

Diseño Instruccional en los Lenguajes de Modelado Educativo¹

Manuel Caeiro Rodríguez, Martín Llamas Nistal, Luis Anido Rifón

Departamento de Enxeñería Telemática
E.T.S.E. Telecomunicación, Vigo, España
{Manuel.Caeiro, Martin.Llamas, Luis.Anido}@det.uvigo.es

Resumen: Los Lenguajes de Modelado Educativo (EMLs, atendiendo a sus siglas en inglés) han sido propuestos para permitir el diseño instruccional de materiales de e-learning. Estos lenguajes se pueden englobar en el movimiento de estandarización de e-learning cuyo objetivo es permitir el modelado de materiales educativos en unidades reutilizables e interoperables. Las propuestas existentes hasta este momento se desarrollaban entorno al concepto de *Objeto de Aprendizaje*, concebido como unidad básica de reutilización basada en el empaquetado de materiales educativos, pero en los que no se tenían en cuenta cuestiones pedagógicas o instructivas. Los EMLs se caracterizan por permitir modelar la utilización instructiva de los materiales educativos considerando la forma en que los aprendices acceden al contenido, la interacción entre los participantes (otros alumnos y profesores), etc. Actualmente, IMS Learning Design es la propuesta de EML más destacada. Este artículo considera tres aspectos principales que deberían considerarse en los EMLs (generalidad, completitud y flexibilidad) y presenta una propuesta para realzar el soporte de diseño instruccional.

Palabras clave: e-learning, aprendizaje colaborativo, lenguajes de modelado educativo, unidad de aprendizaje, objeto de aprendizaje.

Abstract: Educational Modelling Languages (EMLs) have been proposed to support the instructional design of e-learning materials. These languages were developed as part of the e-learning standardization movement to enable the description of educational materials in a reusable and interoperable way. Previous Learning Object specifications are centred on the packaging of educational materials as reusable building blocks and do not consider the way in which instruction or teaching have to be carried out. EMLs are concerned with educational materials, but they also consider the way in which learners use contents, the interaction among participants (other learners and academic staff), etc. Currently, IMS Learning Design (LD) is the most outstanding EML proposal. This paper considers three main concerns that should be considered in EMLs and presents a proposal to enhance e-learning instructional design.

Key words: e-learning, collaborative learning, educational modelling language, unit of learning, learning object.

1. Introducción

El diseño instruccional está presente en mayor o menor grado en cualquier práctica educativa. En un contexto tradicional de enseñanza cara-a-cara, los

profesores pueden involucrarse de forma consciente y reflexiva en el proceso de diseño instruccional como parte de su planificación docente, mientras que otros profesores, aunque nunca han pensado demasiado sobre ello, también toman decisiones de diseño

¹ Artículo seleccionado de SINTICE, VI Congreso Nacional de Informática Educativa (Granada 2005), extendido y revisado para su publicación en IE Comunicaciones.

instruccional cada vez que se preparan para una clase. De forma similar, el diseño instruccional está presente explícita o implícitamente en las prácticas de e-learning. Cualquier sistema de e-learning es el resultado de un diseño instruccional, consciente o inconsciente, con el que se intenta proporcionar ciertas funcionalidades y alcanzar un entorno educativo en el que los alumnos puedan aprender.

El término diseño instruccional¹ [Smith y Ragan 2005] se refiere al proceso sistemático y reflexivo de desarrollar planes educativos en los que se establecen actividades, recursos de información, métodos de evaluación, etc. De esta forma el diseñador instruccional establece cómo será la práctica educativa. Existen numerosas teorías de diseño instruccional en las que se proporcionan indicaciones para diseñar productos instruccionales [Merrill 2001] de acuerdo a distintas teorías sobre el aprendizaje. Estas teorías describen métodos educativos junto con situaciones en las que tales métodos deberían utilizarse [Reigeluth 1999]. Por todo ello, el diseño instruccional se ha concebido como una "ciencia entrelazada" entre las teorías del aprendizaje y las prácticas educativas [Paquette 2003]. En su nivel más simple, una teoría de diseño instruccional puede proporcionar una serie de instrucciones a los profesores. A modo de ejemplo una teoría podría plantear los siguientes pasos: (i) presentar los objetivos de aprendizaje; (ii) proporcionar algunos ejemplos concretos; (iii) poner en marcha las generalizaciones inherentes en los ejemplos concretos; (iv) proporcionar algunos ejercicios en los que se apliquen las generalizaciones; etc. En general, las teorías de diseño instruccional no se limitan a un conjunto de reglas que se suceden en un cierto orden, sino que pueden involucrar indicaciones más abiertas y generales.

Recientemente, el diseño instruccional ha captado numerosos esfuerzos en el proceso de estandarización en e-learning debido al desarrollo de los denominados

“Lenguajes de Modelado Educativo” (EML: *Educational Modelling Languages*) [Koper 2001]. Antes de la aparición de estos lenguajes, las iniciativas de estandarización en este dominio se habían centrado en el desarrollo de especificaciones sobre los “Objetos de Aprendizaje” (*Learning Objects*), concebidos como unidades básicas de agregación de recursos educativos para facilitar su reutilización e interoperabilidad. En otras palabras, en los objetos de aprendizaje se considera solamente las características de los bloques de información que faciliten la búsqueda y la transferencia de recursos educativos [IEEE 2002]. Sin embargo, en general, estas iniciativas no se habían preocupado por la forma en que dichos recursos debían ser utilizados², es decir, sobre el diseño instruccional de los recursos mismos [Downes 2003]. En su propuesta inicial de EML Rob Koper [Koper 2001], uno de los desarrolladores y promotores más importantes de los EMLs, realiza una crítica clave contra la excesiva importancia dedicada a los objetos de aprendizaje como simples “pedazos” de información: "*Muchas veces el aprendizaje no se consigue simplemente a partir de recursos de conocimiento, sino que proviene de las actividades que los aprendices realizan con dichos recursos: solucionando problemas, interactuando con dispositivos reales, interactuando en su situación social y de trabajo...*". Por ello, a diferencia de las especificaciones anteriores sobre objetos de aprendizaje, los EMLs se han enfocado primordialmente a permitir el modelado del diseño instructivo de recursos educativos. Finalmente, el objetivo es permitir que dichos diseños puedan dirigir la ejecución de aplicaciones informáticas que proporcionen las funcionalidades necesarias y gestionen el desarrollo de prácticas educativas de acuerdo a dichos diseños.

Actualmente la especificación de Diseño de Aprendizaje del IMS (LD: *Learning Design*) es la propuesta más destacada de EML. Además, en la práctica, es la especificación de referencia para EMLs

¹ Se utiliza la denominación de diseño instruccional como traducción del término inglés “instructional design”. Ahora bien, es necesario indicar que este término tiene connotaciones negativas en el mundo hispano, dado su carácter implícito de diseño rígido y estricto. Algunos autores consideran más conveniente la utilización del término diseño didáctico, si bien este concepto es más amplio que el que nosotros consideramos, pues en él se pueden tener en cuenta cuestiones afectivas, emocionales, etc.

² Únicamente se habían propuesto especificaciones sobre estructuras de contenidos, en las que se puede indicar una estructura estática (de agrupación jerárquica de contenidos) y otra estructura dinámica (sobre el orden en que dichos contenidos podían ser entregados a los alumnos). Por tanto, el diseño instruccional considerado en estas especificaciones es muy limitado. Las propuestas más importantes de este tipo son ADL SCORM [Dodds 2005] e IMS Content Packaging [Nielsen 2005].

y el motor que aglutina los esfuerzos y proyectos de modelado de diseño instructivo en e-learning. La unidad básica que constituye el núcleo de la especificación IMS LD está constituida por el concepto de "Unidad de Aprendizaje" (UoL: *Unit of Learning*), como recurso básico de aprendizaje –reutilizable e interoperable igual que los Objetos de Aprendizaje– que, además de la colección de recursos de conocimiento, incluye las indicaciones sobre cómo dichos recursos deben ser utilizados tanto por parte de los alumnos como de los profesores (e.g. actividades para solucionar problemas, actividades de búsqueda, actividades de discusión, actividades de evaluación), junto con las aplicaciones, los servicios y demás recursos de apoyo que pudieran ser necesarios.

