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.
12/550,225	08/28/2009	Edward Solovey	B1056/ADBS.209200	1484

121363 7590 11/18/2016

Shook, Hardy & Bacon L.L.P.
(Adobe Systems Incorporated)
Intellectual Property Department
2555 Grand Blvd
Kansas City, MO 64108

EXAMINER	
BORROMEO, JUANITO C	

ART UNIT	PAPER NUMBER
2184	

NOTIFICATION DATE	DELIVERY MODE
11/18/2016	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):

IPDOCKET@SHB.COM
IPRCDKT@SHB.COM
kspringer@shb.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte EDWARD SOLOVEY and BASIL C. HOSMER

Appeal 2016-000615
Application 12/550,225
Technology Center 2100

Before MARC S. HOFF, STEPHEN C. SIU, and JAMES W. DEJMEK, *Administrative Patent Judges*.

SIU, *Administrative Patent Judge*

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the Examiner's Final Rejection of claims 1–19. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

The disclosed invention relates generally to generating a manifestation of a model. Spec. ¶ 1. Independent claim 1 reads as follows:

> 1. A computer-implemented method comprising:
> receiving data comprising a model defining one or more model entities and one or more model services; and
> generating script-based code associated with the model by:

> processing the model to generate for each model entity defined in the model a value object class having support for basic data operations and advanced data operations, the basic data operations including both getter and setter functions for data properties associated with a model entity defined by the model, and the advanced data operations including derived properties, varying properties, constraints, or styles, wherein the styles comprise a collection of user-interface-related attributions associated with elements of the value object class; and
>
> processing the model to generate for each model service a service wrapper object class, the service wrapper object class to provide runtime access to a remote service, the remote service being a web service, a remote object service, or an HTTP service.

The Examiner rejects claims 1–19 under 35 U.S.C. § 103(a) as unpatentable over Gaudette (US 7,958,454 B2; issued June 7, 2011), Scheibli (EP 1486849 A1; published December 15, 2004), and Ortal (US 2007/0209031 A1; published September 6, 2007).

ISSUE

Did the Examiner err in rejecting claims 1–19?

ANALYSIS

Appellants argue that the combination of Gaudette, Scheibli, and Ortal fails to teach or suggest a service wrapper that "provides runtime access to a remote service (e.g., a web service, remote object service, or an HTTP service)." App. Br. 7. We are not persuaded by Appellants' arguments for at least the reasons set forth by the Examiner. *See* Final Act. 2–4, 8; *see also* Ans. 2–3.

For example, Gaudette teaches that "the computing device . . . may include a network interface . . . to interface to a Local Area Network (LAN), Wide Area Network (WAN) or the Internet" and "interfacing the computing device . . . to any type of network capable of communication and performing . . . operations." Gaudette 7:53–55, 7:65 – 8:1. Hence, Gaudette teaches a component that provides runtime access to a remote service (i.e., network capable of performing operations), such as a web service (e.g., a network interface that interfaces the computing device to the Internet). Also, as the Examiner points out, one of skill in the art would have understood that a "service wrapper" is software that "starts executing the . . . service." Scheibli 5:5–6. Combining the known feature of interfacing a computing device with a network (i.e., provide runtime access to a remote service), such as the Internet (i.e., a web service, remote object service, or an HTTP service) with a software component, as disclosed by Gaudette, with the known feature that a "service wrapper" is a known software component that executes services, as disclosed by Scheibli, would have resulted in the mere predictable result of interfacing a computing device with a network, such as a web service (and using a software component), to perform operations (i.e., execute services) with a known software component (that one of skill in the art would have understood to include a "service wrapper") performing known and expected functions of performing operations. Such an expected result would have been obvious to one of ordinary skill in the art. "The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 416 (2007).

Appellants argue that Scheibli fails to disclose or suggest that the service wrapper "is generated by processing a model containing a model service." App. Br. 7. Claim 1 recites data comprising a model defining model entities and services. In other words, claim 1 recites that a model, model entity, and model service, are data. Claim 1 also recites a "service wrapper" that is "generated by processing a model." Hence, claim 1 requires that a "service wrapper" is generated by processing data (a "model" being "data"). We agree with the Examiner that it would have been obvious to one of ordinary skill in the art that software that "executes services" (i.e., a "service wrapper" as disclosed by Scheibli) would be generated by using data. Indeed, Appellants do not explain sufficiently how one of ordinary skill in the art would *not* have utilized data to generate software or that utilizing data to generate software for executing services would have been "uniquely challenging or difficult for one of ordinary skill in the art." *See Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418).

Claim 1, for example, recites processing the model (i.e., processing data) to generate "a value object class" (i.e., data) that supports "advanced data operations," the "advanced data operations" potentially including "styles," wherein the styles comprise a collection of user-interface-related attributions associated with elements of the value object class. In other words, claim 1 recites processing "data" to generate "other data" that potentially supports a "collection of user-interface-related attributions" associated with the "other data." Appellants argue that the combination of Gaudette and Ortal fails to disclose or suggest "that the attribute added by the user is a user-interface-related attribution that is associated with an

element of a value object class generated based on a model entity" or "a model entity . . . is processed to generate a value object class having support for the advanced data operation." App. Br. 8. We are not persuaded by Appellants' argument.

First, as the Examiner indicates, claim 1 does not recite that advanced data operations must include "styles," styles comprising user-interface-related attributions. *See* Ans. 4. Therefore, we are not persuaded by Appellants' contention that the combination of Gaudette and Ortal fails to disclose user-interface-related attributions (i.e., "styles") associated with elements of the value object class, even assuming Appellants' characterization of Gaudette and Ortal to be correct.

In any event, even assuming that claim 1 recites that user-interface-related attributions associated with elements of the value object class are required, we are still not persuaded by Appellants' argument. For example, Gaudette discloses a "GUI tool" for creating a graphical user interface. Gaudette 8:28. The GUI tool of Gaudette is used to "create, modify, and save a graphical user interface" and "can provide a layout editor for placing and arranging user interface elements" or "provide an environment for programming functionality and other portions of the graphical user interface." Gaudette 8:36–43. In other words, Gaudette discloses a "GUI tool" that processes data (e.g., a "model") to generate data (e.g., "value object class") representing a user interface, the generated data supporting "styles" (of the GUI) wherein the "styles" comprise a collection of user-interface-related attributions (i.e., arrangement and attributes of the GUI such as "panels, buttons, text fields" or "programming functionality") each of which are associated with the data representing the user interface (i.e.,

"value object class"). As the Examiner points out, Ortal discloses a "model" associated with an "attribute." Final Act. 4 (citing Ortal ¶ 46). We agree with the Examiner that one of skill in the art would have understood, based at least on Ortal, that a "model" (or data) was known to have associated "attributes." We also agree with the Examiner that combining the known process of processing data to generate data representing "styles" of a user interface, the "styles" including attributes of the user interface (Gaudette) with the known feature that "models" (or data representing a user interface, for example) are associated with "attributes" (Ortal) would have resulted in the mere predictable result of performing a known function of Gaudette to generate data representative of a user interface with desired functions.

Appellants do not provide additional arguments in support of claims 2–18. *See* App. Br. 9.

SUMMARY

We affirm the Examiner's rejection of claims 1–19 under 35 U.S.C. § 103(a) as unpatentable over Gaudette, Scheibli, and Ortal.

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