

Diseño interactivo de cursos adaptativos

J. A. Macías y P. Castells

E.T.S. de Informática, Universidad Autónoma de Madrid

Palabras clave: herramientas de autor, diseño interactivo, software adaptativo, modelo de tareas, modelo de usuario, enseñanza a distancia, interacción persona-computador

Resumen: La capacidad de las aplicaciones educativas para adaptarse al estudiante es una cualidad perseguida desde hace tiempo en el campo de la enseñanza asistida por ordenador. La dificultad que entraña la elaboración de este tipo de productos, y la ausencia de herramientas adecuadas que faciliten esta labor, obstaculizan la participación de instructores y docentes en la elaboración del material. En este artículo presentamos una herramienta de autor, ATLAS, cuyo objetivo es conciliar potencia y facilidad de uso en el diseño de cursos web adaptativos. ATLAS permite la construcción totalmente interactiva de cursos que se adaptan automáticamente a las características del alumno y su comportamiento durante la realización del curso. El diseñador interactúa con la herramienta mediante un lenguaje visual intuitivo basado en la manipulación directa de los elementos implicados en el curso, de forma que la herramienta realiza la transición entre la forma de entender el curso por parte de un profesor, y el modelo de representación del sistema subyacente.

1. INTRODUCCIÓN

La facilidad de difusión que permiten las nuevas redes de comunicación hace que las aplicaciones educativas se enfrenten a una variedad de usuarios y plataformas cada vez mayor. Puesto que es imposible satisfacer las preferencias y necesidades de todos los estudiantes mediante una solución única, se hace necesario producir software flexible, adaptable a distintos tipos de usuario y situación. Dada su complejidad, este tipo de software es difícil de desarrollar, lo cual dificulta la participación directa y activa de los

propios profesionales de la enseñanza en la elaboración del producto, en detrimento del valor pedagógico del mismo.

Existen hoy en día una amplia variedad de herramientas de autor interactivas y editores gráficos de fácil utilización, como WebCT [1], Macromedia Director, o los editores de páginas web, por citar algunos, que simplifican considerablemente el desarrollo del material educativo. No obstante estas herramientas no proporcionan un soporte adecuado para los aspectos dinámicos propios de un curso adaptativo. Además, el tipo de asistencia que proporcionan tiende a estar limitado al trabajo aislado sobre aspectos parciales de un curso, o un subconjunto de ellos, como componentes multimedia, simulaciones, páginas web, índices estáticos, o repositorios de documentos, de manera que la articulación dinámica de todos estos elementos se lleva a cabo mediante herramientas suplementarias no interactivas, o lenguajes de programación.

Por otra parte, durante los últimos años se ha desarrollado otro tipo de herramientas más potentes que sí permiten la construcción de cursos interactivos sofisticados con un gran nivel de flexibilidad, incluyendo capacidades de adaptación al estudiante, utilizando un conjunto relativamente pequeño de primitivas de alto nivel [2, 3, 4]. Sin embargo estas herramientas no son fáciles de utilizar en general, ya que requieren el aprendizaje de lenguajes de especificación propios, y la comprensión de conceptos abstractos no triviales para quien no está necesariamente familiarizado con las complejidades del desarrollo de software. La complementación de este tipo de herramientas con editores basados en lenguajes visuales intuitivos, para aislar al diseñador de la complejidad del sistema subyacente, es un problema difícil cuando se trata de construir cursos dinámicos, ya que el curso como tal no existe en tiempo de diseño.

En este artículo se describe una herramienta de diseño, ATLAS (Authoring Tool for Adaptive Software design), para la construcción interactiva, en un entorno gráfico de edición, de cursos con capacidad de adaptación dinámica al alumno. ATLAS permite el diseño integrado, en un mismo entorno, de los distintos aspectos de los cursos y las relaciones entre ellos: estructura, contenido, presentación, y perfiles de estudiante. Para reducir la necesidad de abstracción por parte del autor, la herramienta ofrece la opción de trabajar en tiempo de diseño sobre una representación de los cursos próxima al resultado final, a partir de ejemplos concretos del perfil de usuario, proporcionados por el diseñador.

