



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.
14/626,735	02/19/2015	Biju Balachandran	000005-047900US	4823
58735	7590	02/01/2019	EXAMINER	
Fountainhead Law Group P.C. Chad R. Walsh 900 LAFAYETTE STREET SUITE 301 SANTA CLARA, CA 95050			KABIR, MOHAMMAD H	
			ART UNIT	PAPER NUMBER
			2199	
			NOTIFICATION DATE	DELIVERY MODE
			02/01/2019	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@fountainheadlaw.com
rbaumann@fountainheadlaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte BIJU BALACHANDRAN and NIRAJ KUMAR¹

Appeal 2018-005707
Application 14/626,735
Technology Center 2100

Before DAVID M. KOHUT, ERIC B. CHEN, and
JOSEPH P. LENTIVECH, *Administrative Patent Judges*.

KOHUT, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants seek our review, under 35 U.S.C. § 134(a), of the Examiner's Final Rejection of claims 3–5, 7, 10–12, 14, 17, 18, and 20.² We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

¹ Appellants identify the real party in interest as “SAP SE, a corporation.” App. Br. 2.

² Claims 1, 2, 6, 8, 9, 13, 15, 16, and 19 are cancelled.

INVENTION

The invention performs an “automatic selection and customization of landscape guides.” Spec., Title (emphasis omitted). According to Appellants, a landscape is an arrangement of servers, and users follow installation guides to perform a landscape’s software upgrades. *Id.* ¶ 1. The invention determines the relevant portions of such guides for users by, in part, identifying the software applications and versions that are already installed. *Id.* ¶¶ 2–3.

Independent claim 7 is illustrative of a disputed claim feature and reproduced, below, with emphasis on this feature.

7. A computer-implemented method comprising:

receiving, by a processor, a customer request to modify a customer landscape that includes a plurality of computer systems, wherein the customer request describes a desired configuration for a software application to execute on a computer system of the customer landscape;

retrieving, by the processor, information describing a current configuration of the software application or the computer system;

determining, by the processor, a software logistics process according to the current configuration and the desired configuration, the software logistics process describing a process to implement the desired configuration of the software application, wherein determining the software logistics process comprises:

comparing, by the processor, a first version identifier for the current configuration against a second version identifier for the desired configuration,

determining, by the processor, the software logistics process based on the comparison, and

wherein the software logistics process is determined to be an installation when the first version identifier is set to NIL;

generating, by the processor, a landscape guide query according to the software logistics process, the customer request, and the information, the landscape guide query being configured to identify a plurality of downloaded universal landscape guides stored in a guides database, the plurality of downloaded universal landscape guides providing instructions to implement the desired configuration of the software application;

customizing, by the processor, a personalized landscape guide from the plurality of downloaded universal landscape guides according to the information describing the current configuration of the software application and a language of the customer, the personalized landscape guide including an impact analysis describing an effect of the software logistics process upon the customer landscape, the personalized landscape guide comprising instructions in the language of the customer;

running, by the processor, the landscape guide query; and

receiving, by the processor, the personalized landscape guide returned from the query.

App. Br. 16–17.

REJECTIONS³

The Examiner rejects claims 3, 4, 7, 10, 11, 14, 17, 18, and 20 under 35 U.S.C. § 103(a) as unpatentable over Cheng (US 6,151,643; Nov. 21,

³ The Final Action’s statements of the rejections fail to account for the Examiner’s subsequent entry of claim amendments. *See* Adv. Act. (Oct. 25, 2017) (entry of claim amendments); Amdt. Under 37 C.F.R. § 1.116 (Sept. 22, 2017). The Amendment “rolled” claims 2, 6, 9, 13, 16, and 19 into claims 7, 14, and 20—that is, canceled claims 2, 6, 9, 13, 16, and 19, added their limitations to claims 7, 14, and 20 (specifically to the ensuing one of these claims), and amended claims 7, 14, and 20 to be

2000), Said (US 2016/0170732 A1; June 16, 2016), Meier (US 2013/0117700 A1; May 9, 2013), Alsina (US 2014/0223423 A1; Aug. 7, 2014), and Back (US 2007/0220507 A1; Sept. 20, 2007). Final Act. 6–14, 17–20 (July 27, 2017).

