



Ingeniería. Investigación y Tecnología

ISSN: 1405-7743

iit.revista@gmail.com

Universidad Nacional Autónoma de México
México

Canales-Cruz, A.; Peredo-Valderrama, R.; Balladares-Ocaña, L.; Peredo-Valderrama, I.; Sossa-Azuela, J.H.

Arquitectura de sistemas tecnológicos para la educación basada en Web

Ingeniería. Investigación y Tecnología, vol. X, núm. 2, abril-junio, 2009, pp. 145-157

Universidad Nacional Autónoma de México
Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=40411431006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Arquitectura de sistemas tecnológicos para la educación basada en Web

Technology System Architecture for Web-Based Education

A. Canales-Cruz

Centro de Investigación en Computación del IPN, México DF.

E-mail: alejandro_canales@cuaed.unam.mx

R. Peredo-Valderrama

Centro de Investigación en Computación del IPN, México DF.

E-mail: peredo@cic.ipn.mx

L. Balladares-Ocaña

Centro de Investigación en Computación del IPN, México DF.

E-mail: ballad@cic.ipn.mx

I. Peredo-Valderrama

Centro de Investigación en Computación del IPN, México DF.

E-mail: peredo@cic.ipn.mx

J.H. Sossa-Azuela

Centro de Investigación en Computación del IPN, México DF.

E-mail: hsossa@cic.ipn.mx

(Recibido: agosto de 2007; aceptado: junio de 2008)

Resumen

En este trabajo se presenta una nueva arquitectura para el desarrollo de sistemas de Educación Basada en Web. Estos sistemas se centran en el estudiante y se adaptan a sus necesidades personales de forma inteligente. La arquitectura se basa en la especificación IEEE 1484 LTSA (Learning Technology System Architecture) y reúne a los modelos de desarrollo de software y diseño instruccional. Por una parte, el modelo de desarrollo de software se encuentra sustentado bajo un Sistema Multi-Agentes (MAS), emplea los métodos y técnicas de la Ingeniería de Dominio para el desarrollo de CARIOO (Componentes de Aprendizaje Reutilizables e Inteligentes Orientados a Objetos). Los CARIOO son un tipo especial de Objetos de Contenido Compartible de acuerdo a SCORM (Sharable Content Object Reusable Model). Por otra, el modelo de diseño instruccional incorpora un modelo mental como los Mapas Conceptuales (MC) para transmitir, construir y generar conocimientos adecuados a este tipo de ambiente.

Descriptores: Huracán, campo de velocidades de vientos huracanados, registros de velocidades de viento, perfil radial de presiones de Holland, Isidoro, Emily, Wilma.

Abstract

In this paper a new architecture for development of Web-Based Education systems is presented. These systems are centered in the learner and adapted to their personals needs in intelligent form. The architecture is based on the IEEE 1484 LTSA (Learning Technology System Architecture) specification and it assembles to software development and instructional design patterns. On the one hand, the software development pattern is supported under a Multi-Agents System, it employs the methods and technical of the Domain

Engineering for development of IRLCOO (Intelligent Reusable Learning Components Object Oriented). IRLCOO are a special type of Sharable Content Object according to SCORM (Sharable Content Object Reusable Model). On the other hand, the instructional design pattern incorporates a mental model as the Conceptual Maps to transmit, build and generate appropriate knowledge to this educational environment type.

Keywords: *Web-Based Education (WBE), multi-agents system, domain engineering, conceptual maps and web semantic.*

Introducción

Actualmente, la educación centrada en el estudiante tiene como objetivo que la secuenciación y autoría de contenido, los modelos pedagógicos y los procesos de evaluación cumplan las metas de aprendizaje de los estudiantes. Además, los repositorios de contenido y evaluaciones deben ser apropiados para los requerimientos particulares de cada individuo. Pero al mismo tiempo, tienen que ser flexibles y disponibles para ajustarse y utilizarse por una amplia comunidad de desarrolladores y estudiantes, respectivamente. Es necesario que el desarrollo de un sistema de Educación Basada en Web (EBW) considere toda la diversidad de requerimientos (tecnológicos y pedagógicos) y proporcione las funcionalidades necesarias a través de los recursos Web.

Asimismo, la ingeniería de dominio tiene que considerar las especificaciones particulares demandadas por los sistemas de EBW, como es la automatización de la generación de componentes reusables de contenido y evaluación, tanto como sea posible. Los componentes ayudan a reducir la complejidad, la administración del cambio y el reuso (Wang, 2005). Mientras que para crear las experiencias de enseñanza-aprendizaje dentro de este ambiente de EBW, es necesario el uso de un modelo mental como los mapas conceptuales.

El propósito de este trabajo es mostrar una nueva arquitectura para el desarrollo de sistemas de EBW que se basa en la especificación IEEE 1484 LTSA (IEEE, 2001). Para lograr esta meta, este artículo se ha organizado como sigue: inicialmente se presenta la arquitectura de sistemas tecnológicos y a continuación se describe cada una las capas que la componen; después, se muestra un sistema de autoría de contenido llamado SiDeC y un sistema de evaluación, que son el resultado de la aplicación de dicha arquitectura.

Arquitectura de sistemas tecnológicos educativos

La arquitectura de sistemas tecnológicos educativos que se presenta, considera una gran parte de los elementos

que intervienen dentro de un ambiente de EBW y pretende sentar las bases para los nuevos desarrollos de este tipo de sistemas. La producción de software educativo bajo esta arquitectura se caracteriza por su colaboración en el aprendizaje de los estudiantes. Básicamente, esta arquitectura presenta avances referentes al estudio de las interacciones entre las entidades del sistema, la incorporación de elementos pedagógicos que intervienen directamente en el aprendizaje de los estudiantes, la descomposición en componentes del sistema de aprendizaje y muestra una metodología para el desarrollo de sistemas de EBW, a través de un modelo de ciclo de vida doble. Estos avances son reflejados en la especificación de cinco capas que se describen en la siguiente figura 1.

