



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.
12/948,635	11/17/2010	Chun E. Ma	CN920090089US1	4228
75949	7590	12/09/2016	EXAMINER	
IBM CORPORATION C/O: Fabian Vancott 215 South State Street Suite 1200 Salt Lake City, UT 84111			PYO, MONICA M	
			ART UNIT	PAPER NUMBER
			2161	
			NOTIFICATION DATE	DELIVERY MODE
			12/09/2016	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):

patents@fabianvancott.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte CHUN E. MA, XIN SHENG MAO, LI YI, and JUN ZHANG

Appeal 2016-000916¹
Application 12/948,635
Technology Center 2100

Before ALLEN R. MacDONALD, JOHN P. PINKERTON, and
GARTH D. BAER, *Administrative Patent Judges*.

BAER, *Administrative Patent Judge*.

DECISION ON APPEAL

¹ Appellants identify International Business Machines Corporation as the real party in interest. Appeal Br. 2.

STATEMENT OF THE CASE

This is a decision on appeal, under 35 U.S.C. § 134(a), from the Examiner's final rejection of claims 1–3 and 5–25, which are all the pending claims. Appeal Br. 10. Claim 4 has been cancelled. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm-in-part.

BACKGROUND

A. The Invention

Appellants' invention is directed to “a method, system and computer program product for creating service mashup instances based on user exploration procedure.” Spec. ¶ 22. Claims 1, 2, 9, and 24 are representative and reproduced below, with emphasis added to the disputed elements:

1. A method for creating a service mashup instance, comprising:

recording at least two services being selected by a user of a computing device during an exploration procedure;

obtaining a relationship between the at least two services;
and

generating the service mashup instance on the computing device *based on the relationship;*

wherein obtaining the relationship between the at least two services comprises determining whether, based on the exploration procedure, the at least two services have a direct connection in a navigation path; and, if so, obtaining the relationship from the direct connection in the navigation path.

2. The method of claim 1, in which obtaining a relationship between the at least two services comprises *retrieving the relationship from a service repository*.

9. An apparatus for creating a service mashup instance, comprising a computing device comprising:

a browser that records at least two services being selected by a user during an exploration procedure;

an analysis component that obtains a relationship between the at least two services; and

a generation component that generates the service mashup instance *based on the relationship*.

24. A computer program product for creating a service mashup instance, the computer program product comprising:

a computer usable storage memory comprising computer usable program code embodied therewith, the computer usable program code comprising:

computer usable program code configured to record at least two services being selected by a user during an exploration procedure;

computer usable program code configured to obtain a relationship between the at least two services by analyzing metadata of the at least two services and deriving the relationship from the metadata; and

computer usable program code configured to generate the service mashup instance *based on the relationship*.

Appeal Br. 24–25, 28–29 (Claims App.).

B. The Rejections on Appeal

The Examiner rejects claims 9, 10, 17, and 21 under 35 U.S.C. § 102(b) as anticipated by Nathan (US 2008/0222599 A1; Sept. 11, 2008).

Final Act. 2.

The Examiner rejects claims 1–3, 5–8, 11–16, 18–20, and 22–25 under 35 U.S.C. § 103(a) as unpatentable over Nathan, in view of Lu (Bin Lu et al., “sMash: Semantic-based Mashup Navigation for Data API Network,” in *WWW 2009*, April 20–24, 2009, Madrid, Spain). Final Act. 4.

ANALYSIS

A. Claims 9 and 11–23

Appellants argue Nathan does not explicitly or inherently disclose “a browser that records at least two services being selected by a user during an exploration procedure,” as recited in independent claim 9. *See* Appeal Br. 12; *see also* Reply Br. 4–7. Appellants further argue that Nathan does not disclose “an analysis component that obtains a relationship between the at least two services,” as recited in claim 9. *See* Appeal Br. 13; *see also* Reply Br. 6–7. Because Nathan does not disclose a device or component that provides a relationship, Appellants argue, Nathan also does not disclose a generation component that generates the service mashup instance “based on the relationship,” as recited in claim 9. *See* Appeal Br. 13; *see also* Reply Br. 6–7.

