

Estudio y evaluación de una herramienta para representar máquinas de estado finitas.

Alejandro Gutiérrez Ferrerías
Master en Software Libre, Universitat Oberta de Catalunya (UOC)
Barcelona, España
agutierrezf@uoc.edu

Resumen / Abstract

En este artículo se presenta los resultados y conclusiones del trabajo de investigación llevado a cabo sobre herramientas informáticas para representación de grafos de autómatas de estado finitos.

El principal resultado de esta investigación es el desarrollo de una nueva herramienta, que permita dibujar el grafo de forma totalmente automática, partiendo de una tabla de transiciones donde se describe al autómata en cuestión.

This article presents the results and conclusions of the research carried out on tools for graph representation of finite state automata.

The main result of this research is the development of a new tool, which allows to draw the graph fully automatically, starting from a transition table which describes the automaton in question.

Introducción

Los objetivos que se han propuestos para realizar este trabajo de investigación son los siguientes:

- Realizar un estudio sobre las herramientas de software existentes que permiten dibujar, editar y manipular grafos en general y/o grafos que representan máquinas de estado finitas en particular.
- Desarrollar, si es necesario, una herramienta de software modular, escalable y fácilmente adaptable y

modificable, que permita dibujar grafos de máquinas de estado finitas (modelo de Moore) de forma automática.

La estrategia que se ha utilizado para alcanzar los objetivos propuestos en la investigación se resume en los siguientes pasos:

- Revisión de las herramientas de manipulación de grafos existentes previamente (Estado del arte).
- Justificación de la necesidad de desarrollar una nueva herramienta informática.
- Selección de la tecnología usada en el desarrollo de la misma.
- Desarrollo de la nueva herramienta de software.
- Conclusiones sobre la investigación realizada.

Estado del arte – Contribuciones

Para cumplir el primer objetivo de la investigación, se ha realizado un estudio del arte de las herramientas de representación de grafos que existen en la actualidad. Se han analizado las herramientas Aisee, GUItar-Fagoo, JFLAP, yFiles, GLEE, Dot-Graphviz y Boole-Deusto. Algunas de ellas tienen una licencia privativa (Aisee, yFile y GLEE) o limitan la libertad de uso comercial (JFLAP y Boole-Deusto) por lo que no son software libre y no son útiles para este proyecto. La restante, GUItar-Fagoo, si usa una licencia de software libre, pero es una herramienta monolítica, que incluye su propia interfaz de usuario, pero no es adaptable para un uso personalizado por el usuario final.

De entre ellas, se ha seleccionado Dot-Graphviz como herramienta para dibujar grafos, debido a que es la única que aglutina las siguientes características que se desean en la herramienta final objeto de esta investigación:

- Puede ser fácilmente adaptado por el usuario para un uso personalizado, debido a su estructura de funcionamiento, donde:
 - La entrada se define en un fichero en un lenguaje propio que define el grafo que se desea generar.
 - La ejecución se realiza mediante una llamada al programa Dot desde la línea de comando. Esta característica hace que sea fácilmente integrable dentro de otro software. Se puede ejecutar tanto en sistemas Windows como Linux.
 - La salida se escribe en un fichero gráfico en un formato definido por el usuario en la llamada al programa.
- Puede generar el dibujo del grafo de forma automática a partir de un fichero que describa las transiciones entre estados del grafo (tabla de transiciones).
- Distribuye los nodos (estados) del grafo de forma ordenada y favoreciendo la legibilidad del mismo.
- Intenta evitar o al menos minimizar los cruces de aristas en el grafo generado.
- Se puede seleccionar el formato gráfico en que se desea obtener el dibujo del grafo generado.
- Tiene una licencia de Software Libre (Eclipse Public License).