Interacciones entre las entidades del sistema

Como resultado del análisis de las interacciones entre las entidades del sistema, se obtuvieron los flujos de información que se muestran en la figura 2. Se concluye que existen cinco tipos básicos de interacción que tienen lugar en el WBE: el estudiante-contenido, estudiante-tutor, estudiante-estudiante, estudiante-universidad y tutor-universidad (Grahame, 1996; Grahame 1989). Con el reconocimiento de que estas interacciones analizan los flujos de trabajo que conforman el modelo de datos a nivel de negocio. Dicho modelo define los objetos de datos clave y sus relaciones entre ellos y con otras áreas de negocio.

Características de diseño relacionadas con el estudiante

El diseño de la arquitectura de sistemas tecnológicos para la EBW es afectado directamente por las necesidades de los estudiantes y, en particular, la naturaleza del aprendizaje humano. En este sentido, se establece un modelo de diseño instruccional para transmitir, construir y generar conocimientos adecuados a este tipo de ambiente. Dicho modelo se basa en la teoría cognitiva de Mapas Conceptuales (MC), que son un modelo

mental sobre la forma en que los seres humanos estructuran una información y tienen la finalidad de ayudar a los estudiantes en la adquisición de conocimiento. Un

MC es una técnica gráfica utilizada durante el proceso de enseñanza-aprendizaje como recurso esquemático o mapa de navegación.

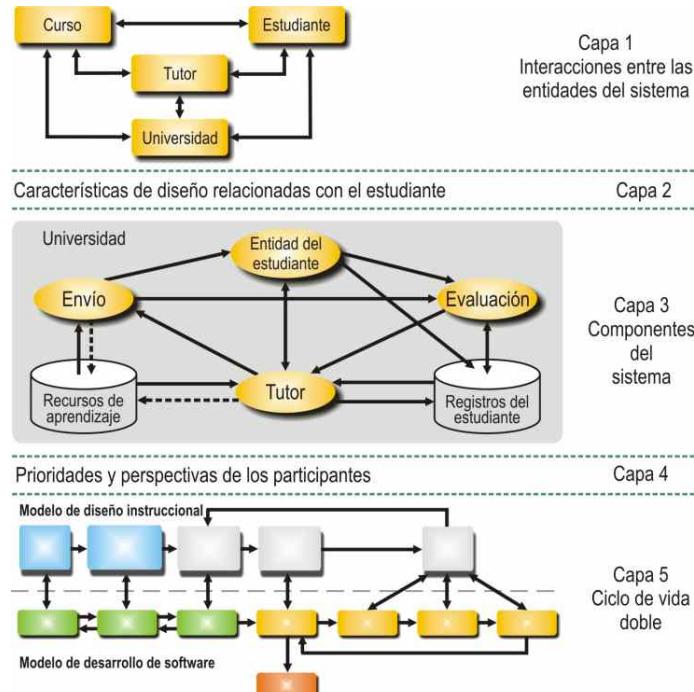


Figura 1. Arquitectura de Sistemas Tecnológicos Educativos para la EBW

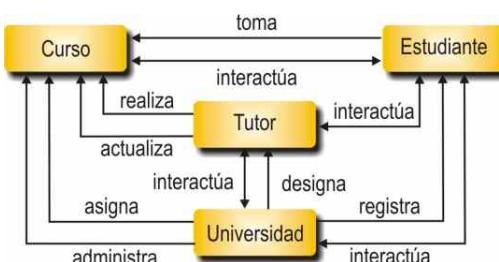


Figura 2. Entidades del sistema de EBW

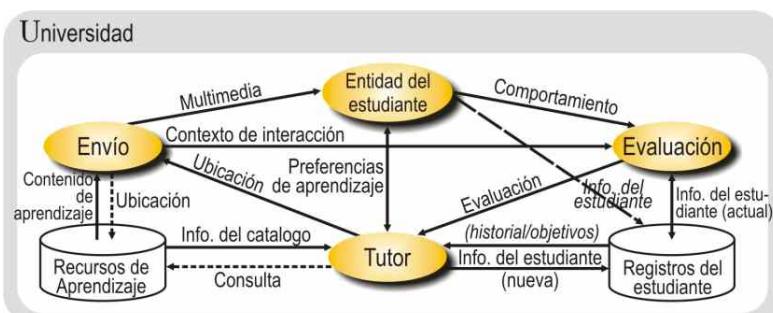


Figura 3. Componentes del sistema

Componentes del sistema

En la figura 3 se muestra la capa de componentes de esta arquitectura, en donde se identifican cinco procesos: entidad del estudiante, evaluación, tutor, proceso de envío y universidad; dos bases de datos: Recursos de aprendizaje y Registros del estudiante; así como catorce flujos de información entre estos componentes.

Primeramente, se considera el proceso de la Universidad porque interviene de forma directa en todo el sistema educativo, de acuerdo con las interacciones entre las entidades del sistema (capa 1). Por otra parte, el proceso del Tutor ha sido dividido en dos: Tutor y Tutor virtual, debido a que este proceso tiene que adaptarse a las necesidades individuales de los estudiantes. Por ejemplo, durante el proceso de enseñanza-aprendizaje algunas decisiones, tales como secuencia, pruebas y actividades, se eligen manualmente por el Tutor, así que la adaptación del curso puede tardar dependiendo de la velocidad de reacción del Tutor y muy probablemente se haga al final del tema.

En este sentido, se presenta un Tutor virtual que realice dichas modificaciones de manera automática cada vez que el estudiante termine una lección, ejercicio o prueba y guarde todos los movimientos realizados dentro del historial (registros del estudiante).

Brevemente, el funcionamiento global de los componentes del sistema tiene la siguiente forma:

- 1) Los estilos de aprendizaje, estrategias, métodos, etc., son negociados entre el estudiante y otro participante, y se comunican como preferencias del aprendizaje.
- 2) *Se recolecta la información de las valoraciones (informales y semiformales) del estudiante en base a su comportamiento, por ejemplo: trayectoria, tiempo empleado, intentos, uso de los diferentes contenidos de aprendizajes, etc. Esta información se guarda directamente en los Registros del estudiante mientras el estudiante interactúa con el sistema.
- 3) El Estudiante es observado y evaluado en el contexto de interacciones multimedia.
- 4) La evaluación produce valoraciones (formales) y/o información del Estudiante.
- 5) La información del estudiante (respuestas escritas, seleccionadas, de voz, etc., y otros detalles más de la evaluación del Estudiante) se guarda en la base de datos (Registros del Estudiante).

