F.3d 1307, 1311 (Fed. Cir. 2014). In this case, the Examiner presents a well-grounded rejection identifying why it is unclear what “invokes the inbound event handler.” Appellants’ arguments do not overcome the rejection and, indeed, demonstrate that the claims are unclear.

Claims 9–16 are further rejected under § 112(b). Claim 9 recites:

9. A system, comprising:
 - a distributed integration framework that integrates a case management system with an external content repository; and
 - a processor programmed to:
 - detect, at an inbound event handler of the distributed integration framework, a case creation event originating from the external content repository that invokes the inbound event handler of the distributed integration framework to cause creation of a case within the case management system;
 - instruct, via the inbound event handler, the case management system to create the case within the case management system; and
 - instruct, via an outbound event handler, the external content repository to create a proxy case to remotely represent the case created within the case management system.

The Examiner concludes that claims 9–16 are indefinite because the limitation *a processor* is singular, meaning there is a single processor or a single computing device performing the actions. However, the claims recite actions performed by event handlers of a distributed integration framework, a case management system, and an external content repository, and it is not clear whether these are parts of a single computing device (such as computer with an extra hard drive content repository and a framework to represent content between the main hard drive and the extra hard drive) or whether they are separate computing devices.

(Non-Final Action 14.)

Appellants disagree and argue:

Appellant [sic] does not claim a single processor carrying out operations on different devices. In contrast, Appellant claims a programmed processor executing within a distributed integration framework that integrates a case management system with an external content repository. This language is also clear if properly considered.

The claimed programmed processor executes the claimed inbound event handler. As clearly taught and claimed by Appellant, the inbound event handler forms a portion of the distributed integration framework that integrates a case management system with an external content repository. . . .

The claims recite the inbound event handler (executed by the programmed processor) to be “detecting” and “instructing.” This is clear within the claim language itself.

(Appeal Br. 43–44.)

Appellants’ Specification discloses:

Figure 6 is a flow chart of an example of an implementation of a process 600 for case management integration with external content repositories. At block 602, the process 600 detects, via a processor executing an inbound event handler of a distributed integration framework that integrates a case management system with an external content repository, a case creation event

originating from the external content repository that is configured via the distributed integration framework to cause creation of a case within the case management system. At block 604, the process 600 instructs, via the inbound event handler, the case management system to create the case within the case management system. At block 606, the process 600 instructs, via an outbound event handler of the distributed integration framework, the external content repository to create a proxy case to remotely represent the case created within the case management system.

(Spec. ¶ 75.)

Thus, paragraph 75 of Appellants' Specification supports Appellants' argument that a programmed processor executes the inbound event handler and, therefore, is part of an integrated process. But this only adds to the lack of clarity. Claim 9 recites "a processor programmed to: detect, at an inbound event handler," "instruct, via the inbound event handler," and "instruct, via an outbound case handler." Appellants appear to read into the claim a requirement that the processor itself executes the inbound event handler. We find no such requirement in the language of claim 9. Therefore, we do not find Appellants' argument persuasive of error. Moreover, even if we were to read claim 9 as proposed by Appellants, it would not address the Examiner's concern regarding execution of the outbound event handler.

In short, the Examiner has "issued a well-grounded rejection that identifies ways in which language in a claim is ambiguous, vague, incoherent, opaque, or otherwise unclear in describing and defining the claimed invention, and thereafter the applicant fails to provide a satisfactory response." *In re Packard*, 751 F.3d at 1311. We are not persuaded that the

Examiner erred in rejecting independent claim 9 and claims 10–16 which depend therefrom, under § 112(b).

The § 101 rejection of claims 1–24

“Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.” 35 U.S.C. § 101. Section 101, however, “contains an important implicit exception: Laws of nature, natural phenomena, and abstract ideas are not patentable.” *Alice Corp. Pty. Ltd. v. CLS Bank Int’l*, 134 S. Ct. 2347, 2354 (2014) (quoting *Assoc. for Molecular Pathology v. Myriad Genetics, Inc.*, 569 U.S. 576, 589 (2013)).

Alice applies a two-step framework, earlier set out in *Mayo Collaborative Services v. Prometheus Laboratories, Inc.*, 566 U.S. 66 (2012), “for distinguishing patents that claim laws of nature, natural phenomena, and abstract ideas from those that claim patent-eligible applications of those concepts.” *Alice*, 134 S. Ct. at 2355.

Under the two-step framework, it must first be determined if “the claims at issue are directed to a patent-ineligible concept.” *Id.* If the claims are determined to be directed to a patent-ineligible concept, then the second step of the framework is applied to determine if “the elements of the claim . . . contain[] an ‘inventive concept’ sufficient to ‘transform’ the claimed abstract idea into a patent-eligible application.” *Id.* at 2357 (citing *Mayo*, 566 U.S. at 72–73, 79).

As an initial matter, we note that all claims are argued together. Therefore, the claims will stand or fall together. *See* 37 C.F.R. § 41.37(c)(1)(iv). We select claim 1 as representative.