Descripción de la herramienta desarrollada

Dot-Graphviz es una herramienta de dibujo de grafos muy genérica y por tanto se hace necesario desarrollar una herramienta software específicamente orientada a la edición, generación y manipulación de grafos de autómatas finitos, de la cual Dot-Graphviz formará parte. El nuevo software desarrollado

permite integrar todas las características que se desean:

- Dibujado automático de un grafo a partir de la tabla de transiciones que lo define.
- Posibilidad de editar y modificar la tabla de transiciones descriptiva de un grafo.
- Posibilidad de guardar la tabla de transiciones de un grafo en disco en un fichero con formato XML.
- Guardar automáticamente el grafo generado en formato gráfico JPEG.
- Posibilidad de cargar en memoria un fichero XML que contenga la tabla de transiciones con la descripción de un grafo.

El nuevo software que se ha desarrollado, utilizando lenguaje Java y la plataforma Eclipse, tiene dos partes, que siguiendo un modelo Vista-Controlador, se mantienen diferenciadas:

- Una interfaz gráfica desde la que el usuario puede llamar las diferentes funciones del programa, y visualizar los grafos y/o datos (clases PFM_UOC, AcercaDe, Fichero, ValidarXML y Tabla_Maquina_Estados). El programa externo Dot-Graphviz tiene un papel principal ya que se encarga de dibujar los grafos.
- Una estructura de datos y las funciones para operar sobre ésta, que modeliza la máquina de estados del autómatas y su funcionamiento (clases Maquina, Salida, Transición y Estado).

En la figura 1 se muestra una vista instantánea de la herramienta desarrollada y en la figura 2 se muestra el diagrama de clases UML del software que se ha desarrollado.

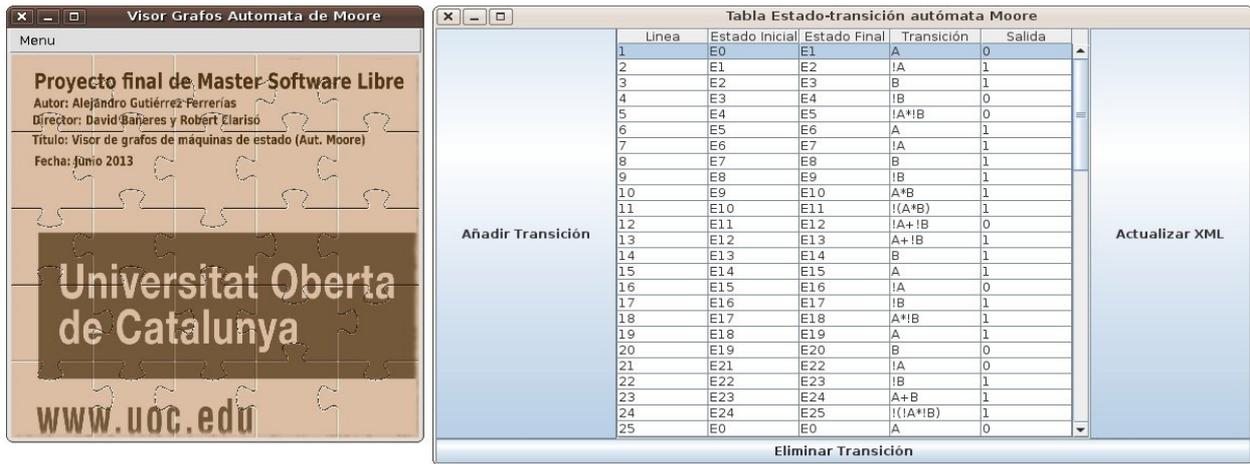


Figura 1 : Herramienta PFM_UOC desarrollada.

