



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.
15/063,521	03/08/2016	RAJESH KHAZANCHI	C421.03	6888
152606	7590	09/22/2020	EXAMINER	
Olympic Patent Works PLLC 4979 Admiral Street Gig Harbor, WA 98332			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	
			MAIL DATE	DELIVERY MODE
			09/22/2020	PAPER

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte RAJESH KHAZANCHI, SERVESH SINGH, KIRAN SINGH,
RISHI SARAF and AGILA GOVINDARAJU

Appeal 2019-003104
Application 15/063,521
Technology Center 2100

Before ST. JOHN COURTENAY III, LARRY J. HUME, and
PHILLIP A. BENNETT, *Administrative Patent Judges*.

BENNETT, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Pursuant to 35 U.S.C. § 134(a), Appellant¹ appeals from the Examiner's decision to reject claims 1–20. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

¹ We use the word “Appellant” to refer to “applicant” as defined in 37 C.F.R. § 1.42(a). Appellant identifies the real party in interest as VMWARE, Inc. Appeal Br. 1.

CLAIMED SUBJECT MATTER

The claims are directed to systems and methods for modularized automated application release management subsystems. Claim 1, reproduced below, is illustrative of the claimed subject matter:

1. An automated-application-release-management subsystem within a cloud-computing facility having multiple servers, data-storage devices, and one or more internal networks, the automated-application-release-management subsystem comprising:
 - a dashboard user interface;
 - an automated-application-release-management controller;
 - an interface to a workflow-execution engine within the cloud-computing facility;
 - an artifact-storage-and-management subsystem; and
 - a set of sets of descriptors, each descriptor including one or more sets of routine and function entrypoints that the automated-application-release-management controller calls in response to callbacks from the workflow-execution engine, each set of routine and/or function entrypoints describing entrypoints within an external module, library, or subsystem, and each descriptor corresponding to a type of task executed by the automated-application release-management subsystem.

Appeal Br. 48 (Claims App.).

REFERENCES

The prior art relied upon by the Examiner as evidence is:

Name	Reference	Date
Sites	US 5,649,203	July 15, 1997
McVeigh et al.	US 2007/0094248 A1	April 26, 2007
Muller	US 2008/0244579 A1	Oct. 2, 2008
Kiriansky et al.	US 2010/0011209 A1	Jan. 14, 2010

Prahlad et al.	US 2010/0332456 A1	Dec. 30, 2010
Muckenhuber et al.	US 2011/0078673 A1	Mar. 31, 2011
Issa et al.	US 7,930,201 B1	April 19, 2011
Krishnan et al.	US 2013/0325789 A1	Dec. 5, 2013
Dvinsky et al.	US 2015/0227362 A1	Aug. 13, 2015

REJECTIONS

Claims 1–20 stand rejected under 35 U.S.C. § 101 as being directed to patent-ineligible subject matter. Final Act. 2–3.

Claims 1 and 2 stand rejected under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, and Kiriansky. Final Act. 5.

Claims 3, 4, 8, and 9 stand rejected under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, and Issa. Final Act. 12.

Claims 5–7 and 10–20 stand rejected under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, Issa and Muckenhuber. Final Act. 15.

REJECTION UNDER 35 U.S.C. § 101

Standard for Patent Eligibility

In issues involving subject matter eligibility, our inquiry focuses on whether the claims satisfy the two-step test set forth by the Supreme Court in *Alice Corp. v. CLS Bank International*, 573 U.S. 208 (2014). The Court instructs us to “first determine whether the claims at issue are directed to a patent-ineligible concept,” *Id.* at 218, and, in this case, the inquiry centers on whether the claims are directed to an abstract idea. If the initial threshold is

met, we then move to the second step, in which we “consider the elements of each claim both individually and ‘as an ordered combination’ to determine whether the additional elements ‘transform the nature of the claim’ into a patent-eligible application.” *Id.* at 217–18 (quoting *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 566 U.S. 66, 79, 78 (2012)). The Court describes the second step as a search for “an ““inventive concept””—*i.e.*, an element or combination of elements that is ‘sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself.’” *Id.* (quoting *Mayo*, 566 U.S. at 72–73).

