

Metamodelo de la especificación OSID versión 2 de OKI

Angels Rius¹, Jordi Conesa¹, Elena García-Barriocanal²

(mriusg@uoc.edu, jconesac@uoc.edu, elena.garciab@uah.es)

¹*Estudis d'Informàtica, Multimèdia I Telecomunicació, Universitat Oberta de Catalunya*

²*Departamento de Ciencias de la Computación, Universidad de Alcalá*

Índice

Resumen.....	3
1. Introducción.....	4
2. Caso de estudio	6
3. Formalización de la especificación OSID de OKI.	8
3.1. OSID Agent	9
3.2. OSID Assesment.....	11
3.3. OSID CourseManagement.....	17
3.4. OSID Grading.....	22
3.5. OSID Scheduling.....	24
3.6. OSID Repository	26
3.7. OSID User Messaging	30
3.8. OSID Workflow.....	32
4. Limitaciones	35
5. Conclusiones.....	37
Bibliographic references.....	38

Resumen

En las últimas décadas dentro del ámbito del e-Learning, la investigación acerca de especificaciones y estándares relacionados con contenidos digitales ha evolucionado notablemente. Aunque se han multiplicado el número de plataformas de sistemas de aprendizaje, las especificaciones y estándares relacionados con procesos para entornos educativos no han tenido la misma suerte.

La especificación Open Service Interface Definition (OSI) de Open Knowledge Initiative (OKI) propone un conjunto de interfaces de servicios web dirigidos a entornos de enseñanza superior con el objetivo de conseguir interoperabilidad entre aplicaciones. Sin embargo esta especificación no ha sido demasiado utilizada, posiblemente por falta de un mapa conceptual que facilite su comprensión y promueva su uso.

En este trabajo se describe un metamodelo de la especificación OKI-OSID mediante UML y un conjunto de reglas. También se presentan una lista de las limitaciones detectadas tras el estudio de dicha especificación.

Palabras clave

Especificación OSID, interfaces de servicios web, especificaciones de procesos de e-learning

1. Introducción

En las últimas décadas la investigación en el ámbito del l'e-learning ha evolucionado notablemente en lo que respecta a especificaciones y estándares relacionados con contenidos digitales, tal como demuestra la existencia de numerosas especificaciones y estándares desarrollados. Algunos ejemplos son: Common Cartridge¹, que es un estándar bajo el cual IMS Global Learning Consortium ha agrupado numerosas especificaciones y estándares para el intercambio de contenidos y actividades (Content Packaging v1.2 (IMS Global Learning Consortium, 2007), Question & Test Interoperability v1 (IMS Global Learning Consortium, 2008) y/o IMS Tools Interoperability Guidelines v1.0 (IMS Global Learning Consortium, 2006)), IEEE Learning Object Metadata v1.0 (IEEE Learning Technology Standards Committee [LTSC WG12], 2002) para la especificación de metadatos relacionados con objetos de aprendizaje o ADL SCORM v1.2 (Advanced Distributed Learning, 2004) para facilitar la interoperabilidad de herramientas de aprendizaje y contenidos educativos a escala global entre otros.

También se han experimentado notables avances en torno a plataformas de sistemas de aprendizaje tal como demuestra la proliferación de este tipo de sistemas, ya sean sistemas de aprendizaje (p.e. Claroline², LAMS³ o SharePointLMS⁴) o sistemas de gestión de cursos (p.e. Dokeos⁵, ILIAS⁶, Moodle⁷ o SAKAI⁸). Sin embargo, no existen mecanismos para la representación de actividades que habitualmente se llevan a cabo en entornos educativos ya sea para preparar el entorno de aprendizaje, ofrecer soporte al proceso de aprendizaje o para evaluar los resultados obtenidos y las competencias adquiridas durante el proceso de aprendizaje. Al conjunto de actividades que tiene este fin le denominaremos en adelante, escena educativa.

Para clarificar más el concepto de escena educativa nada mejor que enumerar algunos ejemplos: la preparación de un curso, la preparación de actividades de aprendizaje, su corrección o el retorno de la evaluación. Disponer de una

¹ <http://www.imsglobal.org/pressreleases/pr080122.html>

² <http://www.claroline.net/>

³ <http://www.lamsinternational.com/>

⁴ <http://www.sharepointlms.com/>

⁵ <http://www.dokeos.com/es>

⁶ <http://www.ilias.de/docu/>

⁷ <http://moodle.org/>

⁸ <http://sakaiproject.org/>

especificación clara, concisa y consistente de escenas educativas facilitaría en gran medida su interpretación automática y la automatización de procesos en entornos educativos.

Sabemos que en el área del BPM (Business Process Management) la automatización de procesos de negocio ha reportado importantes beneficios y existe un lenguaje estándar para la descripción de procesos de negocio, el lenguaje BPMN (Business Process Modelling Notation) (Object Management Group & BPML.org, 2006). Sin embargo, en el ámbito del e-learning no existe un lenguaje similar a BPMN, ni tampoco otros mecanismos de descripción más formal de escenas educativas, ni de los procesos que las componen. Aunque no existen ontologías relacionadas con escenas educativas, existen ontologías para describir entornos educativos. Por ejemplo, para describir contenidos de aprendizaje (Kabel, Wielinga, & de Hoog, 1999), interacciones entre estudiantes y sistemas de aprendizaje en entornos colaborativos (Inaba et al., 2001), tareas de aprendizaje (Mizoguchi, Sinita, & Ikeda, 1996), objetos de aprendizaje y grupos de trabajo (Inaba et al., 2001) así como otros elementos que intervienen en escenarios de entornos colaborativos.