Nuestro sistema pretende poner al alcance de profesores y expertos en la materia las técnicas avanzadas de construcción de cursos adaptativos desarrolladas recientemente en el campo de la enseñanza asistida por ordenador. ATLAS combina la potencia expresiva de un sistema basado en la modelización abstracta del estudiante y sus tareas de aprendizaje, con la

facilidad de uso de la especificación gráfica basada en la manipulación directa, evitando que el diseñador tenga que utilizar lenguajes de especificación textuales. ATLAS está siendo desarrollado sobre la base de otra herramienta preexistente, TANGOW [4], que se utiliza como sistema soporte para la modelización y ejecución de los cursos adaptativos definidos por el diseñador en el entorno ATLAS (ver figura 1). La versión actual utiliza la web como plataforma para la ejecución de los cursos.

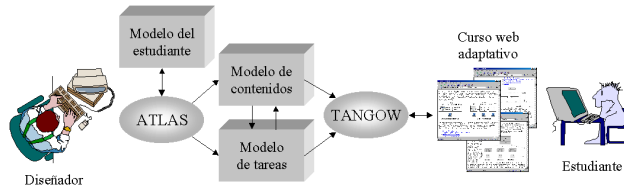


Figura 1. Arquitectura del sistema

2. ANTECEDENTES

La tecnología para el soporte de sistemas hipermedia adaptativos ha progresado considerablemente durante la última década, en particular en el campo de la enseñanza a distancia, con sistemas como DCG [2], ELM-ART [3] y TANGOW [4]. ELM-ART y DCG utilizan una representación explícita de la estructura de conceptos de la materia, en forma de grafo. DCG incluye un planificador que dirige el camino a seguir por el alumno para alcanzar determinados conceptos que se fijan como objetivo, a partir de los conceptos que ya conoce. ELM-ART tiene un sofisticado sistema para estimar el conocimiento adquirido por el usuario en relación con el mapa de conceptos del curso, en función de lo cual el sistema propone de forma dinámica el camino a seguir por el alumno en cada momento. Mientras que en los sistemas anteriores la estructura del curso es fija, siendo el itinerario del estudiante lo que varía, TANGOW genera la estructura del curso en tiempo de ejecución. TANGOW modeliza la actividad del estudiante en forma de una jerarquía de tareas, unidades didácticas, que el estudiante puede realizar. A cada tarea corresponde un conjunto de contenidos en HTML, a partir de los cuales TANGOW compone páginas web en las que se realizan las tareas. La jerarquía de tareas establece que la realización de algunas tareas consiste en llevar a cabo otras subtareas, que a su vez pueden o no ser compuestas. La

estructura de este árbol de tareas es dinámica, y depende de las características del estudiante y su comportamiento durante la realización del curso.

Desde el punto de vista del diseñador, la modelización de cursos adaptativos en TANGOW se puede asimilar a la descripción de las tareas de usuario en ciertas herramientas de construcción de interfaces de usuario basados en modelos, como UIDE [5], MASTERMIND [6] y MOBI-D [7]. Estos sistemas permiten la construcción de interfaces, en un contexto más amplio que el estrictamente educativo, a partir de una modelización de las tareas que el usuario debe poder realizar con la aplicación. Tanto UIDE como MOBI-D incluyen facilidades para la manipulación gráfica de las tareas del usuario. MASTERMIND no proporciona un entorno gráfico de edición del modelo de tareas, pero define un lenguaje muy general para la especificación de las mismas. La novedad principal de nuestro trabajo respecto de estas herramientas es la manipulación explícita de un modelo del usuario en el propio entorno de desarrollo.

La introducción de este tipo de abstracciones en un entorno gráfico plantea nuevas dificultades, ya que los lenguajes visuales son adecuados para transmitir información declarativa, pero no es fácil describir un comportamiento dinámico de esta forma. En este sentido, los sistemas llamados de programación mediante ejemplos [8] resuelven el problema infiriendo la información procedural a partir de ejemplos concretos creados por el diseñador. Hemos adaptado a nuestro contexto este tipo de técnicas, en particular las utilizadas en HANDSON [9], una herramienta que permite la construcción visual de presentaciones dinámicas dependientes de cualquier parámetro de la aplicación, de la plataforma o de la propia interfaz.

3. LA HERRAMIENTA DE AUTOR

ATLAS realiza la transición entre la forma de entender un curso de un profesor y el modelo de representación del sistema subyacente. En este sentido, ATLAS se puede considerar como una interfaz de usuario para el autor de cursos en TANGOW. El entorno de desarrollo presenta al diseñador todos los elementos del curso (tareas, contenidos, perfil del estudiante) como objetos en pantalla que el diseñador puede crear, manipular y asociar unos con otros directamente con el ratón. La figura 2 muestra un estado intermedio en la creación, en este entorno, de un curso sobre programación en Java.

