

Programación orientada a objetos

Instalación y uso básico de BOUML



Patricia López
Computadores y Tiempo Real

Santander, 2009

1



Características

- Herramienta UML 2.0
- Software libre
- Generación de código Java, C++, IDL, PHP y Python
- Multiplataforma: Linux, Windows, MacOS
- Es rápida y no consume mucha memoria
- Extensible: los generadores de código son extensiones (“plug-outs”) incluidas en la distribución
- Instalación : Ejecutar Bouml_4.2_setup.exe y seguir las instrucciones
- Manual de usuario: <http://bouml.free.fr/doc/index.html>

¡No instalar otras versiones en el laboratorio!

- ✦ Al iniciar BOUML nos pide un identificador del desarrollador (BOUML_ID): si varios desarrolladores van a trabajar con el mismo modelo, se debe especificar esta variable de entorno
 - Windows: start > settings > control panel > system > advanced > environmentvariables.
- ✦ Crear el proyecto: *Project => New*
- ✦ Guardar el proyecto: directorio donde se guardará el proyecto.
 - Nota: la carpeta debe llamarse como el fichero .prj
- ✦ Indicar lenguaje de desarrollo:
 - *Languages=> Java default definition*

The screenshot shows the BOUML main interface. On the left is a 'Modelo' (Model) browser showing a tree structure of classes and packages. The central area displays an 'Inheritance Diagram of classes for Classes' with various class nodes and relationships. On the right, there is a 'Documentación' (Documentation) pane. Labels with arrows point to these three main components: 'Modelo', 'Diagramas', and 'Documentación'.

Modelo

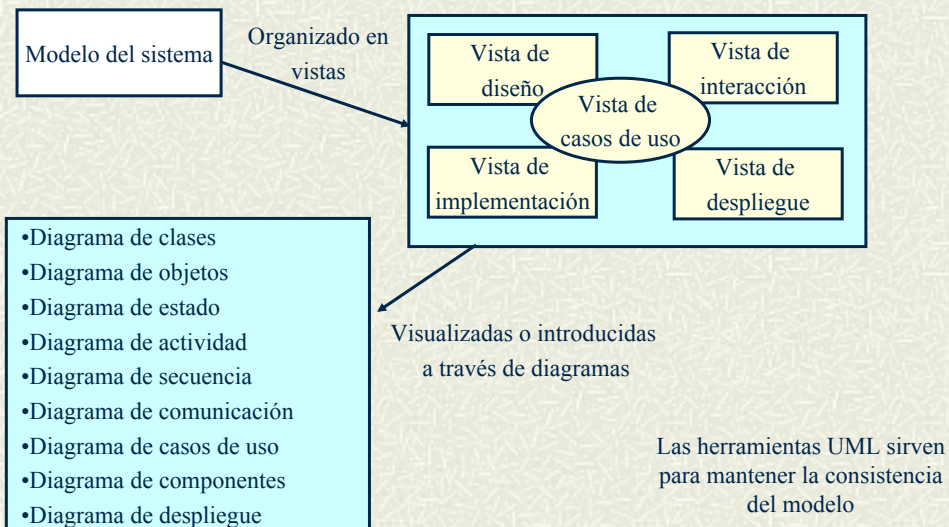
- Simplificación completa y autoconsistente de la realidad, creado para comprender mejor un sistema.
- Captura las propiedades estructurales (estática) y de comportamiento (dinámicas) de un sistema.

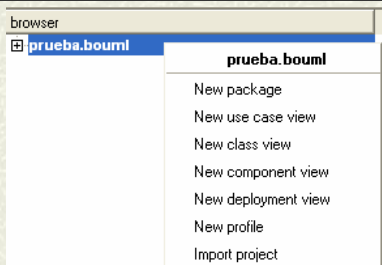
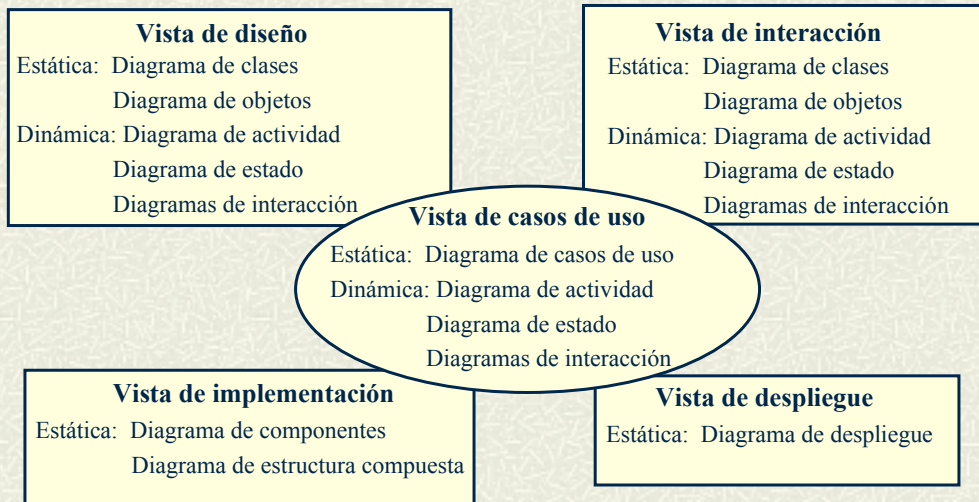
Vista (Arquitectural)