El artículo se estructura de la siguiente manera: la próxima sección introduce los principales aspectos de EMLs relacionados con el diseño instruccional en los que hemos centrado nuestra investigación. En la sección 3 se explica una propuesta para permitir el modelado de estos aspectos. En la sección 4 se muestra un ejemplo de aplicación de la solución propuesta. Finalmente en la sección 5 presentamos las conclusiones.

2. EMLs y Diseño Instruccional

Como hemos indicado, los EMLs se proponen para permitir modelar el diseño instruccional de prácticas educativas. El concepto de práctica educativa se considera de forma abierta y general, bien involucrando desde simples lecciones de corta duración a cursos completos de un año, bien desde cursos tradicionales basados en la entrega de contenidos y la realización de cuestionarios a aproximaciones basadas en la resolución de problemas o realización de proyectos, bien desde cursos de carácter estrictamente individual a esquemas colaborativos basados en la colaboración o la discusión, etc.

La aproximación que siguen los EMLs para permitir el modelado de diseño instructivo se centra en la coordinación. En contra de lo que podría pensarse, la base para permitir el modelado instruccional no se sitúa en cuestiones pedagógicas o tecnológicas. Por el contrario, el modelado se enfoca hacia la descripción

de los elementos que participan en las prácticas educativas junto con la coordinación necesaria para que dichos elementos interactúen de una determinada forma. Como resultado de dicha interacción se espera que los alumnos aprendan. Por ejemplo, el lenguaje debe permitir modelar los contenidos que serán ofrecidos, los participantes que los recibirán (alumnos y profesores), la comunicación que se puede producir entre los participantes, la forma en que los participantes podrán acceder a los contenidos, los objetivos que debe conseguir cada uno de los participantes, etc. De esta forma, abstrayéndose de pedagogía y tecnología, y centrándose en los elementos involucrados y su coordinación, es posible modelar la variedad de prácticas educativas descritas en el párrafo anterior.

En la práctica, para permitir el modelado instructivo los EMLs se organizan de acuerdo a un meta-modelo formal de elementos y relaciones que determinarán la sintaxis y la semántica del lenguaje. Por otra parte, este meta-modelo debe permitir el desarrollo de aplicaciones y sistemas computacionales que soporten el procesamiento de los modelos construidos con el lenguaje. En el caso concreto de IMS LD, el meta-modelo correspondiente [Koper et al. 2003] se organiza alrededor de las definiciones de las actividades a realizar por los participantes (alumnos y personal académico) en entornos concretos. Los entornos reúnen los recursos (e.g. objetos de aprendizaje) y los servicios disponibles para realizar la actividad correspondiente. Las actividades se proponen a los participantes en cierto orden, por ejemplo permitiendo que varias actividades se aborden en paralelo. Esta especificación también considera propiedades, condiciones y notificaciones para permitir la variación del plan prescrito según condiciones particulares durante su ejecución (la ejecución es el acto de crear y manejar una instancia de un modelo de una práctica educativa descrita con un EML).

Nuestra investigación en cuanto a los EMLs se ha centrado en la mejora del soporte de diseño instructivo. Desde un punto de vista computacional, hay una gran cantidad de cuestiones de coordinación que se pueden incluir en el modelado propuesto por los EMLs para potenciar el soporte instruccional final de prácticas educativas. En particular, nuestro trabajo

se basa en el desarrollo de un EML centrado en tres criterios clave:

- *Generalidad*: Este criterio está relacionado con la clase y el tipo de prácticas educativas que un EML debe permitir modelar. Como se describe en un párrafo anterior, un EML debiera permitir el diseño de distintas aproximaciones educativas, considerando diferentes niveles de agregación, diferentes modos de presentación, etc. Por lo tanto, el lenguaje debería permitir el diseño de un currículo completo, un curso sencillo, una pequeña lección, etc. Por otra parte, existe una amplia gama de diversas aproximaciones educativas que consideran varios aspectos a la hora de llevar a cabo la educación: el tipo de contenido, las tareas que tienen que realizar los aprendices, la ayuda proporcionada por los tutores, las evaluaciones, la interacción con los demás, etc. En este momento centramos nuestra atención en el aprendizaje colaborativo, como esquema en el que varios participantes (alumnos y profesores) interactúan entre sí para conseguir que se produzca el aprendizaje. Nuestro propósito final es proponer un EML lo más genérico posible. El punto clave que se persigue con este criterio es no limitar ni condicionar la libertad de los diseñadores para utilizar un determinado EML, sino que con un mismo lenguaje se puedan afrontar la gran diversidad de situaciones posibles. De esta forma se estaría fomentando la reutilización y la interoperabilidad de los recursos de aprendizaje, dado que, por ejemplo, en una misma unidad educativa podrían combinarse distintos tipos de aproximaciones pedagógicas, distintos niveles de agregación, etc.
- *Completitud*: La idea principal de este criterio es permitir modelar los elementos y las distintas formas de coordinación que se pueden considerar en prácticas educativas. Como hemos indicado, la aproximación de los EMLs para permitir el modelado del diseño instruccional se basa en la caracterización de los elementos que participan en la práctica y de la coordinación entre ellos. Por ejemplo, un tipo de coordinación que los EMLs actuales caracterizan es la que se refiere al control del proceso (o flujo de control), que se ocupa de la manera en la cual el control fluye de una actividad a otra, estableciendo el inicio, la finalización y el tiempo de realización para cada

actividad. Actualmente, las tecnologías de la información y la comunicación permiten y soportan numerosas formas de coordinación. En nuestra investigación hemos realizado un extenso trabajo para evaluar la expresividad de los EMLs. Por una parte identificamos los tipos de coordinación que se pueden considerar en las prácticas educativas. A cada uno de los tipos de coordinación lo denominamos 'perspectiva'. Cada perspectiva involucra un propósito particular y específico distinto de las otras perspectivas. En segundo lugar, se caracterizan las distintas formas de coordinación que se pueden considerar en cada una de las perspectivas, identificando y describiendo un conjunto básico de 'patrones de diseño'. Cada uno de los patrones de diseño recoge una solución genérica a un problema de diseño común en la perspectiva. En su conjunto, el grupo de patrones identificados permite describir cualquier forma de coordinación que se pueda considerar en la perspectiva. Finalmente, evaluamos los EMLs tratando de modelar cada uno de los patrones de las distintas perspectivas. De esta forma, obtenemos una indicación sistemática y rigurosa de las capacidades de cada uno de los lenguajes (en el apartado 3.2 se presenta una breve descripción de las perspectivas y los patrones identificados).

- *Flexibilidad*: Este criterio se centra en las modificaciones o las variaciones que se pueden producir durante la ejecución de una práctica educativa. En escenarios educativos tradicionales, los diseños instructivos requieren generalmente de muchos cambios y adaptaciones durante su funcionamiento. Los profesores algunas veces no especifican el plan instructivo de forma completa y detallada antes del inicio de la actividad docente, sino que consideran sólo los puntos principales y proporcionan la especificación adicional durante el "tiempo de ejecución". Por otra parte, aunque el plan educativo se realice de manera completa y rigurosa, durante el desarrollo del mismo suelen surgir numerosos problemas o situaciones no consideradas que requieren el cambio *al vuelo* de dicho plan (e.g. el horario no está bien organizado; algún grupo de trabajo necesita más ayuda). Por lo tanto un EML debería permitir el modelado de diseños educativos que puedan sufrir adaptaciones durante su *ejecución*.

Las ofertas actuales de EML proporcionan una cierta clase de ayuda para la modificación de los diseños instructivos a través del uso de propiedades y de condiciones que se procesan de forma automática. En nuestra propuesta consideramos una mayor especificación de esta capacidad permitiendo indicar los tipos de modificaciones que pueden producirse junto con las condiciones que tienen que producirse o las decisiones que ciertos participantes autorizados (e.g. tutores) tienen que realizar para que dichos cambios puedan producirse.