* Estas son tareas nuevas que se incorporaron al modelo original de IEEE 1484 LTSA.

- 6) El Tutor revisa la evaluación e información del Estudiante, así como las preferencias, el historial y posiblemente, los objetivos de aprendizaje futuros.
- 7) El Tutor virtual revisa el comportamiento e información del Estudiante y de manera automática e inteligente hace modificaciones dinámicas en la secuencia del Curso (personalizado de acuerdo a las necesidades del Estudiante) basándose en el diseño del proceso de aprendizaje.
- 8) El Tutor/Tutor virtual busca en los Recursos de aprendizaje, vía consulta e información del catálogo, un contenido de aprendizaje apropiado para el estudiante.
- 9) El Tutor/Tutor virtual extrae la ubicación de la información de los catálogos disponibles y pasa las ubicaciones (URLs) al proceso de envío o entrega, por ejemplo, un plan para la lección o ligas a contenidos.
- 10) El proceso de envío extrae el contenido de aprendizaje desde los Recursos de aprendizaje, basándose en las ubicaciones (URLs) y transforma el contenido de aprendizaje en una presentación multimedia interactiva y adaptable para el estudiante.
- 11) La Universidad soporta todo el sistema educativo. Esta nueva tarea proporciona una manera de intercambiar información extra entre la Universidad, Estudiantes y Tutores, tal como: respuesta a dudas y comentarios para expresar en general sus opiniones acerca de los cursos, tutores, estudiantes y el sistema educativo.

Es importante la orientación a componentes que se le ha dado a esta capa, ya que la finalidad que se persigue es implementar una familia de aplicaciones (premisa de la ingeniería de dominio).

CARIOO

Los CARIOO se desarrollaron con Flash. Flash es un integrador de medios y tiene un poderoso lenguaje de programación denominado ActionScript (Macromedia, 2006). Este lenguaje es completamente orientado a objetos y permite el desarrollo de los componentes multimedia de contenido y de evaluaciones. En tiempo de ejecución, los componentes cargan objetos de medios de comunicación y ofrecen un ambiente programable y adaptable a las necesidades del estudiante. Flash cuenta con los "Smart Clips" para los elementos de aprendizaje denominados: interacciones de aprendizaje. El objetivo es generar una biblioteca multimedia de CARIOO para los sistemas de EBW con el propósito de separar el contenido del control. En consecuencia, los componentes

utilizan diferentes niveles de código dentro del Flash Player. Con esta estructura, es posible generar componentes especializados que son pequeños, reusables y apropiados para integrarlos a un componente más grande en tiempo de ejecución. La plataforma de desarrollo de CAROO proporciona ciertas funcionalidades de comunicación dentro de la interfaz de Progremación de la Aplicación con el LMS (Learning Management System), el sistema Multi-Agentes y los diferentes marcos de trabajo como AJAX (Crane, 2006), Hibernate (Peak, 2006), Struts (Holmes, 2004), y la descarga dinámica de medios en tiempo de ejecución.

Los CAROO se meta-etiquetan con el propósito de completarlos como una función similar a los códigos de barras de los productos, el cual se utiliza para identificar los productos y determinar ciertas características específicas de ellos. Este contraste se hace con el meta-etiquetado de Resource Description Framework (RDF-XML) (RDF, 2005), lo cual permite cierto grado de inferencia dentro de los materiales a través del empleo de los medios que proporciona la plataforma de Semántica Web.

ActionScript adiciona el componente “WebService-Connector” para conectar a los servicios Web con los CAROO.

El componente “WebServiceConnector” proporciona el acceso a los métodos remotos ofrecidos por un LMS, a través de SOAP (Simple Object Access Protocol). Esto da a un servicio Web la habilidad para aceptar parámetros y regresar un resultado al script, en otras palabras, es posible acceder y unir los datos entre el servicio Web público o el propio con los CAROO. Es posible reducir el tiempo de programación, ya que se usa una simple instancia del componente “WebServiceConnector” para hacer múltiples llamadas a la misma funcionalidad dentro del LMS.

El componente descubre e invoca el servicio Web utilizando SOAP y UDDI (Universal Description, Discovery, and Integration), a través del *middleware* y el servidor JUDDI (Java UDDI). Colocando una capa en tiempo de ejecución entre un cliente de servicio Web y el servidor, se incrementan las opciones para escribir más inteligente y dinámicamente los clientes.

Reduciendo las necesidades de la dependencia del código duro dentro de los clientes de servicios Web. Sólo es necesario utilizar diferentes instancias para cada una de las diferentes funcionalidades. Los servicios Web pueden descargarse y desplegarse dentro de un CAROO.

Prioridades y perspectivas de los participantes

En esta capa se describe el sistema de componentes (capa 3) desde una variedad de perspectivas de acuerdo a los participantes (Universidad, Tutor y Estudiante). Si se analiza el diseño de la capa 3, con certeza se realizará de diferente forma de acuerdo a la percepción de los participantes, por ejemplo, puede enfocarse desde un punto de vista del contenido de aprendizaje, las evaluaciones, centrada en el alumno, la forma de entrega de los contenidos de aprendizaje, la tecnología a emplear, etc.

En este caso, las prioridades y perspectivas del diseño de la capa 3 se plantearon inicialmente tomando en cuenta a todos los elementos que intervienen en el entorno del estudiante (capa 1) y los elementos pedagógicos (capa 2). Estos últimos son reflejados en el hecho de agregar un nuevo flujo de información a la capa 3 (el no. 2), que se encarga de enviar directamente información del estudiante a la base de datos (registros del estudiante), sobre su comportamiento dentro del escenario de aprendizaje. Esta información se analiza y proporciona elementos para decidir sobre la dirección (secuenciación dinámica), que debe tomar la instrucción del estudiante, pero desde una perspectiva pedagógica, la cual, no sólo contempla a las evaluaciones escritas o de otra índole como método para valorar el conocimiento adquirido de los estudiantes.

