

Herramienta de visualización de traductores dirigidos por la sintaxis

Ángel F. Sánchez-Granados
angelfrancisco.sanchez@urjc.es
Universidad Rey Juan Carlos
Móstoles, Madrid, España

Jaime Urquiza-Fuentes
jaime.urquiza@urjc.es
Universidad Rey Juan Carlos
Móstoles, Madrid, España

ABSTRACT

Para los estudiantes de grados afines a la Informática, el estudio de la traducción dirigida por la sintaxis suele resultar complejo de aprender. Para incrementar su motivación y mejorar su experiencia de aprendizaje, en este trabajo se presenta VisTDS, un software de visualización de traductores.

La herramienta, que permite al usuario un alto grado de interactividad, está orientada tanto a docentes como a estudiantes, y ofrece una representación gráfica de los conceptos con los que se trabajan de forma teórica.

Aunque este programa aún se encuentra en desarrollo, los resultados obtenidos hasta el momento son positivos, y han resultado de utilidad para definir las líneas de trabajo futuro.

KEYWORDS

Visualización de software, procesadores de lenguajes, enseñanza de la informática, traducción dirigida por la sintaxis, interacción persona-ordenador

1 INTRODUCCIÓN

Dentro de los programas de grado en Ciencias de la Computación, la asignatura de Procesadores de Lenguajes tiende a resultar difícil para el alumnado dada la complejidad para aplicar los conceptos abstractos que se tratan en situaciones reales. Dentro de esta disciplina, la traducción dirigida por la sintaxis (TDS) es una de las partes más importantes a la vez que compleja.

Por otra parte, se viene demostrando desde hace décadas que la aplicación de técnicas de visualización del software en entornos educativos resulta efectiva, teniendo un claro ejemplo en las animaciones de algoritmos de ordenación realizadas por Baecker [1] en la década de los 60. No obstante, además de la visualización como tal, es importante cómo se aplica [3] [5], ya que este tipo de herramientas resultan de utilidad para mejorar la motivación de los estudiantes [2].

En este trabajo se presenta desde el punto de vista interactivo VisTDS, una herramienta de visualización de traductores dirigidos por la sintaxis, cuyo propósito es ayudar al alumnado en su proceso de aprendizaje.

Este artículo se estructura de la siguiente forma: en la sección 2 se describe la herramienta, en la sección 3 se realizan presentaciones de las diferentes evaluaciones realizadas hasta el momento, y en la sección 4 se ofrecen las conclusiones y trabajos futuros.

2 DESCRIPCIÓN DE LA HERRAMIENTA

La herramienta desarrollada es un software de visualización de TDS cuyo objetivo es el de mejorar el proceso de aprendizaje de la TDS. Esta herramienta es agnóstica al lenguaje de especificación utilizado, evitando de este modo el aprendizaje de un nuevo lenguaje para su uso, y el poder portarlo a los generadores de traductores existentes.

El enfoque de la aplicación es doble: por un lado, pretende ser de utilidad al estudiante en su trabajo autónomo y, por otro lado, servir como herramienta de apoyo a los docentes en el aula.

Este software se compone de una API de anotación que permite generar la información de visualización partiendo de una gramática proporcionada por el usuario, y de una interfaz de visualización que interpretará dicha información.

El flujo completo de funcionamiento es el siguiente: en primer lugar, el usuario escribe una especificación utilizando uno de los lenguajes compatibles con VisTDS. A continuación, la especificación se anota con las sucesivas llamadas a las API, generando una *especificación del traductor anotada*. Tercero, el traductor es generado utilizando esta especificación anotada. Después, el traductor es ejecutado con la cadena de entrada proporcionada por el usuario, generando como resultado un fichero en formato XML con la información de visualización. Por último, este fichero XML se carga en el software de visualización. En este artículo se abordará únicamente la aplicación de visualización puesto es que con la que se produce la interacción.

La ventana principal de la interfaz gráfica se compone de tres partes: la especificación del traductor, la entrada a procesar y el árbol sintáctico enriquecido.

La especificación del traductor, ubicada en la parte superior derecha, muestra todas las reglas y sus acciones semánticas asociadas que componen la especificación. Como se puede observar, mediante el código de colores se pueden diferenciar los símbolos terminales y no terminales. Por otra parte, para facilitar la visualización de las reglas, las acciones semánticas se pueden mostrar u ocultar.

El panel de cadena de entrada muestra los tokens que componen la entrada del traductor y el estado de la ejecución a cada momento. En la figura, nótese que los ítems en color rojo son los leídos y los azules los pendientes de leer. Clicando en un token, el árbol de ejecución se situará en el estado correspondiente. Junto a este panel se encuentran los botones de navegación, que permiten navegar al inicio, al paso anterior, al paso siguiente y al final de la ejecución. De cara a un uso realmente autónomo de la aplicación por parte tanto de docentes como de estudiantes, será necesario permitir la carga de la propia especificación.

La zona principal de la ventana la ocupa el árbol sintáctico enriquecido. Como se ha mencionado, el árbol se va construyendo

a medida que el usuario interactúa con los controles apropiados (tokens de entrada, controles o atajos de teclado). Al igual que sucede con el panel de especificaciones, los símbolos terminales y no terminales, representados mediante los nodos, pueden ser distinguidos a simple vista por el color. Situando el cursor encima de cada nodo se puede consultar el valor del mismo, y al colocarlo encima de un rectángulo gris (que representa una regla) se resalta la regla asociada en el panel de especificación. Asimismo, si en alguno de estos puntos el docente ha incluido alguna anotación de interés, también se mostraría.