3. Una Propuesta de EML para Diseño Instruccional

Esta sección presenta una propuesta de meta-modelo para un nuevo EML que proporcione un mejor soporte para permitir el diseño instruccional de recursos de aprendizaje para e-learning. En la sección anterior describimos los criterios de diseño que guían nuestro trabajo: generalidad, completitud, y flexibilidad. Como en otros EMLs la solución se representa según un meta-modelo enfocado a la descripción de actividades. En este sentido, el trabajo está muy relacionado con las tecnologías de gestión del *workflow*, *groupware*, y las propuestas de interacción hombre-máquina. Especialmente, hemos tomado las ideas clave de las siguientes referencias [Aalst et al. 2003a, Jorgensen 2004, Russell et al. 2004].

El “Escenario Educativo” (ES: *Educational Scenario*) es el elemento central de nuestra propuesta de modelado (cf. Figura 1). Se trata del punto básico de agregación donde se sitúan el resto de los elementos del meta-modelo. Además, cada ES constituye un contexto no accesible desde otros ESs. De esta manera es posible construir espacios educativos bien-estructurados y separados. Un ES reúne los elementos siguientes:

- Escenarios Educativos. Un ES puede contener otros ESs. De la misma forma que un curso puede descomponerse en varias lecciones, un ES padre puede descomponer su objetivo general en varios sub-objetivos más concretos que se corresponderían con varios sub-ESs hijo. Es importante destacar que todos los ESs presentan

la misma organización. Los “Conectores de Control” (ControlConnector) y los “Conectores Temporales” (Temporal Connector) se pueden introducir para permitir relacionar sub-ESs hijo de un cierto ES entre sí. De esta forma es posible modelar escenarios en los que un ES se compone de varios sub-ESs entre los que no se establece ningún orden específico o condiciones temporales, y otros escenarios en los que si se considere la misma descomposición junto con una determinada ordenación o condiciones temporales entre los sub-ESs hijo.

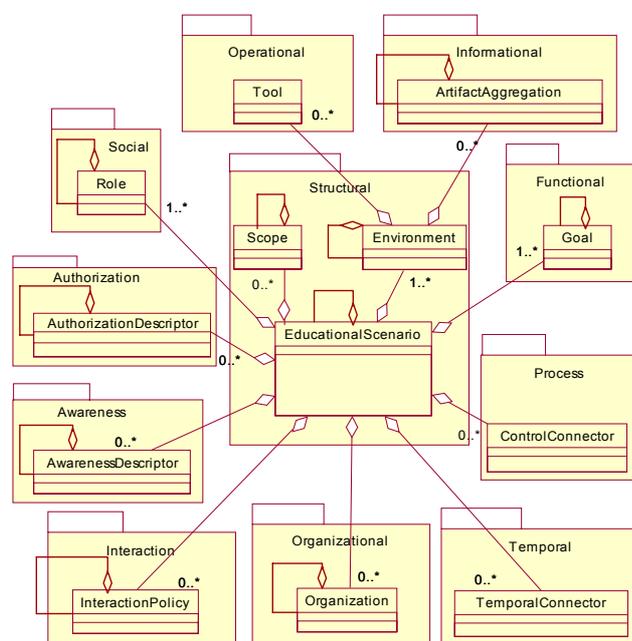


Figura 1. Entidades principales del meta-modelo propuesto

- *Objetivos*. Cada ES necesita considerar uno o varios objetivos (*Goals*) que dirijan la acción de los participantes (alumnos y profesores) en el mismo (cf. Figura 2). En nuestra propuesta consideramos que los objetivos pueden agregarse en estructuras de objetivos compuestos que pueden descomponerse en sub-objetivos, o caracterizarse como objetivos múltiples, que tienen que ser propuestos y alcanzados varias veces. Como ejemplo de objetivos múltiples están los objetivos que tienen que ser realizados por cada uno de los alumnos de un ES. Se deben crear tantas instancias de cada uno de estos

objetivos como alumnos existan. Para determinar el número de veces que se debe proponer un mismo objetivo se utiliza un indicador específico (*GoalMultiplicity*). Sin embargo, en este punto es importante tener en cuenta que no se indica quién debe realizar dicho objetivo. Dicho responsable será determinado finalmente a través del elemento *Role*. Los objetivos tienen propiedades que permiten el modelado de diferentes vocabularios en cuanto a su estado de completitud (e.g. 0..10, No intentado – fallado – superado) junto con las condiciones o procesos que se utilizan para determinar dicho estado de completitud. En relación con sus elementos asociados, los objetivos pueden enlazarse con la especificación de agregaciones de artefactos (*ArtifactAggregation*) de entrada y de salida, indicando los parámetros de entrada y de salida de los mismos. Los conectores de objetivos (*GoalConnectors*) se utilizan para relacionar objetivos y permiten modelar objetivos de tipo obligatorio, opcionales, prohibidos o condicionados (e.g. antagónicos, complementarios) a otros objetivos. Estas conexiones también se utilizan para indicar cómo se debe determinar el estado de completitud de un objetivo en relación al estado de completitud de sus sub-objetivos. En el apartado 4 se presenta un ejemplo de modelado de objetivos utilizando estas entidades.

- **Entornos.** El ES debe incluir algunos entornos (al menos uno) que contengan los recursos, documentos (*Artifacts*) y herramientas (*Tools*), que se podrán utilizar para alcanzar los objetivos previstos en el mismo. Los entornos pueden ser agrupados de forma jerárquica, estando finalmente compuestos por documentos (e.g. páginas del HTML, textos, figuras, etc.) y herramientas (e.g. instalaciones de comunicación, simuladores, procesadores de texto, etc.). Como parte de los entornos de un ES pueden incluirse dos sub-entornos especiales con agregaciones asociadas con cada uno de los objetivos del mismo, que son los parámetros de entrada y de salida (cf. Figura 2)). Estos parámetros se pueden relacionar con condiciones de los conectores de control para determinar cuando se puede iniciar y finalizar un ES determinado. Al igual que en los dos casos anteriores, entre las agregaciones de artefactos (*ArtifactAggregation*) se pueden

establecer conectores de artefactos (*ArtifactConnectors*) que permiten el modelado del flujo de información entre entornos y por extensión entre ESs. Estos conectores de artefactos indican cómo deben copiarse documentos, transferirse, etc. Estos conectores también permiten relacionar los parámetros de entrada de una actividad con los parámetros de salida de una previa, así como con las posibles operaciones en herramientas. El elemento de ámbito (*scope*) se utiliza para modelar elementos relacionados que están situados en uno o varios entornos. Por ejemplo, los contenidos de un curso que deben presentarse en el caso de que el alumno necesite una explicación detallada de los conceptos pueden agruparse como bajo un ámbito “de explicación detallada”. De esta forma, activando dicho ámbito se mostrarán estos contenidos independientemente del entorno en el que se encuentren.

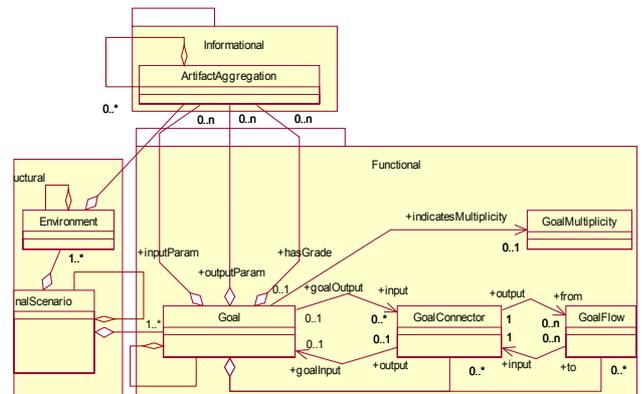


Figura 2. Entidades utilizadas para realizar el modelado de objetivos

- **Roles.** El ES necesita considerar a algunos participantes, por lo menos a uno, para realizar las tareas previstas que conduzcan a la consecución de los objetivos del mismo. El número de participantes no está restringido, y los participantes pueden ser organizados en jerarquías de grupo (cf. Figura 3). Hemos considerado tres roles especiales: (i) el rol *Profesor* no es obligatorio, pero si se incluye, este rol puede modificar el ES prescrito según se desee y controlar su ejecución; (ii) el rol *Responsable* es obligatorio (pero puede ser

realizado por varios participantes), y tiene autorización para definir nuevos ESs hijos, permitiendo una descomposición del actual; (iii) finalmente, puede haber cualquier número de *Participantes*. De esta manera permitimos la descripción de grupos de diversos tamaños y estructuras. De forma similar a los casos anteriores, también se considera la posibilidad de que los participantes cambien de rol en distintos ESs. En ciertas prácticas educativas es habitual que los alumnos tengan que desempeñar roles diferentes (e.g. enseñanza basada en proyectos). Para permitir el cambio de rol a usuarios de acuerdo a unas normas concretas se establecen conectores de roles (*RoleConnectors*). Estos conectores permiten transferir un recurso que realiza un determinado rol en un ES a otro rol (que puede ser un rol compuesto) en otro ES. De forma similar a los objetivos también se incluye un indicador de multiplicidad de roles.