We do not find these arguments persuasive. We agree with the Examiner that Nathan’s visual mashup designer application teaches the claimed “browser.” *See* Final Act. 2 (citing Nathan ¶¶ 15, 20). Appellants’ argument that Nathan’s visual mashup designer application is not a browser is not persuasive as Appellants’ specification indicates that the claimed “browser” need not be a traditional web page browser and may be a special browser for web service browsing or searching. *See* Spec. ¶ 43. As Nathan teaches the visual mashup design application creates a wrapper for a selected

web service or web page, the visual mashup design application is configured to allow a user to browse or search for web services, and thus, Nathan teaches the claimed “browser.”² We also agree with the Examiner that Nathan’s marking of a user’s web service selections for creating a mashup application teaches the claimed “[recording] at least two services being selected by a user during an exploration procedure.” *See* Ans. 11 (citing Nathan ¶ 22). Appellants’ argument that Nathan’s marking of web service selections requires manual selections of web services that Appellants’ claimed invention seeks to avoid is not commensurate with the scope of claim 9, as claim 9 fails to explicitly prohibit such manual steps, and thus, is not persuasive.

Further, we agree with the Examiner that Nathan teaches the system creates wrappers for selected web services (i.e., mashup components), receives input from a user to map output of a first mashup component to input of a second mashup component, and maps a relationship between the first mashup component and the second mashup component via the created wrappers, resulting in a generated mashup application. *See* Ans. 11; *see also* Nathan ¶¶ 21–22. Because Nathan teaches the system obtaining a mapping that defines a relationship between the two selected web services from the user and generating a mashup application based on the obtained mapping, we agree with the Examiner that Nathan teaches “[obtaining] a relationship between the at least two services,” and “[generating] the service mashup instance based on the relationship.” Similar to Appellants’ arguments

² Further, Nathan’s Figure 8 clearly illustrates a screen generated by the visual mashup designer application that is displayed within a browser, with the title “PROJECT NAME - BROWSER NAME.” *See* Nathan, Fig. 8.

regarding Nathan's marking, Appellants' argument that Nathan's analyzing and determining a data relationship and service accessibility between two selected web services requires manual steps that Appellants' claimed invention seeks to avoid is not commensurate with the scope of claim 9, as claim 9 fails to explicitly prohibit such manual steps, and thus, is not persuasive.

Accordingly, we sustain the Examiner's rejection of independent claims 9, 17, and 21. We further sustain the rejection of claims 11–16, 18–20, and 22–23, not argued separately.

B. Claim 10

Appellants argue Nathan fails to disclose an analysis component that “retrieves the relationship from a service repository associated with the analysis component,” as recited in claim 10, because Nathan describes retrieving data from a web service or website, as opposed to a service repository, and because the data being retrieved in Nathan is not a “relationship” between at least two services. *See* Appeal Br. 16; *see also* Reply Br. 9.

We find this argument persuasive. We agree with Appellants that Nathan teaches receiving web services (i.e., mashup components) from a user via a drag and drop interface, and further receiving input from the user to map output of a first mashup component to input of a second mashup component. *See* Nathan ¶¶ 22, 26. As Nathan teaches the system receiving input that describes a relationship between at least two web services from a user, Nathan fails to teach retrieving the relationship from a service

repository, as claim 10 requires. Accordingly, we do not sustain the Examiner's rejection of claim 10.

C. Claims 1, 3, and 5–8

Appellants argue Nathan fails to disclose “recording at least two services being selected by a user of a computing device during an exploration procedure,” “obtaining a relationship between the at least two services,” and generating the service mashup instance on the computing device “based on the relationship,” as recited in independent claim 1 for the same reasons Nathan fails to disclose similar claim limitations recited in claim 9. *See* Appeal Br. 17; *see also* Reply Br. 10. Appellants additionally argue Lu fails to cure any of Nathan's deficiencies. *See* Appeal Br. 17; *see also* Reply Br. 10. Appellants further argue that Nathan and Lu, whether considered individually or in combination, fail to disclose or suggest “obtaining the relationship between the at least two services comprises determining whether, based on the exploration procedure, the at least two services have a direct connection in a navigation path; and, if so, obtaining the relationship from the direct connection in the navigation path,” as also recited in claim 1. *See* Appeal Br. 17–19.