En un nivel más cercano a la implementación, también existen marcos para la construcción de entornos de aprendizaje con una clara orientación a servicios, como ELF (como e-Learning Framework) y la especificación OSID de OKI (Open Knowledge Initiative, 2004).

ELF⁹ es un marco desarrollado por distintas organizaciones internacionales para ofrecer soporte al desarrollo e integración de sistemas en el ámbito del e-Learning, la investigación y administración de la educación. Se compone de tres niveles: 1) servicios comunes, 2) servicios relacionados con el dominio de e-Learning y 3) servicios para usuarios y agentes.

La especificación OSID de OKI¹⁰, a la que nos referiremos en adelante como especificación OKI-OSID, es otro marco dirigido específicamente a las comunidades de enseñanza superior cuyo objetivo es ofrecer un conjunto de interfaces de servicios web para la construcción de sistemas de aprendizaje promoviendo la interoperabilidad entre aplicaciones. A partir de la versión 2.0 este marco incorpora *packages* que facilitan el desarrollo de sistemas de e-learning en lenguajes como Java y C#, pero todavía no existe un metamodelo que describa esta especificación a nivel conceptual y por tanto facilite su uso.

Con el objetivo de proponer algún mecanismo que permita la representación de procesos en entornos educativos y facilitar su posterior automatización, en este artículo se presenta, a modo de prueba de concepto, un metamodelo de la especificación OKI-OSID. Ello permite demostrar su utilidad en la creación de

⁹ <http://www.elframework.org/framework>

¹⁰ <http://www.cen-Itso.net/main.aspx?put=248>

aplicaciones o la incorporación de nuevas funcionalidades en sistemas de aprendizaje existentes y otros sistemas relacionados.

La estructura del presente artículo consta de cinco secciones. En la primera se contextualiza el problema y se determinan los objetivos de este trabajo. En la segunda se describe el caso de estudio a utilizar para probar la factibilidad de la técnica propuesta. En la tercera sección, que constituye el núcleo del artículo, se presenta la formalización de la especificación OKI-OSID. En la cuarta se discuten las limitaciones de la especificación y finalmente, en la última sección, se exponen las conclusiones y el trabajo futuro.

2. Caso de estudio

La asignatura Bases de Datos I (BDI) es una asignatura obligatoria de las titulaciones informáticas de la Universitat Oberta de Catalunya. Dicha asignatura por su carácter práctico utiliza dos tipos de aulas: aulas de teoría y aulas de laboratorio. Así pues un estudiante matriculado en ella, es asignado a un aula de teoría y a un aula de laboratorio. Las aulas de laboratorio admiten un ratio de estudiantes mayor que las de teoría, por lo que es posible que existan varias aulas de teoría asociadas un mismo curso.

Toda asignatura en la UOC es responsabilidad de un profesor, profesor responsable de asignatura (PRA), quién lidera y coordina al equipo docente de la asignatura. Dicho equipo está formado por el PRA y un conjunto de consultores. El consultor es un experto en la materia que imparte y ejerce la función docente tal como se entiende en entornos de enseñanza presencial.

El PRA, entre otras múltiples tareas, se encarga de preparar el curso y asignar un consultor a cada aula del curso. La asignación de un consultor a un aula supone que este acepta la responsabilidad de guiar a los estudiantes de la aula durante el proceso de aprendizaje, corregir sus actividades y resolver sus dudas. A esta tarea en la UOC se le denomina encargo básico de consultoría (EBC).

La escena Asignar Encargos EBC para el curso BDI tiene lugar justo antes del inicio de curso, cuando existe la necesidad de asignar consultores a aulas. Así pues, para cada curso, que viene identificado por la asignatura y semestre, el PRA determina el equipo de personal docente colaborador de teoría y laboratorio y procede, mediante una aplicación, a la asignación de consultores a aulas de la asignatura.

Una vez conocido el número de aulas necesarias, el PRA realiza una propuesta de asignación. El sistema se encarga de notificar individualmente a cada consultor su asignación, la cual debe ser aceptada. Así queda establecido el encargo EBC del

curso, que se registra en el repositorio UOC a efectos de remuneración e historial y finaliza la escena.

A partir de la descripción de esta escena, se observa que intervienen en ella varias figuras docentes: el PRA y el conjunto de consultores y se utilizan varios recursos. Entre los recursos que se requieren para llevar a cabo el proceso están: los datos del curso (*id_asignatura*, *id_semestre*), el número de aulas de cada tipo (*numAulasBDIPorTipoEstablecidas*), la lista de consultores de teoría y laboratorio de la asignatura (*ListaConsultoresPorTipoBDI*), la lista de asignaciones de consultores a aulas (*ListaAsignacionesPorTipoBDI*), las notificaciones de aceptación (*NotificacionAceptacionConsultoriaBDI*), la lista de encargos EBC de BDI (*ListaEncargosEBC_BDI*) y el repositorio UOC de donde se recupera información sobre posibles consultores y se almacenan los encargos EBC asignados. La escena puede representarse como un único proceso denominado *AsignarEncargosEBC_BDI_T10*. Gráficamente se muestra en la figura 1.