The USPTO has published revised guidance on the application of § 101 consistent with *Alice* and subsequent Federal Circuit decisions. USPTO, *2019 Revised Patent Subject Matter Eligibility Guidance*, 84 Fed. Reg. 50 (Jan. 7, 2019) (“Guidance”), updated by USPTO, *October 2019 Update: Subject Matter Eligibility* (available at https://www.uspto.gov/sites/default/files/documents/peg_oct_2019_update.pdf) (“October 2019 Guidance Update”).

Under the Guidance, we first look to whether the claim recites:

- (1) any judicial exceptions, including certain groupings of abstract ideas (*i.e.*, mathematical concepts, certain methods of organizing human activity such as a fundamental economic practice, or mental processes) (referred to Step 2A, prong 1 in the Guidance); and
- (2) additional elements that integrate the judicial exception into a practical application (*see* MPEP § 2106.05(a)–(c), (e)–(h)) (referred to Step 2A, prong 2 in the Guidance).

See Guidance, 84 Fed. Reg. at 52–55. Only if a claim (1) recites a judicial exception and (2) does not integrate that exception into a practical

application, do we then move to Step 2B of the Guidance. There, we look to whether the claim:

(3) adds a specific limitation beyond the judicial exception that is not “well-understood, routine, conventional” in the field (*see* MPEP § 2106.05(d)); or

(4) simply appends well-understood, routine, conventional activities previously known to the industry, specified at a high level of generality, to the judicial exception.

See Guidance, 84 Fed. Reg. at 56.

Examiner’s Findings and Conclusions

The Examiner rejects claims 1–20 as being directed to a judicial exception without significantly more. Final Act. 2–4; Ans. 2–7. Under Step 2A, prong 1, the Examiner determines claim 1 is directed to non-statutory subject matter. Ans. 3. The Examiner determines claims 10 and 19 recite, “respectively a method and computer-stored instructions for the action steps of: *storing, incorporating, accessing and calling*. Per step 2A analysis, the steps as recited when construed from the claim context do not direct interpretation that these amount significantly more than a human sequences of generic actions interactively carried out using a computer, which is typical of a[n] Abstract Idea, defined by the courts.” *Id.*

Appellant’s Contentions

Appellant argues:

The Examiner cites only 35 U.S.C. §101 as authority for the rejections, but 35 U.S.C. §101 does not mention software or hardware, and most certainly does not stand for the proposition that software is unpatentable or that hardware is required for patentability. An automated-application-release-management subsystem within a cloud computing facility it is clearly both a machine and a manufacture and is clearly patentable subject matter under 35 U.S.C. §101.

Appeal Br. 19. Appellant further adds “the Examiner has failed to state a *prima facie* 35 U.S.C. § 101 claim rejection.” *Id.*

Step 2A, Prong 1—The Judicial Exception

Applying the Guidance, we are persuaded the Examiner has erred in rejecting claim 1 as being directed to patent-ineligible subject matter. The Guidance instructs us first to determine whether any judicial exception to patent eligibility is recited in the claim. The Guidance identifies three judicially-expected groupings of abstract ideas: (1) mathematical concepts, (2) certain methods of organizing human activity such as fundamental economic practices and commercial interactions (including . . . advertising, marketing or sales activities or behaviors; business relations), and (3) mental processes. *See* Guidance, 84 Fed. Reg. at 52, 54-55.

We agree with Appellant that the claims before us do not recite an abstract idea within these groupings. The limitations of claim 1 recite a series of computer elements that function as an automated application release management controller. We discern no mathematical concepts recited therein, nor does the Examiner find any such operations. We also do not agree with the Examiner that the claims recite a commercial interaction. The claims make no mention of any commercial practice or business-related concept. Rather, the claims exclusively recite system components performing application management functions. Nor do we find the functions in claim 1 recite any limitations that may be performed practically as a mental process. The claimed steps are not of the sort to be practically performed by a human in their mind, as each relate to how an application release management subsystem is rendered.

