

# Hacia una arquitectura para sistemas de e-learning basada en PoEML

Roberto Pérez Rodríguez, Manuel Caeiro Rodríguez, *Member, IEEE*, Luis Anido Rifón, *Member, IEEE*

**Title—** Towards a PoEML-based Architecture for E-learning Systems.

**Abstract—**The current panorama of Learning Technologies is composed of not interoperable technological domains. Thus, there are web-based systems, m-learning systems, and the recent concept of t-learning systems. In this paper we present a proposal of architecture to support the interoperability of e-learning systems, independently of the access technologies used by participants. That architecture enables the collaboration between participants in the same learning process, which is formalized by using the PoEML Educational Modeling Language.

**Index Terms—**e-learning, ubiquitous computing, Computer Supported Collaborative Learning, interoperability

## I. INTRODUCCIÓN

DURANTE los últimos años, las Tecnologías del Aprendizaje se han convertido en un sector estratégico tanto para empresas como para universidades. *E-learning* es una palabra que engloba numerosos conceptos tales como aprendizaje a distancia, aprendizaje asistido por ordenador, y muchos otros. A pesar de los recientes avances en la estandarización de las Tecnologías del Aprendizaje, los sistemas de *e-learning*, tanto propietarios como *open-source*, son desarrollados sin el suficiente soporte para la interoperabilidad. Las consecuencias de este enfoque son que cada nuevo sistema es desarrollado desde cero, los sistemas no son compatibles, los Contenidos Educativos no son portables entre sistemas y, lo que es peor, las Tecnologías del Aprendizaje no evolucionan al siguiente nivel de desarrollo.

Como en todo ámbito tecnológico que alcanza un cierto estado de desarrollo han aparecido ciertos estándares de facto que intentan hacer compatibles las diferentes tecnologías y unificar esfuerzos. La mayor parte de estas especificaciones se centran en los contenidos de los sistemas de *e-learning*

(tutoriales, lecciones, ejemplos, exámenes, etc.), definiendo estándares de clasificación, creación y distribución de dichos contenidos.

En los últimos años ha aparecido un nuevo enfoque en cuanto a las especificaciones de diseño centrado en el proceso educativo (o de aprendizaje atendiendo a la denominación en inglés de *learning process*). El proceso educativo se refiere a la realización y coordinación de las diferentes actividades de una unidad pedagógica (e.g. seminario, jornada, curso académico), en los que no es suficiente con describir los contenidos sino que hay que especificar, por ejemplo, qué tienen que hacer los alumnos, el orden en el que hay que realizar los contenidos, etc.

En la actualidad, la arquitectura de los sistemas de *e-learning* es dependiente de la tecnología de acceso. De esta manera tenemos sistemas inter-departamentales que dependen de la tecnología de la intranet corporativa, sistemas basados en Web, sistemas basados en la televisión digital, y sistemas basados en dispositivos móviles. Cada uno de estos sistemas presenta habitualmente un diseño monolítico que hace virtualmente imposible su extensión para soportar una tecnología de acceso distinta.

XML es el estándar de mayor utilización para la representación de información de forma independiente de su presentación. Las aplicaciones, al proveer interfaces basadas en XML, pueden ofrecer su funcionalidad independientemente de la tecnología final utilizada por el usuario para acceder a ellas. De esta manera, los Escenarios Educativos pueden contener recursos consumibles desde varias tecnologías de acceso.

En este artículo se presenta una propuesta de arquitectura para sistemas de *e-learning* que no depende de la tecnología de acceso empleada. Para ello se ha optado por una Arquitectura Basada en Componentes, en la cual cada componente del sistema tiene una funcionalidad independiente y realiza una o varias interfaces bien definidas. De esta manera, el soporte de una nueva tecnología de acceso se logra añadiendo el componente que realice la nueva funcionalidad, permaneciendo el resto del sistema sin cambios.

