

Arquitectura de un Diccionario Multilingüe de Términos de Lógica

T. Alsinet¹, R. Béjar¹, C. Fernández¹, F. Manyà¹, G. Vázquez² y J. Tió²

¹Departament d'Informàtica

²Departament d'Anglès i Lingüística

Universitat de Lleida (UdL)

Jaume II, 69, E-25001 Lleida

e-mail:ramon@eup.udl.es

1 Introducción

En este artículo describimos la arquitectura de una herramienta *software* para la gestión de un diccionario multilingüe de términos de lógica. Esta aplicación la hemos diseñado e implementado en el marco del proyecto *Tools for Teaching Logic*. Para obtener más información sobre este proyecto ALFA de la Comisión Europea, así como un enlace al diccionario, se puede acceder al URL <http://aracne.usal.es>.

Los requerimientos funcionales de la aplicación son permitir (i) la creación de una base de términos lógicos en diferentes lenguas, (ii) la traducción de los términos entre lenguas, (iii) el acceso remoto a la aplicación a través de Internet utilizando un navegador y (iv) la unicidad del *software* y la simplicidad de administración del sistema.

El diseño de la aplicación permite aumentar dinámicamente tanto el número de términos lógicos como el número de lenguas. Actualmente, el diccionario contiene más de 500 términos y ofrece traducción a las siguientes lenguas: alemán, catalán, español, holandés, inglés, italiano, portugués y vasco. Para su creación, se partió de una lista de términos en inglés que facilitaron los socios ingleses del proyecto. El resto de socios realizaron la traducción a sus propias lenguas.

Finalmente, cabe mencionar que la herramienta es independiente del dominio y, por tanto, puede utilizarse para crear diccionarios multilingües de términos para otras disciplinas distintas de la lógica.

2 Arquitectura del diccionario

La aplicación ha sido desarrollada siguiendo la tecnología WWW (*World Wide Web*) [1], que es uno de los modelos de arquitectura más empleados actualmente en los sistemas de información distribuidos. En la tecnología WWW, las aplicaciones se dividen en una parte de datos que se almacenan y gestionan en el servidor, en una parte de procesamiento local o aplicación cliente y en una parte de interfaz gráfica de usuario que se ejecuta localmente mediante un navegador.

En nuestro caso, la base de datos almacena los términos del diccionario multilingüe. El acceso a los datos se realiza a través de la conexión a un gestor de base de datos relacional que se encuentra instalado en un ordenador que está permanentemente conectado a la red Internet y que actúa como servidor de la aplicación.

El modelo de datos de la aplicación ha sido definido siguiendo el modelo entidad-relación [2], normalmente utilizado como modelo conceptual en el diseño de sistemas de información [5, 8] basados en gestores de bases de datos relacionales que son interrogados utilizando el lenguaje SQL (*Structured Query Language*) [3]. En concreto, cada concepto lógico ha sido modelizado mediante un identificador único y un descriptor de concepto. Cada lengua ha sido representada como un conjunto de términos lógicos. Finalmente, cada concepto lógico está relacionado con uno o más términos de una misma lengua, y cada

término tiene un significado único, lo cual permite la traducción entre lenguas.

La aplicación cliente se encarga de modificar y consultar el contenido del diccionario de términos lógicos a través del acceso directo al servidor de la base de datos. La comunicación entre la aplicación cliente y el servidor se realiza, a través de la red Internet, empleando la interfaz de comunicación JDBC (*Java Database Connectivity*) [4, 6], que es un estándar industrial para el acceso a bases de datos relacionales en entornos de desarrollo basados en el lenguaje de programación Java. La única información que la aplicación cliente conoce sobre el servidor es su dirección IP (*Internet Protocol*) y el puerto de servicio, siendo el acceso a la base de términos lógicos totalmente transparente al usuario. Por tanto, la arquitectura WWW permite compartir una misma base de términos lógicos entre múltiples usuarios y crear diccionarios alternativos distribuidos en la red. Es decir, cada diccionario está físicamente ubicado en un servidor distinto, por lo que para cada área de la lógica se definen diccionarios especializados, cuyo mantenimiento es responsabilidad de diferentes grupos de usuarios.

La implementación de la aplicación cliente es un *applet* Java [7] que se ejecuta localmente y que el usuario descarga del servidor, cuando se conecta a la página web del diccionario, a través de un navegador con capacidad para ejecutar *applets* Java. El usuario deberá introducir su identificador y palabra de acceso (*password*) para iniciar la conexión entre el *applet* y el servidor de la base de datos. El tipo de operaciones que un usuario puede realizar sobre la base de datos del diccionario depende de los permisos que posee. El servidor se encarga de gestionar los permisos y, una vez ha identificado al usuario, comunica al *applet* las operaciones que aquél puede realizar. En particular, es necesario tener permiso para las operaciones de modificación, adición y eliminación de términos. La consulta se permite a todos los usuarios.

La aplicación ofrece una interfaz gráfica de usuario distinta para cada operación, que se caracteriza por su uniformidad y que ha sido diseñada para facilitar al usuario un acceso cómodo y rápido al diccionario de términos lógicos.

Agradecimientos

Esta investigación ha sido parcialmente financiada por los proyectos ALFA *Tools for Teaching Logic* y CICYT TIC96-1038-C04-03. El segundo autor disfruta de una beca del *Comissionat per a Universitats i Recerca* (1998FI00326). El primer prototipo del diccionario fue implementado por J. Planes.

Referencias

- [1] T. Berners-Lee. Information Management: A Proposal, 1989. Available at <http://www.w3.org/history/1989/proposal.html>.
- [2] C. J. Date. *An Introduction to Database Systems*. Addison-Wesley, sixth edition, 1995.
- [3] C. J. Date and H. Darwen. *A Guide to the SQL Standard*. Addison-Wesley, 1997.
- [4] G. Hamilton, R. Cattell, and M. Fisher. *JDBC Database Access with Java*. Addison-Wesley, 1997.
- [5] M. Page-Jones. *The Practical Guide to Structured Systems Design*. Prentice-Hall, 1988.
- [6] G. Reese. *Database Programming with JDBC and Java*. O'Reilly, 1997.
- [7] G. Steele, J. Gosling, and B. Joy. *The Java Language Specification*. Mc Graw-Hill, 1997.
- [8] E. Yourdon. *Modern Structured Analysis*. Prentice-Hall, 1989.