

JSEM-HP, una herramienta de autor para el desarrollo de sistemas hipermedia adaptativos evolutivos

Fernando Molina Ortiz¹, Nuria Medina Medina², Lina García Cabrera³

^{1,2}Dpto. Lenguajes y Sistemas Informáticos. Universidad de Granada.

³Dpto. Informática. Universidad de Jaén.

¹fmo@ugr.es ²nmedina@ugr.es ³lina@ujaen.es

Resumen En este artículo se muestra el prototipo JSem-HP, una herramienta de autor para el desarrollo de sistemas hipermedia evolutivos adaptativos basada en el modelo SEM-HP. Esta herramienta, realizada en Java, permite al autor definir sistemas hipermedia mediante la creación de estructuras conceptuales modeladas mediante redes semánticas, elegir distintas vistas o presentaciones de éstas, definir las formas en las que éstas pueden ser navegadas y especificar la adaptación al usuario que el sistema llevará a cabo, la cual tiene en consideración el conocimiento del usuario y cómo éste se incrementa con la navegación.

1. Introducción

Estamos construyendo una herramienta de autor para el desarrollo de sistemas hipermedia adaptativos y evolutivos, basados en el modelo SEM-HP [2,5], aunque la versatilidad del diseño ha permitido utilizar el subsistema de memorización en [3] y [6]. En este artículo se hace una descripción somera de la herramienta, continuando y actualizando lo descrito en [4]. En la sección 2 se dan las ideas principales del modelo SEM-HP en el que se basa la herramienta, para pasar a describir someramente el diseño de la herramienta en la sección 3. La sección 4 muestra al prototipo en funcionamiento, y se finaliza con las conclusiones y el trabajo futuro en la sección 5.

2. El modelo SEM-HP

El modelo SEM-HP es un modelo sistémico, evolutivo y semántico para el desarrollo de sistemas hipermedia adaptativos [2,5]. En este modelo un sistema hipermedia esta formado por tres sistemas interrelacionados: conocimiento, navegación y aprendizaje, los cuales además son evolutivos.

2.1. Sistema de conocimiento: memorización y presentación

El *sistema de conocimiento*, que almacena y mantiene el conocimiento del autor tiene dos partes: el *subsistema de memorización*, que estructura el dominio

conceptual y de información mediante una *estructura conceptual de memorización* (ECm), y el *subsistema de presentación* que permite crear diferentes vistas de ésta, a las que llamamos *estructuras conceptuales de presentación* (ECp). Una ECm es una red semántica con dos tipos de nodos: conceptos (ideas etiquetadas) e items (trozos de información). Los conceptos se relacionan entre ellos mediante asociaciones conceptuales, y los conceptos con items mediante asociaciones funcionales. Una asociación se etiqueta mediante una relación que tiene un nombre y una serie de propiedades asociadas (como ciclicidad, transitividad, etc). Los nodos tienen un conjunto de atributos: los conceptos una etiqueta y los items otros como su identificador, el autor, fecha de creación, etc.

2.2. Sistema de navegación

El *sistema de navegación* determina cómo se puede navegar a través de las relaciones conceptuales, reduciendo o extendiendo su navegabilidad en uno o ambos sentidos, y mediante una serie de reglas de orden modificables por el autor formalizadas en lógica temporal. Para cada ECp, el autor puede definir distintas *estructuras conceptuales de navegación* (ECn), que definen diferentes formas de navegación.

2.3. Sistema de aprendizaje

El *sistema de aprendizaje* se encarga de la adaptación al usuario, asistiendo a éste en su interacción con el sistema adecuando la navegación a las necesidades de información y los conocimientos que va logrando el lector. Para esto el autor define las reglas de peso, que calculan el conocimiento que se tiene sobre un concepto a partir del de los items asociados a él, las reglas de conocimiento, que definen qué items y qué nivel de conocimiento ha de tener el usuario para poder acceder a determinado item, y las reglas de actualización, que modelan cómo el conocimiento del usuario crece a medida que éste visita los items de información. Mediante este subsistema el autor puede definir distintas *estructuras conceptuales de aprendizaje* (ECI) para cada ECn. El sistema gestiona un modelo de usuario, que almacena, entre otras cosas, el nivel de conocimiento del usuario, su experiencia en la materia y en el uso del sistema hipertexto, sus preferencias y sus objetivos. Éste modelo de usuario se va actualizando a medida que el usuario navega, o a petición de éste.