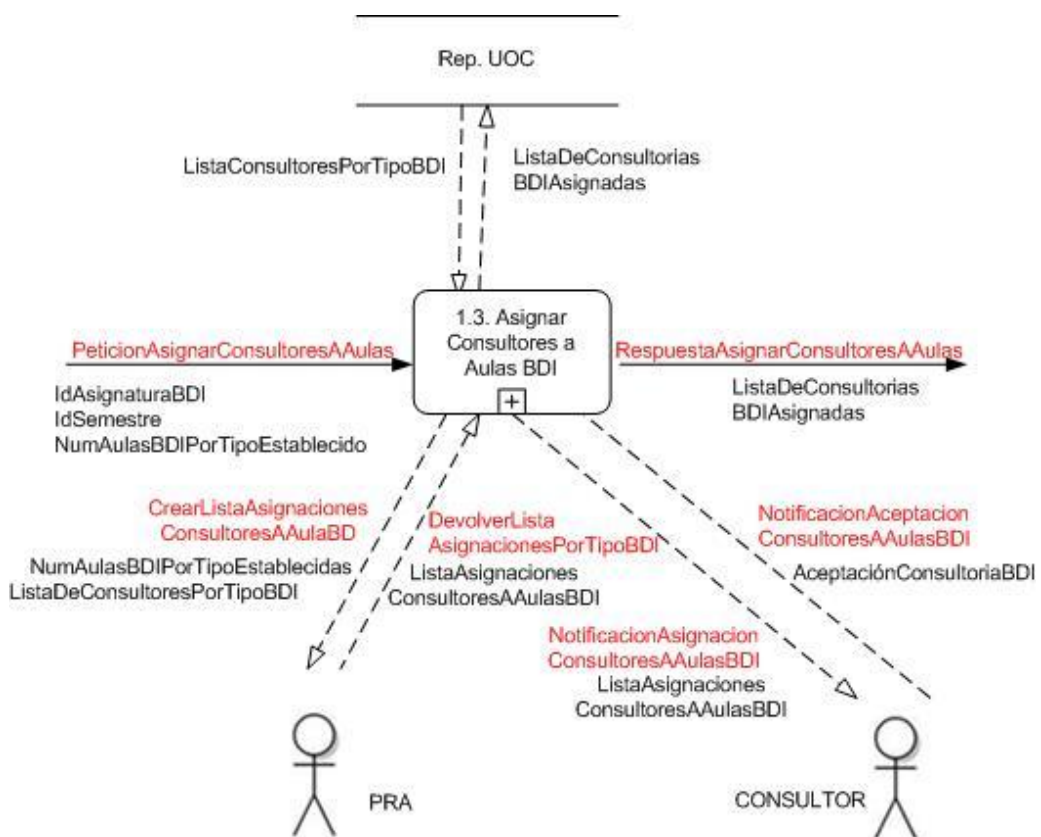


Figura1. Escena Asignar Encargos EBC al curso BDI del semestre Otoño 2010

La notación empleada en la figura 1 utiliza un lenguaje de representación basado en BPMN. Esta notación permite representar procesos en entornos educativos mediante secuencias de procesos, cuya adecuada combinación permite alcanzar el objetivo final de la escena educativa. La secuencia de procesos que describe la escena requiere de la intervención ordenada de participantes o agentes externos y del intercambio de recursos. Para facilitar la descripción de estos diagramas se ha desarrollado una herramienta CASE que ofrece soporte a la especificación de escenas en esta notación. Más información de la herramienta puede encontrarse en (Rius, Conesa, & Gañan, 2010).

3. Formalización de la especificación OSID de OKI.

La especificación Open Service Interface Definition (OSID) define un conjunto de interfaces de servicios web para entornos de enseñanza virtual, dirigidos a la comunidad educativa de estudios superiores y diseñados para conseguir interoperabilidad entre aplicaciones.

La propuesta de OKI, muy próxima a la implementación, consiste en la definición de un conjunto de interfaces organizados en paquetes sin incluir una definición conceptual de los mismos que ayude a entender dicha especificación y promueva su uso.

La especificación se compone de un conjunto de interfaces agrupados por servicios, a cada uno de estos servicios se le denomina OSID. Aunque existen dieciocho OSID, en este trabajo sólo serán analizados algunos servicios de alto nivel: **Agent, Assessment, CourseManagement, Grading, Scheduling, Repository, User Messaging** y **Workflow**; desestimando aquellos OSID más ligados a la implementación: Root, Sql, Shared, Logging, Id, Hierarchy, Filing, Dictionary, Auhorization y Authentication. Más información en (Rius, 2010).

Algunos se definen independientemente (Agent/Repository) y otros hacen referencia a clases externas en su definición. Se han estudiado todos los OSID que aparecen en al tabla 1 a excepción del OSID Authorization. Este OSID se ha desestimado por ofrecer servicios a otro nivel de abstracción.

OSID	OSID a los que referencia para su definición
Agent	
Assessment	Agent, CourseManagement, Grading

CourseManagement	Agent, Scheduling, Repository
Grading	Agent, Assessment, CourseManagement
Scheduling	Agent, CourseManagement
Repository	
UserMessaging	Agent
Workflow	Agent, Authorization

Tabla 1: Relaciones entre OSIDs

Aunque interesa la visión de conjunto, se ha considerado oportuno presentar distintos diagramas, uno para cada OSID, con el objetivo de facilitar la legibilidad y comprensión del metamodelo.

