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.
14/023,549	09/11/2013	Lu Liang	CAM920120063US2_8150-0435	6435

73109 7590 09/06/2018

Cuenot, Forsythe & Kim, LLC
20283 State Road 7
Ste. 300
Boca Raton, FL 33498

EXAMINER	
PAULINO, LENIN	

ART UNIT	PAPER NUMBER
2197	

NOTIFICATION DATE	DELIVERY MODE
09/06/2018	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):

ibmptomail@iplawpro.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte LU LIANG, ZHI QIANG SUN, BRIAN C. SCHIMPF, and YUHONG YIN

Appeal 2017-011589
Application 14/023,549
Technology Center 2100

Before JOHNNY A. KUMAR, JENNIFER L. McKEOWN, and CATHERINE SHIANG, *Administrative Patent Judges*.

McKEOWN, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants[1] appeal under 35 U.S.C. § 134(a) from the Examiner's decision to reject claims 1–11. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

---

[1] According to Appellants, the real party in interest is IBM Corporation. App. Br. 1.

STATEMENT OF THE CASE

Appellants' disclosed and claimed invention is directed to a "[s]oftware lifecycle management includ[ing], searching, using a processor, historical development data including prior development tasks for a software system." Abstract.

Claim 1 is illustrative of the claimed invention and reads as follows:

> 1. A method of software lifecycle management, the method comprising:
>
> searching, using a processor, historical development data comprising prior development tasks for a software system, wherein the searching is performed according to a current development task for the software system;
>
> determining that the current development task has an affinity with a selected prior development task implemented within the software system; and
>
> providing a recommendation for the current development task based upon the selected prior development task.

THE REJECTIONS

The Examiner provisionally rejected claims 1–11 on the ground of non-statutory obviousness-type double patenting as being unpatentable over claims 12–25 of copending Application No. 13/682,308. Final Act. 7.

The Examiner rejected claims 1 under 35 U.S.C. § 101 as directed to non-statutory subject matter. Final Act. 7–9.

The Examiner rejected claims 1, 4, 6, 8, and 9 under 35 U.S.C. § 103(a) as unpatentable over Olson (US 2008/0141214 Al; published June 12, 2008); and Garimella et al. (US 2011/0161913 A1; published June 30, 2011) (hereinafter "Garimella"). Final Act. 9–14.

The Examiner rejected claims 2, 3, 5, and 7 under 35 U.S.C. § 103(a) as unpatentable over Olson, Garimella, and Wagner et al. (US 2005/0262482 A1; published Nov. 24, 2005) (hereinafter "Wagner"). Final Act. 14–17.

The Examiner rejected claims 10 and 11 under 35 U.S.C. § 103(a) as unpatentable over Olson, Garimella, and Travison et al. (US 2008/0222501 A1; published Sept. 11, 2008) (hereinafter "Travison"). Final Act. 17–18.

ANALYSIS

THE OBVIOUSNESS-TYPE DOUBLE PATENTING REJECTION

Claims 1–11

Appellants present no arguments pertaining to the Examiner's provisional obviousness-type double patenting rejection. *See* App. Br. 2, fn. 1 (noting that the provisional obviousness-type double patent rejection is not the subject of the appeal). Accordingly, we summarily sustain this rejection.

THE REJECTION UNDER 35 U.S.C. § 101

Claim 1

Based on the record before us, we are not persuaded that the Examiner erred in rejecting claim 1 as directed to patent ineligible subject matter.

