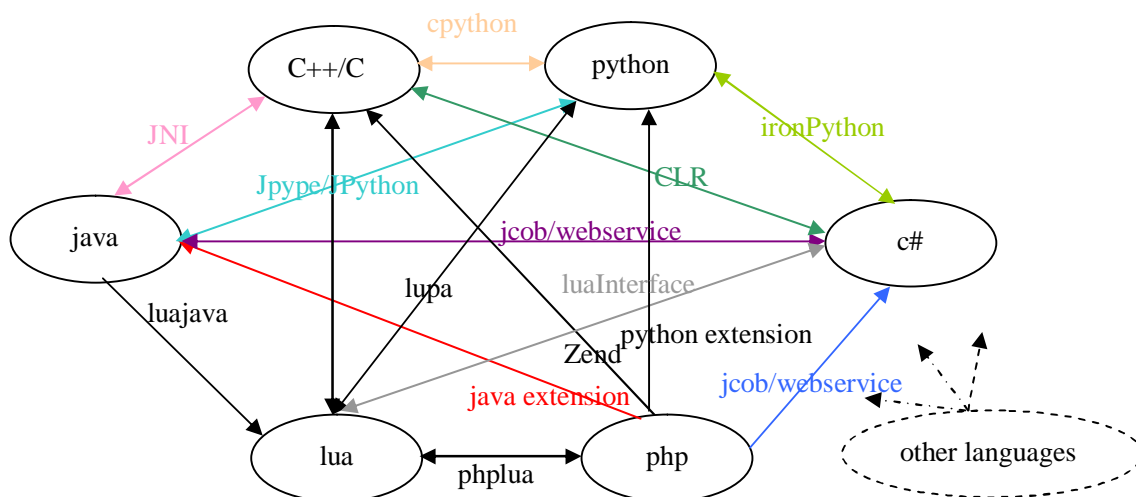




Multi-Language Programming : Object As Medium

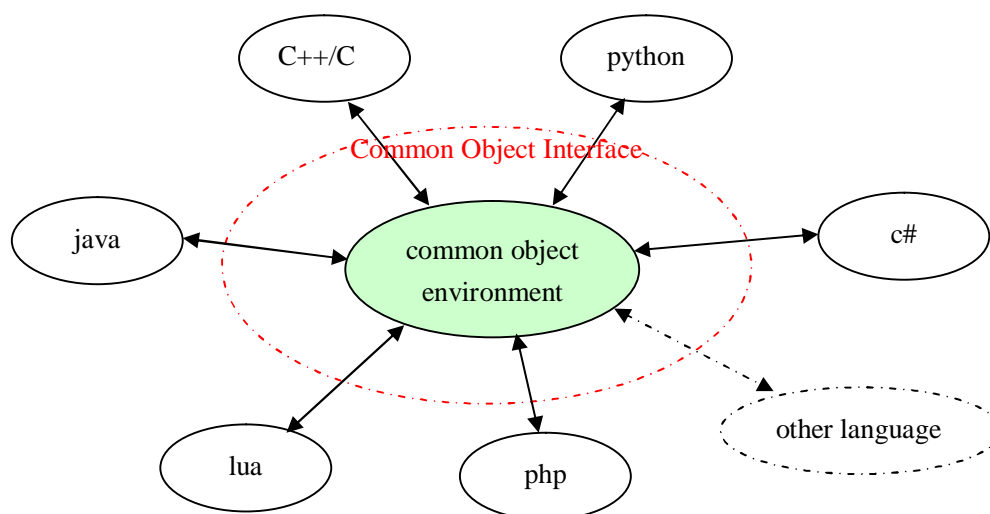
Software developers may encounter the problem to develop applications with multiple languages, maybe for efficiency, simplicity, marketing time, or for reusing libraries developed with other programming languages. There are many methods can be used, for example, if you want application mixed with java, you may use jni interface; mixed with python, you may use cpython interface; mixed with c# may use CLR, and so on. Developers can also use some libraries or middle-wares to simplify programming, such as using ironPython for c# interacting with python, jacob/webservice for java/php calling c#, Jpype/Python for python calling java, luajava for lua calling java, lupa for python calling lua, luaInterface for lua call c#, phplua for php calling lua, and so on. Shown with a figure as follows:



It can be seen from above figure, libraries or middle-wares for multi-language programming are very chaotic. Development will be complicated if three or more different languages are used in one application. Is there a more general way for multi-language programming?