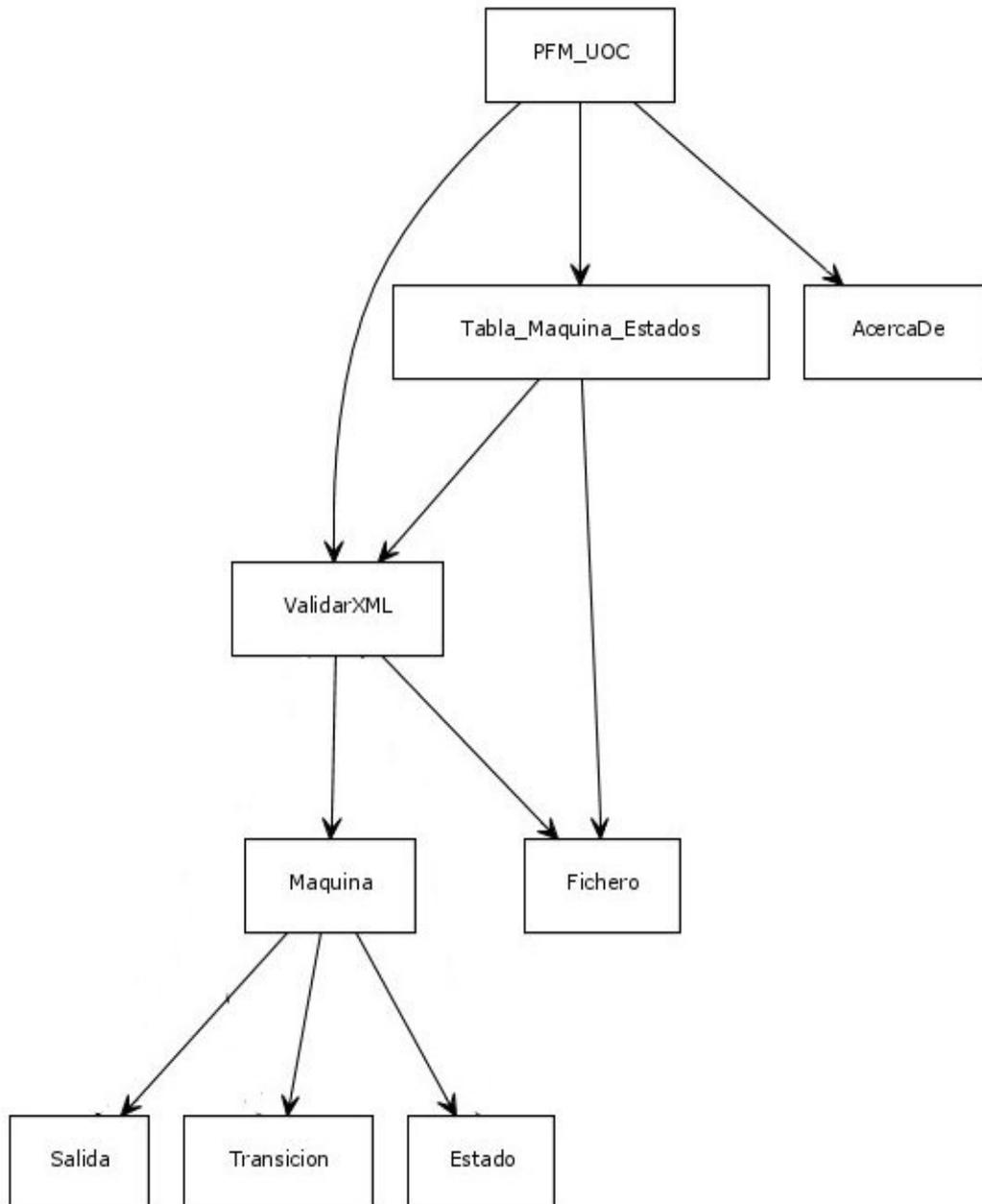


Figura 2: Diagrama de clases UML del software desarrollado PFM_UOC

El programa desarrollado es una aplicación de ventana, y dispone de un menú desde el cual se seleccionan las diferentes opciones que permite la aplicación:

- Editar la tabla de transiciones (permite añadir, eliminar y modificar).
- Cargar una tabla de transiciones desde un fichero XML residente en disco.
- Guardar la tabla de transiciones actual en disco y con formato XML.
- Generar el grafo en formato JPEG, que describe la tabla de transiciones cargada en memoria. Si se hace alguna modificación, se debe actualizar la imagen XML de la tabla de transiciones, antes de generar el grafo.

