

Implementando Ayuda Dinámica en la Plataforma Eclipse

Juan Angel Vanrell Claudio Vaucheret
Departamento de Ciencias de La Computación
Facultad de Economía
Universidad Nacional del Comahue
email: {javanrell, vaucheret}@gmail.com

Resumen

El sistema de ayuda de la plataforma Eclipse¹ define puntos de extensión con los cuales podemos contribuir. Uno de los aspectos menos documentados es la producción de ayuda en forma dinámica. En este artículo trataremos el uso del punto de extensión que es utilizado para la producción en forma dinámica de contenidos. Se describe, a modo de ejemplo, la inclusión de un sistema de ayuda para un entorno de programación del lenguaje Prolog.

1. Introducción

Eclipse es una plataforma universal para la integración de herramientas de desarrollo. Es de arquitectura abierta y extensible basada en componentes añadidos (plug-ins). Inicialmente desarrollado por OTI e IBM evolucionó en un proyecto de código abierto.

La plataforma Eclipse está diseñada para afrontar las siguientes necesidades[3]:

- Soportar la construcción de gran variedad de herramientas de desarrollo.
- Soportar las herramientas proporcionadas por diferentes fabricantes de software independientes (ISV's).

- Soportar herramientas que permitan manipular diferentes contenidos (HTML, Java, C, JSP, EJB, XML, GIF, etc).
- Facilitar una integración transparente entre todas las herramientas y tipos de contenidos sin tener en cuenta al proveedor.
- Proporcionar entornos de desarrollo gráfico (GUI) o no gráficos.
- Ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux.
- Hacer hincapié en que el lenguaje de programación sea Java para la construcción de nuevos plug-ins.

Generalmente, la creación de un plug-in para el uso de una herramienta externa involucra la inclusión de la ayuda que provee dicha herramienta como parte integrada de la plataforma. Esta integración logra un mayor acoplamiento entre la herramienta y Eclipse ya que se utiliza el mismo motor de ayuda provisto tanto para la plataforma como para las herramientas implementadas por medio de los plug-ins. De esta forma los usuarios de la herramienta no necesitan salir del entorno Eclipse para realizar consultas específicas de la misma.

2. Ayuda Estática

La inclusión de la ayuda de la herramienta en la plataforma Eclipse necesariamente involucra los si-

¹<http://www.eclipse.org>

guientes pasos:

1. Si deseamos que la ayuda de la herramienta posea una estructura dentro del motor de ayuda de Eclipse debemos generar un archivo toc.xml que la describa.
2. Habiendo creado el archivo toc.xml si deseamos que la ayuda esté disponible sin intervención del usuario debemos colocar la ayuda dentro del directorio del plug-in que contiene la ayuda del mismo.

Esta aproximación tiene desventajas. En primer lugar, la creación de un archivo toc.xml para la ayuda de la herramienta puede ser tediosa salvo que utilicemos algún programa de generación automática. En segundo lugar si la herramienta se va actualizando al igual que sus archivos de ayuda, deberíamos generar también nosotros nuestras modificaciones correspondientes en el archivo toc.xml o generarlo nuevamente para lograr consistencia. En cuanto al segundo problema, las actualizaciones de la herramienta harían que debamos reemplazar nuestros archivos de ayuda (nueva versión del plug-in o del plug-in de ayuda correspondiente).

Para solucionar este problema existen tres alternativas:

1. Lograr que el proveedor de la herramienta genere el archivo toc.xml necesario para la ubicación del contenido de la ayuda de la herramienta dentro de la plataforma Eclipse, o lo que es igual, genere el plug-in de ayuda.
2. Generar nosotros un plug-in de ayuda para cada nueva versión de la herramienta con los cambios necesarios.
3. Ubicar la ayuda en forma dinámica.

Trataremos la tercer solución como tema en este artículo. Para ver como lograr las dos anteriores ver [1] y [2]. Para ejemplificar vamos a utilizar la construcción de un plug-in de ayuda para Ciao Prolog[4].

