

Herramienta Colaborativa Multidispositivo para la Edición de Modelos basada en EMF

Miguel A. Teruel, Arturo C. Rodríguez, Elena Navarro, Pascual González

LoUISE Research Group, Universidad de Castilla-La Mancha
{MiguelAngel.Teruel, Arturo.Rodriguez, Elena.Navarro,
Pascual.Gonzalez}@uclm.es

Resumen. En este trabajo se presenta un editor de modelos que permitirá el trabajo colaborativo con modelos pertenecientes a cualquier metamodelo creado con EMF. Además, este editor permite su uso desde una gran variedad de dispositivos, facilitando una colaboración efectiva gracias al *awareness* que proporciona tanto sobre el propio proceso de edición como de los usuarios que en ella colaboran. Además, la implementación como servicios webs de la gestión de modelos EMF y del soporte a características de *awareness* permite su reutilización en futuras aplicaciones.

Palabras clave: editor modelos, colaboración, multidispositivo, EMF, Workspace Awareness, servicio web

1 Introducción

Durante los últimos años, estamos viendo cómo la colaboración entre usuarios está llegando a todo tipo de aplicaciones. Ya sean aplicaciones clásicas, como editores de texto [1] y juegos online [6], o las vanguardistas aplicaciones Post-WIMP [3]. Además de éstas, otras aplicaciones orientadas a dominios más específicos, como la edición de modelos, también están siendo adaptadas para que varios usuarios puedan trabajar colaborativamente [5].

No obstante, los editores colaborativos actuales están enfocados a editar modelos que son instancias de un metamodelo concreto, limitando su aplicación a otros dominios o contextos de interés. Por esa razón, en este trabajo se presenta un editor de modelos colaborativo basado en Eclipse Modeling Framework (EMF) [7]. Gracias a su integración con EMF (transparente al usuario), el editor de modelos permite su adaptación a cualquier metamodelo que haya sido diseñado con dicho framework. Dicha integración se realiza por medio de un servicio web que permite independizar la plataforma y tecnología de la aplicación cliente. Junto a este servicio web se ha implementado otro servicio encargado de dar soporte a la colaboración, proporcionando a los usuarios el *awareness* (o consciencia) necesario, tanto de los otros colaboradores como de los elementos de los modelos que están siendo manipulados.

Este trabajo se estructura de la siguiente manera. Tras esta introducción, en la Sección 2, se presentará el servicio web encargado de la gestión de los modelos EMF. A continuación, la Sección 3 presenta el servicio web que ofrece soporte al *awareness*.

La aplicación colaborativa cliente que consume dichos servicios se presenta en la Sección 4. Finalmente, la Sección 5 presenta nuestras conclusiones y trabajos futuros.

2 Arquitectura de un Servicio Web basado en EMF

Eclipse Modeling Framework (EMF) [7] es uno de los frameworks de modelado y generación de código más extendidos en el Desarrollo Dirigido por Modelos. Esta tecnología, desarrollada en Java, se integra en el IDE Eclipse y permite crear, editar e instanciar modelos e implementar de forma automática un esqueleto de código a partir de los mismos. Además, a diferencia de otras herramientas de este tipo, EMF está diseñado para ofrecer de forma real un punto de partida para una programación eficiente. La definición del metamodelo que guía el proceso puede hacerse mediante tres métodos: (1) Esquema XML, (2) diagrama de clases UML y (3) interfaces Java.

Una vez definido el metamodelo, EMF genera de forma automática varios componentes que permiten al usuario la instanciación y el manejo de dichos modelos. Entre estos componentes podemos encontrar un plug-in para Eclipse que permite crear instancias del metamodelo a través de un wizard y las clases java que permiten manejar los elementos del modelo a nivel de programación.

