

Entornos de Autor para el Aprendizaje basado en Internet

Manuel Prieto Méndez
Francisco Ruiz González
Manuel Ortega Cantero
José Bravo Rodríguez

SINOPSIS

Los sistemas de aprendizaje cooperativo basado en computadoras (ACBC) se están desarrollando muy activamente gracias a las facilidades de Internet. Java (Sun Inc.) Visual Basic Scrip (Microsoft Co.) y HTML son ya estándares para construir aplicaciones para World Wide Web (WWW). Nuestro grupo desarrolla y pone a punto un entorno de autor que genera ficheros interpretables con visores Web. El sistema se apoya en el uso de herramientas de programación visual y produce o utiliza recursos multimedia, de hipertexto así como formas y escenarios en HTML para la expresión del material de aprendizaje.

Términos Clave : <aprendizaje> <lenguaje de computación> <computadora>

ABSTRACT

The Computer Based Cooperative Learning Systems (CBCL) are being actively developed thanks to the INTERNET facilities. Java (Sun, Inc.) Visual Basic Sript (Microsoft, Co.) and HTML are standarts to build applications for WWW (World Wide Wed). Our group develops and updates and author environment that generates files which can be interpreted with Web wienfinders. The system relies in the use of visual programming tools and uses or produces multimedia hypertext resources as well as forms and backgrounds in HTML to express the learning materials.

Key Terms : <learning> <computer languages> <computers>

1. Introducción

La actual generación de los Sistemas para el aprendizaje basado en computadoras se encuentra influenciada por el desarrollo de tres elementos principales :

- * La ciencia del conocimiento que promueve nuevos y perfeccionados modelos de la memoria, el aprendizaje y de representación del conocimiento.
- * Las telecomunicaciones con sus mecanismos de transmisión de todo tipo de información de manera casi instantánea y en cualquier lugar del planeta.
- * Los sistemas hipermedia que proporcionan recursos de presentación audiovisual y el manejo de estructuras hipertextuales interactivas que son particularmente útiles para las aplicaciones en educación.

Es evidente que las Ciencias de la Educación están tomando ventajas de las nuevas realizaciones en estos tres campos que harán de la didáctica una ciencia más precisa en los próximos años. Los procesos de enseñanza - aprendizaje se centran cada vez más en los problemas del aprendizaje y se acercan a los resultados de la Informática y en particular de la Inteligencia Artificial y la Ciencia Cognitiva.

La reciente explosión de las capacidades de Internet esta influyendo en la obtención de nuevas técnicas, sistemas y metodologías que mejoren el aprendizaje humano. Los sistemas de Trabajo Cooperativo Basado en Computadoras (TCBC - CSCW) y especialmente los de Aprendizaje Cooperativo Basado en (ACBC - CSCL) se relacionan estrechamente con los avances que se producen en los tres aspectos antes mencionados.

Los instrumentos basados en Internet y en el WWW serán utilizados ampliamente debido a sus bajos costos de explotación, su amplia difusión a nivel mundial y el alto grado de standarización alcanzado en breve tiempo.

Nuestro trabajo actual se centra en el desarrollo de un Sistema de Autor (denominado TOLEDO) destinado a los maestros y profesores que se interesa en la generación automatizada de material de apoyo al aprendizaje que puedan ser interpretados por los visores Web de internet ampliamente difundidos en todo el mundo utilizando básicamente, los estándares de descripción de ficheros HTML.

2. Arquitectura del sistema

Nuestra propuesta tiene dos componentes funcionales principales: El entorno de autor y el generador de código HTML. El trabajo con el sistema tiene lugar en dos etapas: la etapa de desarrollo y la etapa de utilización. El profesor (autor) define y desarrolla la aplicación utilizando el sistema de desarrollo. El alumno es el consumidor de la aplicación a través de los ficheros generados por el sistema y alguno de los visores WWW ampliamente difundidos en Internet.