With regard to step one of the *Alice* framework, the Examiner determines that “[t]he claims are directed to a distributed integration framework having event handlers that send and receive instructions between a case management system and external content repository in response to events in the case management system.” (Non-Final Action 15.) The Examiner also determines that “the claims are directed to manipulation by software of data in one or more computers, which is similar to using categories to organize, store, and transmit information between the computers, which is an abstract idea.” (*Id.* at 16.) The Examiner also determines that “the claims are directed to representing data, such as a document, using proxy documents.” (*Id.*) The Examiner also determines that “the claims are directed to manipulating data by performing specific operations (e.g., create, unfile, delete) on the data based on detecting events.” (*Id.*)

Appellants argue that “the rejection is defective on its face for failing to provide clarity with respect to which judicial descriptor of an abstract idea is being alleged.” (Appeal Br. 134.)

We are sympathetic to Appellants’ argument. However, looking at the Examiner’s determinations more carefully, they reduce to a determination that “[t]he claims are directed to a distributed integration framework having event handlers that send and receive instructions between a case management system and external content repository in response to events in the case management system” and that “represent[s] data, such as

documents, using proxy documents.” (Non-Final Action 15–16.) This subsumes the Examiner’s additional determinations, i.e., that the claims are directed to the manipulation of data.

Regardless, under step one of the *Alice* framework, we “look at the ‘focus of the claimed advance over the prior art’ to determine if the claim’s ‘character as a whole’ is directed to excluded subject matter.” *Affinity Labs of Texas, LLC v. DIRECTV, LLC*, 838 F.3d 1253, 1257 (Fed. Cir. 2016) (quoting *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1353 (Fed. Cir. 2016)). Moreover, because the focus of the analysis is on the claim as a whole, a claim may incorporate multiple abstract ideas, but that “does not render the claim non-abstract.” *RecogniCorp, LLC v. Nintendo Co.*, 855 F.3d 1322, 1327 (Fed. Cir. 2017).

The Specification provides evidence as to what the claimed invention is directed. In this case, the Specification discloses that the claimed “invention relates to case management integration with external content repositories.” (Spec. ¶ 2.) Claim 1 recites “[a] method comprising: detecting . . . a case creation event originating from the external content repository,” “instructing . . . the case management system to create the case within the case management system,” “and instructing . . . the external content repository to create a proxy case.”

Although we and the Examiner describe, at different levels of abstraction, to what the claims are directed, it is recognized that “[a]n abstract idea can generally be described at different levels of abstraction.” *Apple, Inc. v. Ameranth, Inc.*, 842 F.3d 1229, 1240 (Fed. Cir. 2016). In this case, that does not “impact the patentability analysis.” *See id.* at 1241. Moreover, here, the limitations do not recite implementation details.

Instead, they recite functional results to be achieved. In other words, the claims do not recite “a particular way of programming or designing the software . . . , but instead merely claim the resulting systems.” *Id.*

Therefore, we are not persuaded that the Examiner erred in determining that the claims are directed to an abstract idea.

Even if we “defer[red] our consideration of the specific claim limitations’ narrowing effect for step two,” which we may do, *see Bascom Global Internet Servs., Inc. v. AT&T Mobility LLC*, 827 F.3d 1341, 1349 (Fed. Cir. 2016), we would still not be persuaded that the claims recite patent eligible subject matter.

Nonetheless, Appellants argue that the claims are directed to “technological integration of devices by a distributed integration framework that specifically as claimed integrates a case management system with an external content repository.” (Appeal Br. 131–32.) As such, Appellants further argue, the claims are “directed to technology by which the two disparate devices have been functionally integrated by the distributed integration framework” and that this “enhances computing technology itself.” (*Id.* at 132.)

Analogizing to an earlier case, we note that in *Alice*, the Court found claims directed to creating shadow records held independently by separate institutions, adjusting shadow records as transactions are made, and issuing related instructions, to amount to nothing more than “electronic record keeping—one of the most basic functions of a computer.” *Alice*, 134 S. Ct. at 2359. Moreover, Appellants do not indicate where in the Specification anything other than a generic processor programmed to perform the steps of the abstract idea of the invention is disclosed. Indeed, Appellants’

Specification discloses that the invention “may be implemented using any technology appropriate for a given implementation. For example, the respective computing devices may include devices such as a computing devices [sic] (e.g., desktop, laptop, server, etc.), or any other device capable of processing information as described above and in more detail below.” (Spec. ¶ 40.)

The introduction of a generic processor to implement the claimed steps does not alter the analysis at the second step. “[T]hese claims in substance [are] directed to nothing more than the performance of an abstract business practice . . . using a conventional computer. Such claims are not patent-eligible.” *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1256 (Fed. Cir. 2014).

Even considered as an ordered combination, the processor of Appellants’ method adds nothing that is not already present when the steps are considered separately. The claims do not, for example, purport to improve the functioning of the processor. Nor do they effect an improvement in any other technology or technical field. Instead, the claims at issue amount to nothing significantly more than an instruction to apply the abstract idea. That is not enough to transform an abstract idea into a patent-eligible invention. *Alice*, 134 S. Ct. at 2360.

In view of the above, we are not persuaded that the Examiner erred in determining that the claims are directed to patent-ineligible subject matter.

