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

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.

Một plug-in có thể có nhiều loại mở rộng khác nhau. Đối với mỗi loại mở rộng, nó phải đi liền với tập các yêu cầu về cấu hình và trạng thái khác nhau. Chính vì vậy, một plug-in có khả năng mở rộng phải cung cấp nhiều loại tương thích (hay khe cắm - slot) khác nhau đối với các loại mở rộng tương ứng. Các loại tương thích này được gọi là điểm mở rộng (extension point).

Hình 1 minh họa mối qua hệ giữa các thành phần trong mở rộng. Trong trường hợp này, phần mở rộng là các mục trong menu của chức năng Help của Eclipse. Trong mở rộng này, plug-in chính là giao diện người sử dụng của Eclipse, *org.eclipse.ui*, trong đó, các menu sẽ được mở rộng thông qua các điểm mở rộng là *actionSets*. Plug-in mở rộng là giao diện của Help trong Eclipse, *org.eclipse.help.ui*. Để sử dụng chức năng này, plug-in mở rộng này sẽ sử dụng các *actionSets* để mở rộng giao diện bằng cách thêm các mục trong menu của mình, ví dụ như *Help->Help Contents* và *Search->Help.* Mở rộng được định nghĩa bởi plug-in mở rộng.

(Hình 1)

Hình trên cũng chỉ ra các đối tượng callback nằm trong class của mở rộng. Đó là các class để xử lý menu của chức năng Help. Ta có thể thấy các class này được chỉ ra bởi tên trong phần khai báo của mỗi mở rộng. Ví dụ, với mở rộng menu *Help->Help Contents* khai báo class callback của nó là *HelpContentsAction*. Và mở rộng menu *Search->Help* khai báo class callback của nó là *OpenHelpSearchPageAction.*

* + 1. Các thành phần của mở rộng
       1. Plug-in chính (Host plug-in)

Đối với một mở rộng cụ thể, plug-in đóng vai trò chính sẽ cung cấp các điểm mở rộng (extension point) và được mở rộng. Bên cạnh việc cung cấp dịch vụ của riêng nó, plug-in này cũng làm nhiệm vụ điều phối và điều khiển các mở rộng khác.

Trong file manifest của plug-in, các điểm mở rộng được khai báo là một thành phần XML *extension-point*. Như ví dụ bên dưới:

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

<plugin

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_1.gif id="org.eclipse.ui"

name="Eclipse UI"

version="2.1.0"

provider-name="Eclipse.org"

class="org.eclipse.ui.internal.UIPlugin">

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_2.gif <extension-point id="actionSets" name="Action Sets"

schema="schema/actionSets.exsd"/>

<!-- Other specifications omitted. -->

</plugin>

Điểm mở rộng http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_2.gif xác định định danh duy nhất cho điểm mở rộng trong plug-in chính. Để xác định điểm định danh duy nhất toàn cục, định danh mở rộng này cần phải sử dụng thêm một định danh nữa (là http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_1.gif) để hình thành định danh đầy đủ. Ví dụ, định danh đầy đủ của điểm mở rộng *actionSets* là *org.eclipse.ui.actionSets*.

* + - 1. Plug-in mở rộng (Extender plug-in)

Một plug-in có vai trò là mở rộng, khi nó thêm vào một số tính năng cho plug-in chính. Một mở rộng được khai báo bằng cách sử dụng thành phần XML *extension* trong file manifest.

Dưới đây là ví dụ của một plug-in mở rộng, mở rộng *actionSets* bằng cách thêm vào hai mục của menu.

<plugin

id="org.eclipse.help.ui"

name="Help System UI"

version="2.1.0"

provider-name="Eclipse.org"

class="org.eclipse.help.ui.internal.WorkbenchHelpPlugin">

<!-- ... -->

<!-- Action Sets -->

<extension

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_1.gif point="org.eclipse.ui.actionSets">

<actionSet

label="Help"

visible="true"

id="org.eclipse.help.internal.ui.HelpActionSet">

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_2.gif <action

label="&Help Contents"

icon="icons/view.gif"

helpContextId="org.eclipse.help.ui.helpContentsMenu"

tooltip="Open Help Contents"

class="org.eclipse.help.ui.internal.HelpContentsAction"

menubarPath="help/helpEnd"

id="org.eclipse.help.internal.ui.HelpAction">

</action>

<!-- ... other actionSet elements -->

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_3.gif <action

label="&Help..."

icon="icons/search\_menu.gif"

helpContextId="org.eclipse.help.ui.helpSearchMenu"

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_4.gif class="org.eclipse.help.ui.internal.OpenHelpSearchPageAction"

menubarPath="org.eclipse.search.menu/dialogGroup"

id="org.eclipse.help.ui.OpenHelpSearchPage">

</action>

</actionSet>

</extension>

<!-- ... -->

</plugin>

Lưu ý rằng trong plug-in mở rộng, điểm mở rộng *actionSets* được tham chiếu đến định danh ở http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_1.gif.

Các hoạt động mở rộng được chỉ ra ở http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_2.gif, http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_3.gif sẽ tạo ra hai mục menu mới là *Help->Help Contents* và *Search->Help.*

* + - 1. Extension callback

Một đối tượng callback là một đối tượng thuần Java (không phải là plug-in) được gọi bởi plug-in chính trong một số trường hợp nhất định tương ứng với các điểm mở rộng được xác định. Giao diện của các đối tượng callback được cung cấp bởi plug-in chính. Để sử dụng các đối tượng callback này, cần khai báo class đối với một extension cụ thể. Vì việc sử dụng các đối tượng callback này là mở rộng tham chiếu đến giao diện callback, mà thông thường được đóng gói trong plug-in chính, nên có thể nói các plug-in mở rộng cũng phụ thuộc vào plug-in chính.

