

JRubik

1. Introducción

JRubik es un cliente Olap realizado en java/swing sobre componentes del proyecto JPivot.

Este cliente es capaz de conectar a fuentes Olap basadas en el motor relacional Mondrian (en un futuro podrá acceder via XMLA a servidores MS Analysis Server, SAS Olap, ...). Las consultas Olap pueden realizarse por medio del lenguaje MDX (en un futuro cercano se prevee que estos componentes sean compatibles Jolap).

Los componentes principales son: navegador olap, gestor de consultas MDX, visor de tablas, visualizador de gráficos, visor de mapas, menus y marcadores, datos estadísticos ... Estos componentes están basados en componentes JPivot donde se ha realizado la labor de cambiar el interfaz web basado en JSP y TLD por uno hecho en Swing. Ello ha implicado rehacer todo el interfaz de la aplicación y la parte de control de la interacción del usuario.

Además se ha añadido componentes nuevos. Los menús y marcadores que almacenan el estado de los componentes para una consulta dada. Estos pueden ser visibles para todos los usuarios o para uno en particular. El visor de mapas está basado en otro proyecto abierto de visualizador GIS llamado Geoclient que permite generar mapas en SVG que incorporan los resultados de una consulta Olap.

2. Descarga/Instalación

2.1. Requisitos

- Java SDK 1.4.1 o superior
- Adobe SVG Viewer 3.x Plugin (sólo necesario para la utilización de mapas).

2.2. Instalación

No es necesario compilar el proyecto entero ya que se incluyen binarios precompilados en la distribución. Sólo tiene que descargar y descomprimir el último fichero de la distribución: rubik-x.x.x.zip y ejecutar el fichero por lotes correspondiente a su plataforma (.bat o .sh).

2.3. Compilación

El proyecto está preparado para ser compilado usando Ant 1.5.x o superior y Eclipse 3.x IDE. Para la compilación con Ant basta cambiar el directorio raíz del proyecto y ejecutar "ant" en la línea de comandos. Revise los ficheros build.xml y ant.properties para personalizar los parámetros de compilación. Para la instación como proyecto de Eclipse simplemente hacer el directorio raíz del proyecto igual al directorio de proyecto en la definición de nuevo proyecto de Eclipse. Los ficheros .project y .classpath están incluidos en este directorio.