2.4. Tipos de navegación

El modelo SEM-HP permite cuatro tipos de navegación: a) Libre, en la que el usuario navega sobre una ECp y siempre puede acceder a todos los items de ésta. b) Por relación conceptual, en la que se navega sobre una ECn, ajustándose a los enlaces de navegación y a las reglas de orden. c) Por conocimiento, asociada a una ECI en la que sólo son accesibles los items para cuya comprensión el usuario está preparado. d) Resumen de conceptos, en la que el usuario navega una ECm completa, y para cada concepto se muestra un resumen compuesto a partir de los items asociados a éste.

2.5. Evolución

Para dotar de capacidad de *evolución* a los sistemas desarrollados el modelo incorpora la figura del Metasistema. Este proporciona al autor un conjunto de acciones evolutivas (*ACe*) que le permiten modificar los componentes del sistema. Cada *ACe* tiene un conjunto de restricciones que el Metasistema comprobará para determinar si ésta podrá ser llevada a cabo de forma consistente. En ocasiones un cambio en un subsistema genera la necesidad de nuevos cambios en ese subsistema o en otros. Para ésto el Metasistema realiza de forma automática la propagación interna y externa del cambio.

3. Diseño e implementación de JSEM-HP

En esta sección se describe la arquitectura básica del prototipo y sus principales clases. Como se justifica en [4], en la implementación se utiliza el lenguaje Java y la librería de manipulación de grafos JGraph ([1]).

3.1. Estructura general

En la figura 1 se pueden ver las principales clases del prototipo, excluyendo entre otras las utilizadas en el interfaz gráfico. El interfaz *ConceptualStructure* representa una estructura conceptual (EC) de cualquiera de los subsistemas. Éste hereda de *EvolutionarySystem* que define las características que para nosotros ha de tener un modelo para ser evolutivo y permite al Metasistema (*MetaSystem*) aplicar acciones evolutivas sobre él (estas características incluyen la capacidad de acceso orientado a transacciones, de forma que si la postcondición de un acción evolutiva no se cumple podamos volver el sistema a un estado consistente). *CSGraphRO* define las operaciones de acceso al grafo (red semántica) asociado a una EC, y *SemanticDomain* recoge la información asociada a un dominio semántico [4], que se refiere a una parcela de conocimiento común, caracterizada por los atributos de los nodos y de las estructuras conceptuales, las relaciones conceptuales disponibles, etc. *AttributeValuePairs* modela elementos que pueden tener un conjunto de pares atributo-valor, como puede ser el nombre de una EC o de un concepto, la fecha de creación de un ítem, etc. La clase *LogicExpression* representa a un completo subsistema lógico que permite la creación mediante un interfaz gráfico de expresiones lógicas de forma genérica, el cual se utiliza para la reglas de orden, conocimiento y actualización. Las clases *CMem*, *CSpres*, *CSnav* y *CSlearn* representan EC de memorización, presentación, navegación y aprendizaje respectivamente. Observamos que *CMem* está implementada utilizando clases de JGraph, aunque definiendo nuestros propios interfaces para grafos, nodos y asociaciones para hacer la herramienta más independiente de la librería utilizada. La clase *CSbrowse* representa una EC que está siendo navegada por el usuario. No creemos necesario detallar el resto de las clases mostradas en la figura, que, como las descritas hasta ahora implementan conceptos expuestos en la sección 2 de una forma casi directa. En cuanto a la evolución, se sigue el esquema descrito en [4], que emplea una clase para modelar cada acción evolutiva.