En nuestra propuesta, un motor de ejecución de procesos educativos es expuesto como un Servicio Web. De esta manera, la lógica de control de los procesos educativos está centralizada, siendo la misma independientemente de la tecnología de acceso empleada por cada participante.

Roberto Pérez Rodríguez está en el Departamento de Ingeniería Telemática, Universidad de Vigo, 36310 Vigo, España (e-mail: rperez@gist.det.uvigo.es)

Manuel Caeiro Rodríguez está en el Departamento de Ingeniería Telemática, Universidad de Vigo, 36310 Vigo, España (e-mail: Manuel.Caeiro@det.uvigo.es)

Luis Anido Rifón está en el Departamento de Ingeniería Telemática, Universidad de Vigo, 36310 Vigo, España (e-mail: [Luis.Anido@det.uvigo.es](mailto:Luis.Anido@det.uvigo.es))  
DOI (Digital Object Identifier) Pendiente



señalar que cada una de las cuestiones englobadas en un Escenario Educativo se incluye como una entidad separada. De esta manera, se facilita la modificación de Escenarios Educativos mediante el reemplazo de elementos específicos, especificaciones o expresiones, facilitando de esta manera la reutilización. Además, durante el tiempo de ejecución es necesario crear instancias de los Escenarios Educativos y de sus elementos. El número de instancias a crear puede ser determinado estáticamente durante el tiempo de diseño o dinámicamente durante el tiempo de ejecución de acuerdo con el resultado proporcionado por expresiones específicas.

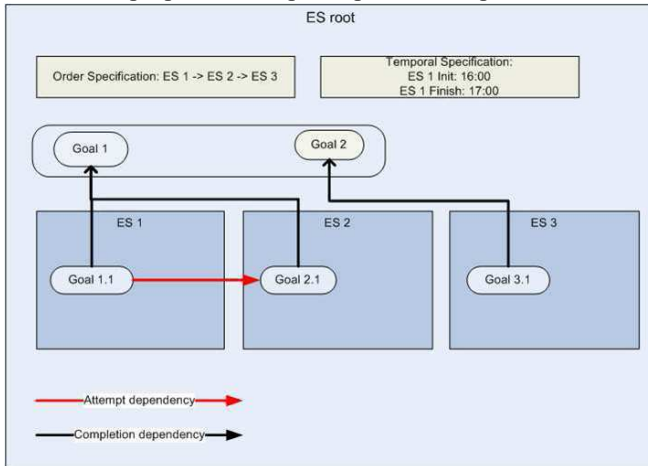


Figura 2. Agregación de Escenarios Educativos y objetivos

### III. ARQUITECTURA

La Figura 4 muestra un diagrama con la arquitectura del sistema. En las siguientes subsecciones se detalla dicha arquitectura.

#### A. La máquina de procesos educativos: PoEML Engine

El motor de ejecución de procesos educativos es el componente central del sistema de *e-learning* [4]. En él se almacena la información relativa a los Escenarios Educativos, participantes, objetivos, y resto de elementos, haciendo evolucionar el estado del sistema dependiendo de los eventos, tanto externos al motor como internos a él, que se producen.

El motor de ejecución se integra en el sistema de *e-learning* a través de una interfaz bien definida basada en Web Services, garantizando el requisito de la conectividad. Al mismo tiempo, los componentes de presentación deben estar lo más desacoplados que sea posible respecto al motor de ejecución, esto implica utilizar un sencillo conjunto de APIs. La escalabilidad también es un requisito fundamental, al ser el motor de ejecución el componente central del sistema de *e-learning*.

