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.
13/746,723	01/22/2013	Majeed M. Arni	AUS920080669US2	4999

37945 7590 11/09/2018
DUKE W. YEE
YEE AND ASSOCIATES, P.C.
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER
BOSWELL, BETH V

ART UNIT	PAPER NUMBER
3623	

NOTIFICATION DATE	DELIVERY MODE
11/09/2018	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):

ptonotifs@yeeiplaw.com
mgamez@yeeiplaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte MAJEED M. ARNI, PETER A. COLDICOTT,
EDUARDO T. KAHAN, and MEI Y. SELVAGE

Appeal 2017-007077
Application 13/746,723
Technology Center 3600

Before JUSTIN BUSCH, JENNIFER L. McKEOWN, and CATHERINE SHIANG, *Administrative Patent Judges*.

BUSCH, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. § 134(a), Appellants appeal from the Examiner's decision to reject claims 1–22, which constitute all the claims pending in this application. We have jurisdiction over the pending claims under 35 U.S.C. § 6(b). We affirm.

CLAIMED SUBJECT MATTER

Appellants' claimed invention relates to computer programs and methods for managing events in a service oriented architecture (SOA) infrastructure. Abstract; Spec. ¶ 3. More specifically, the claims relate to enhancing "the integration of components in a configuration of SOA governance components" (i.e., generating, monitoring, and managing events generated by the components) using plug-ins to the components that produce events including automatically notifying interconnected components when one component generates an event (e.g., a component is added or changed). Spec. ¶¶ 7–9. Claims 1, 13, and 18 are independent claims. Claim 1 is reproduced below:

> 1. A method for managing maintenance of components in a service oriented architecture (SOA) infrastructure, the method comprising the steps of:
>
> a computer detecting a predetermined type of event resulting from one of an addition of a component to the SOA infrastructure or a change to a component of the SOA infrastructure, wherein a repository associated with the component provides a notice of the predetermined type of event associated with a service used by the component to other components in the SOA infrastructure;
>
> responsive to a detected event of the predetermined type of event, the computer determining for each other component of the SOA infrastructure whether a respective component requires an update based on the detected event by querying the each other component of the SOA infrastructure for a service state value of the service, to determine whether the each other component has been updated based on the detected event and the service state value, wherein the detected event is said to exist in an SOA component or the repository associated with the SOA component when the component is aware of or has been notified of the detected event, and has been changed or adapted in view of the detected event; and
>
> responsive to determining that at least one of the each other components has not been updated based on the detected event, the computer automatically updating by notifying the at least one of the each other components of the SOA infrastructure of the predetermined type of event associated with the service used by the respective component to implement a change or modification needed for the respective service or a respective relationship based on the detected event, wherein the at least one

> of the each other components of the SOA infrastructure is adapted to the detected event, transforming the at least one of the each other components respectively from a previous service state value to a new service state value.

REJECTIONS

Claims 1–22 stand rejected under 35 U.S.C. § 101 as being directed to judicially excepted subject matter. Final Act. 6–9.

Claims 1–22 stand rejected under 35 U.S.C. § 103 as obvious in view of Loewy (US 2004/0193703 A1; Sept. 30, 2004) and Zager (US 2002/022952 A1; Feb. 21, 2002). Final Act. 9–14.

ANALYSIS

We have reviewed the Examiner's rejections in light of Appellants' arguments that the Examiner erred. In reaching this decision, we have considered all evidence presented and all arguments Appellants made. Arguments Appellants could have made, but chose not to make in the Briefs, are deemed waived. *See* 37 C.F.R. § 41.37(c)(1)(iv).