Para cada uno de los OSID objeto de estudio se presentará:

- Un diagrama de clases UML, detallando las propiedades de las clases del OSID y mostrando vacías, sin atributos, las clases externas relacionadas.
- Alguna instancia de la clase que se está definiendo, a ser posible referente al proceso de caso de ejemplo de este trabajo, PrepararCursoBDI-0910-UOC.
- Los tipos de relación existentes, entre las clases del OSID en cuestión y clases externas al mismo y sus restricciones.
- Los métodos que, aún teniendo interface en la especificación, no quedan reflejados en el metamodelo, es decir aquellos métodos que no son constructores, ni destructores o bien que no son consultores de propiedades y/o relaciones deducibles del metamodelo.

A continuación en distintos subapartados se presentan los OSID analizados por este orden: *OSID Agent*, *OSID Assessment*, *OSID CourseManagement*, *OSID Grading*, *OSID Scheduling*, *OSID Repository*, *OSID User Managing* y *OSID Workflow*.

3.1. OSID Agent

Este OSID ha sido diseñado para crear, gestionar y consultar agentes, entendiendo a los agentes como individuos o procesos que interactuarán con el sistema, incluídos grupos de ellos.

En la figura 2 se muestra el diagrama de clases correspondiente al paquete Agent de OSID-OKI v2.0.0.

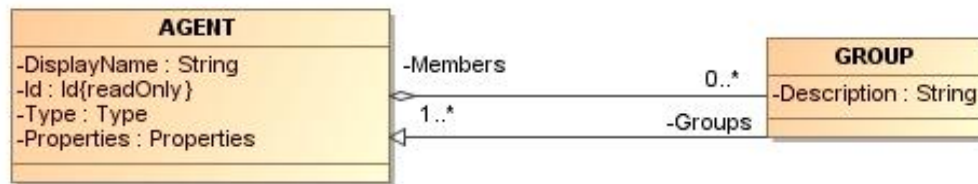


Figura 2. Diagrama de clases del OSID Agent

3.1.1. Clase Agent

Un **Agent** o agente es una entidad (individuo o proceso) que invoca servicios específicos.

Aunque la clase *Agent* mantiene relaciones con muchas otras clases, en este apartado no se mencionan debido a que la navegabilidad es en sentido contrario y se definirán al describir las demás clases.

Ejemplos de instancias de la clase **Agent** en el caso de estudio son el PRA de BDI (*PRA_BDI*), cada uno de sus consultores (*ConsultorTeoriaBDI_1*, *ConsultorTeoriaBDI_2*, *ConsultorTeoriaBDI_3*, *ConsultorLaboratorioBDI_1*) y/o el proceso *AsignarEncargosEBC_BDI_T10*.

3.1.2. Clase Group

Un **Group** o grupo es un tipo de agente concreto que representa la agrupación de uno o más agentes bajo un mismo nombre o descripción.

Los grupos pueden contener otros grupos. Puesto que un grupo es una especialización de agente, hereda todos sus propiedades y además incorpora algún nuevo atributo como la descripción.

Un ejemplo de instancia de la clase **Group** es el equipo de consultoría de BDI (*EquipoConsultoriaBDI*) cuyos miembros son los consultores del curso de BDI.

3.1.3. Tipos de relación

La especificación OKI-OSID tiene en cuenta algunos tipos de relación entre la clase *Agent* y la clase *Group*, por ejemplo, los agentes miembros de un grupo (Members) y los grupos a que pertenece un agente (Groups).

Métodos

Este OSID ofrece interfaces para buscar agentes o grupos por criterio de búsqueda (`getAgentBySearch`, `getGroupBySearch`) o por tipo (`getAgentByType`, `getGroupByType`). También para comprobar si un grupo o miembro está contenido en un grupo (`contains`) y saber de qué grupos es miembro un agente (`getGroupsContainingMembers`).

3.2. OSID Assessment

El OSID **Assessment** tiene como finalidad proporcionar interfaces de servicios para crear, organizar, administrar, evaluar, almacenar y recuperar actividades de curso. En la figura 3 se muestra el diagrama de clases correspondiente a este OSID.

Las actividades llamadas *Assessment* siguen un ciclo de vida, son creadas, publicadas, descargadas para su realización y entregadas para su evaluación.

Se consideran formadas por unidades o componentes reutilizables, ejercicios (*Section*) y preguntas (*Item*) que pueden ser evaluadas independiente. Sin embargo, la entrega se realiza únicamente a nivel de actividad, a pesar de contener ejercicios y preguntas.

Según el momento una actividad puede tener distintos estados. Existe una clase para cada posible estado de una actividad (*Assessment*, *AssessmentPublished* y *AssessmentTaken*). Análogamente sucede lo mismo para sus partes, *Section* y *SectionTaken* e *Item* e *ItemTaken*.

Las actividades entregadas son evaluadas, información que se recoge en la clase *Evaluation*. Aunque las actividades, ejercicios o preguntas puedan ser corregidos de forma independiente, la entrega siempre es de una actividad, contenga uno más ejercicios y una o más preguntas.