En las figuras 3 y 4 se muestran sendos grafos generados usando la herramienta mencionada.

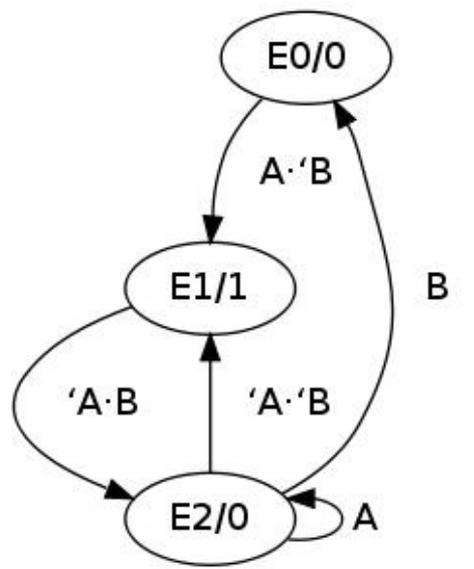


Figura 4: Otro grafo de ejemplo generado con la herramienta.

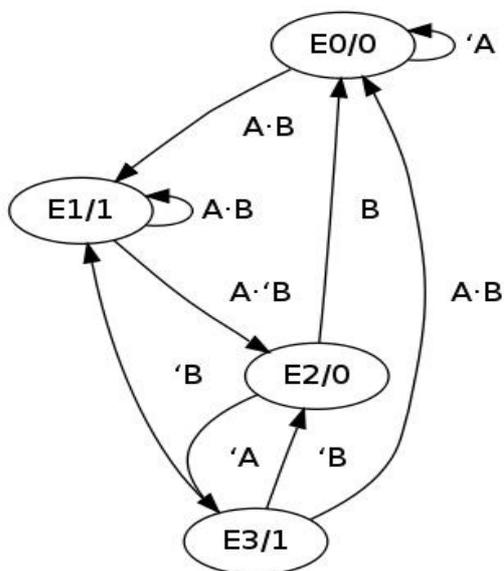


Figura 3: Ejemplo de grafo generado con el programa PFM_UOC

Materiales y métodos

En primer lugar se han determinado la estrategia a seguir, y los métodos de generación de datos y de análisis de los mismos, que se han considerado alcanzar los objetivos de la investigación.

La estrategia de investigación que se ha utilizado en el presente proyecto es “Diseño y Creación” ([5] Oates, 2006, páginas 108-124), la cuál se centra en el desarrollo de nuevos productos de Tecnología de la Información, conocidos con el nombre de “artefactos”. En este caso el artefacto es la herramienta en sí que se ha diseñado.

Los métodos de generación de datos utilizados son “observación” y “cuestionarios”. El método de “observación” se ha usado para extraer datos de los resultados de la batería de pruebas realizadas con el programa real. El método de “cuestionarios” se ha utilizado para obtener, de los usuarios finales de la herramienta, los datos sobre la valoración de la colocación de los nodos y la legibilidad del grafo.

Se ha utilizado el método de análisis de datos cuantitativo para analizar los datos obtenidos debido a que estos se han codificado de forma numérica, lo cuál facilita la automatización de su tratamiento mediante técnicas estadísticas.

Resultados

En este proyecto, se han obtenido tres tipos de resultados:

- Un estudio del arte sobre herramientas de dibujo y manipulación de grafos, y un análisis razonado de cual es la mejor opción para los objetivos propuestos.
- La herramienta de software PFM_UOC desarrollada, cuya utilidad es generar, editar y manipular grafos de máquinas de estados finitas.
- El análisis comparativo de la funcionalidad y facilidad de uso de las herramientas JFLAP, Boole-Deusto y la herramienta desarrollada en el presente proyecto PFM_UOC. La **tabla 1** resume los resultados de este análisis.