Esta forma de trabajar, aunque no carece de una enorme potencia, conlleva dos limitaciones. Por un lado, requiere que el usuario tenga conocimiento acerca del uso de Eclipse. Esto hace que usuarios no ligados de forma directa a la informática encuentren dificultades importantes a la hora de trabajar con la edición de modelos. A fin de solucionar esta limitación, es posible construir software que, usando de base EMF, ofrezca una interfaz adecuada para este tipo de usuarios. Sin embargo, esto nos llevaría a encontrar una segunda limitación, ya que estas aplicaciones standalone sólo pueden ser desarrolladas en Java y son desarrolladas exclusivamente para un metamodelo concreto.

Por ello, hemos desarrollado un servicio web que ofrece una vía de comunicación entre aplicaciones destinadas a trabajar con modelos y el framework EMF (Fig. 1). De esta manera, es posible desarrollar aplicaciones cuya interfaz de usuario y lógica no asociada a la gestión de modelos son totalmente independientes de todo lo relacionado con éstos, permitiendo mejorar así la portabilidad dado que se pueden integrar mediante cualquier tipo de tecnología en esas partes de la aplicación.

De esta manera, sólo el servicio web hace uso de las clases de EMF, actuando sus métodos como una capa intermedia que consumen las peticiones externas, usan el

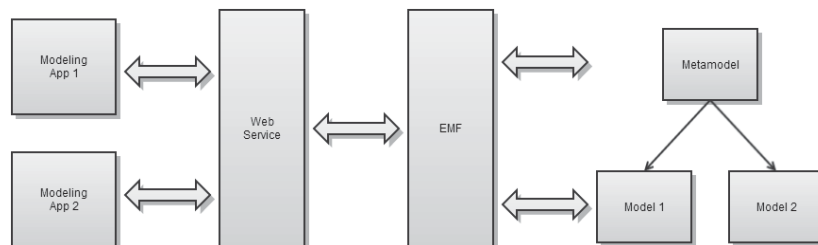


Fig. 1. Arquitectura del servicio web basado en EMF

framework para consultar el modelo en función de la información requerida y devuelven dicha información en formato XML. El servicio web ha sido implementado usando Axis e incluyendo las librerías necesarias del framework EMF en el servidor Apache Tomcat en el que ha sido desplegado. Dicho servicio web se ha desarrollado agrupando los diferentes métodos en dos grupos distintos:

- *Métodos de consulta del metamodelo*: con estos métodos las aplicaciones clientes pueden obtener información sobre las metACLases y las relaciones del metamodelo, saber si una metACLase es abstracta, las cardinalidades de las relaciones, etc.

```
public String getClassInfo(String model,String className,String info)
public String getReferenceInfo(String model,String element,
String reference,String info)
```

El primer método devuelve información concreta de una metACLase del modelo. Para ello se pasa al servicio web el nombre del modelo, el nombre de la clase y la información requerida. El segundo método devuelve información acerca de una referencia del metamodelo. En este caso, además del modelo, la referencia y la información requerida, es necesario saber el elemento del que parte la referencia.

- *Métodos de gestión del modelo*: estos métodos permitirán consultar y modificar objetos del modelo o crear nuevas instancias del mismo.

```
public getElement(String model,String type,String id)
public createElement(String model,String parent,String type,String id)
```

El primer método recupera un objeto del modelo mientras que el segundo crea un nuevo objeto en el modelo. El servicio web se ocupa también de la persistencia del modelo. De esta forma la edición de un modelo determinado no está ligada a un solo usuario, sino que puede hacerse de manera colaborativa.

3 Servicio Web para dar soporte a Workspace Awareness

Paralelamente al desarrollo de la colección de servicios web para EMF, ha sido implementada una nueva biblioteca de métodos web con el objeto de mejorar la interacción colaborativa de los usuarios. Concretamente, este servicio web da soporte a la interpretación *Workspace Awareness* (WA) propuesta por Gutwin [2]. Mediante WA, se pretende dar respuesta a una serie de elementos de los cuales un usuario necesita ser consciente cuando trabaja con un sistema colaborativo. Por ejemplo, WA plantea elementos de consciencia como *quién* hay disponible en el sistema, *qué* están haciendo los demás, *dónde* están trabajando, *cuando* realizaron una determinada acción, etc. Concretamente, los elementos que recoge WA pueden verse en la Tabla 1.