Hasta el momento, el proceso de anotación y el de visualización se realiza de forma independiente. Para mejorar la experiencia de usuario con el manejo de la herramienta, se encuentra en proceso de desarrollo una nueva versión del visualizador que permite desde la interfaz gráfica la carga del traductor y la introducción de la gramática para realizar automáticamente el proceso de anotación y carga que, como se ha mencionado anteriormente, es independiente y debe realizarlo el usuario manualmente.

Esta nueva versión, además de permitir la carga de ficheros XML con anotaciones existentes, tiene cargada la especificación de un traductor para que, con el objetivo de simplificar las pruebas, los usuarios puedan introducir una entrada que se ajuste a esta especificación y generar la visualización directamente.

3 EVALUACIÓN DE LA HERRAMIENTA

En el diseño de VisTDS participaron estudiantes con conocimientos contrastados de la asignatura, que ofrecieron un feedback que resultó muy útil para mejorar la interfaz de usuario.

Debido a las restricciones impuestas a raíz de la pandemia COVID-19, los experimentos controlados se vieron con la obligación de ser pospuestos. Por ese motivo, y de cara a validar si el diseño de VisTDS se ajusta a cumplir su objetivo, se realizó una evaluación analítica del sistema completo [7], del que se exponen resumidamente los resultados obtenidos desde el punto de vista de la interacción para validar la interfaz de usuario, y desde el punto de vista de los estudiantes para validar las capacidades pedagógicas. Finalizadas las restricciones, se ha podido realizar un experimento controlado con el objetivo de validar la utilidad real de la herramienta para el aprendizaje de la TDS. Ambos estudios se han realizado de forma previa a la nueva funcionalidad de introducción de la cadena de entrada.

3.1 Evaluación analítica desde el punto de vista de la interfaz

El estudio de este aspecto se realiza utilizando las reglas de evaluación heurística de Nielsen [6], puesto que es uno de los métodos más utilizados para el estudio de interfaces de usuario.

De este modo, se puede concluir que VisTDS se ajusta a las reglas "Visibilidad del estado del sistema", "Coincidencia entre el sistema y el mundo real", "Reconocer en lugar de recordar" y "Flexibilidad y eficiencia de uso". Por otra parte, la regla "Estética y diseño minimalista" se cumple parcialmente porque, por un lado, la interfaz se ha diseñado evitando elementos redundantes, y por otro lado, es necesario mejorar el zoom automático sobre los diferentes elementos de la interfaz. Por último, se debe mejorar en lo respectivo a la regla "Ayuda y documentación", ya que aunque existe un manual de usuario, ha de ser refinado y ofrecer mayor nivel de detalle.

3.2 Evaluación analítica desde el punto de vista pedagógico

Para la evaluación de este aspecto se han utilizado los principios descritos por Mayer[4] para el diseño de mensajes multimedia. Estos principios describen los requisitos que las herramientas educativas deberían cumplir para un aprendizaje significativo.

De este estudio se extrae que VisTDS cumpliría el "Principio multimedia", el "Principio de contigüidad espacial", el "Principio de personalización", el "Principio de interactividad" y el "Principio de señalización".

3.3 Evaluación empírica

Recientemente se ha realizado un experimento con estudiantes de la asignatura de Procesadores de Lenguajes cuyo objetivo era corroborar si VisTDS resulta de ayuda en el proceso de aprendizaje de la TDS, del que actualmente se están analizando los resultados.

Se ha llevado a cabo durante una sesión ordinaria de clase, habiendo un grupo de tratamiento que utilizó la herramienta y un grupo de control que no la utilizó, ubicados físicamente en diferentes campus de la universidad.

4 CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se ha descrito un software orientado al apoyo en el aprendizaje de la TDS. Actualmente se está desarrollando la funcionalidad de visualización de un traductor utilizando una entrada proporcionada por el usuario.

Posteriormente, se pretende que el software sea capaz de anotar automáticamente las especificaciones sin necesidad de que tenga que realizarlo de forma manual el usuario. Por otro lado, a nivel funcional, se plantea incrementar el número de generadores de gramáticas soportados por la API.

A largo plazo, se plantea liberar los códigos fuentes para permitir que la comunidad puedan mejorar el software. También se contempla el desarrollo de una versión cloud con interfaz web, de forma que los usuarios no tengan que instalar software en su equipo.

REFERENCES

- [1] R.M. Baecker. 1969. *Interactive Computer-Mediated Animation*. Ph. D. Dissertation. M.I.T. Department of Electrical Engineering.
- [2] Chris Evans and Nicola J. Gibbons. 2007. The interactivity effect in multimedia learning. *Computers & Education* 49, 4 (2007), 1147–1160. <https://doi.org/10.1016/j.compedu.2006.01.008>
- [3] C.D. Hundhausen, S.A. Douglas, and J.T. Stasko. 2002. A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing* 13, 3 (2002), 259–290.
- [4] R.E. Mayer. 2008. *Multimedia Learning*. Cambridge University Press.
- [5] Thomas L. Naps, Guido Rößling, Vicki Almstrum, Wanda Dann, Rudolf Fleischer, Chris Hundhausen, Ari Korhonen, Lauri Malmi, Myles McNally, Susan Rodger, and J. Ángel Velázquez-Iturbide. 2002. Exploring the Role of Visualization and Engagement in Computer Science Education. In *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education (Aarhus, Denmark) (ITiCSE-WGR '02)*. Association for Computing Machinery, New York, NY, USA, 131–152. <https://doi.org/10.1145/960568.782998>
- [6] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Seattle, Washington, USA) (CHI '90)*. Association for Computing Machinery, New York, NY, USA, 249–256. <https://doi.org/10.1145/97243.97281>
- [7] A.F. Sánchez-Granados, J. Urquiza-Fuentes, A. García-Oller, and J. M. Loeches-Ruiz. 2021. Evaluación analítica de una herramienta de visualización de traductores dirigidos por la sintaxis.