Contribution ID: **44** Type: **not specified**

Pythonic Interfaces

Monday 9 July 2007 10:00 (30 minutes)

With the evolution of computer systems, software development has become increasingly more complex. One way to deal with this increased complexity is through the use of software libraries. Many object-oriented languages provide special constructs such as abstract classes and interfaces which ensure that components are properly extended and executed. Unfortunately, the Python programming language is devoid of such features. In this article, we present a library extension for Python to include these features into the language and allow for explicit class interfaces and abstract classes. While it has been attempted before, our library provides a simple, elegant, and Pythonic solution to the problem via a pure-Python stand-alone library. By extending the Python language in this manner, we allow developers to define concrete models for libraries and create modular code, while ensuring that software system designs are enforced at run time. We also argue that this provides the Python language another degree of flexibility in a formalized mechanic, as opposed to error-prone traditional "hand-shake"contracts. The usability of our proposed Python extension is demonstrated in a case study of an original game engine framework.

Summary

In this article, we present a pure-Python library extension for the Python language to include class interfaces and abstract classes in a simple, elegant, and Pythonic manner.

Author: Mr HAWKER, Michael (McGill University / Mikeware)

Presenter: Mr HAWKER, Michael (McGill University / Mikeware)

Session Classification: Python Language and Libraries