THE 35 U.S.C. § 101 REJECTION

Step One of Alice Framework

In step one of the *Alice* analysis, we "determine whether the claims at issue are directed to" a patent-ineligible concept, such as an abstract idea. *Alice Corp. Pty. Ltd. v. CLS Bank International*, 134 S. Ct. 2347, 2354–55 (2014). The Examiner determines the claims are directed to the concept of "managing maintenance of components in a service oriented architecture (SOA) infrastructure," which is simply a method of organizing human activities and an idea of itself. Final Act. 6–7; Ans. 12–14 (citing *Electric Power Group, LLC v. Alstom S.A.*, 830 F.3d 1350 (Fed. Cir. 2016)). The Examiner explains that "the focus of the claims is not on such an improvement in computers as tools, but on certain independently abstract ideas that use computers as tools" because, like in *Electric Power*, Appellants' claims rely on generic, conventional computers and technology to implement the abstract idea. Final Act. 14; Ans. 13–14 (citing *Elec.* Power, 830 F.3d 1350). The Examiner finds each of the recited steps are part of the abstract idea. Final Act. 7; Ans. 12. The Examiner finds the dependent claims' method steps also are directed to the abstract idea because they merely include information regarding the services. Final Act. 7; Ans. 12.

Appellants assert the claims are not directed to an abstract idea. *See* App. Br. 9–11; Reply Br. 4–5. Specifically, Appellants argue their claims are unlike the claims held ineligible in *Electric Power* because Appellants' claims recite different elements than those cited in *Electric Power*. Reply Br. 4–5. Rather, Appellants argue the claims do not focus on using a computer in its ordinary capacity but are directed to an improvement to computer functionality and, in particular, a "specific improvement to the way computers operate using a three-step synergistic process." App. Br. 10–11 (citing *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1334 (Fed. Cir. 2016)).

Appellants argue the claims' "event detection" (i.e., the detecting step), "update determination" (i.e., the determining step), and the automatic updating step are not just "'gathering', analyzing', or 'displaying' of information." Reply Br. 4–5. Appellants argue these claimed elements are, respectively, "a specific event-detection resulting from an SOA infrastructure addition/change," "a specific event-detection triggering mechanism that invokes an event-based SOA component update determination with respect to other components of the SOA infrastructure," and "an improvement to the computer system itself . . . in that the other SOA components requiring updates per the detected event are adapted to the detected event by transforming [the components] from a previous service state value to a new service state value." Reply Br. 4–5. Appellants argue these distinctions render claim 1's alleged abstract idea "substantially different from the 'abstract idea'" in the claims in *Electric Power*. Reply Br. 5.

Appellants separately argue that each of the dependent claims "improves the operation of the underlying computer system" by providing an alleged benefit resulting from the additionally recited features. App. Br. 15–20. For example, claim 4 depends from claim 1 and recites the determining step "is based on the computer determining whether each other component has been notified of the detected event." Appellants argue claim 4 improves the operation of the underlying computer because it provides "a *front-end* event-notification determination *before* providing an automatic update notification to a particular component." App. Br. 16.

We agree with the Examiner's conclusions because Appellants' claims focus on managing components in an SOA infrastructure, which simply involves receiving, analyzing, and transmitting information. In particular, Appellants' claims recite a computer receiving information (notice of a predetermined type of event) from one device (repository), transmitting information (query/request for a service state value of a service) to other devices (each other component), receiving information an (service state value) from the other devices (each other component), analyzing the received information (service state values) to determine whether the other devices (each other component) has been updated, and in response to the determination, transmitting information (automatically updating *by notifying* a component of the predetermined type of event *to implement* a change or modification), and another device (the component) stores information (changes a service state from a previous value to a new value).

We are not persuaded by Appellants argument that the claims improve the way computers operate by "using a three-step synergistic process." *See* App. Br. 10–11. Essentially, Appellants' three-step synergistic process is simply the idea of automatically updating other components in response to detecting a change or addition of one component. *See* App. Br. 11.