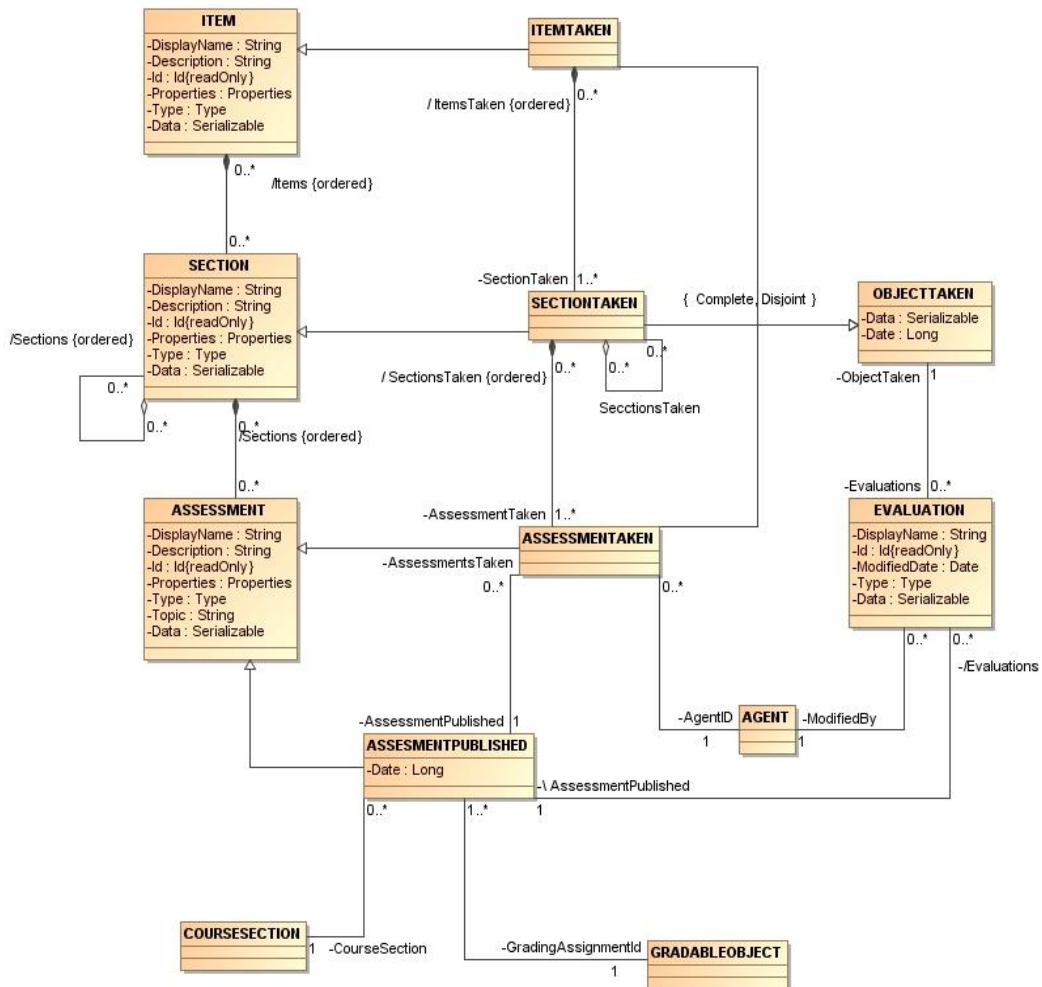


Figura 3. Diagrama de clases correspondiente al OSID Assesment

3.2.1. Clase Assessment

Un *Assessment* o actividad constituye la estructura de organización de más alto nivel de este OSID. Se corresponde con el concepto actividad de curso. Según su complejidad un *assessment* estará formado por *Sections* y cada *section* por *Items*, siendo los primeros equivalentes a ejercicios y los segundos a preguntas.

Cada actividad tiene un nombre, una descripción y un identificador. Es de un tipo determinado y tiene asociado un tópic y su contenido se encuentra en la propiedad *Data*.

En el caso de ejemplo, un *Assessment* sería cualquier actividad del curso de Bases de Datos I, ya sean materiales y recursos docentes de cursos anteriores que se utilizarán como las nuevas actividades (PAC, práctica o examen) creadas a propósito para el nuevo curso.

Ejemplos de instancias de esta clase relativas al caso de ejemplo podrían ser: la primera PAC del curso Bases de Datos I, la segunda práctica y el examen correspondiente al primer turno de examen.

3.2.2. Clase Section

Un *Section* o ejercicio se considera una estructura de organización de nivel intermedio dentro del OSID. Pueden considerarse independientemente o como parte de un *assessment*. Sus propiedades son las mismas que tienen la clase *Assessment* a excepción del tópic.

En el caso de ejemplo, los exámenes de BDI están formados por tres ejercicios, por lo que podríamos citar como posibles instancias de esta clase los ejercicios 1, 2 y 3 del primer turno de examen.

3.2.3. Clase Item

Un *Item* o pregunta es la unidad más básica que los *assessment* pueden contener. Una pregunta puede pertenecer a más de un ejercicio o actividad pero es creada independientemente para ser reutilizada posteriormente. Tiene las mismas propiedades que la clase *Section*.