We do not find these arguments persuasive. We agree with the Examiner that Nathan teaches “recording at least two services being selected,” “obtaining a relationship between the at least two services,” and “generating the service mashup instance . . . based on the relationship,” for the reasons previously discussed with respect to claim 9. We agree with the Examiner that the combination of Nathan and Lu also teaches “determining whether, based on the exploration procedure, the at least two services have a

direct connection in a navigation path; and, if so, obtaining the relationship from the direct connection in the navigation path.” *See* Final Act. 4–5; *see also* Ans. 13–14. More specifically, we agree with the Examiner that Nathan teaches determining whether, based on an exploration procedure, at least two services have a connection. *See* Final Act 4. We also agree with the Examiner that Lu teaches at least two services having a direct connection in a navigation path and obtaining a relationship between the at least two services from the direct connection in the navigation path. *See id.* at 4–5. Appellants’ specification broadly defines “navigation path” as “any data structure that stores services browsed or searched by a user and related information, such as metadata of services.” Spec. ¶ 38. Lu teaches a data API (application programming interface) network constructed and visualized around matched APIs generated from a “fuzzy-match-keyword-search.” *See* Lu, 1133. Since Lu’s data API network is a data structure that stores APIs (i.e., web services) resulting from a user keyword search as well as API-related information, we agree with the Examiner that Lu’s API network teaches the claimed “navigation path.” *See* Ans. 13. Further, Lu teaches a navigation of the data API network via an automatic link of “mashupable APIs” and a detailed mashup candidate recommendation. *See* Lu, 1133. Since Lu teaches identifying whether at least two APIs are “mashupable” (i.e., have a connection within the data API network) via a navigation of the data API network, we agree with the Examiner that Lu also teaches “determining whether . . . the at least two services have a direct connection in a navigation path; and, if so, obtaining the relationship from the direct connection in the navigation path.” *See* Ans. 13. Thus, we agree with the

Examiner that the combination of Nathan and Lu teaches or suggests all of the limitations of claim 1.

Accordingly, we sustain the Examiner's rejection of independent claim 1. We further sustain the rejection of claims 3 and 5–8, not argued separately.

D. Claim 2

Appellants argue Nathan fails to disclose “retrieving the relationship from a service repository,” as recited in claim 2, for the same reasons Nathan fails to disclose an analysis component that “retrieves the relationship from a service repository associated with the analysis component,” as recited in claim 10. *See* Appeal Br. 23; *see also* Reply Br. 12. We agree that Nathan fails to disclose the claimed “retrieving” for the reasons previously discussed with respect to claim 10. Further, the Examiner has failed to establish, on this record, that Lu cures Nathan's deficiency. Accordingly, we do not sustain the Examiner's rejection of claim 2.

E. Claims 24–25

Appellants argue Nathan fails to disclose computer usable program code configured to “record at least two services being selected by a user during an exploration procedure,” “obtain a relationship between the at least two services,” and “generate the service mashup instance based on the relationship,” as recited in independent claim 24 for the same reasons Nathan fails to disclose similar claim limitations recited in claim 9. *See* Appeal Br. 20; *see also* Reply Br. 12. Appellants additionally argue Lu fails to cure any of Nathan's deficiencies. *See* Appeal Br. 20; *see also* Reply

Br. 12. Appellants further argue that Nathan and Lu, whether considered individually or in combination, fail to disclose or suggest “computer usable program code configured to obtain a relationship between the at least two services by analyzing metadata of the at least two services and deriving the relationship from the metadata,” as also recited in claim 24. *See* Appeal Br. 21.

We do not find these arguments persuasive. We agree with the Examiner that Nathan teaches the claimed computer usable program code configured to “record at least two services,” “obtain a relationship between the at least two services,” and “generate the service mashup instance based on the relationship,” for the reasons previously discussed with respect to claim 9. We further agree with the Examiner that the combination of Nathan and Lu teaches the claimed computer usable program code configured to “obtain a relationship between the at least two services by analyzing metadata of the at least two services and deriving the relationship from the metadata.” *See* Final Act. 9–10; *see also* Ans. 14–15. More specifically, we agree with the Examiner that Nathan teaches analyzing a relationship between at least two services and deriving the relationship from parameter information. *See* Final Act 9. We also agree with the Examiner that Lu teaches describing metadata of APIs of the data API network. *See id.* at 9–10. In combining Nathan and Lu, the Examiner’s substitution of Nathan’s parameter information with Lu’s metadata is a mere substitution of one known element for another, yielding predictable results. Such a combination would have been obvious to one of ordinary skill in the relevant art. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007) (“[W]hen a patent claims a structure already known in the prior art that is altered by the mere

substitution of one element for another known in the field, the combination must do more than yield a predictable result.”). Thus, we agree with the Examiner that the combination of Nathan and Lu teaches or suggests all of the limitations of claim 24.

Accordingly, we sustain the Examiner’s rejection of independent claim 24. We further sustain the rejection of claim 25, not argued separately.

DECISION

We affirm the Examiner’s rejection of claims 1, 3, 5–9, and 11–25.

We reverse the Examiner’s rejection of claims 2 and 10.

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-IN-PART