**History of GUI in Java with AWI And Swing**

Danhong Li

Lewis University

**History of GUI in Java with AWI And Swing**

The graphical user interface is the window through which the user interacts with the program, which is more intuitive and easier to operate than the command-line interface. "Java's original GUI library was the Abstract Window Toolkit (AWT)."(Deitel). The swing was added to the platform after AWT, compatible with AWT, and improved AWT at the same time. Let's talk about two different technologies that can be used to create a GUI which are AWT and Swing.

AWT is a basic tool provided by Java to create and set up a Java graphical user interface. AWT is provided by the java.awt package in Java, which contains many classes that can be used to build a platform-independent graphical user interface (GUI). These classes are called components. AWT is a Java platform-independent window system, graphics, and user interface device toolkit. AWT is a part of Java Foundation Class (JFC), which provides a standard API for the Graphical User Interface (GUI) for Java programs. Because Java is a platform-independent programming language, but GUI is often dependent on a specific platform, Java uses the corresponding technology to enable AWT to provide applications with an interface independent of the machine platform, which ensures the same program The GUI has a similar appearance when running on different machines.

Abstract Window Toolkit is a set of tools provided by API to build graphical user interface GUI for Java programs. AWT can be used in Java applets and applications. It supports graphical user interface programming functions including user interface components; event processing model; graphics and image tools, including shapes, colors, and fonts; layout manager, which can carry out flexible window layout and distinguish the size and screen of specific windows Rate has nothing to do; data transfer type, you can cut and paste through the clipboard of the local platform.

However, when Java was launched, AWT was criticized as the weakest component of Java. The most fundamental disadvantage is that AWT only provides a very thin abstraction layer on top of the native user interface. In the second edition of the Java development kit, the AWT device was largely replaced by the Swing toolkit. Swing avoids the drawbacks of AWT by drawing the device itself. Swing calls the low-level routines in the local graphics subsystem instead of relying on the high-level user interface modules of the operating system. The emergence of Swing announced the end of AWT, and there is almost no application of AWT on the GUI.

Java Swing is a part of Java Foundation Classes (JFC). It is an attempt to solve the shortcomings of AWT. SWING solves many shortcomings of AWT. Compared to AWT, Swing is a lightweight component. SWING provides many screen display elements that are better than AWT. They are written in pure Java, so they can run across platforms like Java itself; they are part of JFC. They support the replaceable look and feel and themes. However, Swing does not really use the equipment provided by the native platform but merely imitates them on the surface. This means that you can use any look and feel supported by JAVA on any platform. The disadvantage of lightweight components is slower execution speed. The advantage is that uniform behavior can be adopted on all platforms.

As the first GUI class library in Java, AWT has achieved a breakthrough in Java GUI development from scratch. Although it performed poorly afterwards, it has basically withdrawn from the stage of history, but its historical achievements cannot be obliterated. Swing is part of the Java standard, and many improvements have been made on the shoulders of AWT, but it is too complicated and inefficient. It is criticized and not accepted by programmers.

**Works Cited**

“Abstract Window Toolkit.” *Wikipedia*, Wikimedia Foundation, 21 Aug. 2020, en.wikipedia.org/wiki/Abstract\_Window\_Toolkit.

“The Abstract Windowing Toolkit Group.” *AWT Group*, openjdk.java.net/groups/awt/.

Deitel, Paul, and Harvey Deitel. “Java™ How To Program (Early Objects), Tenth Edition.” *O'Reilly Online Learning*, Pearson, www.oreilly.com/library/view/javatm-how-to/9780133813036/ch25lev2sec1.html.

Kernighan , Brian W. “Is Java Slow.” *Wiki.c2.Com*, wiki.c2.com/?IsJavaSlow.

Siegel, Jerrold. *Java.applet and Java.awt*, www.umsl.edu/~siegelj/newcourse/part4/apawt.htm.

Singh, Chaitanya. “Java AWT Tutorial for Beginners.” *Beginnersbook.com*, 4 June 2015, beginnersbook.com/2015/06/java-awt-tutorial/.