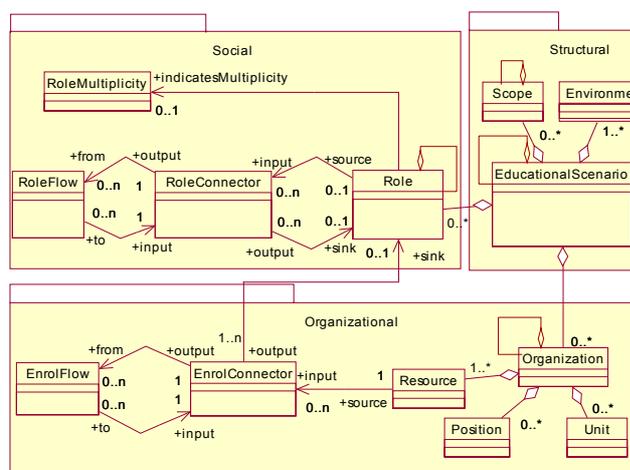


Figura 3. Entidades utilizadas para realizar el modelado de participantes y roles

- **Organización.** La organización (*Organization*) que tiene que desarrollar un ES puede ser determinante para el desarrollo del mismo. En la figura 3 se representan las entidades que permiten modelar la asignación de participantes a roles en base a los recursos de los que disponga dicha organización. Esta asignación se permite a través de la utilización de conectores (*EnrolConnectors*). La organización, u estructura organizativa, también puede influir en las autorizaciones que se conceden a los roles, la

percepción que reciben, y otros asuntos que veremos a continuación.

- **Descripciones de Autorización.** Los descriptores de autorización (*AuthorizationDescriptors*) permiten asignar en un ES a los roles con los permisos que se les quiere proporcionar para que puedan invocar operaciones en las aplicaciones. Normalmente, en escenarios educativos colaborativos no todos los usuarios tienen los mismos privilegios y sólo algunos de ellos pueden realizar operaciones que se consideran críticas (e.g. los profesores o tutores).
- **Descripciones de Percepción.** Los descriptores de percepción (*AwarenessDescriptor*) de un ES permiten modelar la información que debe recogerse de la interacción de los participantes en el ES, cómo debe procesarse y a quién debe enviarse. En muchas prácticas educativas es necesario informar a ciertos usuarios de las acciones que realizan otros usuarios (e.g. un profesor quiere conocer el momento en que sus alumnos están conectados al sistema). Pero, por otra parte, tampoco se debe saturar a los usuarios con demasiada información, que les impida realizar con un esfuerzo razonable sus tareas. Estas descripciones de percepción permiten modelar la información que debe recogerse, como debe procesarse esta información y a quién debe enviarse.
- **Políticas de Interacción.** Un ES puede tener una o varias políticas de interacción (*InteractionPolicy*) que se utilizan para modelar la ejecución de herramientas en los entornos correspondientes del ES. Esta funcionalidad es muy importante en escenarios colaborativos para organizar y coordinar la forma en que los artefactos y las herramientas son gestionadas y controladas. En nuestra propuesta hemos considerado las siguientes formas de gestión y control: (i) control de sesión, para gestionar el inicio y la finalización de herramientas; (ii) control de pertenencia, para invocar operaciones sobre las herramientas en función de condiciones sobre los participantes; (iii) control de turno, para gestionar el acceso dinámico y compartido de los usuarios a ciertas operaciones de las herramientas; (iv) control de conversación, para gestionar el tipo de mensajes que se pueden intercambiar entre varios participantes; (v)

control de versión, para gestionar la generación de versiones de artefactos; y (vi) control de sellos temporales, para gestionar la generación de sellos temporales.

- *Conectores*: Consideramos diferentes tipos de conectores, por ejemplo: *Conector de Rol*, *Conector de Control* y *Conector de Artefactos*. Cada uno de ellos proporciona varios mecanismos para relacionar los correspondientes elementos en los ESs: (i) el *Conector de Flujo* se utiliza para relacionar el secuenciamiento entre los ESs (e.g. secuencia, Union-O, divide-Y, condiciones del parámetro, condiciones temporales, etc.); (ii) el *Conector de Rol* permite la transferencia de participantes a través de roles, apoyando que cierto participante pueda realizar diversos roles en distintos ESs (e.g. libre, *Responsable* decide); (iii) el *Conector de Datos* apoya el flujo de datos entre entornos y ESs (e.g. copia, transferencia, sincronización, etc.). Los *Conectores* utilizan *Reglas* para proporcionar comportamientos condicionados. En nuestra propuesta consideramos que el comportamiento de los conectores puede depender de condiciones fijas o de decisiones que tiene que adoptar una cierta persona o conjunto de personas. De esta forma pretendemos aumentar la flexibilidad del sistema pero manteniendo la misma lógica de relación entre los elementos del meta-modelo.
- *Flujos*. Son simples enlaces usados para conectar los distintos conectores. Los *Flujos* se asocian con los *Conectores*, nunca con otros elementos (*Roles*, *ESs*, o *Entornos*).

Una vez presentadas las entidades más importantes que constituyen nuestra propuesta de meta-modelo, en los siguientes apartados vamos a analizar el mismo respecto a los criterios descritos anteriormente.

3.1. Generalidad

Desde nuestro punto de vista la estructura base del ES proporciona un buen punto de partida para permitir el diseño instruccional de una gran variedad de prácticas educativas. El ES es el elemento principal del meta-modelo, representando una unidad educativa en cualquier nivel de agregación. Hemos tomado esta idea del dominio del *workflow*, donde se ha

establecido que un sistema que no incluya clases separadas para las tareas atómicas y compuestas puede acomodarse más fácilmente a la evolución del tiempo de diseño y del tiempo de ejecución, haciendo el lenguaje más simple y más eficiente [Jorgensen 2004]. La aproximación utilizada es una mezcla del modelo basado en bloques y del basado en flujo de *workflow*. Los ESs se agrupan por medio de una jerarquía de ESs. Los ESs compuestos son o colecciones que tienen una lista desordenada de sub-items, o redes donde se interconectan algunos sub-items. Las colecciones y las redes se refieren así a modelos con diverso grado de estructura. La red de interconexión se compone de dos tipos de elementos: Conectores de Control y Flujos. Cada ES puede tener dos Conectores de Control para describir las condiciones que determinan cuando se puede iniciar y finalizar el mismo. Además de estos, entre los ESs pueden incluirse otros conectores para componer construcciones más complejas.

Por lo tanto, los ESs juegan un papel central, siendo tanto el concepto para la estructuración del diseño de unidades educativas como la piedra angular en la que se pueden apoyar diferentes aproximaciones educativas. Por una parte, la estructura del ES permite describir desde simples unidades de lección hasta cursos completos o planes de estudios. Además, permite representar la agregación de unidades educativas de bajo nivel en componentes más amplios, incluyendo las relaciones entre los elementos involucrados. Por otra parte, la estructura simple conteniendo los objetivos, participantes, y entorno, proporciona un marco básico donde la mayoría de escenarios y de prácticas de aprendizaje se pueden organizar. Hasta la fecha, hemos centrado nuestra atención principalmente en entornos de aprendizaje en colaboración (véase la sección siguiente), pero consideramos que la estructura propuesta va bien a otros enfoques.