Suponiendo el caso de ejemplo, instancias de esta clase serían las preguntas de primer ejercicios del primer turno de examen de BDI en el curso 2009-2010.

3.2.4. Clase AssessmentPolished

Un *AssessmentPolished* o actividad publicada es básicamente un *assessment* con información adicional relativa a su publicación.

Al ser una especialización de la clase *Assessment*, hereda las propiedades de esta clase e incorpora otros nuevos relativos a la publicación de la actividad tal como la fecha de publicación.

En el caso de ejemplo, no existen instancias de esta clase puesto que el proceso de preparación del curso de Bases de Datos I finaliza antes de publicarse ninguna actividad en las aulas, pero suponiendo el curso de Bases de Datos I iniciado, un ejemplo de esta clase podría ser la PAC1 del curso.

3.2.5. Clase ObjectTaken

La clase *ObjectTaken* u objeto evaluable representa una actividad, ejercicio o pregunta entregada por un estudiante para su evaluación.

Se trata de una generalización completa y disjunta de las clases *AssessmentTaken*, *SectionTaken* e *ItemTaken*, por lo que toda instancia de *ObjectTaken* es instancia de alguna de las clases anteriormente mencionadas y cualquier instancia de la clase *ObjectTaken* pertenece a únicamente a una de las clases que generaliza.

Según la especificación OKI-OSID es en tiempo de ejecución que se controlan las autorizaciones para la entrega de actividades según el curso en que está matriculado el estudiante y el aula donde está inscrito.

En el caso de ejemplo, no se consideran instancias de esta clase por no existir actividades entregadas, ya que el curso aún no se ha iniciado. Una vez iniciado el curso, podrían considerarse instancias de esta clase el primer ejercicio del primer turno de examen o la primera PAC entregada por un estudiante concreto.

3.2.6. Clase AssessmentTaken

Un *AssesmentTaken* es una actividad entregada por algún estudiante tras su realización con el fin de ser evaluada. No tiene más propiedades que las heredadas de la clase *ObjectTaken*.

En el caso de ejemplo, no existen instancias de esta clase por el motivo ya mencionado anteriormente, sin embargo una vez iniciado el curso podría considerarse instancia de esta clase la primera PAC o el primer examen realizados por algún estudiante.

3.2.7. Clase SectionTaken

Un *SectionTaken* o ejercicio entregado es un ejercicio realizado y entregado por algún estudiante que forma parte de alguna actividad entregada. Igual que en el caso anterior es una especialización de la clase *ObjectTaken* que no añade atributos específicos a los de la clase que especializa.

En el caso de ejemplo, tampoco existen instancias de esta clase, pero en un estadio más avanzado del curso podría serlo el primer o segundo ejercicio del primer turno de examen entregado por un estudiante determinado.

3.2.8. Clase ItemTaken

Un *ItemTaken* o pregunta entregada es una pregunta que forma parte de algún ejercicio entregado por un estudiante. Es una especialización de la clase *ItemTaken* a la cual le corresponde alguna instancia de la clase *SectionTaken*. Por los mismos motivos que antes, no se presentan sus atributos.

En el caso de ejemplo, no existen instancias de esta clase, pero en caso de existir el detalle de actividades a nivel de pregunta, un ejemplo podría ser la pregunta 1 del ejercicio 1 del examen 1 entregado por un estudiante concreto.

3.2.9. Clase Evaluation

La clase *Evaluation* o evaluación mantiene información relativa a los resultados de las evaluaciones de actividades entregadas por algún estudiante y corregidas por algún evaluador.

En el caso de ejemplo, no existen resultados de la evaluación puesto que no existen actividades entregadas. Sin embargo, suponiendo que un estudiante concreto ha entregado la primera PAC, existiría una instancia de evaluación cuyo contenido estaría almacenado en la propiedad *Data* y la fecha e la última modificación en la propiedad *ModifiedBy*.

3.2.10. Tipos de relación

Los tipos de relación entre clases de este OSID y otras clases externas son los mostrados en la tabla 2. Aunque explícitamente no se muestren las restricciones, para mejorar la legibilidad de la información relevante, mencionar que los tipos de relación *Sections* i *SectionTaken* tiene la propiedad transitiva.

Tipos de relación	Descripción
<i>Sections</i>	<i>Sections</i> o ejercicios que ordenados forman una actividad u otro ejercicio
<i>Items</i>	<i>Items</i> o preguntas que ordenadas forman un ejercicio
<i>GradingAssignmentId</i>	Criterios de evaluación de una actividad publicada
<i>CourseSectionId</i>	Curso donde se está publicada una actividad concreta
<i>AgentId</i>	Estudiante que entrega una actividad
<i>AssessmentPublished</i>	Actividad publicada de referencia correspondiente a una actividad entregada o a una evaluación
<i>AssessmentTaken</i>	Actividad entregada de la que forma parte un ejercicio entregado
<i>AssessmentsTaken</i>	Actividad entregada de una actividad publicada

SectionTaken	Ejercicio entregado del que forma parte una pregunta entregada
SectionsTaken	Ejercicio de una actividad entregada u otros ejercicios
ItemsTaken	Preguntas de un ejercicio entregado
ObjectTaken	Objeto al que se refiere una evaluación
Evaluations	Evaluaciones relativas al objeto entregado o una actividad publicada
ModifiedBy	Agente evaluador de una evaluación

Tabla 2. Tipos de relación de las clases del OSID CourseManagement

La organización de actividades como conjuntos de ejercicios ordenados y de ejercicios como conjunto de preguntas ordenadas requiere saber los ejercicios que forman de una actividad (Sections) u otro ejercicio y su ordenamiento, así como las preguntas que constituyen un ejercicio (ítems) y su orden dentro del ejercicio. La navegabilidad en sentido contrario no tiene sentido debido a que actividades, ejercicios y preguntas pueden existir de forma independiente.