Eventos generados por el participante:

- Comenzar un ES
- Finalizar un ES
- Intentar un objetivo
- Darse de alta/darse de baja

Eventos generados por el motor de ejecución:

- Instanciar un ES

- Instanciar un objetivo
- Dar por finalizado un objetivo
- Hacer accesible/no accesible un ES
- Hacer intentable/no intentable un objetivo

Los componentes de presentación también pueden acceder al motor de ejecución de una manera pasiva simplemente para recuperar información, tanto sobre los Escenarios Educativos como sobre los objetivos. Esta recuperación de información está asociada a la navegación por los Escenarios Educativos:

- Conseguir la información relativa a un Escenario Educativo
- Conseguir la información de los subescenarios de un cierto ES

Un Escenario Educativo tiene que ser instanciado para que un usuario pueda acceder a él. Este caso es similar a la instanciación en un lenguaje de programación orientado a objetos, donde un objeto tiene que ser instanciado antes de poder acceder a sus atributos y métodos. En el caso que nos ocupa, un Escenario Educativo puede estar también en los estados *Iniciado* y *Finalizado*. La Figura 3 muestra todos los posibles estados de ejecución en que puede estar un Escenario Educativo.

Un evento externo al motor de ejecución como el acceso a un Escenario Educativo puede desencadenar varios eventos internos al motor de ejecución como la instanciación de sus Subescenarios Educativos. Al mismo tiempo, se impone la restricción de que dichos Subescenarios Educativos tengan un objetivo propuesto. Este comportamiento es un problema bien conocido, que se resuelve mediante reglas Evento-Condición-Acción:

- Evento: un participante accede a un Escenario Educativo
- Condición: que el Subescenario tenga un objetivo propuesto
- Acción: instanciar el Subescenario

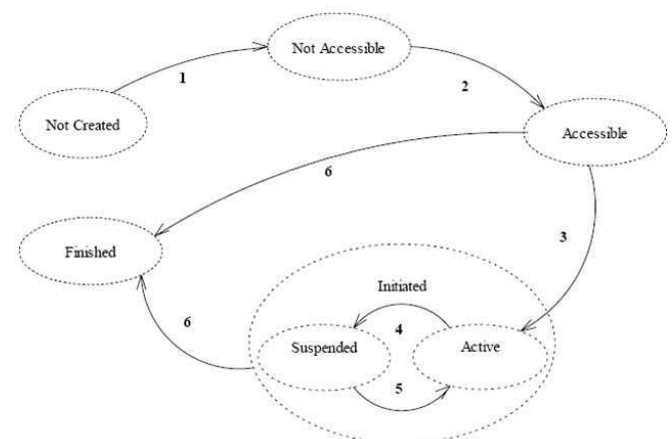


Figura 3. Posibles estados de ejecución de un Escenario Educativo

De una manera similar, el motor de ejecución tiene que manejar los eventos relativos a los objetivos de los Escenarios Educativos. El participante puede generar el evento de intentar objetivo, con los posibles resultados de éxito y fracaso. Este evento genera en el motor de ejecución los eventos de



de ejecución de procesos educativos [6]. El enfoque para esto está basado en la programación orientada a aspectos. El Desarrollo Software Orientado a Aspectos (*Aspect-oriented Software Development*) es un paradigma que permite modularizar asuntos entrecruzados (*crosscutting concerns*), esto es, código que no puede ser capturado en módulos utilizando el mecanismo de descomposición presente en el lenguaje de programación empleado. En el caso de Moodle, las unidades de modularización son scripts PHP, y los nuevos asuntos entrecruzan multitud de scripts PHP pertenecientes al código fuente de Moodle. El procedimiento para añadir un nuevo *asunto* a Moodle es:

- Listar los puntos la ejecución de Moodle donde ese *asunto* tiene que ser tenido en cuenta. Esos puntos se definen como *joinpoints*
- Un *pointcut* se define como un conjunto de *joinpoints*
- El último paso es definir el código *advice* a ejecutar en cada *pointcut*. El código *advice* encapsula las llamadas a los Web Services del motor de ejecución de procesos

#### B. iTV

Consideramos que la experiencia de los participantes en los sistemas de t-learning sería enriquecida al integrar un motor de ejecución de procesos educativos no perteneciente al dominio de la iTV (*Interactive Television*).

