



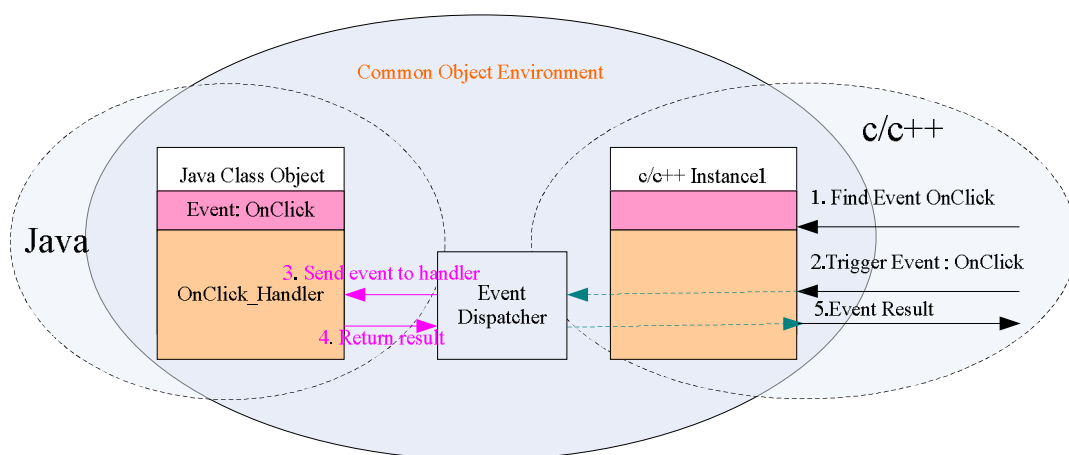
# **Multi-Language Programming : Common Object's Event**

Object may generate or handle events. This feature should continue to be supported in common object environment. Event is not an entity. It is a set of information. If the event information is described by special language, then it can not be triggered or handled by other languages. Therefore, environment should present a common model to describe events and provide a dispatcher to schedule these events.

In the environment, event may be identified by UUID or name. UUID uniquely corresponds to an event. Different languages can use interface of environment to find the event defined, and then trigger the event. Event may also have parameters or have response results from the handler. These parameters should be converted by environment when passed to different languages.

Common object environment should support to define object's event and event handler. Object's event may be handled by itself. For example, we click button, trigger a click event, then the button's click function is called to process the event. Object's event may also be handled by other objects. These objects can register handler function to capture events interested. These two kinds of event handling method should be supported.

Using UUID to identify an event enables events defined in one language to be found or triggered by other defined languages. For events, environment is responsible for parameter conversion between different languages, and dispatching event to objects which may be defined by different languages.



In above figure, Click event and its handler are defined for class object of java. If a c/c++ module needs to trigger the event, it finds the event first. Then it triggers “OnClick” event. Event dispatcher is responsible for routing the event to correct handler, which will call OnClick\_Handler of java. The handler returns results. The dispatcher translates results to c/c++ language format, and then return the results to c/c++ module.

Summary: common object environment supports defining object attributes, functions, and events. It also supports function overloading, event dispatching, and parameters/arguments conversion between different languages. It is an object-oriented runtime environment.