- Proyección de la organización y estructura de un modelo de un sistema, centrada en un aspecto.
- Incluye un subconjunto de los elementos incluidos en el modelo
- Una vista no contiene la semántica completa de la aplicación. Ésta reside en el modelo completo.

Diagrama

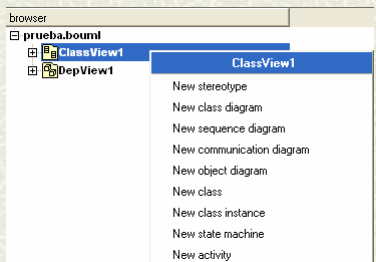
- Representación gráfica de un conjunto de elementos normalmente mostrado como grafo conexo de nodos (elementos) y arcos relaciones).





BOUML no permite añadir elementos de forma arbitraria.

- El modelo se organiza en vistas
- Y opcionalmente en paquetes



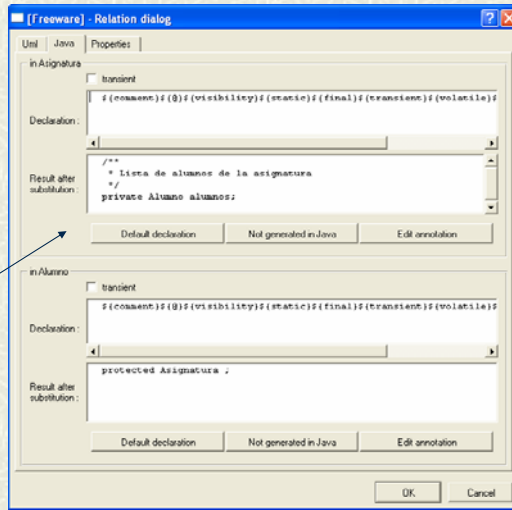
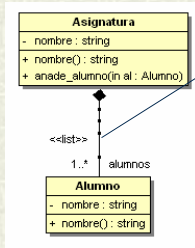
Una vez creada una vista, podemos añadir sus elementos:

- Generalmente a través de diagramas

- # Generar una vista de clases:
 - *Botón derecho en el proyecto => New class view*
- # Generar diagrama de clases:
 - *Botón derecho en la vista de clases => New class diagram*
- # Cambiar las opciones del dibujo (muestra de atributos, signatura completa de operaciones, etc):
 - *Botón derecho en el diagrama de clases => Edit drawing Settings*
- # Crear clases y asociaciones:
 - *Elegir el icono correspondiente en la paleta y pinchar en el área del diagrama*
- # Editar propiedades de las clases (añadir operaciones, atributos, etc):
 - *Doble click sobre la clase en el diagrama o en el modelo*

- # Generamos una vista de desarrollo:
 - *Botón derecho en el proyecto => New Deployment View*
- # La asociamos a la vista de clases:
 - *Botón derecho sobre la vista de clases => Edit => deployment view*
- # Creamos un artefacto (.java) para cada clase:
 - *Botón derecho en la clase => Create source artifact*
- # Elegimos el directorio en que se va a generar:
 - *Botón derecho en el proyecto => Edit generation settings => Directory*
- # Generamos el código :
 - *Botón derecho en el proyecto => Generate => Java*
Aparece una ventana que informa de los errores y warnings encontrados

- Para ver como se va a generar el código de una clase o asociación: *Botón derecho en una clase o asociación => Edit => Java*



Pinchando en el proyecto => *Edit => Edit Generation Settings* podemos cambiar el modo de generación

Para una asociación de multiplicidad a..b el patrón es:

```
/${comment}/${@}/${visibility}/${static}/${final}/${transient}/${volatile}/${stereotype} ${name}/${value};
```

En el ejemplo: `private List alumnos;`

Para que lo genere según Java 5.0:

```
/${comment}/${@}/${visibility}/${static}/${final}/${transient}/${volatile}/${stereotype} <${type}> ${name}/${value};
```

`private List<Alumno> alumnos;`