Se utiliza un servidor DVB (*Digital Video Broadcasting*) para el *streaming*, mientras que los Web Services que provee el motor de ejecución controlan el proceso de aprendizaje. El participante debe contar con un receptor MHP (*Multimedia Home Protocol*).

#### C. M-learning

Se llama *m-learning* al aprendizaje a través del uso de dispositivos móviles. Estos dispositivos móviles son una de las tecnologías más prometedoras para el soporte del aprendizaje, ya que el participante puede acceder al sistema en cualquier tiempo y lugar. Este enfoque se ajusta especialmente bien en escenarios colaborativos, donde los participantes pueden colaborar entre sí mediante el uso de un dispositivo móvil.

Cuando se accede a un Escenario Educativo desde un dispositivo inalámbrico móvil, el sistema tiene que saber cómo ensamblar los contenidos para posteriormente enviarlos al dispositivo móvil. Además, la información relativa a un Escenario Educativo debe ser formateada para ajustarse a dicho dispositivo.

Con nuestro enfoque, tendríamos de un módulo de presentación para *m-learning* capaz de comunicarse con los dispositivos móviles utilizando los protocolos de comunicación empleados en las redes inalámbricas. Este módulo de presentación consumiría los Web Services que provee el motor de ejecución de PoEML. Así, el funcionamiento es muy similar a un sistema basado en Web, ya que hacemos separación del código dependiente del dominio tecnológico (web, dispositivos móviles, iTV) del

código independiente del dominio (el motor de ejecución de procesos educativos).

### V. LA EXTENSIÓN DE MOODLE

La extensión que hemos desarrollado se compone de un nuevo tipo de curso con tres vistas: la vista en tiempo de ejecución (*run-time*), la herramienta de autoría, y la vista de monitorización. En esta sección detallaremos estas tres vistas.

Antes de detallar los diferentes entornos, es importante describir el concepto de *descripción del curso*, así como el de *instancia de curso*. Una descripción de curso es análoga a una clase en la Programación Orientada a Objetos: una guía a partir de la cual los objetos son creados. De esta manera, múltiples instancias de curso pueden ser creadas a partir de una descripción de curso: una por estudiante.

En los cursos Moodle actuales, se necesita simplemente un esquema de base de datos para guardar la descripción de cursos. En nuestro caso, necesitamos un esquema de base de datos para las descripciones de cursos y otro para las instancias de cursos en tiempo de ejecución.

#### A. El entorno de autoría

El proceso de autoría consiste en una serie de operaciones atómicas. En el conjunto de operaciones posibles hay algunas como:

- Añadir/editar/borrar un Escenario
- Añadir/editar/borrar un Objetivo
- Añadir/editar/borrar un recurso/actividad

Los cambios atómicos son validados para asegurar que son consistentes con el diseño general del curso. Después de haber verificado su consistencia, el cambio atómico se compromete contra el esquema de base de datos que contiene las descripciones de cursos. El nivel de granularidad puede ser fijado como se desee: una operación atómica puede consistir en añadir un Objetivo con todos sus campos constituyentes, o puede ser editar un Objetivo previamente creado y cambiar sólo un campo. El proceso de autoría consistente en cambios atómicos presenta algunas ventajas valorables:

- Los cambios atómicos son inmediatamente vistos por otros co-autores
- No hay necesidad de un chequeo de la consistencia como el que se necesita al importar un archivo que contiene una descripción de un curso completo

El chequeo de la consistencia es aún necesario para el soporte de la interoperabilidad. Las descripciones de cursos siguen el modelo de datos de PoEML. Por lo tanto, las descripciones de cursos pueden ser exportadas desde su esquema de base de datos a archivos XML. Así, un archivo XML que contiene una descripción de curso puede ser importado en otro LMS compatible con PoEML.