La programación tiene lugar con los recursos locales del Servidor. El consumo de las aplicaciones se ejecuta bajo el esquema cliente - servidor. En el servidor (o servidores distribuidos) se encuentran los ficheros necesarios. En el equipo cliente (del estudiante) solo deberá residir el Visor Web y, eventualmente, los intérpretes de Escenarios Java o Visual Basic necesarios para ejecutar los programas instructivos.

La figura 1 muestra un esquema general de la etapa de desarrollo.

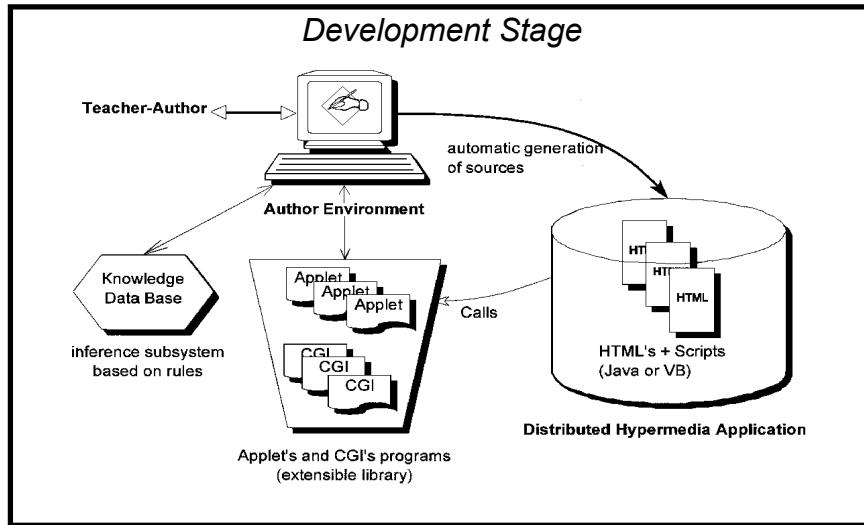


Figura 1

La estructura corresponde a la etapa de utilización se presenta en la figura 2

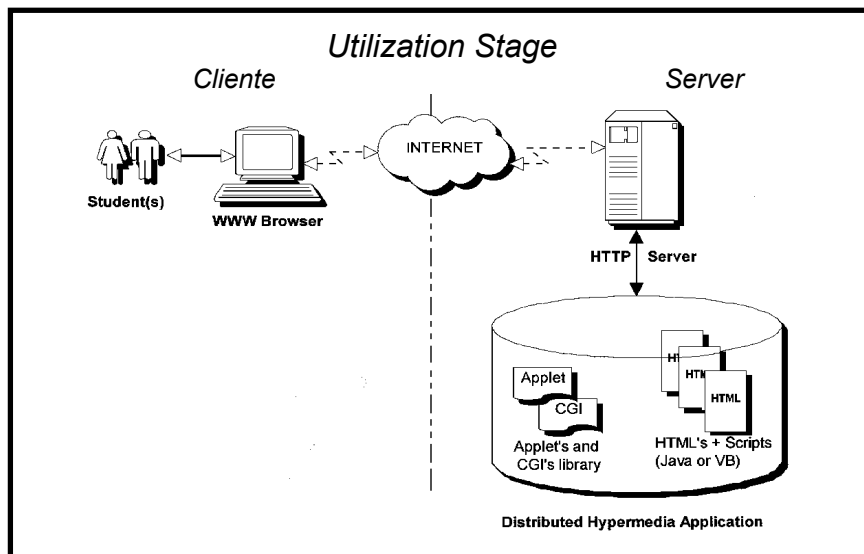


Figura 2

En el sistema, hay tres tipos de objetos que se presentan con diferentes clases de documentos HTML. Los ficheros HTML llaman a otros ficheros de código interpretables en el standar CGI (Common Gateway Interface) o a ficheros ejecutables Appletes.