Although the claims recite a "computer" to perform the detecting, determining, and updating steps, a "repository" from which notices are received, and "component[s] of the SOA infrastructure" from which information is received and to which information is transmitted, these elements do not change the character of the claims as a whole from being directed to an abstract idea. Similarly, to the extent Appellants' assertion that the limitations recited in the dependent claims improves the operation of the underlying computer is an argument that the dependent claims are not directed to abstract ideas, *see* App. Br. 15–20, we disagree. The limitations in the dependent claims (changing a name of a service in addition to a service state value, the determining step being based on particular data, each component having a storage queue and plug-in, generically automating a computer action corresponding to the detected event in response to a trigger, and automating email or task creation as part of the updating step) simply refine the particular data being received, analyzed, and transmitted (service name, query response, task or e-mail generated) or add generically recited elements (storage queue and plug-in). These limitations, therefore, do not change the character of Appellants' claims.

The Federal Circuit has concluded similar concepts were directed to abstract ideas. Specifically, in *Electric Power*, the Federal Circuit concluded claims reciting a method of collecting data from various sources, "detecting and analyzing events" by identifying information in the received data, reporting the event analysis results and visualizations of measurements, aggregating the event analysis information, and providing a composite indicator were directed to an abstract idea because the claims were directed to "collecting information, analyzing it, and displaying certain results of the collection and analysis." *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1351–53 (Fed. Cir. 2016).

Similarly, the Federal Circuit concluded claims directed to collecting and analyzing information and presenting the results were ineligible as claiming no more than an abstract idea. *Content Extraction & Transmission LLC v. Wells Fargo Bank, N.A.*, 776 F.3d 1343, 1347 (Fed. Cir. 2014); *see also Smart Sys. Innovations, LLC v. Chicago Transit Auth.*, 873 F.3d 1364, 1372 (Fed. Cir. 2017) (concluding "claims directed to the collection, storage, and recognition of data are directed to an abstract idea"); *SAP Am., Inc. v. InvestPIC, LLC*, 898 F.3d 1161, 1167 (Fed. Cir. 2018) ("merely presenting the results of abstract processes of collecting and analyzing information . . . is abstract as an ancillary part of such collection and analysis") (quotations omitted).

Moreover, limiting the particular data analyzed does not change the character of the claim. *See Elec. Power*, 830 F.3d at 1353 (stating that "collecting information, including when limited to particular content (which does not change its character as information)" is treated as "within the realm of abstract ideas"). Thus, the particular event type information or service state values transmitted between the repository, computer, and SOA components do not change the character of the claims.

We agree with the Examiner that the claims are directed to an abstract idea and, thus, turn to step 2 of the *Alice* analysis.

Step Two of Alice Framework

In step two of our *Alice* analysis, we determine whether the *additional* limitations, when considered both "individually and 'as an ordered combination'" contain an "inventive concept" sufficient to transform the claimed "abstract idea" into a patent-eligible application. *Alice*, 134 S. Ct. at 2355–58.

The Examiner finds "the computer, repository, computer program product, data storage device(s), and an article of manufacture" are the portions of the claims that are not the part of the abstract idea. Final Act. 8; Ans. 12. The Examiner concludes these elements are insufficient to add significantly more sufficient to render the claims patent eligible because they are generic computer components necessary to perform the claimed steps. Final Act. 8; Ans. 12. The Examiner determines the claims do not recite an improvement in computers, another technology, or another technical field; rather the alleged improvements are to the recited steps/functions in a process and not to any server or other device because the elements just limit the abstract idea to a particular technological environment. Final Act. 7; Ans. 13. The Examiner concludes the dependent claims do not add to the abstract idea but further refine the abstract idea because "they include, for instance, information/data regarding the SOA infrastructure data services." Final Act. 7; Ans. 12. The Examiner concludes the claims, therefore, do not amount to significantly more because the claims: require only a generic computer elements to perform generic, well-understood, routine, and conventional computer functions; do not recite an improvement to another technology or technical field; and are not rooted in computer technology. Final Act. 7–9; Ans. 12–14.

