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.
11/343,340	01/30/2006	Roger R. Webster	B232/ADBS.209190	5081

121363 7590 04/10/2018
Shook, Hardy & Bacon L.L.P.
(Adobe Systems Incorporated)
Intellectual Property Department
2555 Grand Blvd
Kansas City, MO 64108

EXAMINER	
AKINTOLA, OLABODE	

ART UNIT	PAPER NUMBER
3691	

NOTIFICATION DATE	DELIVERY MODE
04/10/2018	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):

IPDOCKET@SHB.COM
IPRCDKT@SHB.COM
mjjordan@shb.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte ROGER R. WEBSTER and JEREMY A. HALL

Appeal 2016-006760[1]
Application 11/343,340[2]
Technology Center 3600

Before ANTON W. FETTING, NINA L. MEDLOCK, and
GEORGE R. HOSKINS, *Administrative Patent Judges*.

MEDLOCK, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1–26. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

[1] Our decision references Appellants' Appeal Brief ("App. Br.," filed September 17, 2015) and Reply Brief ("Reply Br.," filed June 27, 2016), and the Examiner's Answer ("Ans.," mailed April 27, 2016) and Final Office Action ("Final Act.," mailed April 20, 2015).
[2] Appellants identify Adobe System, Inc. as the real party in interest. App. Br. 3.

CLAIMED INVENTION

Appellants' claimed invention "relates to electronic inventory management and in particular to techniques for automatic asset versioning" (Spec. ¶ 1).

Claims 1, 8, 13, and 20 are the independent claims on appeal. Claim 1, reproduced below, is illustrative of the claimed subject matter:

> 1. A method comprising:
> using one or more processors to perform operations of:
> detecting a new asset in a monitored storage location, wherein the detecting is performed by a daemon configured to detect an operating system event that is raised and associated with the monitored storage location, and wherein the operating system event is raised upon a conducting of operations including at least adding the new asset to the monitored storage location;
> requesting advice in a form of a user input as to whether the new asset is to receive a unique identifier; and
> generating the unique identifier in response to favorable advice and recording the unique identifier in metadata associated with the new asset along with versioning information.

REJECTIONS

Claims 1–26 are rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter.

Claims 1–26 are rejected under 35 U.S.C. § 103(a) as unpatentable over Gentner et al. (US 2002/0107886 A1, pub. Aug. 8, 2002) (hereinafter "Gentner") and Mohammed (US 2001/0014968 A1, pub. Aug. 16, 2001).

ANALYSIS

Non-Statutory Subject Matter

Under 35 U.S.C. § 101, an invention is patent-eligible if it claims a "new and useful process, machine, manufacture, or composition of matter." 35 U.S.C. § 101. The Supreme Court, however, has long interpreted § 101 to include an implicit exception: "[l]aws of nature, natural phenomena, and abstract ideas" are not patentable. *See, e.g., Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2354 (2014).

The Supreme Court, in *Alice*, reiterated the two-step framework previously set forth 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 Corp.*, 134 S. Ct. at 2355. The first step in that analysis is to "determine whether the claims at issue are directed to one of those patent-ineligible concepts." *Id.* If the claims are not directed to a patent-ineligible concept, e.g., an abstract idea, the inquiry ends. Otherwise, the inquiry proceeds to the second step where the elements of the claims are considered "individually and 'as an ordered combination'" to determine whether there are additional elements that "'transform the nature of the claim' into a patent-eligible application." *Id.* (quoting *Mayo*, 566 U.S. at 79, 78).

The Court acknowledged in *Mayo*, that "all inventions at some level embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas." *Mayo*, 566 U.S. at 71. Therefore, the Federal Circuit has instructed that claims are to be considered in their entirety to determine "'whether their character as a whole is directed to excluded subject matter.'"

McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1312 (Fed. Cir. 2016) (quoting *Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1346 (Fed. Cir. 2015)).

Here, in rejecting claims 1–26 under 35 U.S.C. § 101, the Examiner finds that the claims are "directed to the abstract idea of organizing human activity and a mathematical algorithm" (Final Act. 2).[3] Paraphrasing the language of the independent claims, the Examiner characterizes the claims as falling into one of three groups, i.e., claims 1–7, claims 8–19, and claims 20–26 (*id.*).

The Examiner states, for example, that claims 1–7 are directed to "detecting a new asset, requesting advice as to whether the new asset is to receive a unique identifier and generating the unique identifier in response to favorable advice and recording the unique identifier associated with the new asset along with versioning information" (*id.*). The Examiner similarly paraphrases the language of claims 8–19 and claims 20–26 in describing the concept to which those claims are directed (*id.*). But, the Examiner does not explain how the claim limitations correspond to the asserted abstract idea. For example, the Examiner does not explain how "detecting a new asset," which claim 1 recites is "performed by a daemon[4] configured to detect an

[3] We note that the pages of the Final Office Action are unnumbered; we refer to the page bearing the "Office Action Summary" legend as page 1, and treat the pages that follow as though consecutively numbered.
[4] A daemon is generally understood to refer to a type of computer program that runs unobtrusively in the background, rather than under the direct control of a user, and is activated by a specific event or condition. *See Newton's Telecom Dictionary*, CMP Books, 22nd ed. (2006) (defining the term "daemon" as "[a] harmless UNIX program that waits in background and runs when a request is made on the port that it is watching").

operating system event," involves human activity and/or a mathematical algorithm. Instead, the Examiner makes the conclusory statement that claims 1–26 are directed to "the abstract idea of organizing human activity and a mathematical algorithm" without identifying how the claim limitations correspond to the asserted type of abstract idea.