Cada actividad se publica en un aula (CourseSection) y se establecen unos criterios de evaluación (GradingAssignmentId). Los estudiantes realizan estas actividades y tras su entrega se obtiene el resultado de su evaluación (Evaluations).

Las actividades entregadas provienen de alguna actividad publicada (AssessmentPublished) realizada y entregada por algún estudiante (AgentId). Puesto que una actividad entregada puede estar formada por un conjunto de ejercicios, que pueden ser evaluados independientemente, interesa saber los ejercicios entregados que constituyen una actividad (SectionsTaken) u otro ejercicio y al revés, a partir de un ejercicio entregado saber la actividad que se va a evaluar (AssessmentTaken). Análogamente, la evaluación de ítems o preguntas de forma independiente requiere saber a qué ejercicio entregado pertenecen las preguntas corregidas (SectionTaken) y viceversa, dado un ejercicio o sectionTaken qué preguntas lo componen (ItemsTaken).

Además, como cualquier actividad entregada se refiere a un objeto entregado (ObjectTaken), la relación ObjectTaken permite saber a qué actividades, ejercicios y preguntas entregados pertenecen las evaluaciones. Una evaluación es llevada a cabo por un evaluador (ModifiedBy) y se basa en una actividad publicada (AssessmentPublished).

En este OSID hay algunos tipos de relación derivados de otros (*items*, *sections*, *itemsTaken*, *sectionsTaken*, *assessmentPublished*) por lo que pueden definirse las correspondientes reglas derivación.

La composición y ordenación de ejercicios dentro una actividad (Sections) o de preguntas u otros ejercicios dentro de un ejercicio (Items) pueden obtenerse a partir de la partes que componen una actividad u ejercicio respectivamente.

Los ejercicios que forman una actividad y su orden se obtiene a partir de todas las partes que forman la secuencia de ejercicios que determina la actividad. Análogamente para las preguntas u ejercicios que componen un ejercicio, así como para la composición de actividades y ejercicios entregados.

Mediante reglas de derivación también puede describirse cuál es la actividad de referencia publicada de una actividad entregada con el fin de proceder a su evaluación completa o parcial (preguntas u ejercicios).

3.2.11. Métodos

La especificación además ofrece interfaces para invocar a métodos consultores de una actividad, ejercicio o pregunta por tipo (`getAssessmentByType`, `getSectionByType`, `getItemByType`). También es posible obtener evaluaciones de un tipo (`getEvaluationsByType`).

3.3. OSID CourseManagement

El objetivo del OSID **CourseManagement** es soportar la creación y gestión del catálogo de cursos.

El catálogo de cursos está organizado principalmente en torno a una jerarquía de tres niveles: **CanonicalCourse**, **CourseOffering** y **CourseSection**. Sin embargo, incluye otras clases como **CourseGroup**, **EnrollmentRecord**, **CourseGradeRecord** y se relaciona con otras clases externas para asociar a cada curso materiales, evaluaciones, tareas o agentes.

En la figura 4 se muestra el diagrama de clases UML correspondiente este OSID.

