Event-Based Programming

David Nandigam


KINECT
for XBOX 360

WHAT IS EVENT-BASED PROGRAMMING?

Event-Based Programming (EBP) is a technique that leverages events and notifications as the primary means to interconnect the main parts of a software system.

WHAT BENEFITS DOES IT BRING?

It reduces or eliminates coupling, yielding simpler parts that can often be developed and tested in isolation from the rest of the system. The bigger the system, the greater the benefits.

HOW ARE EVENT-BASED SYSTEMS DIFFERENT FROM EVENT-DRIVEN SYSTEMS?

An **event-based system** uses events and notifications *internally* to propagate commands and changes throughout the system. An **event-driven system** responds to *externally* generated events. The internal structure of an event-driven system may have little or nothing to do with events and notifications. Applications that run under a GUI operating system (e.g. Windows), are event-driven: they are essentially idle until an event occurs, like a mouse click on a key press. In response to the event, the application does something and then goes idle again, waiting for the next event.

DO I NEED SPECIAL PROGRAMMING LANGUAGES OR TOOLS TO DEVELOP AN EVENT-BASED SYSTEM?

No. All the popular programming languages, including C#, Java, C++, ObjectPascal and others can be used. Although event-based programs tend to be developed in object-oriented languages, older languages such as C or even assembly languages can be used. Fundamentally, the language must support the notion of pointer or reference.

ARE EVENT-BASED PROGRAMS BIGGER THAN TRADITIONAL ONES?

Absolutely. As proof, there is a complex event-based application for Amtrak that runs on a Windows Mobile 6 handheld device. The program interacts heavily with hardware systems, such as Bluetooth devices, laser barcode scanners, WiFi, 3G cellular modems and magnetic stripe readers.