Having determined that claims 1, 10, and 19 do not recite a judicial exception, we conclude our analysis. . *See* Guidance, 84 Fed. Reg. at 54 (“If the claim does not recite a judicial exception (a law of nature, natural phenomenon, or subject matter within the enumerated groupings of abstract ideas in Section I), then the claim is eligible at Prong One of revised Step 2A. This concludes the eligibility analysis, except in the rare circumstance described below.”). We, therefore, do not sustain the rejection of claim 1, nor of independent claims 10 and 19 and the remaining dependent claims.

REJECTIONS UNDER 35 U.S.C. § 103

The Examiner’s Findings and Conclusion of Obviousness

The Examiner rejects claim 1 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, and Kiriansky. Final Act. 5–11. The Examiner finds that Muller generally teaches the preamble and the first four limitations in the body of the claim. Final Act. 5–6. The Examiner notes that although Muller teaches an automatic application management subsystem, it does not explicitly disclose that the application management subsystem is an “application *release* management subsystem,” nor does it expressly teach that the system is “within a cloud-computing facility.” Final Act. 6. The Examiner addresses these deficiencies relying on Krishnan and Prahlad, finding that these references demonstrate that it was known to provide management of applications within cloud computing networks. Final Act. 6 (citing Krishnan ¶¶ 27, 34, 39, 43, 54, 93, 131; Prahlad ¶¶ 157, 158, 276–281, 307).

The Examiner further finds that Muller does not teach the limitation:
a set of sets of descriptors, each descriptor including one or more sets of routine and function entrypoints that the

automated-application-release-management controller calls in response to callbacks from the workflow-execution engine, each set of routine and/or function entrypoints describing entrypoints within an external module, library, or subsystem, and each descriptor corresponding to a type of task executed by the automated-application release-management subsystem.

Final Act. 8. The Examiner finds limitation obvious in view of the combined teachings of Prahlad and Dvinsky, which teach the use of callbacks in support of management of assets or storage resources. Final Act. 9 (finding that the references show that “a callback mechanism being associated with pointer information correlating an entity to be invoked responsive to receiving a callback is a well-recognized practice.”). The Examiner also relies on Sites and Kiriansky, which teach the configuration of entry points of external functions (Sites) and the invocation of external functions via callbacks (Kiriansky).

The Examiner reasons that a person of ordinary skill in the art would have been motivated to combine the pertinent teachings of the references:

It would have been obvious at the time of the effective date of the claimed invention for one of ordinary skill in the art to implement callback in Muller so that information input supporting the callback mechanism or layer in monitoring the recorded entry points associated with the original code would include sets of descriptors (as per Sites), each descriptor including one or more sets of routine and function entry-points which (per teachings of Krishnan, Prahlad, Dvinsky, McVeigh) Muller's automated-application management controller calls in response to callbacks, being a well-known mechanism and that each set of routine and/or function entrypoints underlying a intercepted callback would be entry-points within an external module (as in Sites), library (as in Kiriansky), or subsystem (as in Muller or Prahlad); because configuring callback interception with declared pointer information pre-established by this well-known mechanism enables external code identified by the

pointer information to be enlisted for code transfer or execution control switch, as that would preclude code from being invoked by the original code (at a recorded entry points as in Sites original source code analyzer), the redirection (using this callback intercepting at specific original code locations) to external module or functions, subsystem procedure, such as library per the protection approach by Kiriansky, and also because structure input supporting the mapped entry points and pointer redirection per this callback approach can be transferrable as input file having descriptor information that bears the same pointer information identifying a given external function entry points at which the execution switch, or control transfer (as in Sites) is to execute – function transfer as in Sites - in place of the original call sites being set by the callback mechanism or declaration at their respective and predetermined entry point; i.e. enabling callback to be highly flexible with this type of manageable text form (with user friendly descriptor format provision of text file) into a runtime callback monitoring underlying a secure code execution as in Kiriansky.

Final Act. 10–11; *see also* Ans. 18–19 (providing similar reasoning).