3. Ayuda Dinámica

Ciao Prolog genera documentación en múltiples formatos, incluido HTML, en forma automática con cada versión del mismo. Esta documentación es provista en un directorio denominado `../ciao-1.10p5Win32/doc/reference/ciao_html/` para el caso de Windows y en un directorio similar para Linux, en el cual se encuentra un punto de entrada denominado "ciao_toc.html" que no es más que una tabla de contenidos HTML. Utilizaremos este archivo como conexión entre la plataforma Eclipse y la ayuda de la herramienta Ciao. A continuación se muestra el listado del archivo "toc.xml" que describe la estructura de la ayuda provista por el plug-in.

```
<?xml version="1.0" encoding="UTF-8"?>
<?NLS TYPE="org.eclipse.help.toc"?>

<toc label="Ciao Prolog Table of Contents">
  <topic
    label="Ciao Help"
    href="ciao_toc.html"/>
  <topic
    label="Plugin Ciao Help"
    href="html/index.html">
  <topic
    label="Instalación"
    href="html/install.html"/>
  <topic
    label="Puesta en Marcha"
    href="html/init.html"/>
</topic>
</toc>
```

Como podemos ver la estructura provista es jerárquica, dentro de un toc puede haber uno o más tópicos, dentro de los cuales puede haber a su vez uno o más tópicos y así continuar.

Definimos dos tipos de contribución para la ayuda de nuestro plug-in:

Una ayuda estática: provista por páginas HTML generadas para ayudar al usuario a entender cómo es el uso del plug-in. Esta ayuda puede encontrarse en la carpeta "html" perteneciente a la estructura del plug-in como puede verse en el código xml anterior. La característica principal de estos archivos es que siempre sabemos dónde

se van a encontrar ya que nosotros decidimos su ubicación proveyéndolos junto al plug-in.

```
<!-- ..... Help ..... -->
<extension point="org.eclipse.help.toc">
    <toc
        file="toc.xml"
        primary="true">
    </toc>
</extension>
```

Ayuda dinámica: Este tipo de ayuda no es provista por el plug-in sino que es anexada utilizando un nuevo punto de extensión org.eclipse.help.contentProducer que se utiliza para ubicar en forma dinámica las páginas de ayuda solicitadas.

```
<extension point=
"org.eclipse.help.contentProducer"
    id="ciaoProlog.help.CiaoDynamicHelpProducer"
    name="Ciao Dynamic Help Content">
    <contentProducer producer=
        "ciaoProlog.help.CiaoDynamicHelpProducer" />
</extension>
```

La etiqueta contentProducer indica en su atributo producer la clase que se va a encargar la gestión de ubicación de las páginas solicitadas. Esta clase debe implementar la interfase org.eclipse.help.IHelpContentProducer que define el método getInputStream(String pluginID, String href, Locale locale).

Este método intenta ubicar la ayuda que fue solicitada por el usuario y posee una referencia a href, desde un plug-in con id pluginID. El parámetro locale es utilizado para diferenciar las peticiones de distintos lenguajes de ayuda soportados y puede ser ignorado si no poseemos soporte. Como desarrolladores del plug-in nosotros decidimos donde es que deseamos que se comience la búsqueda y que hacer en caso de que falle.

Es importante destacar que todas las peticiones a la ayuda serán evaluadas por este método.

El uso que damos a este método en el plug-in Ciao es el de ubicar la ayuda propia del Ciao, que no se encuentra en una dirección estática sino que varía en base al directorio en que el usuario colocó en Ciao.

Ciao provee un directorio html en el cual se encuentra la ayuda en formato HTML para el uso del lenguaje y referencias para el uso de sus predicados predefinidos. En este directorio se encuentra un archivo denominado "ciao_toc.htm" que provee la entrada a toda la ayuda disponible.

Conociendo esto indicamos en nuestro archivo de contenido el nombre de dicho archivo de forma que lo encuentre cuando lo busquemos en dicha carpeta:

```
<topic label="Ciao Help" href="ciao_toc.html"/>
```

Luego basta con indicar donde debe ser buscada la ayuda en primera instancia (en este caso damos como ubicación un campo que el usuario del plug-in ha definido previamente en la página de preferencias correspondiente) y en caso de no ser encontrada retornamos un valor nulo que será interpretado por Eclipse como "buscar en el directorio por default".

```
public InputStream getInputStream(String
    pluginID, String href, Locale locale){

    Path path = new Path(CiaoPrologPlugin.getDefault().
        getPreferenceStore().getString(
            IPrologPreferenceFields.
                PREF_CIAO_HELP_PATH));

    IPath iPath = path.append(IPath.SEPARATOR +
        new Path(href).lastSegment());

    try{
        return new FileInputStream(iPath.toFile());
    }

    catch(FileNotFoundException fnf){
        // El archivo es buscado en la
        // documentación interna (estática)
        return null;
    }

}
```