* + - * 1. Đối tượng dịch vụ chi tiết

Mỗi đối tượng callback có một vai trò nhất định trong mở rộng. Trong định nghĩa XML về mở rông, callback được xác định bởi các thành phần XML con. Ví dụ, với điểm mở rộng *actionSets*, một callback được khai báo là *action*. Có thể có nhiều đối tượng callback trong *actionSets*, mỗi đối tượng phục vụ một mục trong menu hay các nút bấm.

Thông thường, trong các khai báo của các điểm mở rộng thường bao gồm một thuộc tính để xác định tên của class callback. Ví dụ với plug-in *help* UI, thuộc tính của class *action* là “*class*” như được chỉ ra ở http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_4.gif.

Nhưng khi plug-in mở rộng xác định các đối tượng callback cần thiết, và khai báo các class của nó, các đối tượng này lúc này chỉ tồn tại là kết quả của hành động cụ thể của plug-in chính. Ví dụ ở http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_4.gif, class callback *action* cho menu *Search->Help* được xác định bởi plug-in mở rộng *org.eclipse.help.ui* sẽ là *OpenHelpSearchPageAction* (package *org.eclipse.help.ui.internal*). Nhưng giá trị callback tương ứng được tạo bởi plug-in chính *org.eclipse.ui.* Trong trường hợp này, giá trị của callback sẽ được tạo lần đầu tiên khi mục menu *Search->Help* được gọi.

* + - * 1. Đối tượng dịch vụ không chi tiết

Lưu ý rằng không phải tất cả các thành phần XML dùng để định nghĩa một mở rộng đều tương ứng với một vai trò callback. Một số thành phần có thể được mô tả thuần, cung cấp một số tham số cho plug-in để hình thành các UI tương ứng, hoặc để tạo ra các đối tượng bên trong như là một phần của mở rộng. Trong ví dụ trên, thành phần *actionSets* bản thân nó không định nghĩa đối tượng callback. Nhưng thành phần gây ra một đối tượng bên trong được hình thành trong plug-in *org.eclipse.ui* lại biểu diễn cho *actionSets* đó.

* + 1. Mối quan hệ giữa Plug-in và Mở rộng

Một hoạt động mở rộng là một khái niệm chung trong Eclipse và để hiểu rõ một cách khái quát, ta có thể tóm tắt lại mối quan hệ có thể tồn tại giữa đối tượng plug-in, điểm mở rộng, và đối tượng callback.

1. Có thể có nhiều điểm mở rộng có thể tồn tại trong một plug-in chính
2. Một plug-in có thể vừa là một plug-in chính, chỉ ra các điểm mở rộng, vừa là một plug-in mở rộng, mở rộng các plug-in khác.
3. Nhiều plug-in có thể mở rộng một điểm mở rộng cho trước
4. Một plug-in cho trước có thể một điểm mở rộng cho trước nhiều lần
5. Một plug-in mở rộng có thể bao gồm nhiều mở rộng của các plug-in chính khác nhau.
6. Một hành động mở rộng của một điểm mở rộng bởi một plug-in cụ thể có thể tạo ra nhiều đối tượng callback.
7. Một plug-in có thể xác định mở rộng thông qua điểm mở rộng của riêng nó.
   * 1. Lược đồ điểm mở rộng

Một mở rộng cụ thể được xác định bởi một thành phần cấu hình XML trong plug-in mở rộng. Các thành phần này cung cấp các thông tin cần thiết để khởi tạo các đối tượng callback cần thiết cho phần mở rộng, cũng như các thông tin cần thiết để tùy chỉnh giao diện và hành vi của các plug-in chính. Khi một điểm mở rộng được tạo ra, ngoài việc khai báo trong file manifest của nó, cũng cần xác định cú pháp cấu hình cho phần mở rộng cho điểm mở rộng đó. Cú pháp này là một lược đồ XML và lưu trữ trong một tập tin có đuôi *.exsd*, ví dụ, *actionSets.exsd*. Các định nghĩa lược đồ này sau đó được tài liệu hóa của plug-in.

Dưới đây là một bản giản lược của file *actionSets.exsd:*

<schema targetNamespace="org.eclipse.ui">

<element name="extension">

<complexType>

<sequence>

http://www.eclipse.org/articles/Article-Plug-in-architecture/images/tag_1.gif <element ref="actionSet" minOccurs="1" maxOccurs="unbounded"/>

</sequence>

<attribute name="point" type="string" use="required"> </attribute>

<attribute name="id" type="string"> </attribute>

<attribute name="name" type="string"> </attribute>

</complexType>

</element>

<element name="actionSet">

<complexType>

<sequence>

<element ref="menu" minOccurs="0" maxOccurs="unbounded"/>

<element ref="action" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

<attribute name="id" type="string" use="required"> </attribute>

<attribute name="label" type="string" use="required"> </attribute>

<attribute name="visible" type="boolean"> </attribute>

<attribute name="description" type="string"> </attribute>

</complexType>

</element>

<element name="action">

<complexType>

<choice>

<element ref="selection" minOccurs="0" maxOccurs="unbounded"/>

<element ref="enablement" minOccurs="0" maxOccurs="1"/>

</choice>

<attribute name="id" type="string" use="required"> </attribute>

<attribute name="label" type="string" use="required"> </attribute>

<attribute name="toolbarPath" type="string">

<attribute name="icon" type="string"> </attribute>

<attribute name="tooltip" type="string"> </attribute>

<attribute name="class" type="string"> </attribute>

</complexType>

</element>

</schema>

Lược đồ định nghĩa điểm mở rộng *actionSets* như là một thành phần gồm các thuộc tính và một chuỗi *actionSet*, và một *actionSet* bao gồm một số thuộc tính và chuỗi *menu* và *actions*.