3.2. Completitud

Para ofrecer una visión y soporte completos, hemos clasificado la casuística involucrada en escenarios educativos soportados por las TIC, haciendo especial énfasis en situaciones basadas en aprendizaje colaborativo. Las actividades colaborativas comprenden diferentes tipos de tareas (e.g. atómicas,

compuestas, colecciones), interacciones (e.g. comunicación, cooperación), etc. Nuestro análisis se basa en la identificación de perspectivas y para cada una de ellas, en el estudio de los posibles patrones que puedan aparecer en la misma. Como resultado de este análisis, obtenemos un conjunto de situaciones o casos de uso que conducen el desarrollo de nuestro modelo.

Entendemos por perspectiva (también podríamos denominarla aspecto o dimensión) como una característica que incluye un propósito determinado y que puede ser analizada de forma separada de otras perspectivas. Algunas de estas perspectivas relacionadas con el trabajo colaborativo han sido ya identificadas en otros dominios relacionados: *workflow* [Aalst et al. 2003b], *groupware* [Lonchamp 1998], interacción persona-computador [Pinelle et al. 2003], etc. Por su parte un patrón puede ser entendido como una solución general a un problema común [Alexander, 1977]. Los patrones pueden identificarse como soluciones comunes a un determinado problema dentro de una perspectiva.

Hemos llevado a cabo un análisis de los dominios mencionados de trabajo (colaborativo) y de Aprendizaje Colaborativo basado en Computador, obteniendo como resultado la identificación de las siguientes perspectivas:

- La *perspectiva funcional* caracteriza las tareas a ser realizadas y su descomposición en unidades más pequeñas. Se relaciona con objetivos y tareas atómicas y compuestas.
- La *perspectiva procesal* (o *flujo de control*) está relacionada con el orden relativo en el que deben llevarse a cabo las tareas y conseguir los objetivos.
- La *perspectiva temporal* que determina la prerreación temporal entre los objetivos que deben ser llevados a cabo, en secuencia o en paralelo. Se relaciona con el acoplamiento de tareas identificado en [Pinelle et al. 2003].
- La *perspectiva de información* (o *flujo de datos*) describe los artefactos utilizados (e.g. objetos educativos) y el flujo de artefactos entre tareas. Algunos tipos de flujo de artefactos entre tareas son la transferencia síncrona de datos entre una tarea que finaliza y otra que se inicia, o la transferencia asíncrona de datos concerniente a la copia o transferencia.

- La *perspectiva operacional* se ocupa de las aplicaciones y los servicios que se pueden utilizar en las tareas (e.g. simuladores, editores, servicios de comunicación y colaboración). Estos sistemas pueden ser muy diferentes en cuanto a su naturaleza, pero desde un punto de vista de modelado solo nos interesa su funcionalidad y los detalles técnicos de funcionamiento debieran ser transparentes. Por tanto, en esta perspectiva sólo estamos interesados en modelar la funcionalidad de las herramientas para después permitir su gestión e integración mediante otras perspectivas (e.g. autorización, percepción, interacción).
- La *perspectiva organizacional* describe la estructura de los participantes, *roles* y grupos y su asignación a tareas concretas. Dicha organización puede ser tenida en cuenta para tomar decisiones en las otras perspectivas. Por ejemplo, para decidir la asignación de un recurso a una tarea de acuerdo con la posición organizativa del mismo.
- La *perspectiva de autorización* se ocupa de los permisos que los participantes poseen para acceder a recursos o utilizar determinadas operaciones en las herramientas.
- La *perspectiva de percepción* se ocupa de indicar qué información producida durante la ejecución de un diseño instructivo debe recogerse, cómo debe procesarse y a quién debe enviarse. En definitiva, esta perspectiva trata de cómo se hace visible o disponible a ciertos participantes lo que otros participantes han hecho o están haciendo. En el caso de prácticas educativas la posibilidad de realizar este tipo de descripciones es muy interesante.
- La *perspectiva de interacción* se ocupa del control en la invocación de operaciones en las herramientas utilizadas en las prácticas educativas. Durante el desarrollo de un curso puede ser interesante automatizar ciertas funciones que supongan la invocación de operaciones determinadas en las herramientas. Un simple ejemplo es el establecimiento de condiciones para fijar el inicio y la finalización de un servicio de comunicación.

En el meta-modelo propuesto se ha intentado capturar los requisitos involucrados en cada una de estas perspectivas. En particular, se ha realizado una

descomposición del mismo en paquetes que recoge cada una de ellas.

3.3. Flexibilidad

El meta-modelo propuesto está basado en objetos y orientado a estados para mejorar su flexibilidad. Todos los elementos del meta-modelo derivan del elemento abstracto *Entidad* (c.f. Figura 4).

Una *Entidad* contiene atributos, operaciones, permisos, eventos y está vinculada a otras entidades. A continuación se realiza una descripción de estos elementos:

- **Atributos.** Existen al menos dos atributos incluidos en todas las entidades del meta-modelo: estado y meta-datos. (a) El atributo estado que se usa para mantener el estado actual del elemento. Todos los elementos serán asociados con un modelo de diagramas de estado para soportar la ejecución. Las operaciones disponibles estarán restringidas por el estado. (b) El atributo metadatos se usa para mantener información que describe el elemento (e.g. versión, fecha de creación, etc.). Para ello utilizaremos el estándar LOM [IEEE 2002].
- **Operaciones.** Se utilizan para cambiar el estado del elemento. Las operaciones integradas por defecto están destinadas al cambio del valor de los atributos (*cambiarEstado*, *cambiarMetadatos*), y para cambiar las reglas (*añadirRegla*, *borrarRegla*). La gestión de cualquier elemento por parte del motor de ejecución debe llevarse a cabo mediante llamadas a sus operaciones. Por lo tanto, es posible permitir que algunos de los participantes lleven a cabo cierto control de la ejecución.
- **Permisos.** Permiten agrupar las credenciales necesarias para poder invocar una o varias operaciones de cada entidad. Por ejemplo, se pueden definir permisos de edición que permitan realizar modificaciones en el modelo respecto a una entidad determinada. Estos permisos pueden ser asignados a ciertos participantes mediante elementos del paquete de autorización.
- **Eventos.** Cada entidad del meta-modelo tiene la capacidad de generar un conjunto determinado de eventos, que se generan como consecuencia de la

realización de determinadas operaciones y del cambio de estado.

Para especificar la autorización a participantes para la realización de determinadas operaciones se utilizan políticas específicas [Kagal 2002]. La estructura de políticas incluye las nociones de derechos, prohibiciones, obligaciones y dispensas. Una *Política* se define mediante la relación de un *Rol* a una *Regla* determinada, la cual especifica las condiciones que gobiernan las interacciones del *Rol*. Como consecuencia, el *role* es autorizado (o prohibido, o obligado o dispensado) para llevar a cabo determinada *Acción* si se cumplen determinadas condiciones. En general, las políticas son los mecanismos para regular de forma dinámica el comportamiento de los componentes del sistema sin cambiar el código y sin la necesidad de requerir el consentimiento de los componentes que se gobiernan. Cambiando las políticas, un sistema puede ajustarse de forma continuada para acomodarse a las variaciones de las restricciones y condiciones externas. Consideramos que las políticas son los mecanismos para asignar autorizaciones para el control y modificación de los diseños durante la ejecución.