The Examiner determines that claim 1 is directed to the abstract idea without significantly more. Final Act. 8; Ans. 2. Specifically, the Examiner explains that claim 1 is directed to an idea of itself (Final Act. 8) and further elaborates that claim 1 is directed to the abstract idea of "automat[ing] the process of providing recommendation to a developer by searching historical development data, determining affinity between current development task and a selected prior development task, and providing a recommendation

based on the selected prior development task." Ans. 6–7; *see also* Ans. 5–7 (citing *Accenture Global Services, GMBH v. Guidewire Software, Inc.*, 728 F.3d 1336 (Fed. Cir 2013) as support). *See also* Final Act. 8 (citing and relying on *In re Maucorps*, 609 F.2d 481, 482, 486 (CCPA 1979) and *SmartGene, Inc. v. Advanced Biological Labs. SA*, 555 F. App'x 950 (Fed. Cir. 2014)).

The Examiner further determines that

> The additional limitations that the searching, determining and providing is implemented by a processor in a method for software lifecycle management but the mere recitation of "a processor" is akin to adding the words "apply it" with a computer in conjunction with the abstract idea. However, such limitations are not enough to add significantly more to the method for searching, determining and providing. The steps describe nothing more than a computer's basic function of searching, determining and providing information and do not meaningfully limit the performance of processor.

Final Act. 9.

Appellants argue that the Examiner's reliance on *Accenture* is misplaced because claim 1 merely involves an abstract idea, but is not directed to an abstract idea. Reply Br. 3; App. Br. 11, 13–14. Appellants allege that the Examiner erred in finding that the claimed invention is *directed to* "'searching, determining, and providing information.'" App. Br. 13. According to Appellants, "[i]n this instance, the claims certainly do more than 'simply instruct' one to implement 'searching, determining and providing information' on a generic computer. Thus, the claims are <u>not</u> 'directed to' the Examiner's alleged abstract idea." App. Br. 14.

Appellants further contend that

> The Examiner's analysis fails in that it does not appreciate that the claimed invention, as a whole, is directed to more than "generating tasks [based on] rules ... to be completed upon the occurrence of an event." In this instance, the claimed invention contemplates finding a prior development task for a software system based upon a search using a current development task in the same software system. Once a prior development task is found, a determination is made whether the prior development task has an affinity with the current development task and a recommendation is provided on that basis. Accordingly, the claimed invention results in an *improvement to the development of software*, which is a technical task necessarily rooted in computer technology. Consequently, the Examiner cannot properly assert that the facts of the present application are sufficiently similar to the facts of Accenture Global Services so as to rely upon the rationale provided by the Federal Circuit.

Reply Br. 3–4; App. Br. 14; *see also* App. Br. 11 (distinguishing *Maurcorps* because the claimed invention is directed to the technological improvement, "i.e. the creation of software.").

We find Appellants' arguments unpersuasive. Notably, Appellants do not identify improvements to the *functioning of the computer itself*, or meaningful limitations that do more than generally linking the use of the abstract idea to a particular technological environment. *See Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1336 (Fed. Cir. 2016)(finding the claims are not directed to an abstract idea because "the plain focus of the claims is on an improvement to computer functionality itself, not on economic or other tasks for which a computer is used in its ordinary capacity."); *Electric Power Group, LLC v. Alstom S.A.*, 830 F.3d 1350, 1354 (Fed. Cir. 2016)(distinguishing and discussing *Enfish*).

While developing software generally may be considered a technological task, claim 1 does not identify any improvement in the

functioning of either software or hardware of a computer.[2] Rather, as the Specification explains, the claimed invention is directed to improving the efficiency for the mental processes of a software developer. Spec. ¶¶ 1–2; Ans. 6–7 (explaining that the claims merely automate a process). *See also OIP Techs., Inc. v. Amazon.com, Inc.*, 788 F.3d 1359, 1363 (Fed. Cir. 2015) ("relying on a computer to perform routine tasks more quickly or more accurately is insufficient to render a claim patent eligible")