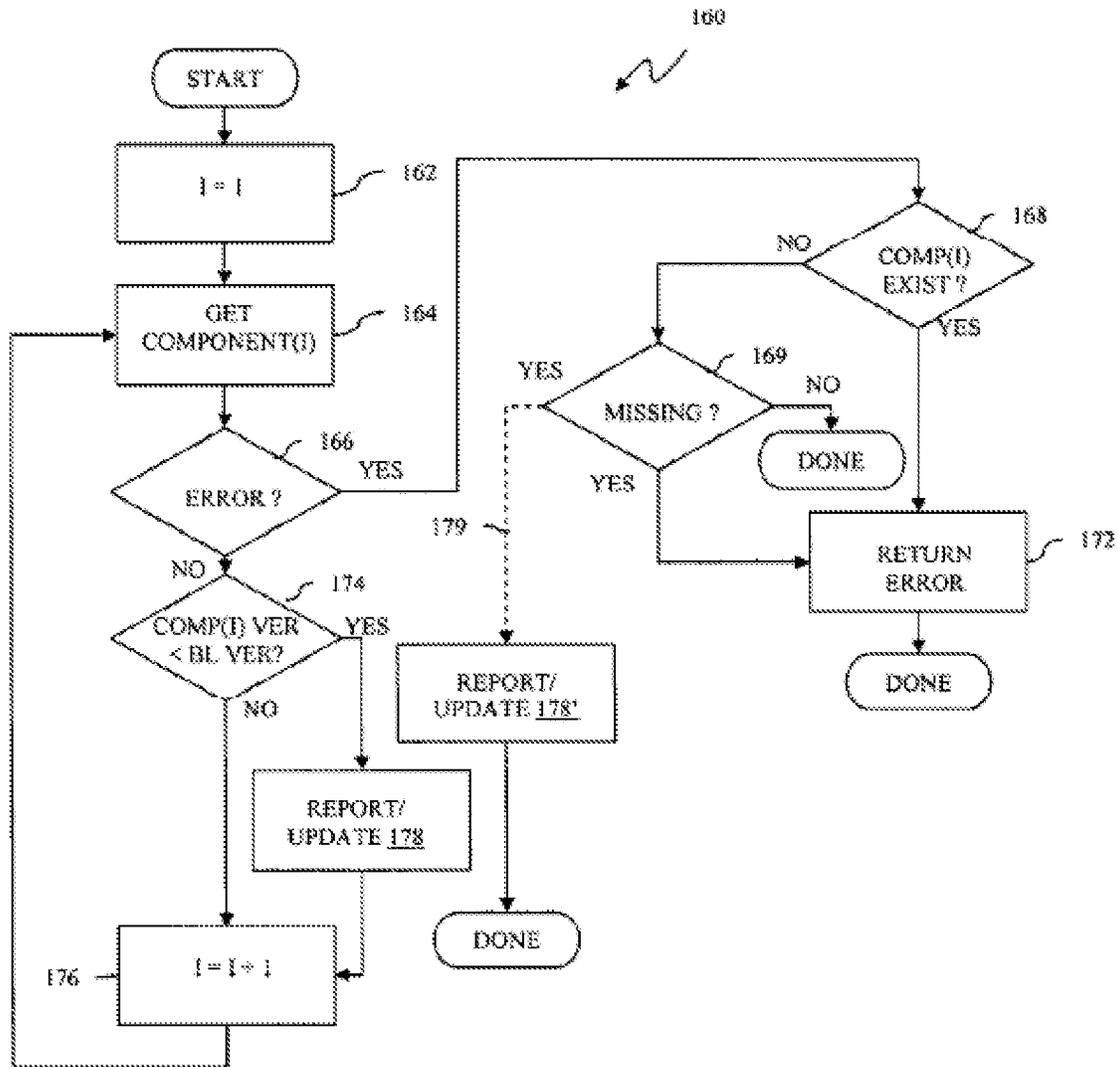
The Examiner rejects claims 5 and 12 under 35 U.S.C. § 103(a) as unpatentable over Cheng, Said, Meier, Alsina, Back, and McCaleb (US 6,751,794 B1; June 15, 2004). Final Act. 15–17.