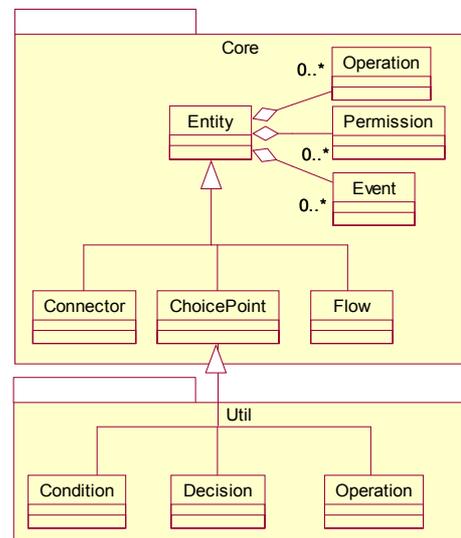


Figura 4. Entidades básicas del meta-modelo

En la figura 4 se representan otras entidades básicas de la especificación, como son los conectores (*Connectors*), los flujos (*Flows*) y los puntos de elección (*Choice Point*). Estos elementos también son clave para proporcionar flexibilidad en el meta-modelo. En el caso de los conectores y los flujos,

como decíamos antes, nos permiten establecer relaciones entre los elementos de los modelos, siendo posible utilizar distintos tipos de relación. Por su parte los puntos de elección nos permiten modelar aquellas partes de una práctica educativa en las que es necesario hacer una elección. Esta abstracción sirve de punto de encuentro entre distintas formas de realizar una elección: (i) mediante una condición (*Condition*) que puede ser evaluada de forma automática; (ii) mediante una decisión (*Decision*) que tiene que ser tomada por una persona o por un grupo de personas (en este caso puede ser necesario considerar mecanismos de votación); o (iii) como resultado de una operación (*Operation*) que también puede evaluarse de forma automática. La utilización de una abstracción común para estas tres entidades nos permite utilizar cualquiera de ellas en los modelos, según lo que convenga en cada caso, mejorando la flexibilidad de los mismos.

4. Un Ejemplo de Uso

En esta sección mostramos un ejemplo de modelado utilizando algunos de los elementos introducidos en el meta-modelo. El ejemplo se basa en la asignatura “Ingeniería del Software I” que pertenece al segundo curso de los estudios de Ingeniería de Telecomunicación de la Universidad de Vigo. En esta asignatura, después de ciertas clases iniciales, los alumnos deben realizar un conjunto de varias prácticas de diseño y programación. En concreto se propone la realización de tres prácticas que comparten la misma estructura. En la Figura 5 presentamos un modelo para articular cada una de estas prácticas como un ES llamado *Práctica de Ingeniería del Software*. Actualmente estamos en el proceso de definición de la notación empleada en la figura. Esta notación está basada en una propuesta de *workflow* [Jorgensen 2004]. En la figura únicamente se ha incluido la representación parcial de *Objetivos*, *Roles*, *Recursos*, *sub-ESs*, y *conexiones de sub-ESs*. Por simplicidad no se han incluido las conexiones de datos, descripciones, políticas, etc.

En la figura 5 se muestra el esquema de ESs propuesto para modelar una *Práctica de Ingeniería del Software*. Se muestra un ES padre que se descompone en cinco sub-ESs. Los actores de cada ES se representan en la parte izquierda del esquema.

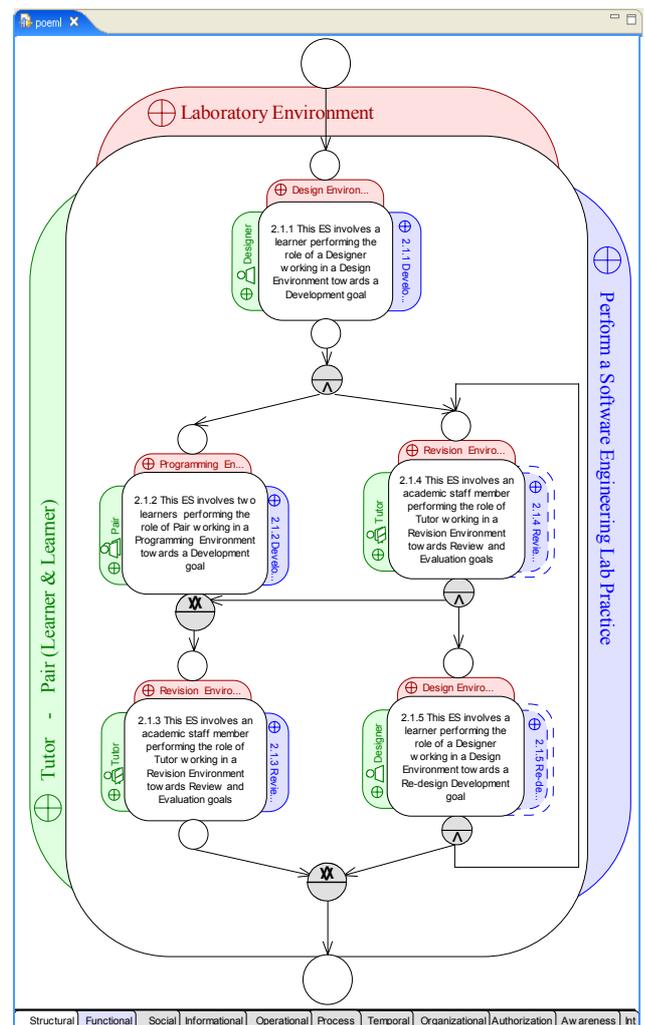


Figura 5. Ejemplo de un escenario educativo para una práctica de ingeniería del software

Como se puede ver, la *Práctica de Ingeniería del Software* atañe a dos tipos principales de actores: *Pareja* y *Tutor*. Una pareja es un grupo compuesto de dos actores: *Diseñador* y *Programador*. Estos actores son transferidos desde los roles del curso (i.e.: *Alumnos* y *Tutores*) utilizando *conectores de Rol* y *Flujos*. Hemos decidido asignar *Alumnos Padre* a *Diseñadores* y *Programadores*, y *Tutores Padre* a *Tutores* directamente. Cada una de las prácticas debe ser realizada por una *Pareja* de alumnos, en la que un alumno ejerce de *Diseñador* y otro alumno de *Programador*. Los alumnos de cada *Pareja* se mantienen en cada una de las tres *Prácticas ESs*, pero cambiando el rol *Alumnos* (cf. Figura 6). A saber, el *Diseñador* en la primera *Práctica* será el *Programador* en la segunda, y el *Programador* de la

primera será el *Diseñador* en la segunda. El rol de cada *Alumno* en la tercera *Práctica* se decide en función de los resultados previos. En relación con los ES hijos, los roles se transfieren a cada sub-ES directamente.

Como hemos dicho, el ES relativo a la *Práctica de Ingeniería del Software* se descompone en cinco sub-ESs conectados por *Conectores de Control y Flujos*.

El conector $\text{\textcircled{\small \wedge}}$ representa la lógica AND, en este caso un *and-split* indicando que se deben activar los dos caminos de salida. El conector $\text{\textcircled{\small \vee}}$ representa la lógica OR, en este caso un *or-join* indicando que se puede iniciar el siguiente ES cuando ambas líneas de entrada hayan finalizado o si tras finalizar una se sabe que la otra no va a finalizar nunca (esta situación se puede producir con ESs optativos). El conector $\text{\textcircled{\small \emptyset}}$ representa una operación vacía. El primer Sub-ES (2.1.1) está dedicado al diseño de la solución de la práctica propuesta. Esta tarea debe ser llevada a cabo por el rol *Diseñador* de la *Pareja*, utilizando un *Entorno de Diseño* para producir un documento de *Diseño*. Cuando el primer sub-ES finaliza, dos sub-ESs se inician: *Programación* (2.1.2) y *Evaluación del Diseño* (2.1.4). La sub-ES *Evaluación del Diseño* es asignada al *Tutor*, el cual es responsable de la revisión del diseño propuesto. Como resultado se produce un documento de *Evaluación del Diseño*. Esta revisión puede ser utilizada por el *Diseñador* para modificar y actualizar el documento inicial de *Diseño* en la siguiente sub-ES: *Revisión de Diseño* (2.1.5). Tanto el ES 2.1.4 como el 2.1.5 son optativos. En paralelo con estos dos sub-ES, la *Pareja* debe programar el diseño propuesto en la sub-ES *Programación*. Esta tarea puede ser asignada al *Programador* únicamente, pero preferimos que ambos participantes colaboren en las labores de programación debido a motivos educativos. Además, con nuestro meta-modelo se podría indicarse que la *Pareja* tuviese la capacidad de decidir la descomposición de esta tarea en varias sub-tareas para organizar y gestionar las actividades de programación (e.g. cada miembro de la *Pareja* puede programar diferentes funciones y posteriormente integrarlas). Esto es posible permitiéndole al rol *Programador* especificar nuevas sub-ESs. Cuando la *Pareja* finaliza la ES *Programación*, la sub-ES *Evaluación de Programación* (2.1.3) se activa. Esta sub-ES se asigna al *Tutor* para revisar y evaluar los

resultados *Diseño* y *Programa* producidos por la *Pareja* y producir de este modo la *Evaluación del Programa*.