We find that the Examiner has not sufficiently established that the claimed invention is directed to an abstract idea. Therefore, we do not sustain the Examiner's rejection of claims 1–26 under 35 U.S.C. § 101.

Obviousness

We are persuaded by Appellants' argument that the Examiner erred in rejecting independent claims 1, 8, 13, and 20 under 35 U.S.C. § 103(a) because Mohammed, on which the Examiner relies (*see* Final Act. 4–5 (citing Mohammed ¶¶ 27, 41, 43); *see also* Ans. 4–5), does not disclose or suggest "a monitored storage location," i.e., "detecting a new asset in a monitored storage location, wherein the detecting is performed by a daemon configured to detect an operating system event that is raised and associated with the monitored storage location," as recited in claim 1, and similarly recited in claims 8, 13, and 20 (App. Br. 16–18; *see also* Reply Br. 7–9).

Mohammed is directed to a system configured to automatically upgrade a software component if an operating system upgrade is detected (Mohammed, Abstract). More specifically, Mohammed discloses that a detection module, e.g., a service daemon, is invoked each time the system is booted, and is configured to detect an operating system upgrade and invoke an upgrade module if a system upgrade is detected (*id.* ¶¶ 16, 19). Mohammed discloses, with reference to Figure 2, that on system startup, the detection module determines the version of the current operating system, i.e., the current OS platform (*id.* ¶ 20). Next, the detection module checks the operating system version (i.e., software OS platform) under which the currently installed software components were loaded; if the current OS platform matches the software OS platform (i.e., indicating an operating system upgrade has not occurred), the detection module exits without performing a software upgrade (*id.*). Otherwise, an upgrade command is issued to invoke the upgrade module to perform the software update (*id.* ¶ 21; *see also id.* ¶¶ 27, 41, 43).

Mohammed discloses that, in one embodiment, when the detection module is initially installed, certain setup steps are performed, including updating entries in the operating system registry; the updated registry entries include a "Platform" entry that identifies the software OS platform for which the currently loaded software components are configured (*id.* ¶ 23). Mohammed details the operation of the detection module, with reference to Figures 3A–3C, and discloses that the detection module retrieves the current OS platform, e.g., via a GetVersionEx (&osVersionlnfo) method, and stores the value in a parameter, "NewPlatform"; the detection module next retrieves the software OS platform, e.g., from the operating system registry, and stores the retrieved value in a parameter "OrigPlatform" (*id.* ¶ 25). The values of NewPlatform and OrigPlatform are compared, and if the two are not equal, a software update may be performed automatically or as indicated by the user (*id.* ¶¶ 29–32).

Responding to Appellants' arguments, the Examiner takes the position that "in order for the program to detect that an operating system (inherently stored at a specific location in the computer) has been upgraded from a previous version, the specific location must be monitored by the detection module" (Ans. 4). However, we find nothing in the cited portions of Mohammed, nor does the Examiner point to anything, that discloses or suggests that the "specific location in the computer" where the operating system is "inherently stored," i.e., the storage location at which the new asset is detected, is monitored by the detection module, as the claims require.

Indeed, the Examiner acknowledges that Mohammed explicitly states "that in the Windows operating system, the method that may be used to retrieve the current OS platform is GetVersionEx(&osVersionlnfo)" and also states that the detection module "retrieves a value representing the operating system platform (software OS platform, which may be stored in the operating system registry) under which software was loaded and stores the retrieved value into a parameter OrigPlatform" (*id.* at 4–5 (citing Mohammed ¶ 25)).

The Examiner asserts that Mohammed, thus, discloses the claimed "monitored storage location" (*id.* at 5). However, we agree with Appellants that even if the Examiner intends that the registry be considered the "monitored storage location," the value representing the software OS version (which stored as a registry entry) cannot reasonably be considered a "new asset," within the meaning of the claims (Reply Br. 9). Retrieving a value representing the OS platform to determine if an OS upgrade has occurred, as disclosed in Mohammed, also is not the same as "detecting a new asset in a monitored storage location, wherein the detecting is performed by . . . detect[ing] an operating system event that is raised and associated with the monitored storage location," as recited in claims 1, 8, 13, and 20 (App. Br. 17–18; *see also* Reply Br. 8–10).

In view of the foregoing, we do not sustain the Examiner's rejection of independent claims 1, 8, 13, and 20 under 35 U.S.C. § 103(a). For the same reasons, we also do not sustain the rejection of dependent claims 2–7, 9–12, 14–19, and 21–26. *Cf. In re Fritch*, 972 F.2d 1260, 1266 (Fed. Cir. 1992) ("dependent claims are nonobvious if the independent claims from which they depend are nonobvious").

DECISION

The Examiner's rejection of claims 1–26 under 35 U.S.C. § 101 is reversed.

The Examiner's rejection of claims 1–26 under 35 U.S.C. § 103(a) is reversed.

REVERSED