Appellant’s Argument

No “automated-application-release management” in Prior Art

Appellant offers lengthy arguments against the rejection, but the central contention Appellant makes is that none of the references “is in any way, related to automated-application-release-management subsystems or to management of application releases in a cloud-computing environment.”² Appeal Br. 29–36. In support, Appellant cites various passages from the Specification describing an example of an “automated-application-release-management subsystem within a cloud-computing facility.” Appeal Br. 24–29. With respect to Muller, Appellant argues “Muller appears to be directed

² Appellant argues the rejections under § 103 as a group, with claim 1 being representative. Our decision with respect to claim 1, therefore, is dispositive as to the remaining claims.

to a system that allows a user to create virtual machines within some type of computing facility.” Appeal Br. 29. Similarly, Appellant reproduces a passage from Krishnan and argues “Krishnan discusses a document-control system, an extract-transform-load framework, and various other types of services, but nowhere mentions anything related to management of application releases within a cloud-computing facility.” Appeal Br. 29.

We are not persuaded the Examiner erred in determining Muller and Krishnan teach or suggest an “automated-application-release-management subsystem” as recited in claim 1. Appellant’s argument intimates that the meaning of “automated-application-release-management” is narrowly confined to the exemplary embodiment provided in the Specification. However, aside from quoting passages from the Specification, Appellant does not offer any proposed definition for the phrase. Rather, Appellant’s position appears to be that Muller and Krishnan do not teach systems having all of the characteristics described in Appellant’s Specification, and therefore cannot teach an “automated-application-release-management subsystem,” as recited in claim 1.

In the Answer, the Examiner provides a detailed response (Ans. 7–21) to Appellant’s argument, explaining that under its broadest reasonable interpretation, an automated-application-release-management subsystem is “a subsystem of a larger cloud-based infrastructure, including aspects of application workflow provisioning and resources management” (Ans. 12), and further explains the basis for concluding the recited “automated-application-release-management subsystem” is obvious over the cited references:

[T]he claimed language (AARMS) [automated-application-release-management subsystem] in its entirety in this case is perceived – via BRI [broadest reasonable interpretation] – as a subsystem (with no expressed dependency to a larger system that bears . . . relationship to [a] cloud-computing facility (see Prahlad, Krishnan, Dvinsky) and that includes automated application, management and administering of application resources (see Muller) such as release control (see McVeigh).

Ans. 13. Thus, the Examiner explains that under its broadest reasonable interpretation, an “automated-application-release-management subsystem” is a subsystem that automates application management, including application distribution.

Appellant does not challenge the Examiner’s interpretation of “automated-application-release-management subsystem,” nor does Appellant provide any arguments countering the Examiner’s additional explanation and proffered reasoning. Further, and as we noted above, Appellant does not offer its own definition for this phrase. In the Reply Brief, Appellant merely states:

The Examiner’s assertions and commentary with respect to the Examiner’s 35 U.S.C. § 103 rejections, which apparently begin on page 8 of the Examiner’s Answer, make even less sense to Appellants’ representative that those made with respect to the 35 U.S.C. § 101 rejections. Rather than spend many pages of text based on speculations and assumptions with regard to what the Examiner may have meant by many of these assertions and commentary, Appellants instead continue rely on the arguments made in the Appeal Brief.

Reply Br. 7. Thus, the additional explanation and reasoning set forth in the Answer, including the Examiner’s interpretation of “automated-application-release-management subsystem” stands unchallenged in this record, and Appellant has not sufficiently explained why the Examiner’s interpretation

of “automated-application-release-management subsystem,” and the applicability of the cited references is in error.

Appellant’s arguments in the Appeal Brief are all premised on an unduly narrow interpretation of “automated-application-release-management subsystem,” and the arguments distinguish the prior art based on that narrow interpretation. Because we agree with the Examiner that the scope of “automated-application-release-management subsystem” is not limited to the embodiments described in the Specification, Appellant’s arguments are not commensurate with the broadest reasonable interpretation of the argued limitation.

*Appellant’s Argument
No Motivation to Combine References*

Appellant also contends the Examiner’s rationale for combining the references is insufficient:

Appellants’ representative cannot even parse the statement, let alone discern anything in the statement that would suggest any way to combine these disparate references or any reason for one skilled in the art to even consider the use of the disparate references together, let alone to try to combine them. The Examiner’s attempt to motivate the combination of unrelated references falls short.