Los entornos se muestran esquemáticamente en la parte superior de la representación. Utilizamos diferentes notaciones para identificar las herramientas (e.g. *Entorno de Diseño*, *Entorno de Programación*), documentos (e.g. *Especificación de Prácticas*, *Diseño*, *Programa*, *Evaluación de Diseño* y *Evaluación de Programación*), y sub entornos (*Ficheros de Pruebas*). Los documentos deben ser transferidos entre diferentes sub-ESs. No hemos representado esta transferencia, pero resulta obvia considerando los recursos definidos en cada sub-ES. El documento *Especificación de la Práctica* debería incluirse en todas las sub-ESs, pero no se hace así por limitaciones de espacio. El único punto interesante se refiere a las sub-ES *Programación* y *Revisión de Diseño*. Ambos ESs utilizan el documento *Diseño*, el cual puede ser modificado y actualizado en la sub-ES *Revisión de Diseño*. Por lo tanto, ambos documentos pueden actualizarse cuando el *Diseñador* ha realizado algún cambio en el ES *Revisión de Diseño*. Esta conexión debería representarse en la figura con un *Conector de Artefactos* apropiado.

4.1. Transferencia de Roles

Como hemos introducido, los actores son transferidos de un ES a otro mediante el uso de los *conectores de rol y flujos*. La Figura 6 representa la transferencia de los roles *Alumno* desde el ES padre representando la asignatura *Ingeniería del Software I* a cada uno de los tres sub-ESs hijos *Práctica de Ingeniería del Software* (2.1, 2.2, 2.3). Como se ha comentado en la sección previa, los roles del ES padre *Alumno* deben ser asignados a *Diseñadores* y *Programadores* de acuerdo con la estructura de la *Pareja*. Los participantes *Pareja* se mantienen en cada una de las tres ES *Práctica*, pero cambiando el rol *Alumno*. A saber, el *Diseñador* en la primera *Práctica* será el *Programador* en la segunda, y el *Programador* en la primera será el *Diseñador* en la segunda. El rol de cada *Alumno* en la tercera *Práctica* se decide de acuerdo con los resultados previos. En relación con los ES hijos, los roles se transfieren a cada sub-ES de forma directa, de acuerdo con los roles que se hubiesen definido. En la Figura 4 se representa este

comportamiento utilizando diferentes tipos de círculos (*conectores de rol*) y flechas (*flujos*).

El *conector*  representa la lógica XOR, donde cada elemento del *flujo de entrada* se entrega a uno sólo de los *flujos de salida*. En este caso, cada *flujo de salida* de dicho conector se debe activar de forma alternativa. El *conector*  representa la lógica AND, donde cada elemento del *flujo de entrada* se entrega a todos los *flujos de salida* simultáneamente.

El *conector*  representa una operación vacía. Directamente transfiere los elementos desde el *flujo de entrada* al *flujo de salida*. Finalmente, el *conector*  representa un punto en el que es necesario realizar una elección (en este caso una condición) sobre las transferencias entre los flujos de entrada y de salida. De acuerdo a las características de estos elementos el modelo propuesto representa el comportamiento requerido. Las transferencias entre tutores se realizan de forma directa. En cualquier caso es interesante destacar que el tutor tiene que ser el mismo en los tres ESs.

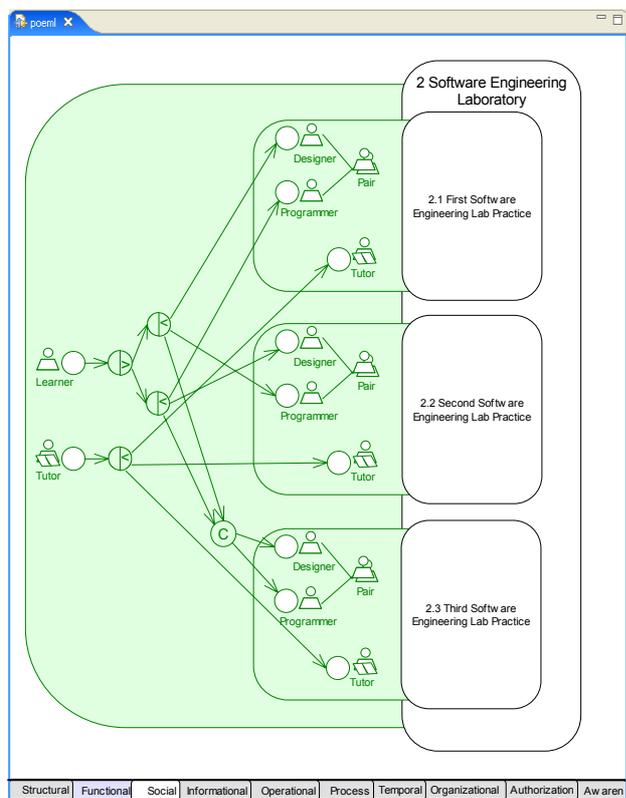


Figura 6. Transferencia de roles entre ESs

4.2. Transferencia de Objetivos

En este apartado vamos a representar el modelado de los objetivos, esto es, de la perspectiva funcional (c.f. figura 7).

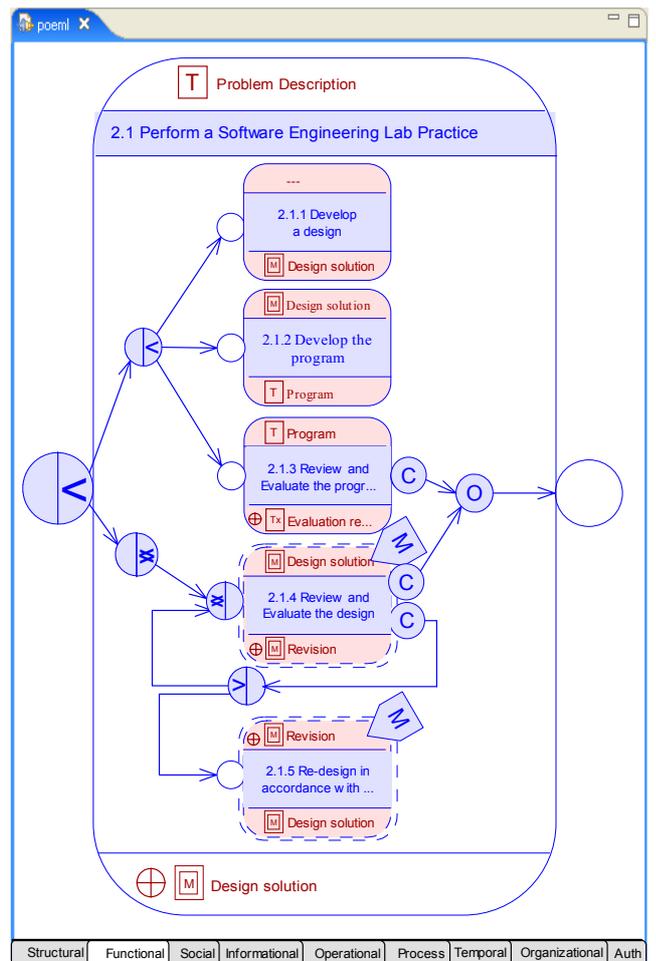


Figura 7. Transferencia de objetivos

Se identifica un objetivo padre (2.1) que tiene como parámetro de entrada (representado en la parte superior) un fichero de texto en el que se contiene la descripción de la práctica a realizar y como parámetro de salida (representado en la parte inferior) un documento multimedia con todos los productos generados durante la práctica (diseño, código, revisiones, etc.). Este objetivo global se descompone en cinco sub-objetivos:

- Sub-objetivo 2.1.1. Este sub-objetivo propone la realización del diseño de una solución a la práctica propuesta. Incluye como parámetro de

salida una solución de diseño. No contiene ningún parámetro de entrada porque la información requerida para afrontar este objetivo ya está disponible en el objetivo padre (2.1)

- Sub-objetivo 2.1.2. Este sub-objetivo plantea el desarrollo de un programa. Su parámetro de entrada es un documento de diseño y como parámetro de salida tiene que proporcionar un programa de acuerdo a dicho documento y a las especificaciones del problema.
- Sub-objetivo 2.1.3. Este sub-objetivo indica que es necesario realizar una revisión y una evaluación del programa. El parámetro de entrada es un programa y como parámetro de salida se debe proporcionar un documento de revisión del mismo y una evaluación.
- Sub-objetivo 2.1.4. Este sub-objetivo propone la realización de una revisión y una evaluación de un diseño que se plantea como solución a un problema. El parámetro de entrada de este objetivo debe ser el diseño que se va a evaluar y como parámetros de salida deben proporcionarse un documento de revisión y una evaluación.
- Sub-objetivo 2.1.5. Este sub-objetivo plantea la realización de un re-diseño de un problema en base a un diseño original y a una revisión. Estos artefactos constituirán los parámetros de entrada, mientras que como salida debe proporcionarse un nuevo diseño.