En conclusión, la capa 4 proporciona un análisis sobre el diseño de la capa 3 desde la perspectiva de un sistema adaptable e inteligente (secuenciación dinámica) y en donde intervienen principalmente los procesos de Entidad del Estudiante y Tutor. Dicho análisis es importante porque servirá para la implementación de la capa 3. Cabe recalcar que el diseño de la capa 3 se mantiene, lo único que cambia es el énfasis hacia ciertos procesos y flujos de información que juegan un papel importante de acuerdo a nuestra perspectiva y que representan una innovación dentro de esta arquitectura que se presenta.

Ciclo de vida doble

La arquitectura de sistemas tecnológicos para la EBW considera los aspectos pedagógicos, o dicho de una manera más formal, la inclusión de un modelo de diseño instruccional que permita traducir principios pedagógicos de aprendizaje, en una estrategia para el desarrollo de contenido de aprendizaje y actividades (Díaz-Barriaga, 2002). La incorporación de un modelo de diseño instruccional dentro del proceso de desarrollo de software

para la EBW, genera la necesidad de un ciclo de vida con esta característica. Pero al revisar los estándares internacionales ISO/IEC 9001 (ISO, 2000), 90003 (ISO, 2000), 12207 (ISO, 2002) y 15504 (ISO, 2004) de ingeniería de software que rigen la industria en este campo, éstos sólo contemplan básicamente el desarrollo de software desde una perspectiva tecnológica y no profundizan dentro de los requerimientos del campo de aplicación del software, como en este caso es la educación. En todos los estándares de ciclos de vida citados, sólo en la sección de requerimientos es posible incluir los requerimientos pedagógicos.

En la figura 4 se muestra el ciclo de vida doble. Este ciclo une el modelo de desarrollo de software y el modelo de diseño instruccional, lo que es una innovación en el proceso de desarrollo de software actual, ya que hasta

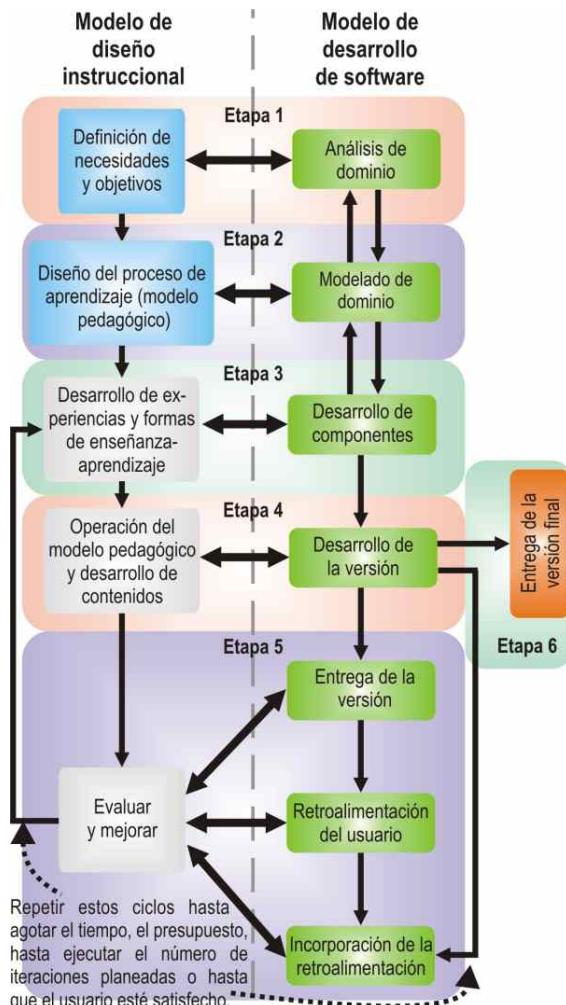


Figura 4. Ciclo de vida doble

el momento esta característica no es considerada en los estándares citados. El propósito de este ciclo de vida es no sólo desarrollar software educativo basado en cuestiones tecnológicas o computacionales, sino además, agregar elementos cognitivos que colaboren en la adquisición de conocimiento de los estudiantes. La unión de estos modelos representa un avance dentro del desarrollo de software educativo, debido a que el ciclo de vida doble integra los principios pedagógicos de aprendizaje dentro del desarrollo de software, ya que se considera que intervienen de manera directa para que se lleve a cabo el proceso de enseñanza-aprendizaje. El modelo de diseño instruccional se basa en los MC, mientras que el modelo de desarrollo de software se rige por la ingeniería de dominio.

El ciclo de vida doble proporciona al desarrollador la facilidad para ir y venir desde un extremo pedagógico a otro tecnológico para el diseño de la aplicación. Esto se debe a que el software tiene una finalidad educativa, por lo que los principios pedagógicos guiarán el desarrollo tecnológico del software. Además, este ciclo de vida doble ayuda al rápido desarrollo y entrega de partes del software.

En la primera etapa del ciclo de vida doble se definen los objetivos del software y se hace un análisis del dominio, lo que equivale a obtener los requerimientos del software tanto pedagógicos como tecnológicos. Para la segunda etapa se obtiene un modelo del dominio, que es el proceso para desarrollar un modelo estructural suficientemente genérico que represente el dominio. Además, se elige un modelo pedagógico para el desarrollo de cursos, que en este caso son los MC, aunque cabe aclarar que se puede utilizar otro modelo pedagógico.

En la tercera etapa se desarrollan las actividades de aprendizaje y el desarrollo de los componentes reutilizables del sistema. Puede observarse que el ciclo de vida doble permite en estas tres etapas retroceder y avanzar entre ellas durante el desarrollo de software con la finalidad de incorporar la información necesaria para diseñar el software hasta alcanzar las metas trazadas para cada etapa.

En la etapa 4 se desarrolla una versión del software que posteriormente en la quinta etapa se muestra al usuario y se refina el producto en función de la retroalimentación del usuario. Este ciclo se repetirá hasta alcanzar un software adecuado, ya que la característica del ciclo de vida doble es precisamente ser evolutivo. Esta característica es muy importante, ya que proporciona la posibilidad de cambiar la dirección del producto a medio camino, en respuesta a las peticiones del usuario. Si se emplea con cuidado la entrega evolutiva, puede

mejorar la calidad del producto, reducir el tamaño del código y producir una distribución más uniforme de los recursos de desarrollo y prueba (Mc Connell, 1997). Finalmente, en la sexta etapa se entrega la versión final al usuario.