We similarly agree with the Examiner that *SmartGene*, while not precedential, is persuasive in this case. The Federal Circuit concluded the claims at issue in *SmartGene* were patent ineligible because they did "no more than call on a 'computing device,' with basic functionality for comparing stored and input data and rules, to do what doctors do routinely." *SmartGene*, 555 Fed. Appx. at 954. In the instant case, claim 1 recites a processor to compare stored and inputted data, i.e. historical developmental tasks and current developmental task, to determine if there is a similarity, i.e. affinity, to do what a developer does routinely. *See, e.g.*, Final Act. 8–9; *see also* Spec. ¶ 2 (noting that developers would gain software historical data from other developers). In other words, the claimed steps do not rely on an inventive device or technique for displaying information or new techniques for analyzing information, but rather constitute a generic recitation of steps

[2] With respect to the recited "recommendation," we note that the Specification explains the recited "recommendation" may merely be displaying "a list of references such as links, to records located within the historical development data." Spec. ¶ 52. Moreover, claim 1 does not require providing the recommendation to be performed by a processor. Nor does claim 1 require the developer to implement the provided recommendation.

for manipulating data. *See also Electric Power Group*, 830 F.3d at 1354 (citations omitted) (discussing that the Federal Circuit has "treated analyzing information by steps people go through in their minds, or by mathematical algorithms, without more, as essentially mental processes within the abstract-idea category.").

Moreover, the Federal Circuit has found claims that simply manipulate data, like those at issue here, are patent ineligible. *See, e.g., Intellectual Ventures I LLC v. Erie Indem. Co.*, 850 F.3d 1315, 1327 (Fed. Cir. 2017) ("creating an index and using that index to search for and retrieve data"); *Electric Power Group*, 830 F.3d at 1353 (collection, manipulation, and display of data); *Intellectual Ventures I LLC v. Capital One Bank (USA)*, 792 F.3d 1363, 1370 (Fed. Cir. 2015) (customizing information and presenting it to users based on particular characteristics); *Content Extraction and Transmission LLC v. Wells Fargo Bank, National Ass'n*, 776 F.3d 1343, 1347 (Fed. Cir. 2014) ("collecting data, . . . recognizing certain data within the collected data set, and . . . storing that recognized data in a memory").

Accordingly, for the reasons discussed above, we affirm the Examiner's rejection of claim 1 as directed to patent ineligible subject matter.[3]

[3] Upon further prosecution, the Examiner should consider whether dependent claims 2–11 should also be rejected as directed to non-statutory subject matter.

THE REJECTION BASED ON OLSON AND GARIMELLA

Claims 1, 4, 6, 8, and 9

Based on the record before us, we are not persuaded that the Examiner erred in rejecting claims 1, 4, 6, 8, and 9 as unpatentable over Olson and Garimella.

Appellants contend that Olson fails to teach or suggest "searching, using a processor, historical development data comprising prior development tasks for a software system, wherein the searching is performed according to a current development task for the software system," as recited in claim 1. App. Br. 15–16; *see also* App. Br. 21. Appellants, in particular, note that Olson generally describes "several factors associated with 'best practices,' but "none of these correspond to the claimed 'a current development task for the software system.'" App. Br. 16.

We disagree. The Examiner explains that Olson's collected data including information about activities (i.e. tasks) such as edit file, open file, save file, build started, build ended, debug and design edit corresponds to the claimed historical development data comprising prior development tasks for a software system. Ans. 8–9; Olson ¶ 24; *see also* Spec ¶¶ 24, 27 (identifying a change request and change set as examples of development tasks). This collected information is used, based on a user request, to form best practices, which as Appellants acknowledge can be based on several factors. *See* App. Br. 16. For example, Olson teaches that best practices may be presented for *project activities*. Olson ¶ 20; Ans. 8; *see also* Olson ¶ 10 (discussing collecting historical task information to provide insight for a *given task*). In other words, Olson teaches searching the collected information, i.e. historical development data comprising prior development

tasks, based on a project task to obtain the best practices for that project task. Olson also discloses comparing a *current* project with the best practices. Olson ¶¶ 2, 27– 29. These teachings would at least suggest searching based on a current project task to a skilled artisan. Final Act. 10–11; Ans. 9–10. As such, we agree with the Examiner that Olson teaches this disputed limitation.

