

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

*Ex parte* KIT CHATSINCHAI, BINO GEORGE, KISHAN THOMAS,  
KUI GONG, and BRAIN M. BUESKER

---

Appeal 2007-4418  
Application 10/119,618  
Technology Center 2100

---

Decided: June 4, 2008

---

Before JEAN R. HOMERE, JAY P. LUCAS  
and CAROLYN D. THOMAS, *Administrative Patent Judges*.

LUCAS, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF CASE

Appellants appeal from a final rejection of claims 1 through 24, and 28 under authority of 35 U.S.C. § 134. The Board of Patent Appeals and Interferences (BPAI) has jurisdiction under 35 U.S.C. § 6(b).

Appellants' invention relates to a system and method for simulating user inputs to a user interface, as typing and mouse movement, for any application in which user input would be required, such as testing a program.

In the words of the Appellants:

Accordingly, Appellant has invented and disclosed a method and system under which an event queue receives events that are sent from a user interface and an event queue to which said events are sent from said user interface and a RobotCreator tool also receives the events as those events are submitted to said event queue. The RobotCreator tool automatically converts the events into robot commands to that the robot commands then form a macro which can substitute for user input events made using said user interface during execution of an application. (Appellants' specification, paragraph 0007 and 0018).

(Br., p. 4, middle).

Claim 1 is exemplary:

1. A workstation comprising a system for generating a set of robot commands based on user entry events in a user interface, said system comprising:

a Java language operating environment comprising a java.awt.Robot class that provides robot commands that correspond to particular user input events;

an event queue to which said events are set from said user interface; and

a RobotCreator tool for receiving said events as those events are submitted to said event queue, wherein said RobotCreator tool converts said events into robot commands using said java.awt.Robot class;

wherein said robot commands then form a macro which can substitute for user input events made using said user interface during execution of an application.

In rejecting the claims on appeal, the Examiner relies upon the following prior art:

Steven, John et al. “jRapture: A Capture/Replay Tool for Observation-Based Testing”, 2000, ISSTA ’00, ACM, pp. 158-167.

“The American Heritage College Dictionary”, 4<sup>th</sup> edition, pg. 544.

“Microsoft Computer Dictionary”, 5<sup>th</sup> edition, pg. 322.

The Examiner rejects the claims on appeal as follows:

Claims 1 to 24 and 28 stand rejected under 35 U.S.C. § 102(b) for being anticipated by jRapture.<sup>1,2</sup>

Appellants contend that the claimed subject matter is not anticipated by jRapture for failure of the reference to teach claimed limitations, as will be discussed below. The Examiner contends that each of the claims is properly rejected.

We reverse.

---

<sup>1</sup> It is noted that a rejection of claims 1, 11, 21, 23, and 24 under 35 U.S.C. § 112, ¶ 1 has been withdrawn. (Ans., p. 2, bottom).

<sup>2</sup> It is further noted that the record indicates that the change to claim 11 noted by Appellants (App. Br., p. 3; Reply Br., p. 2) has indeed been entered, as indicated in the Office communication dated 6/1/2007.

## ISSUE

The issue is whether Appellants have shown that the Examiner erred in rejecting the claims under 35 U.S.C. § 102(b). The issue turns on whether jRapture teaches the RobotCreator tool as claimed.

## FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence.

1. Appellants have invented a method and system to input commands to a computer program that simulate the actions of a user using that program, especially with regard to the user typing data and making mouse movements to control the program's user interface. (Spec, ¶ 19). These input commands, called "events", are stored in an event queue prior to execution. The present invention modifies the event queue using a Java postEvent Method that exists in the java.awt.EventQueue class (in the Java Runtime Environment (JRE)) to intercept these events and send them to be translated into robot commands. (Spec, ¶ 22, 23). The purpose of the robot commands is to simulate the user input commands, events, to facilitate testing, for preparing demos or interactive tutorials. (Spec, ¶ 74).