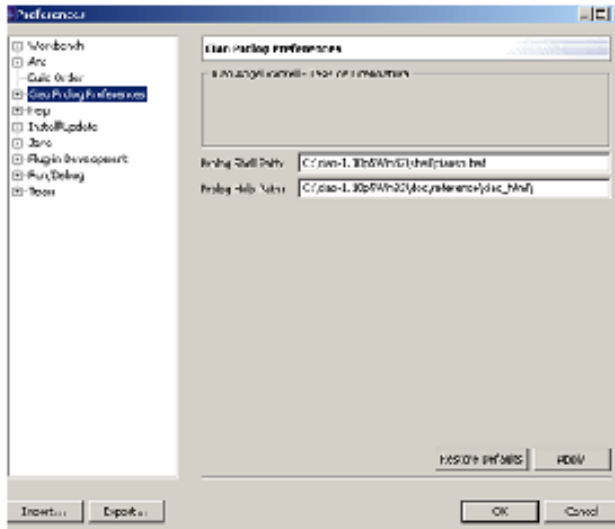


Figura 1: Localización del Archivo Índice

Una vez definida la ubicación del punto de entrada (ver figura 1) por el usuario se obtiene control sobre la posible ubicación de referencias en el mismo directorio por lo cual cada solicitud a una referencia dentro de la tabla de contenidos (o las páginas que se alcancen desde ella) será buscada de la misma forma en el directorio indicado y, de ser encontrada, será abierta en el mismo motor de ayuda de la plataforma como podemos ver en la figura 2.

4. Conclusiones

Si bien el caso descrito esta basado en una herramienta en particular (Ciao), el mismo procedimiento puede ser duplicado para cualquier herramienta que posea ayuda en HTML.

La desventaja al utilizar este tipo de ayuda es la pérdida de flexibilidad brindada por la estructura jerárquica que poseen los archivos toc.xml, con lo cual no es posible recorrer la ayuda de la herramienta en forma aleatoria sino que debemos atenernos a los enlaces que ella posee.

Como ventajas dejamos de preocuparnos por el uso de distintas versiones de la herramienta y ubicaciones

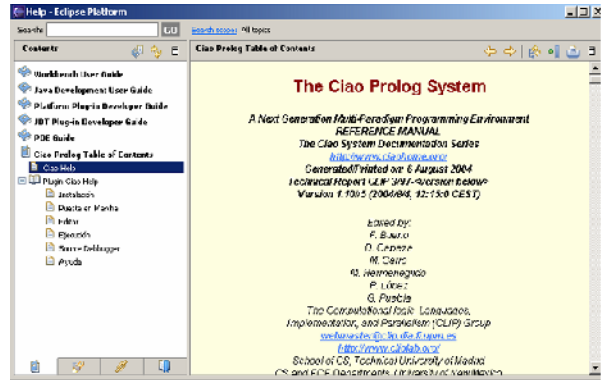


Figura 2: Visualización de la Ayuda Dinámica

de instalación ya que es el usuario quien decide inicialmente, y por única vez, donde va a encontrar el motor de ayuda la ayuda necesaria.

Referencias

- [1] Greg Adams. Help - part 1 contributing a little help. <http://www.eclipse.org/articles>, August 2002.
- [2] IBM. *Eclipse- Platform Plug-In Developer Guide*. IBM, included documentation eclipse platform edition, 2003.
- [3] Object Technology International (OTI). *Eclipse Platform Technical Overview*. OTI, <http://www.eclipse.org/whitepapers/eclipse-overview.pdf> edition, 2003.
- [4] F. Bueno D. Cabeza M. Carro M. Hermenegildo P. López G. Puebla. *The Ciao Prolog System - A Next Generation Multi-Paradigm Programming Environment*. Grupo Clip, <http://www.ciaohome.org/>, the ciao system documentation series edition, August 2004.