Tabla 1. Elementos de Workspace Awareness

Tiempo	Categoría	Elemento	Preguntas específicas
Presente	Quién	Presencia Identidad Autoría	¿Hay alguien en el espacio de trabajo? ¿Quién está participando? ¿Quién está haciendo eso?
	Qué	Acción Intención Artefacto	¿Qué están haciendo? ¿Con que objetivo realiza esa acción? ¿Con que objeto están trabando?
	Dónde	Localización Mirada Vista Alcance	¿Dónde están trabajando? ¿Dónde están mirando? ¿Qué pueden ver? ¿Qué pueden alcanzar?
Pasado	Cómo	Histórico de acciones Histórico de artefactos	¿Cómo ocurrió eso? ¿Cómo llegó el artefacto a ese estado?
	Cuándo	Histórico de eventos	¿Cuándo ocurrió ese evento?
	Quién	Histórico de presencia	¿Quién estuvo aquí y cuándo?
	Dónde	Histórico de localización	¿Dónde ha estado esa persona?
	Qué	Histórico de acción	¿Qué ha estado haciendo esa persona?

Los elementos anteriormente mostrados han sido implementados como un servicio web Axis2 con motor de BBDD MySQL. Al igual que ocurría con el servicio de edición de modelos, la utilización de tecnología de servicios web permite implementar aplicaciones que necesiten soporte de WA utilizando cualquier tecnología. A continuación, a modo de ejemplo, se muestran los métodos correspondientes a los dos primeros de los elementos de WA relacionados con el presente (*quién – presencia y quién - identidad*):

```
public bool isAnyone()
public Participant[] whoIsParticipating()
public void addParticipant(Participant p)
public void removeParticipant(Participant p)
```

El primero de estos cuatro métodos web (isAnyone) proporciona información acerca de si hay alguien disponible en el sistema con quien colaborar, mientras que el segundo (whoIsParticipating) informa acerca de quiénes son esos participantes. Para el correcto funcionamiento de estos métodos web, otros dos métodos auxiliares han sido implementados (addParticipant y removeParticipant). El primero de estos dos métodos se utiliza para comunicar al servidor que un participante se ha conectado al sistema, mientras que el segundo permite eliminar a un participante.

4 Cliente multidispositivo para la edición de Modelos

Una vez se han definido los servicios web que permiten la edición de modelos con soporte a los diferentes elementos de WA, se presenta a continuación la aplicación cliente creada para su utilización. Concretamente, dos aplicaciones cliente usuario distintas han sido implementadas. La primera de ellas se basa en una interfaz web pensada tanto para dispositivos de escritorio como para móviles, en la que se centra



Fig. 2. Editor de modelos ejecutado sobre distintos dispositivos

este trabajo, mientras que la segunda está pensada para ser usada por hasta cuatro usuarios simultáneamente en una mesa Microsoft PixelSense.

Este editor de modelos ha sido implementado como aplicación web con el fin de que pueda ser soportado por una gran variedad de dispositivos (Fig. 2). Concretamente, fue implementado usando una combinación de tecnologías web como ASP.net y JavaScript, las cuales han permitido una total integración con los dos servicios web desarrollados, así como dotando a la aplicación de compatibilidad con cualquier dispositivo que soporte HTML5.

Gracias a esta aplicación, varios usuarios pueden trabajar de forma colaborativa sobre un mismo modelo instancia de un metamodelo EMF. La aplicación cliente accede al modelo mediante el servicio web detallado en la sección 2. Esto permite la separación de la interfaz de la aplicación del metamodelo concreto empleado, haciendo así la aplicación compatible con cualquier metamodelo diseñado con EMF. Además, es importante resaltar que dicha aplicación se ha desarrollado utilizando las heurísticas descritas en un trabajo previo [4]. Estas heurísticas se han definido con dos objetivos distintos: (i) identificar metáforas que asocian tipos de gestos con tareas concretas de edición; (ii) permitir, gracias a dichas metáforas, que personas sin conocimientos de edición de modelos puedan participar en dicho proceso.