El proceso de generación automática de ficheros HTML se basa en el uso de una base de conocimientos en forma de reglas de producción. Nuestro grupo estudia actualmente la conveniencia de incorporar modelos de los Sistemas Tutoriales Inteligentes y en particular de modelación del estado del conocimiento del estudiante.

3. El Entorno de Autor

El entorno de autor utiliza objetos visuales disponibles en forma de iconos en una caja de herramientas. Las instancias se colocan en el espacio de trabajo. Los diseñadores (profesores) utilizan técnicas de programación visual propias de los Sistemas de Desarrollo Rápido de Aplicaciones (DRA - RAD).

La caja de herramientas del Entorno contiene elementos de las siguientes categorías :

- Componentes para el control de flujo de la aplicación (Objetos de Control): Permitir describir la secuencia del programa en estilo imperativo. Algunos ejemplos son : Comenzo ; IrA ; Ejecutar ; LamarA ; Iteración.
- Componentes Audiovisuales Simples (Objetos Simples): Son contenedores de información estática, por ejemplo Audio , Vídeo , Imagen , TextoPlano , TextoIn.
- Objetos de estructura compleja (Objetos Complejos): se componen de otros objetos simples(o complejos) en relaciones internas y pueden ser usados por el diseñador para

representar enlaces hipertextuales. Algunos modelos son : Menu , Envoltorio , Enlace_Libre...

- Programas utilitarios (Servicios Utilitarios) : son programas que facilitan trabajos colaterales. Se pueden usar utilitarios distribuidos con TOLEDO o bien importarlos libremente según la conveniencia o comodidad del autor. Entre los utilitarios, se destacan un editor de textos, editor de imágenes y con editor de ficheros HTML.

4. Herramientas de desarrollo

Nuestro sistema está diseñado para su utilización en entornos educativos. Los profesores deben crear aplicaciones educativas para internet de la manera más cómoda y fácil posible. Por esta razón, se han elegido como herramienta de desarrollo unos lenguajes y entornos muy extendidos y de fácil aprendizaje, frente a otras opciones mucho más potentes, pero de un carácter marcadamente más profesional. Un sistema TOLEDO está siendo desarrollado en entorno Windows® de 16 y de 32 bits, utilizando Visul Basic® 4.0.

Es muy importante que las aplicaciones generadas por el sistema autor puedan ser utilizadas con las herramientas habituales de trabajo en internet, en concreto, con los visualizadores (browsers) WWW disponibles en el mercado. Por esta razón, dichas aplicaciones fuentes (AF's) están formadas por colecciones de diversos tipos :

- Documento HTML que son el soporte básico de almacenamiento de la información y flujo de programa. En cierto sentido, nuestro sistema es parecido a los generadores automáticos de código fuente existentes para ciertos lenguajes de programación clásicos. Estos documentos admiten hipertextos y multimedia (audio, vídeo,...).

- Programas CGI (common gateway interface). Estos programas son lanzados (ejecutados) por los documentos HTML, reciben datos de ellos y les devuelven datos o generan nuevos documentos HTML (ver figura 1). Se dispone de una librería predefinida de programas CGI's (programados también en Visual Basic 4.0). El profesor puede ampliar esta librería incorporando nuevos componentes con posterioridad o desarrollándolos (esta es otra razón que nos inclina a utilizar el citado lenguaje, dada su sencillez de uso).
- Applets (programas) en el lenguaje Java (tm), conocido lenguaje de programación orientado a objetos que permite el desarrollo de aplicaciones distribuidas en Internet. Una de las más inmediatas aplicaciones del potencial de este lenguaje para fines educativos consiste en la posibilidad de visualizar simulaciones. Los visualizadores que utilizan este lenguaje permiten la incorporación dinámica de protocolos, el reconocimiento dinámico de tipos y, en resumen, la creación de contenidos dinámicos.