La Figura 5 muestra el proceso de autoría de varios expertos trabajando en el mismo diseño. Los co-autores realizan cambios atómicos en el diseño del curso, y esos cambios son comprometidos contra el esquema de base de datos de las descripciones de cursos. El diseño del curso puede ser testado



La Figura 8 muestra la monitorización de un objetivo. En la parte superior de la página, podemos ver los posibles estados de un Objetivo y el número de estudiantes por estado. Se usa un histograma para mostrar esa información de una manera gráfica. En la parte inferior de la página, podemos ver una tabla detallando el estado de cada Objetivo para cada estudiante enrolado en el curso.

Las Variables y las Expresiones de Datos pueden ser también monitorizadas. Las Variables pueden servir para expresar la nota de un estudiante en un cierto cuestionario. Las Expresiones de Datos sirven para componer decisiones que dependen del rendimiento del alumno, por ejemplo, una decisión que depende de la nota de un alumno en un cuestionario.

La otra vista de monitorización disponible es la dirigida por estudiante. Esa vista es apropiada para monitorizar la progresión de un alumno a través de los itinerarios del curso.

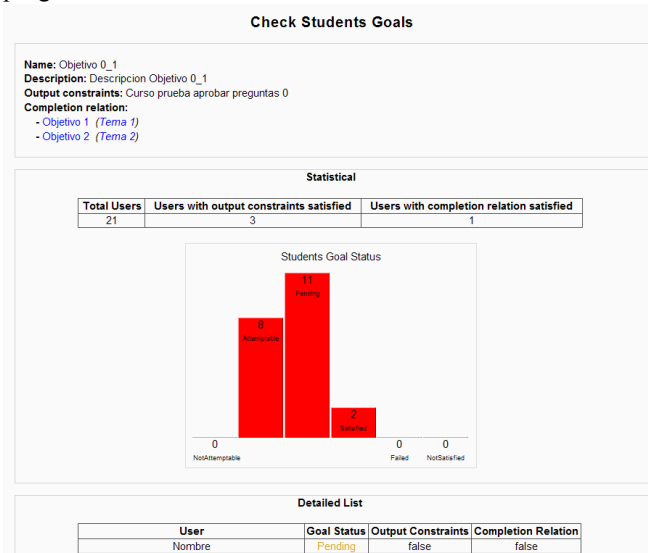


Figura 8. Captura de pantalla del entorno de monitorización.

## VI. TRABAJO RELACIONADO

### A. CopperCore

CopperCore [7] es la implementación de referencia de un IMS Learning Design engine y también provee un player basado en ese engine. El objetivo fue proporcionar una fuente de información para implementar otros players compatibles con LD. En la práctica ha sido visto como el player de referencia y herramienta para ejecutar diseños y para validar otras herramientas de LD tales como editores, herramientas de autoría, etc.

Los objetivos de CopperCore incluían testear si el diseño de la API permitía que un thin client fuese construido encima del engine a un coste reducido; y dar una indicación de la funcionalidad que se esperaba de los IMS-LD players. Rendimiento y escalabilidad no son contempladas por CopperCore, aunque se espera que sea uno de los objetivos de los IMS-LD players en producción.

A nivel interno CopperCore utiliza una máquina de estados finita (FSM - Finite State Machine) para evaluar propiedades y secuenciar condicionalmente el contenido. Por cada UoL importada se crea una FSM y un árbol de actividades es adjuntado a cada estado y mostrado al usuario cuando tal estado es alcanzado.

