

HERRAMIENTAS UML PARA EL MODELADO DE SISTEMAS

Ricardo Armando Barrera Cámara
 Jesús Alejandro Flores Hernández
 José Alonso Pérez Cruz*

Resumen

El siguiente trabajo muestra el resultado de evaluar herramientas de software con licenciamiento de Software libre, disponibles para el modelado de sistemas de información con Lenguaje de Modelado Unificado, para lo cual se considero el licenciamiento, sistema operativo, plataforma, diagramas que genera, y generación de código.

Introducción

Los seres humanos utilizamos diversos modelos para poder representar elementos o concepciones en diversas áreas como pintura, artes plásticas, construcción, sistemas de software e incluso en nuestra vida cotidiana. El modelado de sistemas es una actividad importante e imprescindible en la construcción de cualquier tipo de sistema, está considerado como un plano que nos muestra la estructura genérica o abstracta de algún sistema.

Debido a importancia que ha tomado el hecho de modelar y diseñar antes de construir se han desarrollado diversas herramientas de software que facilitan esta actividad. Al elegir una herramienta de software se debe considerar elementos como son: el tipo de licenciamiento, plataforma, sistema operativo, generación de código.

Lenguaje de modelado unificado

El lenguaje de modelado unificado está compuesto por diversos elementos gráficos que se combinan para formar diagramas, cuenta con reglas para combinar tales elementos[1]. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se le conoce como modelo. El Modelo UML es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio, describiendo lo que se supone hará el sistema, pero no como implementarlo.

Los diagramas más importantes y sus funciones se enlistan a continuación:

- Diagramas de clases: Permite representar los atributos y acciones de un objeto, así como identificar relaciones entre objetos.
- Diagrama de objetos: Muestran entidades que tienen valores específicos y acciones.
- Diagramas de casos de uso: describe las acciones de un sistema desde la visión del usuario, aunque también puede representar la visión del programador.
- Diagrama de Estados: Muestra los estados que toma un objeto.
- Diagrama de Secuencias. Muestra la interacción de los objetos en base a tiempo, aunque realmente es una vista estática.
- Diagrama de Colaboraciones: Representan la manera en que los elementos del sistema trabajan en conjunto.
- Diagrama de Componentes: muestran el producto final del software.
- Diagramas de Distribución: Muestran la arquitectura física de un sistema de información:

Licenciamiento de software

Una licencia de software otorga al usuario el derecho legal a utilizar un software, existen básicamente dos tipos de licencias: Licencia de software libre y Licencias de software comercial.

Según la Fundación de Software Libre (Free Software Foundation) “el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software”, se refiere a cuatro libertades para los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad informática se beneficie.

En las licencias de software comercial una compañía o corporación, posee los derechos de autor sobre un software, negando o no otorgando los derechos de usar el programa con cualquier propósito, de estudiar cómo funciona el programa y adaptarlo a las propias necesidades y además el acceso al código fuente es una condición previa, se pueden distribuir copias, o mejorar el programa y hacer públicas las mejoras, para poder realizar los cambios, el acceso al código fuente es un previo requisito. Así, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución[2].

Elementos de evaluación de herramientas

Se consideraron aspectos como:

- El Sistema operativo, es el software encargado de la administrar y coordinar el hardware con programas de una manera transparente, nos indica la portabilidad de la herramienta.
- La plataforma, se considera el software sobre el cual se ejecuta la aplicación sobre el sistema operativo.
- Diagramas soportados, los tipos de diagramas son muy diversos y algunas de las herramientas ofrecen una amplia o mínima variedad de estos, según sea el estándar de la Herramienta en el cual este basado.
- Generación de código. La herramientas de modelado ofrecen una diversidad de diagramas generan código en algún lenguaje de programación, lo que nos permite que a partir del modelo gráfico se construya software.

Análisis de herramientas de software libre

La siguiente tabla muestra las herramientas de software con los sistemas operativos que predominan en la actualidad, en donde la celda llena indica el sistema soportado.

*Docentes de la Dependencia Área Ciencias de la Información en la Universidad Autónoma del Carmen.

Herramienta	Windows	Linux	Unix	MacOS
ArgoUML				
BOUML				
FuJaba				
Día				
GModeler				
MonoUML				
Papyrus				
StarUML				
TCM				
Umbrello				
Netbeans UML				

Tabla 1 Comparación de herramientas por sistema operativo

La tabla siguiente muestra una relación entre herramientas y plataformas.