Una AF generada por TOLEDO tiene siempre un documento HTML cabecera (head document), que permite a los usuarios comenzar la visualización (ejecución de la aplicación). A partir de este documento, los usuarios pueden interactuar con el sistema mediante las nuevas posibilidades que ya ofrecen los visualizadores Web. Se podrá acceder mediante las opciones de hipertexto a otros documentos HTML, o lanzar programas CGI`s y/o applets Java preconstruídos.

En entornos educativos es fundamental la interacción del usuario(el estudiante) con el sistema. En nuestro caso, esta interacción se realiza vía internet y se tiene el riesgo de que el ancho de banda(velocidad disponible para las comunicaciones) haga más lenta la interacción. Una muy buena solución de este problema es utilizar Java (o Visual Basic Script). Los

applets o programas fuentes tienen un tamaño pequeño y una vez enviados desde el servidor HTTP al cliente, liberan a dicho servidor y a la red de comunicaciones, ya que se ejecutan en la máquina cliente.

El uso de un lenguaje de programación - como los dos citados - permite eliminar la necesidad de empleo de ejecutables CGI, pero preferimos conservar esta opción para dar una mayor versatilidad y posibilidad a los profesores - autores.

La utilización de todas estas opciones nos permiten disponer de un sistema de autor con las siguientes características :

- Aprovechamiento máximo de las posibilidades educativas de hipertexto y multimedia.
- Máxima estandarización (empleo de HTML y sus extensiones)
- Facilidad de desarrollo de aplicaciones de enseñanza a distancia.
- Incorporar posibilidades de cooperación de espacios de trabajo comunes (gestionados en el servidor web) y el uso de útiles de comunicación, que forman parte de las bibliotecas de componentes reconstruidos disponibles.

5. Modelo de Datos

Las características de nuestro sistema necesitan una gestión de datos con las siguientes funcionalidades :

- Almacenamiento y recuperación de datos multimedia : texto, imágenes, audio, vídeo,..
- Opciones de navegación por un hiperespacio
- Utilización de objetos complejos, incluyendo hipertexto

Para satisfacer estas necesidades, nuestro sistema de autor dispone de las características de hipertexto del lenguaje de marcas HTML (referencia X), gestionadas directamente mediante el empleo de URL's (Unified Resource Locatio) por los servidores HTTP.

Pero, además se necesita disponer de unas estructuras

de datos que permitan almacenar y gestionar con eficiencia toda la información . La mejor opción es que dichas estructuras estén basadas en un modelo de datos hipermedia con orientación a objetos. Por estas razones hemos elegido el HM - Data Model (Maurer, Scherbakov, Srinivasan, 1993) en el que realizamos algunos cambios para adaptarlo a nuestro sistema.