Appellants argue the claimed steps are not conventional or routine because the steps are novel and non-obvious and, therefore, add significantly more to the abstract idea. App. Br. 11–12, 14–15. In particular, Appellants argue the claims recite that a repository provides notice of a type of event to other SOA components, which is not a routine computer operation because it is a "novel use of a 'repository' associated with an added/changed component." App. Br. 11–12. Appellants also argue the Examiner's characterization of the recited "repository" as a generic computer component fails to consider the particularly recited usage of the repository. Reply Br. 3–4. Appellants also argue the particular aspects of the recited determining step (i.e., querying each other SOA component to determine whether those components have been updated based on the service state value and the detected event) are not routine or generic computer implemented functions. App. Br. 12.

Appellants also argue the claims add significantly more to the abstract idea because the claims improve the underlying computer itself. App. Br. 14 (citing *DDR Holdings, LLC v. Hotles.com, L.P.*, 773 F.3d 1245 (Fed. Cir. 2014). Specifically, Appellants argue it was previously "extremely *difficult* to add a new product or modify an existing service pertaining to a particular SOA governance flow." App. Br. 14 (emphasis added).

Appellants argue "[t]he claimed repository-based event-notification provides . . . integration facilitation by specially invoking/triggering" the claimed determining step, simplifying the integration of new or changed components into the SOA architecture. Reply Br. 3–4 ("Typically, such new product will have to interact with and be integrated to each of the SOA governance components that are already in the configuration. Accordingly, integration of the new product tends to become exponentially complicated."). Appellants argue the claimed computer-based solution is a specialized automatic update determination and, therefore, necessarily rooted in computer technology to overcome a specific problem arising in realm of computer networks. App. Br. 14. Appellants argue the "three-way synergistic interplay between the [three] claimed steps" is an inventive process solving the problem of integrating new products into an SOA infrastructure "that was previously *tedious and complex*." Reply Br. 5–6. Appellants argue the claims improve "the technical field of 'SOA infrastructure.'" Reply Br. 7–8

An inventive concept "cannot be furnished by the unpatentable law of nature (or natural phenomenon or abstract idea) itself." *Genetic Techs. Ltd. v. Merial L.L.C.*, 818 F.3d 1369, 1376 (Fed. Cir. 2016); *see also Alice*, 134 S. Ct. at 2355 (explaining that, after determining a claim is directed to a judicial exception, "we then ask, '*[w]hat else* is there in the claims before us?'" (emphasis added, brackets in original) (quoting *Mayo*, 566 U.S. at 78)). Instead, an "inventive concept" is furnished by an element or combination of elements that is recited in the claim *in addition to* the judicial exception and sufficient to ensure the claim as a whole amounts to significantly more than the judicial exception itself. *Alice*, 134 S. Ct. at

2355 (citing *Mayo*, 566 U.S. at 72–73); *see BSG Tech LLC v. BuySeasons, Inc.*, 899 F.3d 1281, 1290 (Fed. Cir. 2018) (explaining that the Supreme Court in *Alice* "only assessed whether the claim limitations *other than the invention's use of the ineligible concept* to which it was directed were well-understood, routine and conventional," (emphasis added)).

On the other hand, "[i]f a claim's only 'inventive concept' is the application of an abstract idea using conventional and well-understood techniques, the claim has not been transformed into a patent-eligible application of an abstract idea." *BSG Tech*, 899 F.3d at 1290–91 (citing *Berkheimer v. HP Inc.*, 881 F.3d 1360, 1370 (Fed. Cir. 2018)). "[I]t is irrelevant whether [the claimed abstract idea] may have been non-routine or unconventional as a factual matter . . . narrowing or reformulating an abstract idea does not add 'significantly more' to it." *BSG Tech*, 899 F.3d at 1291.

We agree with the Examiner that the three recited method steps (i.e., the detecting, determining, and updating steps) are part of the abstract idea. Therefore, the question is whether there are *additional* elements recited in the claims that add significantly more to the abstract idea. The claims recite a "computer," a "repository," "data storage devices," an "article of manufacture," and SOA components that perform the detecting, determining, and updating steps. However, as stated by the Examiner, these elements are generally and generically recited, and Appellants' Specification suggests that these are well-known elements being used as intended and may be comprised of software, hardware, or a combination of software and hardware. *See* Spec. ¶¶ 16–22, Figs. 1–2. In other words, Appellants' claims invoke computers merely as tools to receive, analyze, and transmit information in a particular way, thereby implementing the abstract idea of managing SOA components. *See BSG Tech*, 899 F.3d at 1286. Simply using generic computing devices is not "an improvement in computers as tools," like those claims found patent-eligible. *Elec. Power*, 830 F.3d at 1354.