Los siguientes players utilizan CopperCore:

- SLED [8], Service-based Learning Design system. Está enfocado en el uso de web services: la comunicación entre el engine y el player es realizada usando web services. Herramientas adicionales y para usuarios finales (e.g. sistemas de conferencia) fueron construidas orientadas a web services. SLED presenta mayor funcionalidad que el básico CopperCore engine. Por ejemplo, el CopperCore engine sólo es capaz de indicar que el diseño de aprendizaje necesita un foro, mientras que SLED muestra el foro, ejecutándolo a través del servicio de conferencia.
- El incluido en el proyecto Reload [9]. Este proyecto provee una familia de herramientas relacionadas con las especificaciones IMS, desde el ampliamente utilizado IMS-CP packager hasta un IMS-LD editor. Su implementación está basada en las plataformas Java y JBoss. Su funcionalidad incluye: importar/borrar de Learning Designs en el CopperCore engine con una simple interfaz gráfica, leer automáticamente un Learning Design y poblar el engine con un usuario activo por cada rol encontrado dentro del manifiesto (pueden ser añadidos usuarios a medida también), etc.

### B. GRAIL

GRAIL (Gradient-lab RTE for Adaptive IMS-LD in .LRN) es el entorno de ejecución de IMS-LD implementado en .LRN. Ha sido concebido para ser utilizado dentro del contexto de una comunidad .LRN, un conjunto de usuarios compartiendo recursos tales como documentos, foros, calendarios, etc. Un curso es simplemente una instancia de una de esas comunidades. Como suele ser el caso, y IMS-LD no es una excepción, las especificaciones no incluyen detalles de cómo debe ser implementado el RTE (Run-Time Environment). Esto normalmente se traduce en un amplio conjunto de decisiones que tiene que tomar el equipo de diseño. Tienen que ver con importantes aspectos de la usabilidad y efectividad del entorno de ejecución y por lo tanto tienen que ser cuidadosamente consideradas.

Otros entornos de ejecución de IMS-LD como CopperCore utilizan un modelo de máquina de estados finitos (FSM) para evaluar propiedades y secuenciar el contenido condicionalmente. Para cada Unidad de Aprendizaje (UOL-Unit Of Learning) importada, una máquina de estados finitos es creada y un árbol de actividades es creado para cada estado y mostrado al usuario cuando el estado es alcanzado.

La evaluación de una condición debe conducir a un árbol de actividades nuevamente calculado, conteniendo sólo aquellos





**Roberto Pérez Rodríguez** es Ingeniero de Telecomunicación (2008) en la especialidad de Telemática por la Universidad de Vigo. Actualmente es estudiante de doctorado en el Departamento de Ingeniería Telemática de la Universidad de Vigo y realiza labores docentes como Profesor Invitado en dicho departamento. Su labor investigadora está centrada en la utilización de Arquitecturas Orientadas a Servicios para proveer servicios de e-learning.



**Manuel Caeiro Rodríguez** es Ingeniero de Telecomunicación (1999) y Doctor Ingeniero de Telecomunicación (2007) por la Universidad de Vigo. Actualmente es Profesor Contratado Doctor en el Departamento de Ingeniería Telemática de la Universidad de Vigo realizando tareas docentes relacionadas con la ingeniería del software y la arquitectura de ordenadores. En cuanto a la investigación su interés principal se centra en la aplicación de las Tecnologías de la Comunicación y la Información a la educación, en especial en el marco de los lenguajes de modelado educativo.



**Luis Anido Rifón** es Ingeniero de Telecomunicación *cum laude* (1997) en las especialidades de Telemática y Comunicaciones, y Doctor Ingeniero de Telecomunicación *cum laude* (2001) por la Universidad de Vigo. Actualmente es Profesor Titular de Universidad en el Departamento de Ingeniería Telemática de la Universidad de Vigo y ocupa el puesto de Director del Área de Innovación Educativa de la Universidad de Vigo. Ha recibido varios premios por el W3C, la Royal Academy of Sciences y la Colegio Oficial de Ingenieros de Telecomunicación. Es autor de más de 180 artículos en revistas y congresos. Es también el director del Programa Gallego de Investigación en TIC, secretario técnico de AENOR CTN71 SC36 y el director de la delegación española de ISO/IEC JTC1 SC36.