



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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO. Includes fields for EXAMINER (LIAO, JASON G), ART UNIT (2156), PAPER NUMBER, NOTIFICATION DATE (03/27/2015), and DELIVERY MODE (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):

PTOMAIL@nixonvan.com
pair\_nixon@firsttofile.com

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

*Ex parte* DRAGOMIR NIKOLOV, ALEXANDER PANKOV, and  
JOCHEN WIEDMANN

---

Appeal 2013-000274  
Application 12/314,706  
Technology Center 2100

---

Before JEAN R. HOMERE, CATHERINE SHIANG, and  
JESSICA C. KAISER, *Administrative Patent Judges*.

KAISER, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's Final Rejection of claims 1–13. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

*Exemplary Claim*

Exemplary claim 1 under appeal reads as follows:

1. A method for manipulating at least one JAXR registry object that is associated with a SOA registry by an application,

the at least one JAXR registry object being accessible through a Java API for XML registries (JAXR), the method comprising:

defining a mapping between at least one JAXR registry object type and at least one application object type, wherein the mapping is based on at least one Java annotation that is associated with the at least one application object type;

instantiating the at least one application object type to provide the application with at least one application object;

parsing the at least one Java annotation;

associating the at least one application object to at least one JAXR registry object of the at least one JAXR registry object type, the association between the at least one application object and the JAXR object formed based on the parsed at least one Java annotation; and

manipulating the at least one application object through the application, the association between the at least one application object and the at least one JAXR registry object causing a related manipulation of the at least one JAXR registry object through the JAXR based on the defined mapping, the related manipulation interfacing with the SOA registry.

### *Rejections on Appeal*

The Examiner rejected claims 1 and 13 under 35 U.S.C. § 112, ¶ 2 as being indefinite. (Final Act. 5.)

The Examiner rejected claims 1, 2, 4–6, 8–10, 12, and 13 under 35 U.S.C. § 103(a) as being unpatentable over the combination of JAXR,<sup>1</sup> JavaAnnotations2,<sup>2</sup> and Chitre.<sup>3</sup> (Final Act. 6.)

---

<sup>1</sup> Java™ API for XML Registries (JAXR), Version 1.0, Proposed Final Draft (Apr. 10, 2002). The Examiner cited additional references as evidence of inherent properties and intended use of JAXR. (Final Act. 6.)

<sup>2</sup> Annotations, The Java™ Tutorials, <https://web.archive.org/web/20090831141542/http://java.sun.com/docs/books/tutorial/java/javaOO/annotations.html> (last visited Mar. 6, 2015).

The Examiner rejected claims 3, 7, and 11 under 35 U.S.C. § 103(a) as being unpatentable over the combination of JAXR, JavaAnnotations2, Chitre, and additional prior art references. (Final Act. 17, 19.)

### ISSUES

The dispositive issues raised by Appellants' contentions are:

Did the Examiner err in finding claims 1 and 13 indefinite for using the term "Java"?

Did the Examiner err in finding that the combination of JAXR, JavaAnnotations2, and Chitre teaches or suggests "the association between the at least one application object and the JAXR object formed based on the parsed at least one Java annotation," as recited in claim 1?

### ANALYSIS

#### INDEFINITENESS REJECTION OF CLAIMS 1 AND 13

The Examiner finds claims 1 and 13 indefinite for using the trademark "Java" to identify a programming language. (Final Act. 5.) The Examiner cites to the decision in *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982), for the proposition that "[w]here a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph." (*Id.*)

Appellants contend the use of Java in the claims does not render the claims indefinite per se; instead "[a] basic question to be resolved is whether

---

<sup>3</sup> Chitre et al., US 7,899,917 B2; issued Mar. 1, 2011.

the use of ‘Java’ in the context of these claims and this application [is] indefinite, such that a person of ordinary skill in the art could not discern the metes and bounds of the claimed subject matter.” (App. Br. 8–10.)

Appellants further contend “Java is a popular programming language that was originally developed by Sun Microsystems in the early 1990s and released in 1995,” and thus “a person of ordinary skill in the art would understand what is meant by ‘Java annotation’ and . . . ‘Java API for XML registries.’” (*Id.* at 11–12.)

We agree with Appellants. This case is closely analogous to *Ex Parte O’Farrell*, Appeal No. 2011-011075, 2014 WL 2112325, at \*4 (PTAB May 19, 2014) (non-precedential). The *O’Farrell* panel held that the recitation of the trademark “Java” was not indefinite because “‘Java’ is a well-known and widely-used term identifying a particular programming language.” *Id.* Similarly, we find one of ordinary skill in the relevant art would have understood “Java” in the context of the “Java annotation” and “Java API for XML registries” to refer to the programming language known by that name at the time of Appellants’ invention. Accordingly, we reverse the rejection of claims 1 and 13 under § 112, second paragraph.

#### OBVIOUSNESS REJECTION OF CLAIMS 1, 2, 4–6, 8–10, 12, AND 13

The Examiner rejected claim 1 under 35 U.S.C. § 103(a) based on the combination of JAXR, JavaAnnotations2, and Chitre. (Final Act. 5.)

Appellants contend that the Examiner erred because the combination fails to teach or suggest “the association between the at least one application object and the JAXR object formed based on the parsed at least one Java annotation,” as recited in claim 1. (App. Br. 18–20.) The Examiner finds

that JavaAnnotations2 teaches “in Java, annotation processing may play multiple roles including those of a preprocessor or runtime processing” and that the `@override` annotation “indicates overriding elements in a superclass.” (Final Act. 8–9.) The Examiner further finds “[t]he fact that a particular method of association may be chosen for compilation rather than another method means that that particular method may execute in the future. If that method deals with the association of objects, then the Java Annotation caused that association.” (Ans. 9.)

Appellants argue “the `@override` command does not play a part in how objects are associated.” (Reply Br. 9; *see also* App. Br. 20.) We agree with Appellants that the JavaAnnotations2 reference does not teach the argued limitation.

JavaAnnotations2 teaches `@Override` is used to “inform[] the compiler that the element is meant to override an element declared in a superclass,” and “[i]f a method marked with `@Override` fails to correctly override a method in one of its superclasses, the compiler generates an error.” (JavaAnnotations2 at 3.) The Examiner finds that the method chosen over another method using `@override` could deal with an association of objects. (Ans. 9.) The Examiner, however, has not identified any teaching or suggestion to use the `@override` annotation in that way (i.e., forming an association between objects based on a parsed `@override` annotation). Because the Examiner has not adequately explained that finding to support an obviousness conclusion, we reverse the Examiner’s § 103 rejection of claim 1.

Claim 13 recites a similar limitation as claim 1, and claims 2, 4–6, 8–10, and 12 depend from claim 1. Accordingly, we do not sustain the Examiner’s § 103 rejection of those claims for the same reasons as claim 1.

OBVIOUSNESS REJECTION OF CLAIMS 3, 7, AND 11

Claims 3, 7, and 11 depend directly or indirectly from claim 1. The Examiner rejected those claims under 35 U.S.C. § 103(a) as being unpatentable over the combination of JAXR, JavaAnnotations2, Chitre, and additional prior art references. (Final Act. 17, 19.) We note that the Examiner does not rely on the additional references for the disputed claim limitation and thus does not cure the deficiency discussed *supra* with respect to claim 1. Accordingly, we do not sustain the Examiner’s § 103 rejections of claims 3, 7, and 11 for the same reasons as claim 1.

DECISION

We reverse the Examiner’s rejections of claims 1–13.

REVERSED

kme