Moreover, simply using generic computers to automate a process does not confer eligibility onto an otherwise abstract idea because it does not improve a computer or technology, but rather improves the process itself. *See Gottschalk v. Benson*, 409 U.S. 63, 67 (1972) (explaining that the claimed steps could easily "be carried out in existing computers long in use, no new machinery being necessary"). Appellants' arguments that the claims solve a technical problem arising in computer networks and improve the underlying computer or the technical field of SOA infrastructure are unpersuasive. *See* App. Br. 14; Reply Br. 3–8. Appellants arguments that the claims include significantly more because automatically updating SOA components when an SOA component is added or changed was "previously tedious and complex," "extremely difficult," and "complicated" and the claimed process "simplif[ies] the integration" are unpersuasive because using a computer to automate a process does not add significantly more to the abstract idea. *See Intellectual Ventures I LLC v. Capital One Bank (USA)*, 792 F.3d 1363, 1370 (Fed. Cir. 2015) ("[O]ur precedent is clear that merely adding computer functionality to increase the speed or efficiency of the process does not confer patent eligibility on an otherwise abstract idea.").

Appellants' argument that the repository (associated with the SOA component) providing the notice is a novel use of a repository also is unpersuasive. The claims do not require that the repository is a separate device or that the repository performs any function other than providing the notification to the computer; the repository is simply a label for a module, program, or device that provides notice to the recited computer. More importantly, the repository simply "provides a notice of the predetermined type of event." This function is part of the alleged abstract idea that involves transmitting and receiving information. We do not see how the fact that the repository transmits information, a generic computing function, adds significantly more to the abstract idea.

Appellants also describe how many of the limitations additionally recited in the dependent claims allegedly improve the recited process of managing the SOA components and assert the limitations are "not a routine/generic computer-performed step/action." App. Br. 15–20. Appellants argue each of the dependent claims, therefore, "improves the operation of the underlying computer." App. Br. 15–20.

As mentioned above with respect to step one of the *Alice* analysis, the limitations in the dependent claims (changing a name of a service in addition to a service state value, the determining step being based on particular data, each component having a storage queue and plug-in, generically automating a computer action corresponding to the detected event in response to a trigger, and automating email or task creation as part of the updating step) simply refine the particular data being received, analyzed, and transmitted (service name, query response, task or e-mail generated) or add generically recited elements (storage queue and plug-in). These limitations, therefore, are a part of the abstract idea and do not provide elements in *addition* to the abstract idea sufficient to transform the claims into eligible subject matter. *See Alice*, 134 S. Ct. at 2355.

Pre-emption

Appellants also argue the claims are patent-eligible because the claims do not preempt an abstract idea. App. Br. 12–14; Reply Br. 6–7 (citing *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2354 (2014)). Specifically, Appellants argue the claims do not preempt methods of managing SOA components not: invoked by a specified detected event, using service state values, or notifying a particular component to implement a change. App. Br. 12–13. Appellants essentially argue their claims do not preempt methods of managing SOA components as long as the methods do not use the same data as claimed.

We are not persuaded in this case. The claims limit the identified abstract idea only by limiting the content of the information received, analyzed, and transmitted. Moreover, although "preemption may signal patent ineligible subject matter, the absence of complete preemption does not demonstrate patent eligibility. . . . Where a patent's claims are deemed only to disclose patent ineligible subject matter under the *Mayo* framework, as they are in this case, preemption concerns are fully addressed and made moot." *Ariosa Diagnostics, Inc. v. Sequenom, Inc.*, 788 F.3d 1371, 1379 (Fed. Cir. 2015); *see also OIP Techs. v. Amazon.com, Inc.*, 788 F.3d 1359, 1362–63 (Fed. Cir. 2015) ("And [the fact] that the claims do not preempt all price optimization or may be limited to price optimization in the e-commerce setting do[es] not make them any less abstract.").