Sistema de Desarrollo de eCursos

El Sistema de Desarrollo de eCursos (SiDeC) es una herramienta que facilita la autoría de contenido de aprendizaje y tiene la finalidad de solucionar la carencia técnica de los profesores, al simplificar la creación y publicación de cursos dentro de un ambiente de EBW. Además, la estructura y empaquetamiento del contenido multimedia se logra a través del uso de los CARIOO, a un nivel muy bajo de granularidad. En la figura 5 se muestran los grandes pasos que sigue el SiDeC para la construcción de los materiales educativos.

De acuerdo con la arquitectura de sistemas tecnológicos educativos para la EBW, el SiDeC se construyó

bajo la especificación de la capa 2 “Componentes del sistema”, (figura 3).

El SiDeC es una herramienta de meta datos para la generación de CARIOO y cursos en línea (figura 6). Estos cursos cumplen con las especificaciones de los CARIOO y con los elementos de aprendizaje de los modelos de SCORM (agregación de contenido, secuenciación y navegación y ambiente de tiempo de ejecución) (ADL, 2004). Los meta datos representan la descripción específica del componente y sus contenidos, tales como: título, descripción, descriptores, objetivos de aprendizaje, tipo de elemento y derechos de uso. La herramienta de meta datos proporciona plantillas para todos los meta datos y almacena cada componente en el SiDeC o en otro repositorio estándar de IMS/IEEE.

Hasta el momento, las plantillas de las lecciones del SiDeC se basan en la teoría cognitiva de los MC, pero en el futuro se considerarán otras teorías como: Aprendizaje Basado en Problemas (ABP), el método de casos, el método por proyecto, etc.

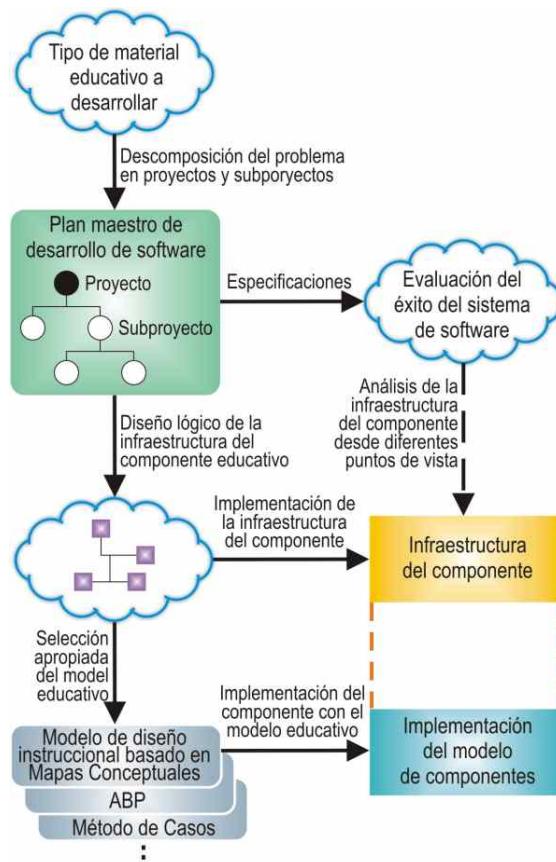


Figura 5. Método de desarrollo de materiales educativos

La figura 7 ilustra como el SiDeC implementa a los MC como un mapa de navegación, mostrando a los estudiantes un panorama global de los objetos de contenido a lo largo de las experiencias de aprendizaje. Estas experiencias siguen la estrategia instruccional, la cual contempla la modificación de la estructura del contenido de aprendizaje a través de la entidad Tutor/Tutor virtual. Tales modificaciones tienen el objetivo de proporcionar aprendizaje significativo y enseñar a pensar a los estudiantes (Díaz-Barriga, 2002).

Basándose en los MC, el SiDeC representa la estructura del curso que el estudiante sigue. El proceso de entrega

identifica un contenido de aprendizaje para el estudiante. El contenido de aprendizaje tiene un CAROO asociado con él. Después, el proceso de entrega lanza el CAROO y lo presenta al estudiante (figura 3). La figura 7 muestra cómo se organiza la estructura del curso y cómo el contenido de aprendizaje puede interpretarse dentro de un árbol de contenido de aprendizaje. Un árbol de contenido de aprendizaje es una estructura conceptual las actividades de aprendizaje administradas por el proceso de entrega de cada estudiante. Esta información se encuentra dentro del *imsmanifest.xml* que está definido dentro del modelo de agregación de contenido de SCORM (ADL, 2004).



Figura 6. Contenido de aprendizaje generado por el SiDeC

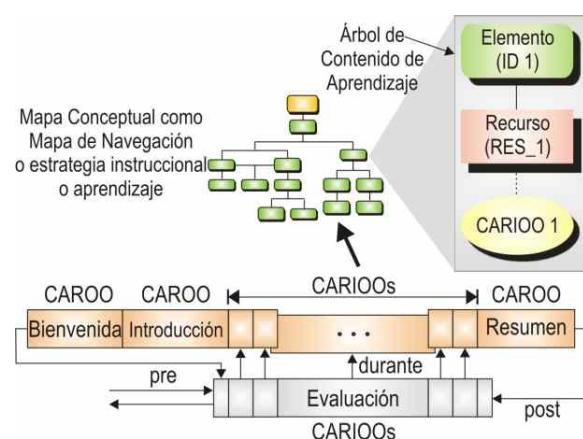


Figura 7. La estructura del curso

Sistema de evaluación

El sistema de evaluación se diseñó bajo la misma filosofía utilizada en el SiDeC. La funcionalidad del sistema de evaluación deposita dentro del análisis del perfil del estudiante que se construye durante las experiencias de enseñanza-aprendizaje. El perfil se basa en las métricas que se obtienen del comportamiento del estudiante en tiempo de ejecución. Estas medidas se almacenan dentro de los registros del estudiante que componen el perfil (figura 3). La generación de las nuevas secuencias de los cursos está en función de los resultados obtenidos, además del cálculo del nivel de adaptación.