Conclusión

En este proyecto se ha desarrollado una nueva herramienta especializada en la representación y manipulación de grafos de autómatas finitos, basados en autómatas de Moore, la cual provee un entorno gráfico para editar, representar, salvar y guardar los grafos mencionados.

Las principales características de la herramienta desarrollada son:

- Se ha usado la herramienta Dot-Graphviz para la generación de los grafos, la cual, gracias a su versatilidad se ha integrado en el software desarrollado.
- Se ha utilizado lenguaje Java sobre plataforma Eclipse.
- Se ha desarrollado siguiendo un modelo Vista-Controlador, de forma que se separe el comportamiento de la presentación de resultados.
- Se ha utilizado el formato XML para guardar datos intermedios.
- Se ha utilizado el formato gráfico JPEG para los grafos generados.

Como aspectos de mejora y ampliación, se puede citar:

- Ampliar la herramienta para que pueda representar otros tipos de autómatas (Mealy, Máquinas algorítmicas...etc).

Referencias

- [1] Koutsofios E., North S.C., “Drawing Graphs with Dot”, www.graphviz.org/Documentation/dotguide.pdf, *Octubre 1993*.
- [2] Almeida A., Moreira N., Reis R., “GUItar and Fagoo: Graphical interface for automata visualization, editing and interaction”, Conferencia-simposium de informática “Inforum”, Septiembre de 2010 (Braga-Portugal)
- [3] Losacco M., Rodger S.H., “FLAP: A Tool for Drawing and Simulating Automata”, ED-MEDIA 93: World Conference on Educational Multimedia and Hypermedia , 1993.
- [4] García Zubía J., Sanz Martínez J., Sotomayor B., “BOOLE-DEUSTO: la aplicación para sistemas digitales ”, VII Jornadas de Enseñanza Universitaria de la Informática (JENUJ), 2001.
- [5] Oates B.J., “Researching Information Systems and Computing”, SAGE, 2006
ISBN 978-1-4129-0223-6

Característica	JFLAP	PFM_UOC	B-Deusto
Método para definir el grafo del autómata	<i>El usuario lo define mediante un editor gráfico</i>	<i>El usuario lo define mediante una tabla de transiciones entre estados</i>	<i>El usuario lo define mediante un editor gráfico</i>
Permite realizar simulaciones del funcionamiento del autómata	<i>Si, permite la simulación interactiva del funcionamiento del autómata</i>	<i>No</i>	<i>Si, permite la simulación interactiva del funcionamiento del autómata</i>
Además del autómata de Moore, admite otros tipos de autómata	<i>Si, admite una gran cantidad de tipos de autómatas (Mealy, Turing...etc)</i>	<i>No</i>	<i>Si, pero sólo el modelo de Mealy</i>
El grafo del autómata se genera de forma automática.	<i>No, el grafo lo dibuja el usuario de forma manual desde el editor.</i>	<i>Si, la disposición de los nodos es automática y se evita el cruce de arista de forma automática</i>	<i>No, el grafo lo dibuja el usuario de forma manual desde el editor.</i>
Permite guardar la representación de los autómatas generados en un fichero en disco y en formato XML.	<i>Si</i>	<i>Si</i>	<i>No</i>
Se puede ejecutar en entornos Windows y Linux	<i>Si, se ejecuta sobre la máquina virtual de Java</i>	<i>Si, se ejecuta sobre la máquina virtual de Java</i>	<i>No, se distribuye en versión ejecutable para Windows. Aunque puede funcionar en Linux usando la herramienta Wine.</i>
Se distribuye bajo una licencia de software libre	<i>No. Se distribuye bajo un tipo de licencia Creative Commons que no admite la libertad de uso con fines comerciales.</i>	<i>Si, Dot-Graphviz se distribuye bajo la licencia pública Eclipse.</i>	<i>No, no permite el uso comercial.</i>

Tabla 1: Resumen de resultados del análisis comparativo de JFLAP, Boole-Deusto y PFM_UOC