Thus, the claimed limitations, considered both individually and together, do not add significantly more to the abstract idea and, therefore, do not render the subject matter patent eligible.

Summary of Analysis of § 101 Rejection

For the above reasons, we determine the Examiner did not err in rejecting claims 1–22 under 35 U.S.C. § 101 as being directed to judicially excepted subject matter.

THE 35 U.S.C. § 103 REJECTION OF CLAIMS 1–22

The Examiner finds the combination of Loewy and Zager teaches or suggests every limitation recited in claims 1–22. Final Act. 9–14. Of particular note, the Examiner finds Zager teaches or suggests the determining step. Final Act. 11 (citing Zager ¶ 77, Abstract, Figs. 9, 28–29); Ans. 18, 20–22. The Examiner finds Zager's Abstract and paragraph 77 "clearly show a service oriented architecture (SOA) infrastructure and state level updates involved with a repository."

Appellants argue, among other things, Zager does not teach or suggest determining which of the other components needs to be updated based on the detected event and a queried service state value. App. Br. 25–28. In particular, Appellants argue Zager's cited portions merely describe, in response to a system model's state changing, identifying model objects likely to be affected by the change and displaying an alarm and the affected objects. App. Br. 27–28. Appellants argue identifying affected managed objects and displaying an alarm do not teach or suggest making any type of determination of whether particular components need to be updated, let alone making such a determination based on the service's server state value obtained from querying the components and the detected event. App. Br. 27–28.

Zager discloses automatically discovering and manually inputting information at runtime to prepare a model for a complex system that represents system resources and service-relationships among the resources. Zager, Abstract. Zager's software agents "report changes in state of the managed resources to the model, which updates itself" and identifies other resources that may be affected by the state change due to the resources' service-relationships. Zager ¶ 77, Abstract, Figs. 9, 28–29. Zager then generates an alarm that groups related resources, identifies any root cause, and displays the resources affected by the change. Zager ¶ 77, Abstract, Fig. 9.

We are persuaded the Examiner failed to demonstrate Zager teaches or suggests the determining step. In particular, the Zager's cited portions do not teach or suggest "querying the each other component of the SOA infrastructure for a service state value of the service." For purposes of this Decision, we assume Zager's managed objects teach or suggest the recited components of the SOA infrastructure. Zager teaches identifying service-relationships between managed objects. Zager ¶ 77, Abstract. Zager also teaches detecting a change in a managed object's state. Zager ¶ 77, Abstract. The Examiner, however, has not sufficiently explained, nor is it apparent from reviewing the cited portions of Zager, how identifying related managed objects teaches or suggests querying *other managed objects* for a service state value of the service, let alone using that returned value as a basis for determining whether the other managed object needs to be updated. Instead, Zager merely teaches identifying objects affected by a fault detected with respect to a related managed object.

For the above reasons, we are persuaded the Examiner erred in rejecting claim 1 as obvious in view of the combination of Loewy and Zager. Claims 2–12 ultimately depend from and incorporate the limitations of claim 1. Independent claims 13 and 18 recite a commensurate limitation to the determining step. Claims 14–17 and 19–22 ultimately depend from and incorporate the limitations of claims 13 and 18, respectively. Thus, for the same reasons, we are persuaded the Examiner erred in rejecting claims 2–22.

SUMMARY

We affirm the Examiner's decision to reject claims 1–22 under 35 U.S.C. § 101.

We reverse the Examiner's decision to reject claims 1–22 under 35 U.S.C. § 103.

DECISION

Because we affirm at least one ground of rejection with respect to each claim on appeal, the Examiner's decision to reject claims 1–22 is affirmed. *See* 37 C.F.R. § 41.50(a)(1).

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