El sistema de evaluación combina los CARIOO, meta-etiquetas adicionales y una plataforma de agentes en Java. También, se consideran algunas tecnologías del campo de la inteligencia artificial para recrear un ambiente de semántica Web. La semántica Web se dirige a ayudar a los usuarios a lograr sus actividades en línea y ofrece grandes ventajas, tales como: reducción de la complejidad para los desarrolladores potenciales, estandarización de las funcionalidades y atributos, definición de un conjunto especializado de APIs.

Todos los recursos tienen un URI (Universal Resource Identifier). Un URI puede ser una URL (Unified Resource Locator) o algún otro identificador único. Un

identificador no necesariamente permite el acceso a un recurso. La capa de XML se usa para definir los meta datos de SCORM de los CARIOO que se utilizan para intercambiar datos sobre la Web. En otras palabras, la capa de XML corresponde al lenguaje utilizado para definir la estructura de los meta datos (XML, 2003) y la capa de RDF representa el lenguaje utilizado para describir toda la información y las clases de meta datos (RDF, 2005). Un CARIOO meta-etiquetado refleja la esencia del uso de una ontología (meta-ontología) que se dedica a definir la semántica para establecer el uso de las palabras y términos en el contexto del vocabulario. Su nivel lógico corresponde al razonamiento utilizado para establecer la consistencia y la exactitud del conjunto de datos y para inferir conclusiones que no están expuestas explícitamente.

En resumen, los componentes y la operación del SiDeC y el sistema de evaluación se muestran en la figura 8. Básicamente, el sistema de evaluación se cumple a través de dos fases.

La primera fase es soportada por el LMS y se dedica a presentar los contenidos que se realizan con el contenido de los CARIOO. Las evaluaciones se realizan calificando los CARIOO y, en algunos casos, por simuladores basados en los CARIOO. Estos procesos se despliegan a través del marco de trabajo de AJAX.

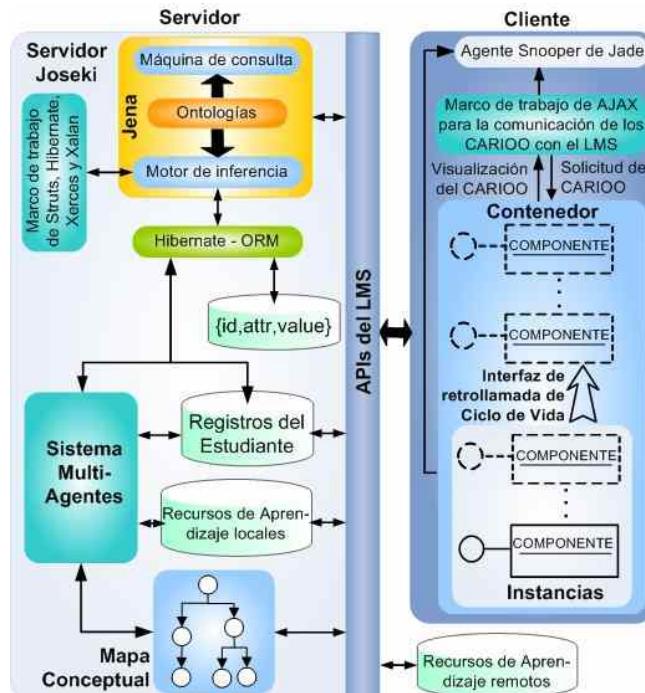


Figura 8. Plataforma de Semántica Web para la EBW

La segunda fase analiza los Registros del estudiante que se lleva a cabo por el servidor basado en JADE y el sistema muti-agentes. Esta plataforma de agentes consta de siete agentes: Snooper, Buffer, Learner, Evaluation, Delivering, Coach, e Info. La idea fundamental es automatizar el análisis del estudiante a través del Tutor y dar resultados parciales que puedan ser útiles para la instrucción final del estudiante. Estos agentes se implementan como programas Java Beans, que están incrustados en las aplicaciones que se ejecutan tanto del lado cliente como del lado servidor. Los agentes emplean la secuenciación dinámica para cambiar la secuencia del curso o las evaluaciones. La secuenciación se define por la estrategia instruccional basada en MC y emplea la navegación/secuenciación de SCORM. Una vez que se recibe la información necesaria (secuencia, tipo de CAROOO y localización, etc.), esto se representa como una cadena construida dinámicamente por la máquina de inferencia basada en reglas conocida como JENA (JENA, 2005) y el servidor JOSEKI (JOSEKI, 2005) para generar retroalimentación dinámica.

Un aspecto importante dentro del desarrollo, tanto del sistema de evaluación como del SIDE, es la persistencia de los datos. Para ello, se utiliza el marco de trabajo de Hibernate. Hibernate es un proyecto ambicioso que se dirige a solucionar completamente el problema de la persistencia de los datos en Java (objetos) que se manejan dentro de las aplicaciones y que se guardan en una base de datos relacional (Peak, 2006). ORM (Object/Relational Mapping) es el nombre que se le da a la solución automatizada para este problema.

Generación de materiales educativos

Tanto el sistema SIDE como el sistema de evaluación se implementaron usando Struts (Holmes, 2004) y bajo el patrón de diseño Modelo-Vista-Controlador (Model-View-Controller, MVC por sus siglas en inglés), mejorando la reusabilidad y mantenibilidad del sistema. Los componentes CAROOO se dividen en dos tipos fundamentales: indivisibles y compuestos, bajo el patrón de diseño de composición, la utilización de este patrón de diseño permitió simplificar la Interfaz de Programación de Aplicaciones (Application Programming Interface, API por sus siglas en inglés), para la manipulación y composición de los componentes CAROOO. Los componentes indivisibles permiten conformar unidades más complejas denominándose compuestos a partir de las unidades indivisibles, separando la parte de contenido/navegación, por medio de un componente contenedor que se encarga de cargar los contenidos y la

navegación, y posibilitando la modificación de un *ims-manifest.xml* personalizado para adecuarlo a las verdaderas necesidades del estudiante. Todos los componentes CAROOO tienen meta datos con información adicional como: nivel, objetivos, autor, etc. Con la finalidad de que por medio del motor de inferencias puedan reconfigurarse para adecuarlos a las verdaderas necesidades del estudiante en tiempo de ejecución.