El espacio de trabajo está formado por tres áreas principales: modelo de tareas, modelo de contenidos, y modelo del estudiante. En la primera (parte izquierda), el diseñador define la estructura del curso mediante un modelo de las tareas del estudiante, que siguiendo el modelo TANGOW, están

organizadas jerárquicamente. En el área de contenidos (superior derecha), se incluyen los componentes disponibles, fragmentos de HTML, que recogen los contenidos de las distintas partes del curso. En el área del modelo del estudiante (inferior derecha) se detallan las características de éste que se consideran relevantes para el curso en construcción, y que han de condicionar su estructura. Este modelo consiste en un conjunto de atributos simples y compuestos estructurados en forma de un árbol que el diseñador puede editar.

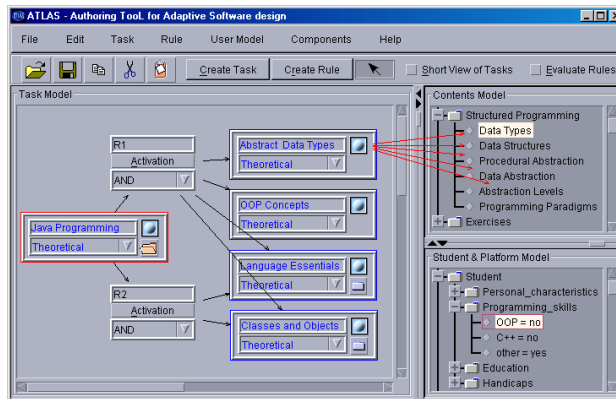


Figura 2. El entorno de desarrollo ATLAS

El diseñador asigna a cada tarea las unidades de contenido que le corresponden, arrastrando con el ratón sobre la tarea objetos del área de contenidos del curso. El árbol de tareas se construye pulsando y arrastrando el ratón entre tareas, asociando cada tarea con sus subtareas. En la figura 2, el diseñador ha descompuesto una unidad (tarea) de alto nivel sobre Programación en Java en cuatro subtareas: Tipos Abstractos de Datos, Conceptos de POO, Aspectos Básicos del Lenguaje, y Clases y Objetos en Java.

Las tareas admiten más de una forma de descomposición. Entre cada tarea y sus subtareas se interpone un nodo (R1 y R2 en la figura) para cada ramificación. Estos objetos representan el elemento adaptativo que permite condicionar la estructura del curso a las características del estudiante y a la actividad realizada por éste en tiempo de ejecución (ver [4]). Su papel consiste en controlar cuál de las descomposiciones se aplicará en tiempo de

ejecución cuando, como en el ejemplo, se ha definido más de una alternativa. La activación de una u otra viene determinada por un predicado asociado a cada ramificación. Estas condiciones se editan seleccionando y arrastrando atributos del modelo de usuario sobre un diálogo de edición de condiciones, que se abre al pulsar el ratón sobre el icono que representa la ramificación condicionada (ver figura 3). ATLAS permite combinar predicados simples de comparación entre propiedades del estudiante, parámetros de las tareas y/o valores literales, en forma normal disyuntiva.

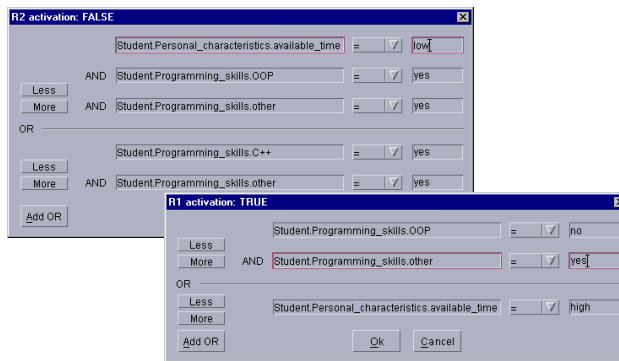


Figura 3. Construyendo condiciones de activación

Para simplificar el tratamiento del aspecto adaptativo, ATLAS permite trabajar sobre perfiles concretos del usuario proporcionados por el diseñador, de forma que la herramienta abstrae y generaliza después el modelo de tareas construido para los casos particulares. Para ello el diseñador edita en primer lugar el modelo del estudiante, del cual depende la estructura de tareas, añadiendo nuevas propiedades o suprimiéndolas, e introduciendo valores. A partir de los ejemplos proporcionados de esta forma, ATLAS instancia la jerarquía de tareas para el perfil descrito, permitiendo que el diseñador edite directamente la estructura resultante, en lugar de la representación inicial, más general y abstracta. En la figura 2 esta opción está desactivada. Al seleccionar el modo “Evaluate Rules” en la barra de herramientas, desaparecería la ramificación R2 para la tarea “Java Programming”, por no cumplirse su condición de activación (fig. 3) con los valores proporcionados en el modelo del estudiante. Cuando el diseñador trabaja en este modo, el sistema se encarga de generalizar el diseño y mantener la consistencia.