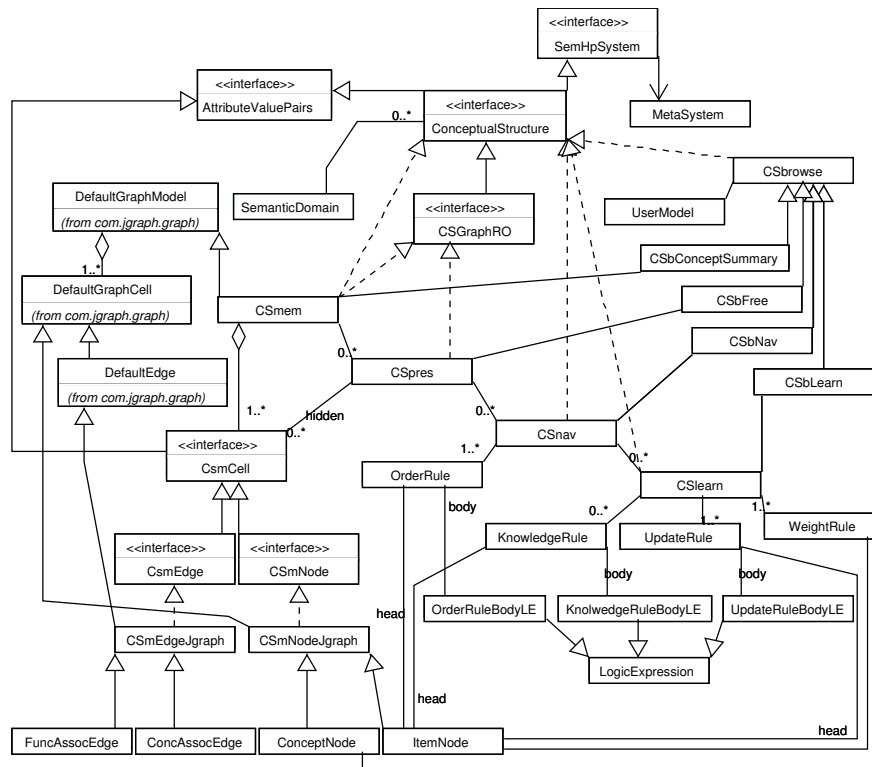


Figura1. Principales clases de SEM-HP

4. Ejemplo de utilización

En esta sección mostramos un ejemplo muy sencillo de uso del prototipo. En la figura 2, pantalla de la izquierda podemos observar una ECM al ser creada por el autor, en la que se están editando el item I1 y una asociación funcional. En la pantalla de la derecha vemos cómo el autor crea una ECp a partir de la ECM anterior, ocultando algunos de los elementos de ésta.

La figura 3 nos muestra a la izquierda cómo el autor puede extender o limitar la navegabilidad de las asociaciones conceptuales en el subsistema de navegación, el cual también permite editar las reglas de orden. A la derecha, vemos el subsistema de aprendizaje, en el que el autor está creando una ECl asociada a la EC anterior. En el ejemplo, el autor está editando una regla de conocimiento asociada al item I2 (historia de la Alhambra), que indica que para visitarlo el usuario ha de tener un conocimiento sobre I3 (introducción a Granada) mayor o igual que bajo, y sobre I1 (historia de Granada) mayor que medio.