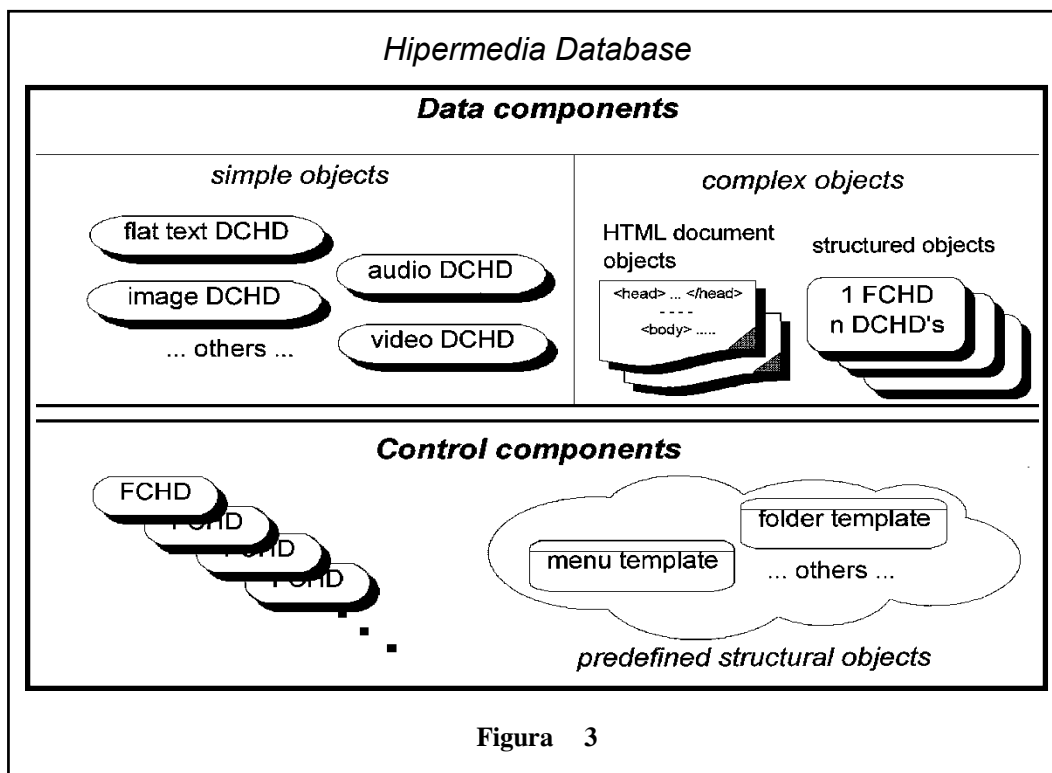


Figura 3

En el HM-Data Model, una base de datos hipermedia es una colección de objetos hipermedia complejos llamados S-colección. Cada S-colección tiene un identificador único (un nombre) y una cabecera (información que se visualiza al “entrar” a la colección. Existen nodos primitivos (sin estructura interna) o nodos con estructura interna formada por un conjunto de otras S-colecciones asociadas entre sí

por ciertos enlaces hipermedias. Este tipo de estructura (ver figura 1) permite una gestión mucho más efectiva y sencilla de hiperespacio (un sistema hipermedia puede llegar a tener cantidades muy grandes de nodos de información y de enlaces entre ellos).

En este caso tenemos que almacenar dos clases de información :

-
-
- a) los datos de la aplicación educativa, es decir, todo lo que los estudiantes podrán visualizar y observar.
 - b) los datos de control de la aplicación.

Los datos de la aplicación pueden ser objetos simples (ficheros de texto plano, audio, vídeo, imagen) o complejos. Los objetos simples no tienen estructura interna significativa, es decir, dentro de ellos no existen otros objetos más simples, ni tienen hipervínculos. En la AF (aplicación fuente) por cada objeto simple, el sistema de autor genera un documento HTML contenedor de datos (data container HTML document DCHD) : que incluye los TAG's que permiten visualizar dicho objeto y retornar al objeto desde el que se llamó.

Los objetos complejos tienen una estructura interna significativa, incluyendo otros objetos más simples y/o hipervínculos a otros objetos. Por cada objeto compuesto el SA genera una colección de DCHD's y un documento HTML de control de flujo (flow control HTML document : FCHD) que contiene los TAG's encargados de controlar el flujo de ejecución de la aplicación, es decir, incluye los hipervínculos disponibles entre los objetos incluidos en el objeto complejo y los criterios de secuenciación, iteración y bifurcación elegidos.

El mayor grado de libertad disponible para definir la estructura interna de un objeto complejo es posible definiéndolo de tipo documento HTML fuente ¹ (source HTML document : SHD), los cuales pueden

¹ Los objetos son de clase simple o compleja en cuanto a si tienen o no estructura interna. Pero en cuanto a su tipo son : texto plano (simple), audio(simple), vídeo (simple), imagen(simple), fuente HTML (complejo). El resto de objetos complejos son realmente una S - Colección de objetos.