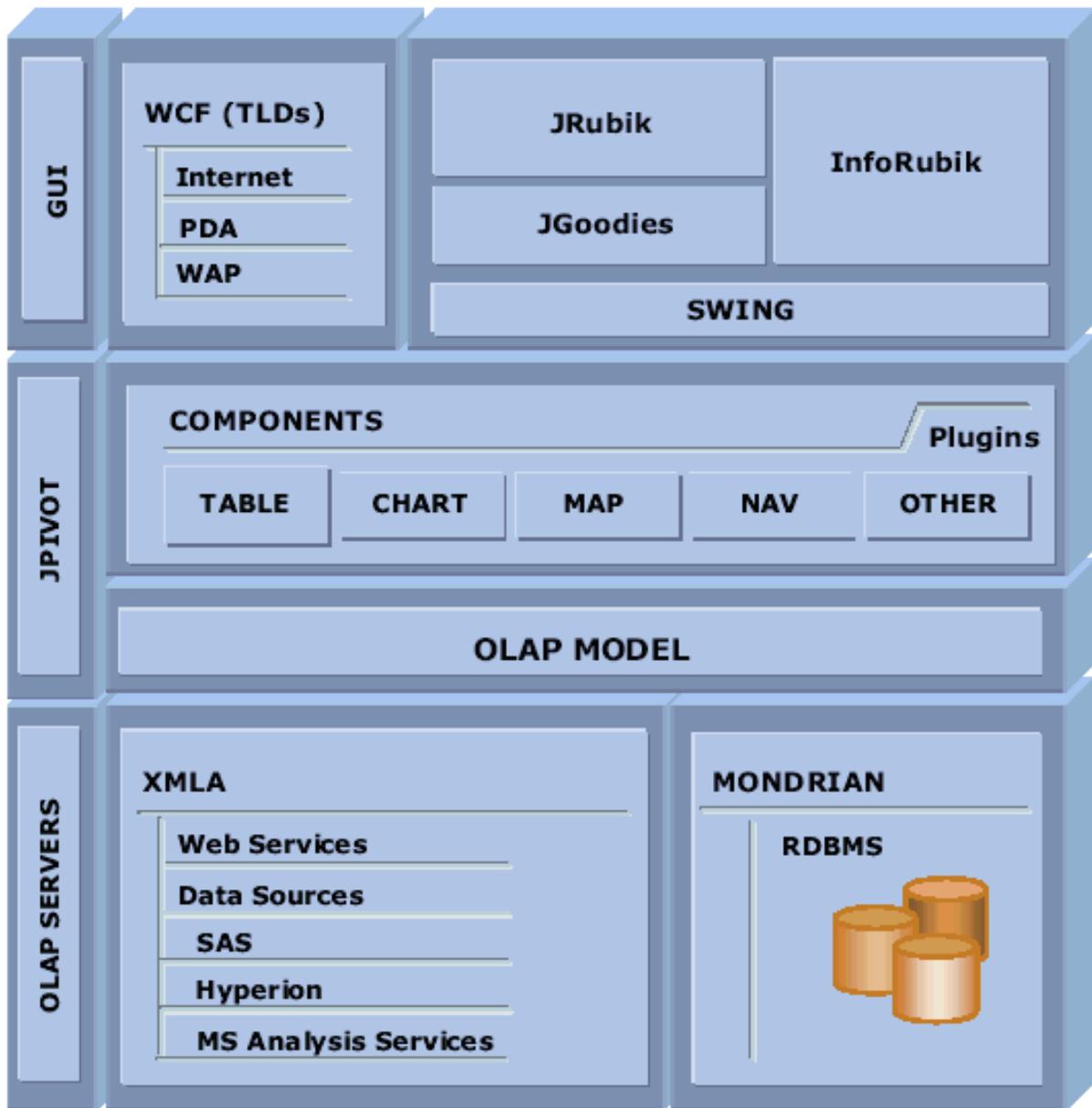
2.4. Ficheros de configuración

Existen dos ficheros básicos de configuración: log4j.properties y mondrian.properties. El fichero log4j.properties configura las propiedades de la aplicación en materia de logs (que utiliza el componente log4j). Una peculiaridad es que la aplicación principal utiliza log4j para la visualización de mensajes en algunos componentes (status bar, ...). Para ello JRubik declara una categoría log4j: "org.aeat.etss" con nivel de log de información (LEVEL=INFO). Si se quiere cambiar en nivel de log, a uno con menor nivel de detalle bastaría con especificarlo explícitamente en el fichero de configuración.

El fichero mondrian.properties define la propiedades del motor Olap Mondrian. Para más información, ver Download/Install (4. Configure the execution environment for testing) en la página web de Mondrian.

3. Arquitectura

3.1. Capas y componentes



Componentes de la Arquitectura

3.2. Estructura de plugins

La aplicación principal y el resto de componentes se han desarrollado utilizando el patrón plugin. Los plugins se almacenan en el directorio: `./lib/plugins`. Un plugin podrá ser

eliminado de la aplicación simplemente con mover su .jar correspondiente del directorio de plugins.

3.3. Componentes

El componente **tabla** es el visor principal de la aplicación. Permite al usuario trabajar con el resultado de la consulta olap, visualizándolo de manera tabular. Podrá navegar a través de dimensiones, jerarquías, miembros, cambiar el orden, eliminar dimensiones El nivel de interactividad de este componente es similar al equivalente de JPivot, pero han sido añadidas alguna funcionalidad nueva como el soporte drag&drop (para mover dimensiones y miembros, cambiar el eje de una dimensión, eliminar elementos ...). Los datos mostrados por este componente pueden ser impresos y exportados a formato PDF, XML, HTML y Excel (o copiados al portapapeles).

El visor **gráfico** utiliza la librería JFreeChart para construir todo tipo de gráficos (sectores, barras, líneas, 3D ...). Por el momento, este componente no permite ningún tipo de navegación o interacción OLAP. Todos los gráficos pueden ser impresos o exportados a formato XML y HTML (también se puede copiar&pegar).

El **navegador** es un árbol que permite al usuario seleccionar las dimensiones y medidas de la consulta olap sin necesidad de escribir la consulta MDX olap directamente. Todas las selecciones del componente tienen una traducción MDX que pueden ser visualizar o modificadas utilizando el editor de consultas MDX. Este componente tiene capacidades drag&drop y menús contextuales.

El usuario podrá almacenar sus consultas favoritas como un **bookmark** para uso futuro. Existen menús predefinidos que utilizan el mismo método de almacenamiento, pero que no pueden ser modificados por el usuario desde la aplicación (Un método simple para personalizar los menus es crear los bookmarks y luego copiar ficheros y directorios desde ./resources/marcadores/user_name a ./resources/mdxmenu).

El componente **mapa** compone, a partir de un mapa esceficiado, los datos olap provenientes de varias medidas con una dimensión geographical. Es necesario definir la dimensión geográfica, para ello se debe especificar en el esquema Mondrian utilizando una etiqueta parameter. Para más información, acudir al fichero de esquema Foodmart incluido en el directorio ./resources/catalogs y localizar la dimensión Customers en el cubo Sales. Nuevos ficheros con mapas pueden añadirse al directorio ./resources/map/maps. Actualmente este componente no soporta múltiples dimensiones geográficas en el mismo cubo. Remember, for Map visualization is necessary to install the Adobe SVG Viewer 3.x plugin.

Existe también un frame, que permite visualizar un conjunto de estadísticas básicas sobre la consulta en curso. Esta es una version alfa y no ha sido probada intensivamente.

JRubik

El componente "**Drill-through**" se utiliza para mostrar los datos originales utilizados para calcular los datos agregados en la consulta OLAP. El método de trabajo es similar al Drill-through de JPivot. Los datos mostrados pueden imprimirse e importarse a PDF, XML, HTML and Excel (o copiados al portapapeles).

Otros componentes: El monitor de memoria está basado en `org.apache.batik.util.gui.MemoryMonitor` de Batik.