En la figura 9 se muestra la generación de materiales educativos en tiempo de ejecución tanto para el SiDeC como para el sistema de evaluación. A continuación, se describe brevemente el funcionamiento global del proceso:

- 1) Cuando el tutor desea crear ya sea un contenido de una lección o una evaluación, accede a dichos sistemas desde un navegador Web como cliente. Ambos sistemas tienen la capacidad de ir mostrando al usuario los materiales que va creando en el instante mismo, tal y como se presentarán a los estudiantes, esto se logra gracias al empleo de componentes CAROOO para la previsualización. Los estilos de aprendizaje, estrategias, métodos, etc., son negociados entre el estudiante y otro participante, y se comunican como preferencias del aprendizaje.
- 2) Se toman métricas del estudiante a lo largo de los diferentes cursos por medio de los componentes CAROOO y es evaluado en el contexto de interacciones multimedia.
- 3) La evaluación produce métricas adicionales y/o información del estudiante, la información del estudiante se guarda en una base de datos del historial (registros del estudiante).
- 4) Continuando con la presentación de componentes CAROOO de contenidos y evaluación por medio del proceso de envío, con la finalidad de presentar una interacción de captura de métricas del estudiante en tiempo de ejecución a lo largo del curso .
- 5) El tutor revisa la evaluación e información del estudiante, así como las preferencias, el historial y posiblemente los objetivos de aprendizaje futuros.
- 6) El tutor busca los recursos de aprendizaje vía consulta e información del catálogo para un contenido de aprendizaje apropiado.
- 7) El tutor extrae la ubicación de la información de los catálogos disponibles (URLs) y pasa las ubicaciones al proceso de envío o entrega; por ejemplo, un plan para la lección.
- 8) El proceso de envío extrae el contenido de aprendizaje desde los recursos de aprendizaje, basándose en las ubicaciones (URLs) y transforma el contenido de

aprendizaje en una presentación de contenidos multimedia interactivos para el estudiante, en función de las métricas medidas del estudiante.

Conclusiones

Este trabajo ha presentado una instancia de los sistemas de EBW. Se ha enfocado en la reusabilidad, accesibilidad, durabilidad e interoperabilidad de los contenidos de aprendizaje, los cuales se construyeron como CARIOO, como el principal componente para la entrega de contenido de enseñanza y evaluación.

Los CARIOO ofrecen una interfaz común y una funcionalidad tal, que se aprovechan para la automatización de la autoría de contenidos de aprendizaje que se

entrega a través de una secuencia dinámica. Los CARIOO aceptan retroalimentación por medio de las valoraciones hechas por la plataforma Multi-Agentes. La información proporcionada se considera como datos en bruto porque se basan en parámetros obtenidos del comportamiento del estudiante.

Un modelo distribuido de espacio de conocimiento compuesto por los CARIOO integra diferentes marcos de trabajo. El desarrollo de este modelo se apoya en el SiDeC y el sistema de evaluación. Considerando la especificación y el empaquetamiento del contenido de aprendizaje del estándar de SCORM. El uso de estos sistemas permite reducir drásticamente el tiempo de desarrollo de autoría de contenido de aprendizaje.

