http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin\_architecture.html#3.

1. MÔ HÌNH PLUG-IN ECLIPSE

Plug-in trong Eclipse là một thành phần cung cấp một loại dịch vụ nhất định trong môi trường lập trình Eclipse. Tương tự, Eclipse cung cấp cơ sở để hỗ trợ kết nối và thực thi một tập các plug-in, hình thành một môi trường lập trình đồng nhất. Khi một chương trình Eclipse được thực thi, plug-in sẽ là một đối tượng được thể hiện trong các plug-in class. Plug-in class sẽ cung cấp cấu hình và hỗ trợ quản lý cho các đối tượng plug-in. Plug-in class phải được kế thừa từ *org.eclipse.core.runtime.Plugin*.

Trong cấu trúc thư mục của Eclipse, các plug-in đã được cài đặt sẽ được chứa trong thư mục *plugins*. Plug-in được mô tả bởi một file manifest XML, gọi là *plugin.xml*. File manifest sẽ chỉ ra cách để Eclipse kích hoạt plug-in đó.

Nội dung truyền vào file manifest của plug-in bằng cách lập trình thông qua *plug-in registry API*. Các đặc tả của plug-in được lưu trữ trong bộ nhớ gọi là *plug-in registry*. Khi Eclipse thực thi, nó sẽ khởi tạo đối tượng của plug-in bằng cách sử dụng plug-in registry API, API này sau đó sẽ đoạn mã được cung cấp để lấy thông tin về plug-in.

Một đoạn manifest plug-in có dạng như sau:

<?xml version="1.0" encoding="UTF-8"?>

<plugin

name="JUnit Testing Framework"

id="org.junit"

version="3.7"

provider-name="Eclipse.org">

<runtime>

<library name="junit.jar">

<export name="\*"/>

</library>

</runtime>

</plugin>

File manifest này mô tả một plug-in cung cấp dịch vụ kiểm thử *JUnit*. Trong đoạn manifest trên, ta chú ý rằng mỗi plug-in có một số định danh duy nhất (thuộc tính XML *id*). Giá trị định danh này sẽ được sử dụng để tham chiếu đến plug-in. Giá trị này cũng có thể được sử dụng để truy cập đến các đối tượng đang thực thi của plug-in, bằng đoạn mã sau:

*Plugin p = Platform.getPlugin(pluginID);*

Các đối tượng plug-in được quản lý bởi quá trình thực thi của Eclipse, và được truy cập như giới thiệu ở trên. Nó không được tạo ra bởi các chương trình ứng dụng.

* 1. Cài đặt và sử dụng plug-in

Để sử dụng một plug-in cần sao chép các tài nguyên tạo nên plug-in (manifest file, file jar, và các file khác) vào trong thư mục *plugins* ở đường dẫn cài đặt Eclipse. Plug-in này sẽ được kích hoạt khi Eclipse thực thi chương trình, và có yêu cầu thực hiện chức năng liên quan đến plug-in đó. Kích hoạt plug-in đồng nghĩa với nạp các class và khởi tạo các biến đối tượng ban đầu.

Chức năng chính của một class plug-in là thực hiện xử lý trong quá trình kích hoạt và bỏ kích hoạt, ví dụ như cấp phát và giải phóng tài nguyên. Với một số plug-in đơn giản, như *JUnit* trên, không yêu cầu kích hoạt hay bỏ kích hoạt, và do đó không cần cung cấp plug-in class. Trong trường hợp này, Eclipse sẽ tự động cung cấp plug-in class mặc định cho đối tượng của plug-in.

Nếu plug-in có liên quan đến kích hoạt hay bỏ kích hoạt, lập trình viên cần sử dụng subclass *org.eclipse.core.runtime.Plugin*, cung cấp cách thức kích hoạt hay bỏ kích hoạt, như là *startup* hay *shutdown*, bao gồm cả tên của một plug-in cụ thể nào đó như là giá trị cho thuộc tính *class* tương ứng trong file manifest.

Trong mô hình Eclipse, một plug-in có thể liên quan đến plug-in khác theo dạng quan hệ sau:

* Phụ thuộc (Dependency): Vai trò trong mối quan hệ này là plug-in phụ thuộc và plug-in tiên quyết. Một plug-in tiên quyết sẽ hỗ trợ chức năng cho plug-in phụ thuộc.
* Mở rộng (Extension): Vai trò trong mối quan hệ này là plug-in chính và plug-in mở rộng. Plug-in mở rộng sẽ bổ sung chắc năng cho plug-in chính.

Mối quan hệ này được đặc tả trong file manifest thông qua các thành phần XML là *requires* và *extension*.

* 1. Phụ thuộc (Dependency)

Khi một plug-in phụ thuộc vào chức năng của các plug-in khác, mối quan hệ này sẽ được đặc tả bởi thuộc tính *requires* trong file manifest của plug-in. Dưới đây là ví dụ:

<?xml version="1.0" encoding="UTF-8"?>

<plugin

id="com.bolour.sample.eclipse.demo"

name="Extension Processing Demo"

version="1.0.0">

<runtime>

<library name="demo.jar"/>

</runtime>

<requires>

<import plugin="org.eclipse.ui"/>

</requires>

</plugin>

Trong ví dụ này, plug-in *com.bolour.sample.eclipse.demo* được khai báo là phụ thuộc vào plug-in UI của Eclipse *org.eclipse.ui.*

Mối quan hệ phụ thuộc này được xác định trong file manifest ở cả quá trình thực thi và quá trình biên dịch. Tại thời điểm thực thi, Eclipse sẽ đảm bảo các plug-in tiên quyết luôn khả dụng đối với các plug-in phụ thuộc khi nó được kích hoạt. Tại thời điểm biên dịch, Eclipse sẽ được dẫn hướng để cung cấp đường dẫn class (classpath) để biên dịch các plug-in phụ thuộc bằng các file jar và tất cả các plug-in tiên quyết của nó.

* 1. Extension

Khi các thành phần của plug-in được sử dụng bởi người sử dụng, một hay nhiều các giao diện người sử dụng (UI) của plug-in sẽ được thêm vào trong Eclipse. Quá trình này gọi là mở rộng (extension). Tuy nhiên, quá trình này không chỉ giới hạn cho các yếu tố UI. Bất cứ plug-in nào cũng có thể cho phép các plug-in khác mở rộng nó bằng cách thêm vào một số thành phần. Một mở rộng được xác định bởi plug-in mở rộng (plug-in extender) và khiến plug-in chính (host plug-in) thay đổi chức năng của nó. Thông thường, việc thay đổi chức năng này có thể là thêm vào chức năng mới hoặc tùy chỉnh chức năng có sẵn thông qua các dịch vụ cung cấp bởi plug-in mở rộng.

Trong trường hợp đơn giản, một mở rộng thêm một hoặc nhiều đối tượng *callback* vào môi trường, thông qua đó để plug-in chính và plug-in mở rộng liên lạc với nhau.

Tuy nhiên, lưu ý là các mô hình mở rộng là khá chung, và không nhất thiết phải yêu cầu một plug-in mở rộng cung cấp cho đối tượng callback tùy chỉnh.