Appeal Br. 38.

However, we are not persuaded by Appellant’s argument because the Examiner articulates reasoning for why it would have been obvious to combine the teachings of the references:

It would have been obvious at the time of the effective date of the claimed invention for one of ordinary skill in the art to implement callback in Muller so that information input supporting the callback mechanism or layer in monitoring the recorded entry points associated with the original code would

include sets of descriptors (as per Sites), each descriptor including one or more sets of routine and function entry-points which (per teachings of Krishnan, Prahlad, Dvinsky, McVeigh) Muller's automated-application management controller calls in response to callbacks, being a well-known mechanism and that each set of routine and/or function entrypoints underlying a intercepted callback would be entry-points within an external module (as in Sites), library (as in Kiriansky), or subsystem (as in Muller or Prahlad); because configuring callback interception with declared pointer information pre-established by this well-known mechanism enables external code identified by the pointer information to be enlisted for code transfer or execution control switch, as that would preclude code from being invoked by the original code (at a recorded entry points as in Sites original source code analyzer), the redirection (using this callback intercepting at specific original code locations) to external module or functions, subsystem procedure, such as library per the protection approach by Kiriansky, and also because structure input supporting the mapped entry points and pointer redirection per this callback approach can be transferrable as input file having descriptor information that bears the same pointer information identifying a given external function entry points at which the execution switch, or control transfer (as in Sites) is to execute – function transfer as in Sites - in place of the original call sites being set by the callback mechanism or declaration at their respective and predetermined entry point; i.e. enabling callback to be highly flexible with this type of manageable text form (with user friendly descriptor format provision of text file) into a runtime callback monitoring underlying a secure code execution as in Kiriansky.

Final Act. 10–11, Ans. 18–19.

The standard for determining whether a claim is obvious is “an expansive and flexible approach.” *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415 (2007). Here, Appellant’s position appears to be that this reasoning is unintelligible and therefore cannot support an obviousness rejection. However, aside from asserting that it cannot be understood, Appellant does

not address the reasoning in a substantive way. That is, Appellant does not explain *why* the proffered reasoning is deficient; Appellant merely argues that the reasoning “simply makes no sense from either a technical or computational standpoint.” Appeal Br. 41. However, we find the proffered reasoning to be reasonable and sufficiently grounded in evidence drawn from the record to justify the combination. Specifically, we understand the Examiner’s reasoning as a finding that combining the teachings of the references is within the level of skill in the art and involves no more than combining known technologies, as described in the various references, to achieve predictable results. *See KSR*, 550 U.S. at 417. As such, Appellant’s arguments are not persuasive of error.

Because we do not find Appellant’s arguments persuasive of Examiner error, we sustain the rejection of claim 1 under 35 U.S.C. § 103. We similarly sustain the rejection of independent claims 10 and 19 rejected under 35 U.S.C. § 103. We also sustain the obviousness rejections of dependent claims 2–9, 11–19, and 20, not argued separately.

CONCLUSION

We reverse the rejections of claims 1–20 under 35 U.S.C. § 101 as being directed to patent-ineligible subject matter.

We affirm the rejection of claims 1 and 2 under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, and Kiriansky.

We affirm the rejection of claims 3, 4, 8, and 9 under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, and Issa.

We affirm the rejection of claims 5–7 and 10–20 under 35 U.S.C. § 103 as being unpatentable over Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, Issa, and Muckenhuber.

Because we have affirmed at least one ground of rejection for each claim on appeal, we affirm the Examiner's decision to reject the claims. 37 C.F.R. § 41.50(a)(1).

DECISION SUMMARY

Claims Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1–20	101	Eligibility		1–20
1, 2	103	Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky	1, 2	
3, 4, 8, 9	103	Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, Issa	3, 4, 8, 9	
5–7, 10–20	103	Muller, Krishnan, Prahlad, Dvinsky, McVeigh, Sites, Kiriansky, Issa, Muckenhuber	5–7, 10–20	
Overall Outcome			1–20	

TIME PERIOD FOR RESPONSE

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).

AFFIRMED