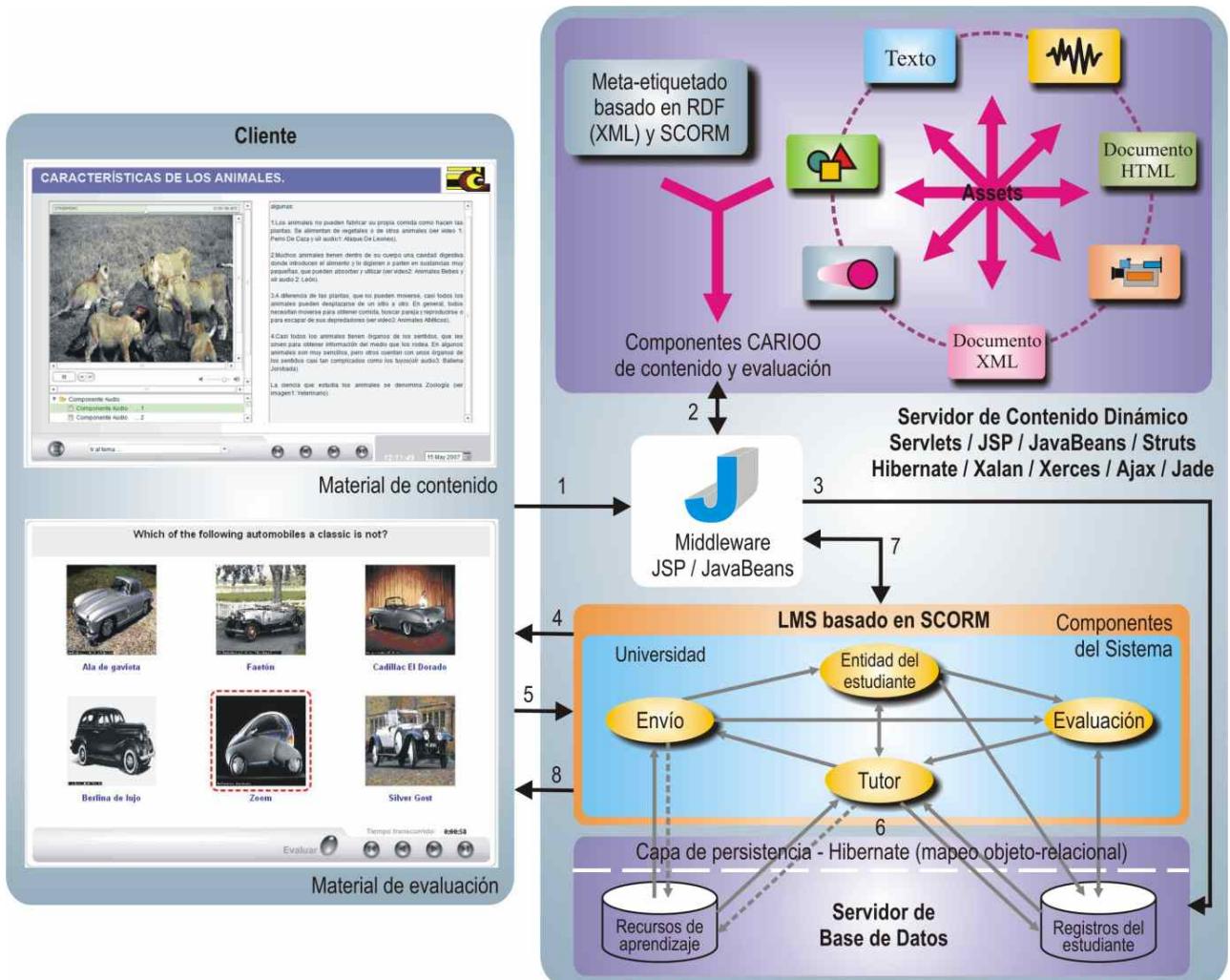


Figura 9. Generación de materiales educativos en tiempo de ejecución

La plataforma multi-agentes se utiliza para la implementación de un *middleware* abierto. La meta técnica de la plataforma es clara: desplegar tecnologías de información distribuidas para que la disponibilidad de los servicios pueda ser más eficiente y flexible. Entre otras ventajas del uso de agentes se tiene que la integración de sistemas puede desempeñarse en un alto grado, la funcionalidad de los CARIOO se incrementa sustancialmente; así como la disponibilidad de implementar diferentes técnicas, estilos de aprendizaje, estrategias instruccionales y técnicas de interacción.

Actualmente, se está trabajando en agentes inteligentes que utilizan XML como un lenguaje de comunicación y en la segunda generación de servicios Web. Asimismo, en los estilos de aprendizaje que se ajusten a la EBW. Finalmente, se están desarrollando nuevos componentes CARIOO y se está actualizando constantemente los ya existentes.

Agradecimientos

Los autores agradecen al CIC-IPN, al proyecto SIP-IPN 20071166 y a CONACYT por el apoyo económico para el desarrollo de este trabajo bajo el proyecto 143049.

Referencias

- ADL. Advanced Distributed Learning Consortium [en línea]. 2004. Disponible en: <http://www.adlnet.org>
- Crane D. *Ajax in Action*. Greenwich. EU. Publicaciones Manning. 2006.
- Díaz-Barriga F. *Estrategias docentes para un aprendizaje significativo*. Segunda edición. México DF. Editorial Mc Graw Hill. 2002.
- Grahame M. *Three Types of Interaction*. EU. The American Journal of Distance Education. 1989. Pp. 1-6.
- Grahame M. *Distance Education: a System View*. Belmont, CA. Wadsworth, EU. 1996. Pp. 128-132.
- Holmes J. *Struts: The Complete Reference*. California, EU. Publicaciones Mc Graw Hill-Osborne. 2004.
- IEEE 1484.1-2001. Standard for Learning Technology - Learning Technology Systems Architecture (LTSA) [en línea]. 2001. Disponible en: <http://ieee.ltsc.org/wg1>
- Internationals Standards Organization 9001 y 90003. Quality Management Systems and Software Engineering—Guidelines for the Application of ISO 9001:2000 to Computer Software. 2000 Edition.
- Internationals Standards Organization 12207. Systems and Software Engineering—Software Life Cycle Processes. 2002 Edition.
- Internationals Standards Organization 15504. Information Technology—Process Assessment. 2004 Edition.
- JENA. Jena Semantic Web Framework [en línea]. 2005. Disponible en: <http://jena.sourceforge.net/>
- JOSEKI Server. A SPARQL Server for Jena [en línea]. 2005. Disponible en: <http://www.joseki.org/>
- Macromedia, Inc. ActionScript [en línea]. Febrero de 2006. Disponible en: <http://www.macromedia.com>
- Mc Connell S. *Desarrollo y gestión de proyectos informáticos*. España. Editorial Mc-Graw Hill. 1997.
- Peak P. *Hibernate Quickly*. Greenwich, EU. Publicaciones Manning. 2006.
- RDF. RDF Specification [en línea]. 2005. Disponible en: <http://www.w3.org/RDF/default.htm>
- Wang A. *Component-Oriented Programming*. Georgia, EU. Publicaciones John Wiley & Sons, Inc. 2005. Pp.3-5.
- XML. Extensible Markup Language, W3 Consortium [en línea]. 2003. Disponible en: <http://www.w3.org/XML>

Semblanza de los autores

Alejandro Canales-Cruz. Obtuvo el grado de doctor en ciencias de la computación por el Centro de Investigación en Computación del IPN. Sus líneas de investigación son EBW, multimedia, sistemas inteligentes y desarrollo de software seguro. Asimismo, cuenta con publicaciones y participaciones en congresos en el ámbito nacional e internacional. Actualmente es candidato al Sistema Nacional de Investigadores.

Rubén Peredo-Valderrama. Maestro en ciencias de la computación egresado del Centro de Investigación en Computación del IPN. Sus líneas de investigación son: sistemas multi-agentes, EBW y contenidos de aprendizaje multimedia. Actualmente es candidato al Sistema Nacional de Investigadores, cuenta con alrededor de 20 publicaciones y congresos en el ámbito nacional e internacional.

Leandro Balladares-Ocaña. Es maestro en ciencias de la computación por el Centro de Investigación en Computación del IPN. Cuenta con diversas publicaciones en revistas de arbitraje internacional y congresos nacionales e internacionales. Sus áreas de investigación son: Interacción Hombre-Máquina (HCI), desarrollo de tecnologías basadas en el Web para fomentar el aprendizaje, aplicaciones con DSPs.

Iván Peredo-Valderrama. Ingeniero egresado de la UAM-Iztapalapa. Obtuvo el grado de maestro en ciencias de computación en 2007. Durante los últimos 2 años ha enfocado su trabajo en CBT y posteriormente EBW.

Humberto Sossa-Azuela. Es profesor titular del Centro de Investigación en Computación. Pertenece al Sistema Nacional de Investigadores. Tiene más de 30 publicaciones internacionales con arbitraje riguroso y más de 100 trabajos en conferencias nacionales e internacionales.