INDEPENDENT CLAIMS 7, 14, AND 20

Appellants argue claims 7, 14, and 20 as a group as they contain similar limitations and were rejected by the Examiner over the same references. App. Br. 12–13. Appellants argue that the Examiner fails to show Back, one of the applied references, teaches “wherein the software logistics process is determined to be an installation when the first version identifier is set to NIL,” as recited in each of the claims. *Id.*

As the Examiner and Appellants dispute whether this limitation is taught by the software upgrade process of Back’s Figure 7 (*see infra*), we begin with an explanation of this process and its steps as shown by Figure 7 (the illustrated process flow), reproduced below.

independent. Adv. Act. (Cont’n Sheet). Our listing of the rejections corrects the statements of the rejections to: (i) add, if a rejected claim now recites or incorporates the limitations of a rolled claim, the prior art references cited against the rolled claim; (ii) account for changes in claim dependencies; and (iii) remove claims 2, 6, 9, 13, 16, and 19.



Back's Figure 7, above, is a flow chart of operations for determining components in need of installation or an upgrade.

The process automates a “baseline” software upgrade, installing the minimum components (software applications) and updating to the minimum versions (of those components/applications) needed by an integrated software program. Back ¶ 43. Generally speaking, the process determines the baseline components/versions lacked by the system (*id.* ¶¶ 45–46) and then obtains those components/versions (*id.* ¶¶ 47–48). The process first “calls” the system’s “version repository 24” for “component information”

about at least one “component(i)” (Step 164). *Id.* ¶¶ 33, 45; *see also id.* ¶¶ 20–22 (general description of the repository 24), 27 (“version repository 24 may contain . . . a first table [of] the actual version numbers of components installed . . . [and] other tables [of] baselines”). If the call results in an error (Step 166:Yes), then the process determines whether “the error . . . indicates that there is no component information corresponding to the [component(i)]” (Step 168). *Id.* ¶ 45. If so, i.e., component information does not “exist” for the component(i) (Step 168:No), then the process determines whether the component(i) is a baseline component (Step 169). *Id.* If so, i.e., a baseline component is “missing” (Step 169:Yes), then the process prompts the system to install the missing baseline component (Steps 178’ and 179) and reports the above-noted error (Step 172). *Id.* ¶¶ 45, 48.

The Examiner finds the disputed limitation is taught by the above steps. Ans. 5–6. Specifically, the Examiner finds that Appellants’ Specification describes the claimed “version identifier . . . set to NIL” as being merely an indicator that software is not installed. *Id.* at 6. The Examiner accordingly interprets the claimed “version identifier . . . set to NIL” as being data that indicates a target software upgrade will require a software installation. *Id.* The Examiner finds Back’s reported error (Step 172) teaches such an identifier *when* reported as a “particular error code” (Back ¶ 33) indicating that a baseline component(i) is “missing” (Step 169:Yes and 172) and will accordingly be installed (Steps 178’ and 179). Ans. 5–6 (citing Back ¶¶ 34–35).