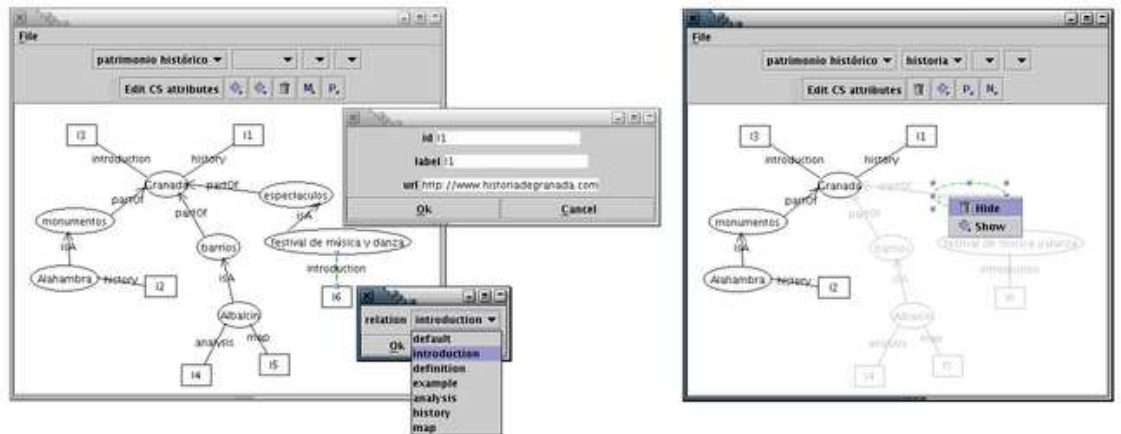


Figura2. Subsistema de memorización. Subsistema de presentación

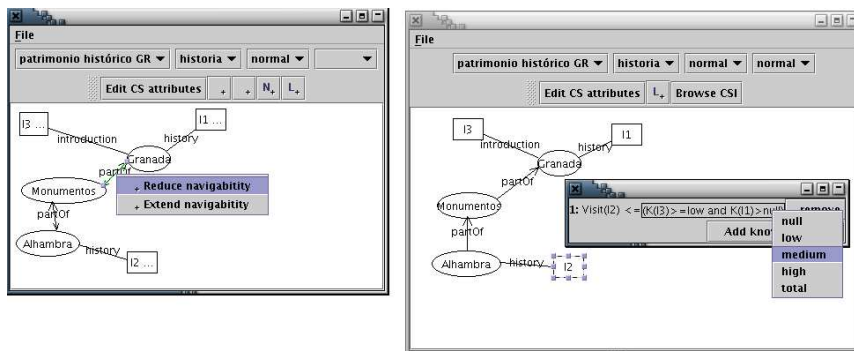


Figura3. Subsistema de navegación. Subsistema de aprendizaje

En la figura 4 se ve cómo el usuario navega por conocimiento la ECI de la figura 3, suponiendo que ha señalado como meta adquirir un conocimiento total sobre I2. Al seleccionar el ítem I3, éste se visualiza en el navegador de la derecha, marcándose este ítem algo más oscuro para indicar que el conocimiento sobre éste ha aumentado. El ítem I2 está oculto y deshabilitado, ya que el usuario todavía no tiene el conocimiento necesario sobre I1 para poder verlo, según la regla de conocimiento mostrada antes. El ítem I1 está marcado como interesante, ya que conocer más sobre éste llevará al usuario a poder visitar I2 y alcanzar su meta.

5. Conclusiones y trabajo futuro

Se ha dado una introducción al modelo SEM-HP, describiendo un prototipo de herramienta de autor basado en el modelo. La herramienta permite la creación

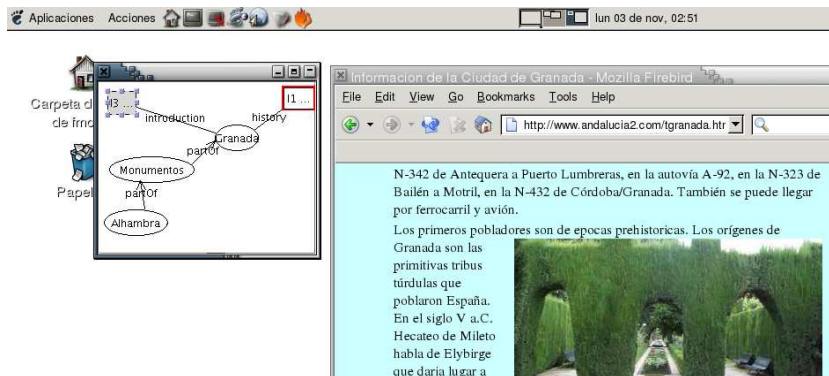


Figura4. Navegación por el usuario

de sistemas hipermedia de acuerdo con el modelo, y se ha intentado hacer flexible, de forma que sea sencillo añadir nuevas acciones evolutivas, nuevos atributos a las EC o a los elementos de ésta, etc. Este afán de flexibilidad se extiende a la implementación de un subsistema lógico extensible que permite la edición interactiva de distintos tipos de fórmulas lógicas, incluyendo las reglas de orden, actualización y conocimiento.

El trabajo futuro se centra en finalizar algunos aspectos de la implementación del prototipo, y en utilizar éste en casos reales para validar el modelo SEM-HP tanto en la creación y evolución de sistemas hipermedia como en la adaptación al usuario.

Referencias

1. Alder, Gaudenz. "The JGraph Swing Component". <http://www.jgraph.com>
2. García-Cabrera, L; Rodríguez-Fórtiz, MJ; Parets-Llorca, J. "Evolving hypermedia systems:a layered software architecture" *Journal of Software Maintenance and Evolution: Research and Practice*. John Wiley & Sons, Ltd. 17 pgs.
3. Hurtado-Torres, M.V.; Parets-Lorca J. *Evolutionary Information and Decisión Support Systems: An integration Based on Ontologies*. Lecture Notes in Computer Science LNCS 2178, pp 146-159, 2001.
4. Molina Ortiz F.; García Cabrera L.; Medina Medina N.; Hurtado Torres M^aV. "Editor de Estructuras Conceptuales Evolutivas: Consideraciones Prácticas". III Jornadas de trabajo DOLMEN. Pp 77-83. VII Jornadas de Ingeniería del Software y Bases de Datos. El Escorial, 18 de Noviembre de 2002.
5. Medina-Medina, N.; García-Cabrera, L.; Rodríguez-Fortiz, M.J.; Parets-Llorca, J. *Adaptation in an Evolutionary Hyperpedia System: Using Semantic and Petri Nets. Adaptative Hypermedia and Adaptative Web-Based Systems (second international conference, AH 2002)*. En *Lectures Notes in Computer Science*. Vol. 2347. Pp: 284-295. Ed LNCS Editorial, Germany. ISSN: 0302-9743
6. Paderewsky-Rodríguez, P.; *Un Modelo Arquitectónico Evolutivo para Sistemas Software Basados en Agentes*. Tesis Doctoral. Universidad de Granada, 1995.