Appellants next assert that Olson fails to teach the recited prior and current development tasks for the same software system. App. Br. 16–19. According to Appellants, Olson compiles industry wide information and, as such, "it is not clear that the data associated with 'best practices' refer to the current software system being developed." App. Br. 16; *see also* App. Br. 19 ("any teachings regarding prior software projects within an industry do not necessarily (i.e., inherently) correspond to limitations (i.e., the prior development task) associated with the same software system.); Reply Br. 4–6.

We find this argument unpersuasive. We recognize that Olson discusses industry wide best practices, however, Olson also teaches collecting data from one subscriber and obtaining best practices from the stored data for that one subscriber. *See, e.g.*, Olson ¶ 13, ¶ 21, Abstract (describing that information is collected for a software development project with one or more users working on the project); Olson ¶ 10 (noting it is desirable to collect data from tools used for software development). Olson also expressly teaches that the described system may function with only one organization as a subscriber. Olson ¶ 30.

Similarly, we find Appellants' contention that the combination of Olson and Garimella fails to teach or suggest "determining that the current

development task has an affinity with a selected prior development task implemented within the software system" unavailing. App. Br. 22–23. For example, Olson teaches the best practices may be generated "based on project phase, project tasks, or any other project related attribute." Olson ¶ 21; Ans. 10. Further, the best practices "show the timeline of a project and the important tasks and phases that should be completed." Olson ¶ 21. In other words, in generating the best practice data for presentation to the user, Olson teaches that a current development task has an affinity with the generated best practices, i.e. selected prior development tasks. *See* Final Act. 10–11; Ans. 10; Olson ¶¶ 20, 21, 27; *see also* Olson ¶ 28 (teaching that Olson's system compares a current project with best practices constantly).

Moreover, the Examiner additionally relies on Garimella as teaching this limitation. *See* Final Act. 11. Namely, Garimella teaches "searching a service repository, where the query is configured to find an existing service that fulfills the requirements of a particular service solution component/ business task." Garimella ¶ 75. Garimella teaches determining an affinity between current and prior business tasks.

Finally, Appellants allege that Olson and Garimella fail to teach or suggest providing a recommendation for the current development task based upon the selected prior development task." App. Br. 24. According to Appellants, neither Olson nor Garimella disclose that the recommendation is "based on the selected prior development task." *Id.* As noted above, however, the Specification explains that the recited "recommendation" may be simply displaying the prior development task and any linked records. Spec. ¶ 52 ("In one aspect, the recommendation is a list of references such as links, to records located within the historical development data.").

Olson-teaches displaying the generated best practices, including "the timeline of a project and the important tasks and phases that should be completed." Olson ¶ 21. Olson's system may also include an alert module "that constantly compares the progress of the current project with best practices and alerts a user if there is a deviation or if the deviation is more than a predetermined value." Olson 28; Final Act. 11. Moreover, Garimella generates "based on the gathered information" and other parameters, "a 'skeleton' of service artifacts for fulfilling the functionality of the selected service solution component," i.e. a recommendation for the current development task based upon the selected prior development task. Garimella ¶ 88; Final Act. 11. As such, we are not persuaded of error in the Examiner's findings.

With respect to claim 4, Appellants assert that

> nothing in the Examiner's analysis refers to "a current change set" or "prior change sets." Instead, the Examiner only refers to "project data," and the Examiner presents no evidence to establish that project data refers to "a current change set." Moreover, the Examiner presents no evidence to establish that "metrics and best practices" refer to "prior change sets."

App. Br. 26. Appellants also contend that the Examiner incorrectly relies on Olson's metrics as change sets. App. Br. 26.