Appellants argue that Back’s returned error does not teach the claimed “version identifier . . . set to NIL” because: “[T]he claims recite a version

identifier whose value is explicitly set to NIL. Given this explicit population of the version identifier . . . , in no sense would any ‘ERROR’ be returned.” App. Br. 12–13 (emphasis omitted); *see also* Reply Br. 7. Appellants further argue that Back’s only illustrated identifiers are not used as the returned error, contending: “Figure 3B of the Back reference does describe . . . a ‘component ID’ field 56. Despite the apparent availability of such an ID field, however, [Back] chooses instead to rely upon the ERROR-based mechanism.” App. Br. 13 (emphasis omitted); *see also* Reply Br. 7.

We are unpersuaded because Appellants’ arguments do not address the Examiner’s specific finding that the claimed “version identifier . . . set to NIL” can be interpreted as an indication that software is not installed. *See* Final Act. 4, 20–21; Ans. 5 (citing Spec. ¶ 24); *see also* App. Br. 6–9 (“Summary of Claimed Subject Matter” section stating Spec. ¶¶ 7, 24, 33 describe the claimed “version identifier . . . set to NIL”). Consistent with the Examiner’s claim interpretation, Appellants’ Specification states: “[t]he software logistics process [is] an installation when the first version identifier is set to NIL” (Spec. ¶ 7); “version is NIL[,] e.g., the software application is not installed” (*id.* ¶ 24); “version is NIL, implying that no versions . . . have been installed” (*id.* ¶ 33). *See also In re Hyatt*, 211 F.3d 1367, 1372 (Fed. Cir. 2000) (“[D]uring examination proceedings, claims are given their broadest reasonable interpretation consistent with the specification.”). Appellants fail to rebut the Examiner’s claim interpretation with sufficient reasoning or evidence. App. Br. 12–13; Reply Br. 7. Instead, Appellants present a conclusory assertion that the “version identifier . . . set to NIL” (claim 7) must be interpreted as an explicit NIL value of a populated ID field. *Id.* In light of the Specification disclosures, we disagree that the

claimed “first version identifier” must be interpreted as a populated value of a field.

Appellants’ arguments also fail to address the Examiner’s findings regarding Back’s paragraphs 33–35. Ans. 6–7, 10. As discussed *supra*, Back therein describes a “particular error code” that indicates a component is “missing” and will accordingly be installed. Back ¶¶ 33–35; *see also supra* 5 (Examiner’s findings). By indicating the repository 24 is not cached with version information for the called component(i) (Back ¶ 33), the “particular error code” indicates that a corresponding index of the repository 24 is set to NIL with respect to version information (i.e., indicates a lack of such information) because the component is not installed. *See also* Ans. 10 (addressing Back ¶¶ 31–35). Thus, even assuming the claimed “version identifier . . . set to NIL” must be something other than an error code (as is argued), the Examiner has shown that Back’s above index is set to NIL with respect to version information (Ans. 6–7, 10). Appellants do not argue that Back’s above index fails to teach a “version identifier . . . set to NIL” as claimed.

For the foregoing reasons, we are unpersuaded of error in the obviousness rejection of independent claims 7, 14, and 20 over Cheng, Said, Meier, Alsina, and Back. Accordingly, we sustain the rejection.

DEPENDENT CLAIMS 3–5, 10–12, 17, AND 18

Claims 3–5, 10–12, 17, and 18 each depend from one independent claims 7, 14, and 20 addressed above. Appellants merely separate these claims into three groups—(i) claims 3, 10, and 17; (ii) claims 4, 11, and 18; (iii) claims 5 and 12—and reassert the above-addressed arguments for each group. App. Br. 14–15; Reply Br. 8–9. Accordingly, we sustain the

Appeal 2018-005707
Application 14/626,735

obviousness rejections of: claims 3, 4, 10, 11, 17, and 18 over Cheng, Said, Meier, Alsina, and Back; and claims 5 and 12 over Cheng, Said, Meier, Alsina, Back, and McCaleb.

DECISION

The Examiner's decision rejecting claims 3–5, 7, 10–12, 14, 17, 18, and 20 is affirmed.

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