Las relaciones entre los parámetros de entrada y de salida no se indican debido a las limitaciones en el espacio. En cualquier caso se representarían con entidades pertenecientes a la perspectiva informativa.

Los conectores de objetivos de entrada se utilizan para indicar si los objetivos son obligatorios, opcionales o están prohibidos. Un conector *and-split*

 se utiliza para indicar objetivos obligatorios (en el ejemplo los objetivos 2.1.1, 2.1.2 y 2.1.3). El objetivo 2.1.4 es opcional porque está enlazado con su padre

por medio de un conector *or-split* . Por su parte, el objetivo 2.1.5 solo puede realizarse si el conector 2.1.4 alcanza un cierto estado de completitud descrito

en su condición de salida . En otro caso este objetivo está prohibido puesto que no tiene ninguna conexión con su objetivo padre. En caso de que se

cumpla la condición anterior el objetivo 2.1.5 será obligatorio y el objetivo 2.1.4 tendrá que realizarse otra vez. Ahora bien, esto será posible siempre que se cumplan las condiciones de multiplicidad

representadas en el indicador .

Los conectores de objetivos de salida se utilizan para indicar cómo se debe calcular el estado de completitud del objetivo padre. En el ejemplo, sólo los objetivos 2.1.3 y 2.1.4 se van a utilizar para este propósito.

5. Conclusiones

Las teorías de diseño instruccional proporcionan métodos y estrategias para determinar cómo debería llevarse a cabo la instrucción. En su conjunto proporcionan herramientas para guiar el desarrollo de planes educativos en los que se indica cómo deben desarrollarse las prácticas educativas. En el dominio de los sistemas de aprendizaje electrónico, los EMLs han sido concebidos para permitir modelar de forma computacional dichos planes, independientemente de la aproximación pedagógica. Por otra parte, se están desarrollando herramientas que permiten la ejecución de estos lenguajes proporcionando soporte a distintas pedagogías. En este sentido, desde el punto de vista de la estandarización, consideramos que los EMLs están llamados a desempeñar un papel fundamental en el futuro del e-learning dado que facilitan tanto la reutilización de los contenidos como de las aplicaciones.

En nuestro trabajo pretendemos contribuir al desarrollo de los EMLs, aunque no desde la perspectiva de estandarización. Nuestro interés principal se centra en el desarrollo de un lenguaje que facilite la práctica educativa. De este modo, se ha discutido y presentado una propuesta que proporciona una solución general, completa y flexible. En cuanto a la completitud, el meta-modelo propuesto permite la descripción de currículos completos, cursos o pequeñas lecciones. Por otro lado, se permite el control computacional de las diferentes perspectivas que hemos identificado en la caracterización de la coordinación de prácticas educativas. Además, dicha coordinación se ha separado en partes claramente delimitadas, proporcionando una clara división de deberes. Esta separación junto con los conectores y puntos de elección, permiten proporcionar una

solución flexible que permite considerar mecanismos de control automáticos y manuales.

No podemos terminar sin comentar la complejidad del meta-modelo. En este sentido, se debe tener en cuenta que este lenguaje no se propone con la intención de que sea utilizado por diseñadores instructivos o profesores directamente. Su propósito es permitir el desarrollo de aplicaciones software para dichos usuarios facilitando los criterios de generalidad, completitud y flexibilidad.

Agradecimientos

Queremos agradecer al *Ministerio de Educación y Ciencia* por su contribución a este trabajo a través del proyecto “*MetaLearn: metodologías, arquitecturas y lenguajes para sistemas E-learning adaptativos*” TIN2004-08367-C02-01.

Referencias

- [Aalst et al. 2003a] Aalst, W. M. P. van der, Hofstede, A. H. M. *YAWL: Yet Another Workflow Language*. Technical Report FIT-TR-2003-04. (2003).
- [Aalst et al. 2003b] Aalst, W. M.P. van der, Weske, M., Wirtz, G. *Advanced Topics in Workflow Management: Issues, Requirements, and Solutions*. *International Journal of Integrated Design*, 7(3). (2003).
- [Alexander 1977] Alexander, C. *Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York. (1977).
- [Dodds 2004] Dodds, P. (Ed.) *SCORM Content Aggregation Model*. Versión 2004, IMS (2004).
- [Downes 2003] Downes, S. *Design Standards and Reusability*. Available at <http://www.downes.ca/cgi-bin/website/view.cgi?dbs=Article&key=1059622263> (2003). Last access on May 2nd, 2005
- [IEEE 2002] IEEE. *IEEE Standard for Learning Object Metadata*. Available at <http://ltsc.ieee.org/wg12/> Last access on May 2nd, 2005
- [Jorgensen 2004] Jorgensen, H. D. *Interactive Process Models* Ph. D. Thesis, Norwegian University of Science and Technology. (2004).
- [Kagal 2002] Kagal, L. *Rei: A Policy Language for the Me-Centric Project*. HP Labs Technical Report, HPL-2002-270. (2002).
- [Kirschner 2002] Kirschner, P. A. (Ed.) *Three Worlds of CSCL. Can We Support CSCL*. Open University of the Netherlands, The Netherlands. (2002).
- [Koper 2001] Koper, R. *Modeling units of study from a pedagogical perspective – The pedagogical metamodel behind EML*, Open University of the Netherlands. (2001).
- [Koper et al. 2003] Koper, R., Olivier, B. & Anderson, T. (Eds.) *IMS Learning Design Information Model*, IMS Global Learning Consortium. (2003).
- [Lonchamp 1988] Lonchamp, J. *Process Model Patterns for Collaborative Work*. Proceedings of the 15th *IFIP World Computer Congress, Telecooperation Conference*, Austria. (1998).
- [Merrill 2001] Merrill, D. *Components of instruction toward a theoretical tool for instructional design*. *Instructional Science*, 29(4), 291-310. (2001).
- [Nielsen 2005] Nielsen, B. (Ed.) *IMS Content Packaging Information Model*, Versión 1.2, IMS Global Consortium (2005).
- [Paquette 2003] Paquette, G. *Educational Modeling Languages, From an Instructional Engineering Perspective*. In McGreal, R. (Dir.). *Accesible Education Using Learning Objects*. (2003).
- [Pinelle et al. 2003] Pinelle, D., Gutwin, C., Greenberg, S. *Tasks with the Mechanics of Collaboration*. *ACM Transactions on Computer-Human Interaction*, 10(4), 281-311. (2003).
- [Reigeluth 1999] Reigeluth, C. *Instructional design theories and models: A new paradigm of instructional theory*. Lawrence Erlbaum Associates. (1999).
- [Russell et al. 2004] Rusell, N., ter Hofstede, A. H. M., Edmond, D., & Aalst, W. M. P. van der. *Workflow Data Patterns*, Technical Report, FIT-TR-2004-01, Queensland University of Technology. (2004).

[Smith y Ragan 2005] Smith, P. L., Ragan, T. J..
Instructional Design. Third Edition, Wiley Jossey-
Bass Education. (2005).

[Strijbos et al. 2003] Strijbos, J. W., Kirschner, P.,
Martens, R. L. (Eds.). *What we Know about CSCL
in Higher Education*, Kluwer, Dordrecht. (2003)