Finalmente, esta aplicación utiliza el servicio web descrito en la sección 3, para permitir a los usuarios trabajar colaborativamente proporcionándoles el awareness necesario tanto de los usuarios como de los elementos del modelo que estén editando. De hecho, directamente ligadas al awareness, la aplicación dispone de las siguientes capacidades relacionadas con el WA descrito por Gutwin (ver Tabla 1):

- *Usuarios conectados*: Muestra una lista de los usuarios conectados (*quién - identidad*). Además, el icono del menú de usuario cambia en función de si hay alguien más en el sistema, mostrando uno o varios usuarios (*quién - presencia*).
- *Historial de usuarios*: Mediante el menú de usuario se puede obtener información acerca qué está haciendo un usuario (*qué - acción*), así como obtener una lista de las acciones pasadas (*qué - histórico de acción*).

- *Historial de elementos*: Al igual que ocurre con los usuarios, pueden consultarse todos los cambios que han acontecido sobre un elemento de un modelo (*cómo – histórico de artefacto*).

5 Conclusiones y Trabajos Futuros

En este trabajo se presenta una propuesta la edición de modelos colaborativa multi-dispositivo. Para dar a dicha edición de modelos colaborativa, dos servicios web (que pueden ser reutilizados por otras aplicaciones) han sido desarrollados. Uno de dichos servicios permite la edición de modelos EMF de forma independientemente de la plataforma o tecnología utilizada. El otro servicio desarrollado soporte la interpretación de WA definida por Gutwin para permitir el trabajo colaborativo.

Paralelamente a esta aplicación multidispositivo, se está desarrollando una segunda que se ejecuta sobre Microsoft PixelSense, haciendo uso de los servicios web descritos. De esta manera, las dos aplicaciones podrán comunicarse, permitiendo a los usuarios de ambas trabajar de forma colaborativa sobre un mismo modelo.

Agradecimientos. Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad, así como por los fondos FEDER de la Unión Europea mediante el proyecto inspire (TIN2012-34003). Además, ha sido financiado por el Ministerio de Educación, Cultura y Deporte mediante una beca FPU (AP2010-0259). Nos gustaría agradecer el trabajo desarrollado por Almudena Alcaide Sánchez y Miguel Saiz Sánchez en el transcurso de su Trabajo Fin de Grado y que ha ayudado a hacer realidad las ideas aquí presentadas.

Referencias

1. Dekeyser, S., Watson, R.: Extending Google Docs to Collaborate on Research Papers. , Toowoomba, Australia (2006).
2. Gutwin, C., Greenberg, S.: A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Comput. Support. Coop. Work* 11 (3), 411–446 (2002).
3. Jetter, H.-C., Zöllner, M., Gerken, J., Reiterer, H.: Design and Implementation of Post-WIMP Distributed User Interfaces with ZOIL. *Int. J. Hum. Comput. Interact.* 28 (11), 737–747 (2012).
4. Lopez-Jaquero, V., Navarro, E., Montero, F., González, P.: Metamodels Infrastructure and Heuristics for Metamodel-Driven Multi-touch Interaction. 14th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2013). pp. 210–227, Springer, Cape Town, South Africa (2013).
5. Mougnot, A., Blanc, X., Gervais, M.-P.: D-Praxis : A Peer-to-Peer Collaborative Model Editing Framework. 9th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS'09). pp. 16–29, , Florence, Italy (2009).
6. Nardi, B., Harris, J.: Strangers and friends: collaborative play in world of warcraft. 20th anniversary conference on Computer supported cooperative work (CSCW'06). p. 149, ACM Press, Banff, Canada (2006).
7. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. Addison-Wesley Professional (2008).