2. The robot commands are a standard class of commands in the JAVA language, `java.awt.Robot`, which became part of the JAVA language with Java Development Kit (JDK) 1.3. The commands correspond to keystrokes on a keyboard, or movements of a mouse. (Spec, ¶ 4). They are difficult to write by hand, so this program provides a RobotCreator, (Spec, Fig 1a, b, or c, Item #100) a device that will capture the events as described just above and create the standard robot commands for the user automatically. (Spec, ¶ 25). The robot commands may be played over and over again, for example to test a program, or for other purposes. (Spec, ¶ 39).

*jRapture*

3. jRapture teaches a Capture/Replay Tool that captures “interactions between a Java program and the system, including GUI, file, and console inputs ... and on replay it presents each thread with exactly the same input sequence it saw during capture”. (P. 158, col 1, top). It is useful for testing software, as it captures the actions of beta users so they can be replayed and reviewed later by trained testing personnel to validate software behavior. (P. 158, col. 2, middle). The Java program is captured by a modified Java API. Then the system uses the captured modified Java API class to construct a system interaction sequence (SIS), which represents the inputs to the Java program together with auxiliary information necessary for correct playback. (P. 159, col. 2, middle). The jRapture SIS is created by an SISBuilder, which receives the thread of executed statements and inputs, stores them, and prepares them for replay

as needed. (P. 162, col. 1, middle). jRapture stores and replays the entire interaction of the inputs with the computer, not just the inputs. The replaying of the interactions may be performed on another computer entirely, which may have different capabilities than the captured platform. (P. 159, col 1, top).

4. The jRapture document holds a copyright of 2000. It states, “We are exploring the use of a new Java API called Robot, which is currently under development, for addressing the aforementioned problems with visualization during replay. It should be noted, however, that handling inputs to the native window manager poses problems that cannot be addressed completely without writing platform specific code to provide inputs to the window manager. The current implementation of jRapture does not include such code.”

#### PRINCIPLES OF LAW

Appellants have the burden on appeal to the Board to demonstrate error in the Examiner’s position. See *In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006)

Both anticipation under 35 U.S.C. § 102 and obviousness under § 103 are two-step inquiries, in which the first step is a proper construction of the claims and the second step requires a comparison of the properly construed claim to the prior art. *Medichem S.A. v. Rolabo S.L.*, 353 F.3d 928, 933 (Fed. Cir. 2003).

It is axiomatic that anticipation of a claim under § 102 can be found only if the prior art reference discloses every element of the claim. See *In re King*, 801 F.2d 1324, 1326 (Fed. Cir. 1986) and *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1457 (Fed. Cir. 1984).

## ANALYSIS

From our review of the administrative record, we find that the Examiner has presented a prima facie case for the rejections of Appellants' claims under 35 U.S.C. § 102. The prima facie case is presented on pages 3 to 7 of the Examiner's Answer.

In opposition, Appellants argue that “jRapture further does not teach or suggest the claimed operating environment that includes a Robot class as claimed.” (Br., p. 11, middle). In further explanation, Appellants explain, “Thus, jRapture fails to teach or suggest the claimed “RobotCreator tool for receiving said events as those events are submitted to said event queue, wherein said RobotCreator tool converts said events into robot commands using said java.awt.Robot class”. (Br., p. 12, middle) We agree, primarily because the reference does not teach a Java environment with the Robot class commands, java.awt.Robot, as claimed. We note in further support of Appellants' assertion that the reference specifically states that as of the time of publication of the reference, the Robot commands were not finalized and therefore were not incorporated in the jRapture system. (See FF4). We find that this is an important omission in the anticipation of the claims.

Appeal 2007-4418  
Application 10/119,618

Therefore, it follows that Appellants have shown that the Examiner erred in finding that jRapture anticipates claims 1 through 24, and 28.

#### CONCLUSION OF LAW

Based on the findings of facts and analysis above, we conclude that the Examiner erred in rejecting claims 1 through 24 and 28.

#### DECISION

The Examiner's rejection of claims 1 to 24 and 28 is reversed.

REVERSED

pgc

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins CO 80527-2400