² Realmente es una lista doble enlazada.

ser editados directamente con un editor HTML de los disponibles en el mercado. Esta opción le confiere al profesor - autor una libertad total, pero incorpora el riesgo de elaborar aplicaciones educativas que desorienten al estudiante, este problema es conocido como pérdida en el hiperespacio (referencia).

En el campo de la educación este problema es especialmente grave. Existen propuestas muy interesantes que lo abordan realizando una fusión de los sistemas tutoriales con el hipertexto (Pérez y otros, 1995). Dado que es el profesor - tutor el que debe indicar, con base a consideraciones didácticas y pedagógicas, como puede navegar el alumno por la aplicación, en el sistema TOLEDO se ofrece un conjunto de "estructuras de navegación" preconstruidas, que ayudan al profesor - tutor a crear y controlar el flujo de la aplicación.

Los ejemplos más habituales son : menú, carpeta (folder)² , todos - con- todos (envelope)

Además de los FCHD's creados para control de flujo interno de un objeto complejo, el SA genera otros FCHD's que controlan el flujo general de la aplicación, con opciones similares a los habituales en la programación estructurada :

```

Iniciar      (BEGIN)
Ir_a         (GOTO)
Llamar_a     (CALL/GOSUB)
Iteración    (Repetir_hasta :      DO_WHILE,
Repetir : mientras : DO_UNTIL)
Bifurcación (Si_entonces : If : THEN, Según _
valor : SELECT _CASE

```

6. Conclusiones y perspectivas

Internet es un soporte natural a nivel mundial para los usos educativos. Sus costos relativamente bajos, su amplia utilización proveen una plataforma común para las aplicaciones instructivas para la educación a distancia y el aprendizaje cooperativo.

TOLEDO posee una filosofía simple basada en tres hechos principales :

- 1) Permite a los autores producir material instructivo interactivo, hipertextual y multimedia

en un entorno de Desarrollo Rápido de Aplicaciones (DRA).

- 2) Generación automática de ficheros HTML (y algunas extensiones) .
- 3) Aplicaciones fáciles de usar en standares de WWW.

TOLEDO será extendido con algunas facilidades adicionales provenientes de las experiencias de los Sistemas Tutoriales Inteligentes. La versión en preparación permitirá asimismo la preparación de actividades para el aprendizaje cooperativo.

Referencias

- HTML Working-Group of the Internet Engineering Task Force (1995) : Hipertext Markup Language - 2.0 W3 Consortium
- Berk, E.; Devlin, J. Editors (1991): Hypertext/Hipermedia Handbook. *McGraw Hill Software Engineering Series*.
- Gronbaek, K. ; Hem, J.A.; Madsen, O.L.; Sloth, L. (1994): Cooperative Hypermedia Systems: A Dexter-Based Architecture. *Communications of ACM*, vol. 37, 2.
- Halasz, F.; Schwartz, M. (1994): The Dexter Hypertext Reference Nodel. *Communications of ACM*, vol. 37,2
- Maurer., H.; Sherbakov, N.; Srinivasan, P. (1993): A new Hypermedia Data Model. Proceedings of DEXA'93, Springer.
- Maurer.; Andrews, K.; Sherbakov, N. (1995): Embedding Courseware into the Internet: Problems and Solutions. Graz Technological University, Austria.
- Pérez, T.A.; Gutiérrez, J.; Lopistéguy, P. (1995): An Adaptativa Hypermedia System. *7th World Conference on Artificial Intelligence in Education (AIED '95)*. Charlottesville, VA (USA).
- Prieto, M. ; Ruíz, F. ; Ortega, M. y Bravo, J. (1996) Author System for Web-Based Learning. *European Conference on Artificial Intelligence y Education (EURO AI DE)*, Lisboa.
- Verdejo, M.F.; Cerri S.A. (Eds.) (1994): Collaborative Dialogue Technologies in Distance Learning. *NATO ASI series F*, vol. 133, Springer - Verlag.