While Olson does not expressly recite the term "change sets," Olson describes that the best practice metrics may include information about activities such as *edit file*, open file, save file, build started, build ended, debug and *design edit*." Olson ¶ 24 (emphasis added). The Examiner finds, and we agree that these teachings would "require[] a change in the current software development project." Ans. 12. Also, as discussed above, Olson

teaches comparing current development tasks with past development data. As such, we are not persuaded of error in the Examiner's findings.

Accordingly, for the reasons discussed above, we affirm the Examiner's rejection of claims 1, 4, 6, 8, and 9 as unpatentable over Olson and Garimella.

THE OBVIOUSNESS REJECTION BASED ON OLSON, GARIMELLA, AND WAGNER

Claims 2, 3, 5, and 7

Based on the record before us, we are not persuaded that the Examiner erred in rejecting claims 2, 3, 5, and 7 as unpatentable over Olson, Garimella, and Wagner.

Appellants contend that Wagner's dependency changes are distinct from the change request recited in claim 2. App. Br. 27–29. In particular, Appellants contend that

> A "change request" is a development task for a software system - it is a request to change a particular aspect of the software system. On the other hand, the "dependency changes" of Wagner refer to changes in the dependency relationships of software systems. Referring to paragraphs [0023]-[0024] of Wagner, many components of software applications are dependent upon other components. For example, a source file may be dependent upon a class file. In Wagner, these dependencies are embodied in a "dependency representation." The teachings of Wagener involve using these dependency representations during software development to keep track of how various components of the software applications relate to one another.

App. Br. 28.

Appellants' argument, fails to explain why changes to the dependency relationships within a software system would not satisfy the recited change request. Namely, these changes would change "a particular aspect of the

software system." The Examiner, in particular, cites Wagner's paragraph 18 that describes that these dependency changes may occur when a software developer alters source code. Wagner 18; Final Act. 14; *see also* Final Act. 15 (explaining that Wagner's teaching improves "the responsiveness of integrated development environment through improved dependency handling to incorporate the comparison of a change request to one made to a software project before the current request in order to see if it is the same request or different in order to eliminate double the work if it determines that the change request is the same."). As such, we are not persuaded of error in the Examiner's rejection.

Accordingly, for the reasons discussed above, we affirm the rejection of claims 2 and 3 as unpatentable over Olson, Garimella, and Wagner. We also affirm the rejection of claims 5 and 7, not argued with particularity.

THE OBVIOUSNESS REJECTION BASED ON OLSON, GARIMELLA, AND TRAVISON

Claims 10 and 11

Based on the record before us, we are not persuaded that the Examiner erred in rejecting claims 10 and 11 as unpatentable over Olson, Garimella, and Travison.

Appellants argue that Travison fails to teach "matching a minimum percentage of the current development task with the prior development task." App. Br. 29; *see also* Reply Br. 10. Specifically, Appellants allege that Travison only compares a percentage of a failure. *Id.* Because "a 'failure' is not comparable to a 'development task,'" Travison cannot be this cannot be combined with Olson and Garimella to teach the limitations of claim 10. *Id.*

The Examiner, however, finds that "Travinson's current failure and historical failures are comparable to a development task because current failure and historical failures pertains to software testing, software testing is a task done during the lifecycle of software development." Ans. 12. Moreover, as discussed above, Olson teaches comparing current and prior development tasks. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *See In re Keller*, 642 F.2d 413, 426 (CCPA 1981); *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986). As such, we are not persuaded of error in the Examiner's findings.

Accordingly, for the reasons discussed above and by the Examiner, we affirm the Examiner's rejection of claims 10 and 11 as unpatentable over Olson, Garimella, and Travison.

DECISION

We affirm the Examiner's decision to provisionally reject claims 1–11 on the ground of nonstatutory double patenting.

We affirm the Examiner's decision to reject claim 1 as directed to patent ineligible subject matter.

We affirm the Examiner's decision to reject claims 1–11 as unpatentable over the cited combinations of prior art.

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)(2016).

AFFIRMED