The § 102(b) rejection of claims 1, 2, 9, 10, 17, and 18

Claim 1 recites, in relevant part, “detecting . . . a case creation event . . . that invokes the inbound event handler . . . to cause creation of a case,” and “instructing . . . the case management system to create the case.”

Johnson discloses “[a] system for managing content and work items in a plurality of disparate content repositories and workflow systems.”

(Johnson, Abstract.) Johnson further discloses that its “software system . . . supports the creation of instances of virtual repositories that include content, content organizing structure (e.g. folders, folder hierarchies, taxonomies)” (*Id.* ¶ 25.) Johnson also discloses that “[i]n a preferred embodiment, subscriptions to content, [and] content organizing structures . . . can be created and persisted in the system. A subscription is a request to track one or more items and monitor it for change” (*Id.* ¶ 60.) Additionally, “[w]hen the items of a subscription are to be monitored for change . . . , an event is created. The event provides access to the subscription the event is for” (*Id.* ¶ 64.) Johnson further discloses that its system can host “event handler . . . modules that handle . . . change events of the system. Each handler receives the event and has a chance to respond These handlers may perform any appropriate action” (*Id.* ¶ 69.)

The Examiner finds that “the handler [of Johnson] receiving the event and responding is equivalent to invoking the event handler . . . , and the change notification is equivalent to a case creation event.” (Non-Final Action 27, citing Johnson ¶¶ 64 and 69.) The Examiner also finds that “creation of content is equivalent to creating a case (i.e., a computer file).” (*Id.* at 28, citing Johnson ¶ 25.)

Appellants argue that “the event that is disclosed within [the] Johnson reference appears to be configured to provide ‘access to’ a subscription, not creation of a case as claimed. . . . Stated differently, creating an event to provide access to a subscription is not equivalent to detecting a case creation event that is configured . . . to cause creation of a case within a completely separate case management system.” (Appeal Br. 60.) “Appellant [sic] claims ‘detecting’ the specific case creation event itself.” (*Id.* at 57.)

The Examiner disagrees and refers to pages 88–89 of the Final Action dated June 12, 2014. (Answer 51–52.) In that Final Action, at page 89, the Examiner found that

the step only requires detecting a case creation event. The detecting can occur regardless of the origin of the case creation event or what the event was configured to do in some other part of the system or another computing device. What the event is configured to do does not alter whether the event is detected, so long as it is detected.

As noted above, claim 1 recites “detecting . . . a case creation event . . . that invokes the inbound event handler . . . to cause creation of a case,” and “instructing . . . the case management system to create the case,” i.e., to create the case associated with the detected event/occurrence.

The portions of Johnson relied on by the Examiner disclose that “[a] subscription is a request to track one or more items and monitor it for change” (Johnson ¶ 60), and that “[w]hen the items of a subscription are to be monitored for change . . . , an event is created. The event provides access to the subscription the event is for” (*Id.* ¶ 64.) In other words, Johnson discloses detecting a change in a stored item.

But claim 1 also requires “instructing . . . the case management system to create the case,” i.e., to create the case associated with the detected

event/occurrence. For this, the Examiner relies on the disclosure in paragraph 25 of Johnson. As noted above, paragraph 25 of Johnson discloses that its “software system . . . supports the creation of instances of virtual repositories that include content, content organizing structure (e.g. folders, folder hierarchies, taxonomies)” (Johnson ¶ 25.) But the Examiner does not sufficiently explain where Johnson discloses that the creations of virtual repositories are the result of instructions for the case management system to create the case associated with the detected event/occurrence.

In view of the above, we agree that the Examiner erred in rejecting claim 1 under § 102(b). Independent claims 9 and 17 contain similar language and the Examiner relies on similar arguments in rejecting those claims under § 102(b). (*See* Non-Final Action 30–40.) Thus, for similar reasons, we find that the Examiner erred in rejecting claims 9 and 17, and dependent claims 2, 10, and 18 under § 102(b).

The § 103(a) rejections of claims 3–8, 11–16, and 19–24

In rejecting dependent claims 3–8, 11–16, and 19–24 under § 103(a), the Examiner additionally relies on Kandasamy and Sun Microsystems. (Non-Final Action 41–59.) However, the Examiner does not rely on either Kandasamy or Sun Microsystems to cure the above noted deficiency in Johnson. Therefore, for the reasons discussed above, we agree that the Examiner erred in rejecting claims 3–8, 11–16, and 19–24 under § 103(a).

DECISION

The Examiner's provisional rejections of claims 1–24 on the grounds of nonstatutory double patenting are affirmed.

The rejection of claims 1–24 under 35 U.S.C. § 112(a) is reversed.

The rejections of claims 1–24 under 35 U.S.C. § 112(b) are affirmed.

The Examiner's rejection of claims 1–24 under 35 U.S.C. § 101 is affirmed.

The Examiner's rejection of claims 1, 2, 9, 10, 17, and 18 under 35 U.S.C. § 102(b) is reversed.

The Examiner's rejections of claims 3–8, 11–16, and 19–24 under 35 U.S.C. § 103(a) are reversed.

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).

AFFIRMED