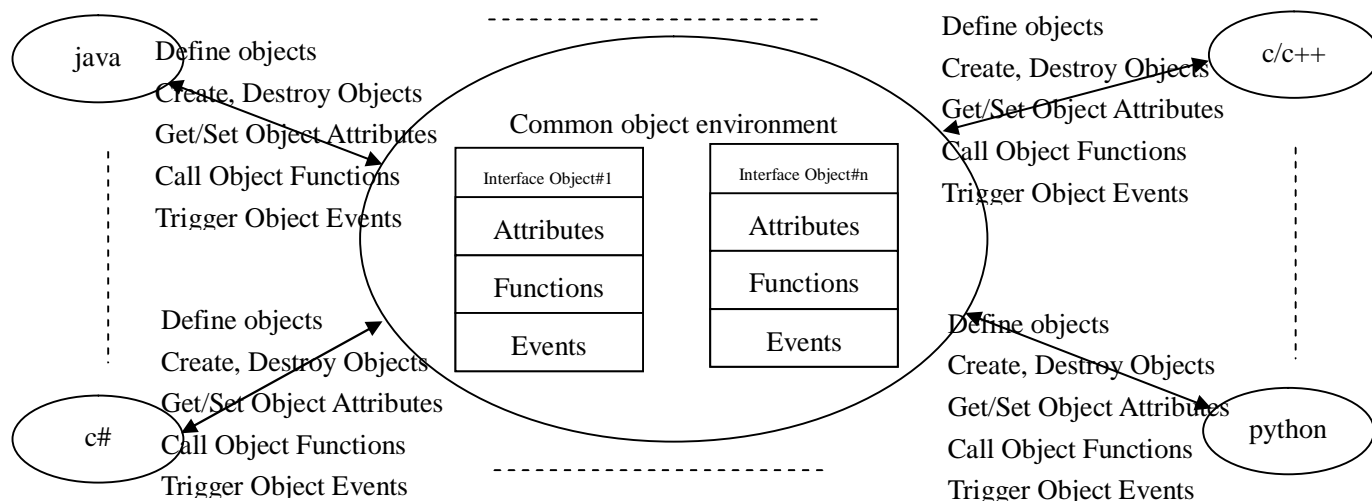
For any language, no matter c++/c, java, python, or c#, the concept of common is the object: python object, java object, c/c++ object, c# object etc. These objects locate within the scope of special language, and are opaque to other languages. Thus, if application needs to access objects defined in other languages, it can only through the low-layer of the interface calls presented by the corresponding languages.

No matter what language, objects have similar features. They have attributes, functions, may trigger events and process events. For objects, their behavior is same, no matter we use which language to describe them. If we construct an environment to manage the objects which will be accessed by other languages, and special programming language is only used to describe these objects, we might find a common and simple method to realize multi-language programming. As shown below:



Common object environment manages interface objects. It can present a common object interface to different languages, which functions and access mode are similar and the difference for languages is only in syntax.

Common object interface presents methods to define object attributes, functions, and event, methods to create and destroy object, and methods to call object's functions and trigger their events. For different languages, common object interface is mapped with specific syntax of the language. Further illustrate is as follows.



Using common object environment and interface will be a common and simplified method for multi-language programming. It has the following advantages:

1. For programmer, common object interface for different languages are very similar and easy to master, the difference is only syntax.
2. The complexity does not increase with the number of languages.
3. Components or libraries developed with one language can be called by another languages. Programmer need only create interface objects, Instead of rewriting for different languages.

4. Applications or products can uses number of languages freely.
5. Common languages environment can present most common functions of languages, such as libraries, distributed functions, etc.

Conclusion : Using objects as medium and developing common object environment and interface libraries or middle-ware, is one way is to simplify multi-language programming.