Herramienta	Java	C++	Gtk+	Flash MX	Mono	Eclipse	Delphi
ArgoUML							
BOUML							
FuJaba							
Día							
GModeler							
MonoUML							
Papyrus							
StarUML							
TCM							
Umbrello							
Netbeans UML							

Tabla 2 Comparación de herramientas por plataforma

Debido a la amplia gama de diagramas y para efectos de un análisis se utilizo la siguiente nomenclatura para cada tipo de diagrama: Casos de uso (CU), Clases (CL), Secuencia(S), Colaboración (CO), Actividades(A),

Despliegue (D), Objetos=(O), Paquetes (P), Componentes (COM), Estados (E), Estructura (ET).

Herramienta	CU	CL	S	CO	A	D	O	P	COM	E	ET
Argo UML											
BOUML											
FuJaba											
Día											
GModeler											
MonoUML											
Papyrus											
StarUML											
TCM											
Umbrello											
Netbeans UML											

Tabla 3 Comparación de herramientas por generación de diagramas

La siguiente tabla muestra los lenguajes de programación que las herramientas de modelado analizadas pueden generar código.

Herramienta	Java	C++	C#	PHP	g++	Python	ILD	ECM Ascript	Ruby	No Genera
ArgoUML										
BOUML										
FuJaba										
Día										
GModeler										
MonoUML										
Papyrus										
StarUML										
TCM										
Umbrello										
Netbeans UML										

Tabla 4 Comparación de herramientas por generación de código

Conclusiones

El Lenguaje Unificado de Modelado (UML), fue creado por la necesidad de utilizar un lenguaje estándar para el modelado de sistemas, debido a que existía una diversidad de modelos o métodos.

En cuestiones de desarrollo de sistemas, para que el producto final sea funcional, es necesario que durante el proceso de desarrollo se realicen ciertas actividades que permitan el buen funcionamiento de un sistema, por lo que el aplicar el modelado de sistemas nos dará la pauta para definir la estructura que tendrá nuestro sistema. Para realizar el modelado existen diversos tipos de diagramas y cada uno tiene los elementos requeridos para satisfacer cada una de las necesidades del sistema según su estructura, las herramientas son muy diversas y su licenciamiento puede variar.

Se han creado herramientas de software para una variedad de sistemas operativos dentro de los cuales destacan Windows y Linux, en particular la mayoría de las herramientas fueron generadas en Java, además los requerimientos de instalación en cuanto a hardware no son muy primordiales para la mayoría de las herramientas. En cuanto a la utilización de las herramientas, independientemente del lenguaje que soporte o del sistema operativo, cada una tiene los elementos y características que satisface la necesidad de sus usuarios.

De las herramientas el 32% trabajan sobre el sistema Operativo Windows, 29% en Linux, 26% en Unix y el 13% MacOS.

Los sistemas operativos en los cuales pueden ser instaladas las herramientas, se observo que el promedio de sistemas operativos pueden ser instaladas es de tres. Siendo BOUML y NetBeans UML con soporte para todos los sistemas operativos, Umbrello con dos y StarUML con un sistema operativo.

Las plataformas requeridas para el funcionamiento de las herramientas analizadas fueron muy diversas y amplias. El 34% requiere Java, el 26% requiere C++, 8% requiere Gtl+ ,8% requiere Flash MX, el 8% requiere Mono, 8% requiere Eclipse, 8% requiere Delphi.

El numero de diagramas que pueden generar se encuentran en un rango de 2 a 11 diagramas, en donde Netbeans UML es el que más

diagramas genera, en tanto que GModeler y MonoUML los que menos diagramas generan.

Las herramientas que presentan mayor diversidad en cuanto a la generación de código en diversos lenguajes de programación destacan ArgoUML y BOUML por generar código en 5 lenguajes y FuJaba, Día, GModeler. MonoUML, Papyrus solo en un lenguaje.

Referencias

[1]J. Schmuller, *Aprendiendo UML en 24 Horas*: Prentice Hall, 2001.
 [2]C. Segovia. "Tipos de Software y Licencias," <http://www.alberguedenaufraigos.com.ar/inst/tiposoft.html>.
 ArgoUML <http://argouml.tigris.org>
 BOUML <http://bouml.free.fr>
 Dia <http://live.gnome.org/Dia>
 FuJaba <http://www.wcs.uni-paderborn.de/cs/fujaba/>
 GModeler <http://www.gskinner.com/gmodeler/index.html>
 MonoUML <http://www.monouml.org/>
 Netbeans UML <http://uml.netbeans.org/>
 Papyrus <http://www.papyrusuml.org/>
 StarUML <http://www.staruml.com>
 TCM <http://www.home.cs.utwente.nl/~tcm/>
 Umbrello <http://uml.sourceforge.net/index.php>