La salida de ATLAS es un modelo del curso listo para ser procesado por TANGOW, que genera dinámicamente el curso tal como lo ve finalmente el alumno, dando soporte a su ejecución.

4. CONCLUSIONES

El desarrollo de productos educativos de calidad pasa necesariamente por involucrar de forma directa al propio colectivo docente en la elaboración del material educativo. Más allá de la selección o elaboración de textos y recursos multimedia, es importante que esta participación se plasme en la definición de aspectos como la estructura y las características funcionales de los cursos, así como en la interacción entre el software y el estudiante. Creemos que el desarrollo de herramientas interactivas que soporten expresamente estos aspectos contribuirá al avance en esta dirección, permitiendo a los diseñadores dedicar más tiempo y atención a los aspectos pedagógicos, y menos a la resolución de problemas técnicos.

La adaptatividad de los cursos construidos, es decir, la toma en cuenta de las características del alumno y su forma de trabajar, permite aumentar el alcance de los cursos y favorece su difusión a través de la red, además de ser un elemento clave en la calidad de la docencia, sea o no sobre soporte informático. La descripción y utilización explícita de modelos de usuario en un entorno gráfico, para definir capacidades adaptativas en las aplicaciones construidas, es un aspecto novedoso tanto en el campo de las tecnologías de la educación, como en el de las interfaces de usuario.

Son muchas las posibilidades de continuación del trabajo presentado. Por ejemplo, actualmente los contenidos de los cursos se elaboran con editores de HTML externos al sistema. Nuestros planes para el futuro inmediato incluyen el desarrollo de una herramienta de edición de contenidos que soporte la adaptatividad en la presentación de los mismos. Tenemos previsto también incorporar al sistema otros modelos que puedan condicionar la estructura y presentación de un curso, tales como la plataforma, el contexto de uso, o las características de los datos manejados por la aplicación.

ATLAS está siendo desarrollado en Java, JDK 1.2. Existe una versión de la herramienta disponible en <http://astreo.ii.uam.es/~atlas>.

5. REFERENCIAS

1. M.W. Goldberg and S. Salari. *An update on WebCT (World-Wide-Web Course Tools)- a Tool for the Creation of Sophisticated Web-based Learning Environment*. Proceedings NAUWeb'97 - Current Practices in Web-based Course Development. Flagstaff, Arizona, 1997.
2. J. Vassileva. *Dynamic Courseware Generation on the WWW*. Proceedings 8th World Conference of the AIED Society. Kobe, Japan, 1997, pp. 498-505.
3. G. Weber and M. Specht. *User modeling and Adaptive Navigation Support in WWW-based Tutoring Systems*. Proceedings 6th International Conference on User Modeling (UM97). Sardinia, Italy, 1997.
4. R.M. Carro, E. Pulido, P. Rodríguez. *Dynamic generation of adaptive Internet-based courses*. Journal of Network and Computer Applications, v. 22, 1999, pp. 249-257.
5. N. Sukaviriya, J. Foley, T. Griffith. *A Second Generation User Interface Design Environment: the Model and the Run-Time Architecture*. Proceedings ACM International Conference on Computer-Human Interaction (InterCHI'93). Amsterdam, 1993, pp. 375-382.
6. P. Szekely, P. Sukaviriya, P. Castells, J. Muthukumarasamy and E. Salcher. *Declarative Interface Models for User Interface Construction: The Mastermind Approach*. En "Engineering for Human-Computer Interaction", L. Bass and C. Unger, eds. Chapman & Hall, 1996, pp. 120-150.
7. A. R. Puerta. *A Model-Based Interface Development Environment*. IEEE Software, v. 14, nº 4, 1997, pp. 40-47.
8. A. Cypher, ed. *Watch What I Do: Programming by Demonstration*. The MIT Press, 1993.
9. P. Castells and P. Szekely. *Presentation Models by Example*. En "Design, Specification and Verification of Interactive Systems '99", D.J. Duke and A. Puerta, eds. Springer-Verlag, Viena, 1999, pp. 100-116.