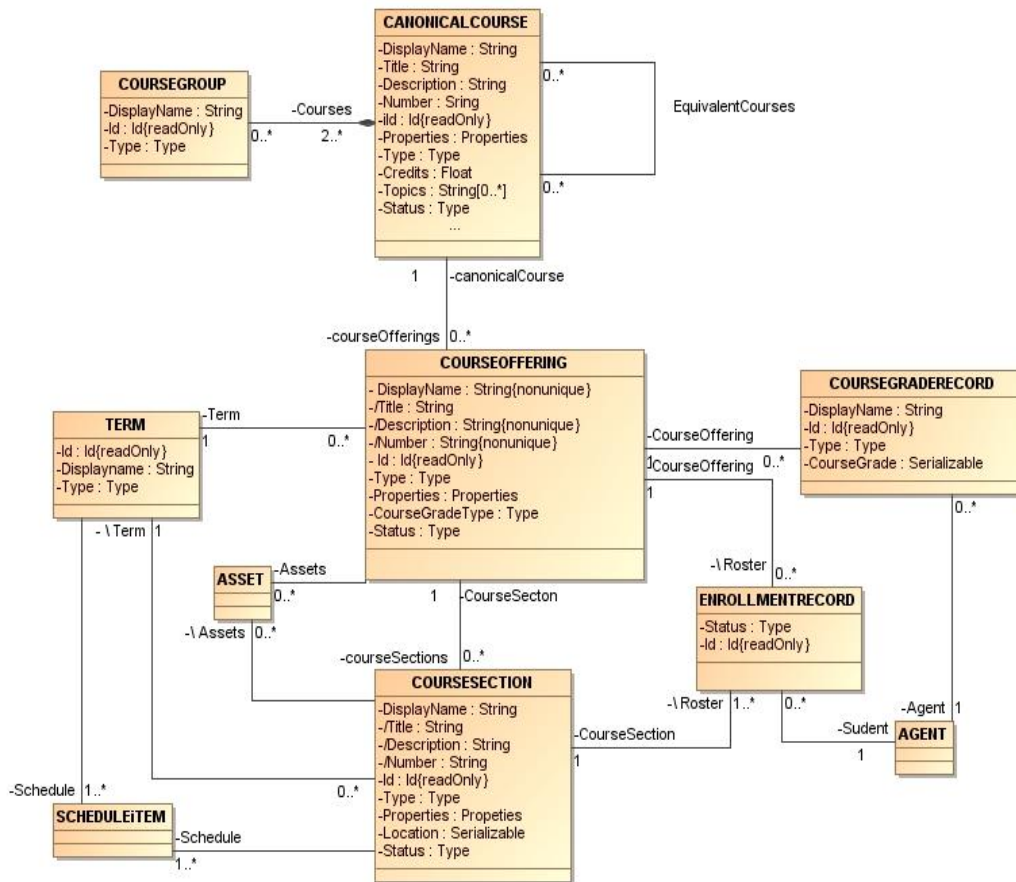


Figura 4. Diagrama de clases del OSID CourseManagement

3.3.1. Clase CanonicalCourse

Un *CanonicalCourse* o curso canónico representa el concepto de asignatura a nivel de plan de estudios, independientemente de cursos académicos concretos y aulas de impartación, y se relaciona con otras asignaturas del mismo plan de estudios.

Todo curso canónico se caracteriza por tener asociada una descripción, un título, un número de curso, un número de créditos y una lista de tópicos. Además tiene otras propiedades propias del curso como el tipo de curso y el estado del mismo.

Un ejemplo de instancia de esta clase para el caso de ejemplo es la asignatura Bases de datos I (*BDI*) que se imparte en dos titulaciones: la Ingeniería Informática de Gestión y la Ingeniería Informática de Sistemas.

3.3.2. Clase CourseOffering

Un *CourseOffering* o curso ofrecido es la concreción de un curso canónico en un periodo académico y con un tipo de evaluación concreto.

Además de las propiedades derivadas de la clase *CanonicalCourse* (Title, Description, Number) que pueden redefinirse, tiene otras como son: el tipo de evaluación, el tipo de curso y el estado del curso.

Un ejemplo de instancia de esta clase es el curso de Bases de datos I para el semestre Otoño 2010.

3.3.3. Clase Term

Un *Term* es un periodo académico durante el cual están planificadas un conjunto de tareas necesarias para la impartición de un curso.

Una instancia de esta clase es el semestre Otoño 2010 (*T10*).

3.3.4. Clase CourseSection

Un *CourseSection* o aula es la materialización de un curso que se imparte, fijando localización y planificación.

Al igual que para los cursos ofrecidos, las propiedades Title, Description, Number pueden ser derivadas de las correspondientes en la clase *CanonicalCourse* o redefinidas al crear el aula. La localización del curso es fundamental en esta clase y viene determinada por el valor de la propiedad *Location*. Además, tiene asociadas otras propiedades como son el tipo o el estado del curso.

Para el caso de estudio existirían tantas instancias de *CourseSection* como aulas de Bases de datos, por lo que se crearían instancias de aulas de tipo Teórico e instancias de aulas de tipo Laboratorio, tantas como se requieran de cada tipo.

3.3.5. Clase CourseGradeRecord

Un *CourseGradeRecord* o registro de evaluación de un curso determina el tipo de evaluación. Se define a partir del agente evaluador y el curso.

En el caso de ejemplo, no existen instancias de esta clase, ya que un registro de evaluación queda definido cuando un estudiante entrega una actividad una vez iniciado el curso. El registro de evaluación de la PAC1 entregada por un estudiante concreto sería una instancia de esta clase.

3.3.6. Clase EnrollmentRecord

Un *EnrollmentRecord* o registro de matrícula almacena información sobre la matriculación de estudiantes a cursos.

Cada registro de matrícula se caracteriza por un identificador y un indicador sobre el estado de la matrícula.

En el caso de ejemplo, las matrículas asociadas a cada aula del curso Bases de Datos I serían posibles instancias de la clase. Además, se podría obtener información sumariada a partir de la unión de todas las matrículas de estudiantes inscritos en aulas del curso.

3.3.7. Clase CourseGroup

Un *CourseGroup* o grupo de cursos es una agrupación de dos o más cursos canónicos dentro de un plan de estudios de acuerdo con algún criterio (cadena de prerequisites, correquisitos o tipo de especialización).

En el caso de ejemplo, no se menciona ningún grupo de asignaturas en las que se incluya el curso Bases de Datos I, puesto que la escena se circunscribe a la preparación de un único curso.

3.3.8. Tipos de relación

Los tipos de relación entre las clases del OSID *CourseManagement* y otras clases externas relacionadas con ellas, se presentan en la tabla 3. Las restricciones de estos tipos de relación no se muestran explícitamente para mejorar la legibilidad de la información. Sin embargo, mencionar que los tipos de relación *EquivalentCourses*, *Term* y *Assets* tienen la propiedad simétrica y que *EquivalentCourses* a su vez tiene la propiedad transitiva.

Tipo de relación	Descripción
<i>EquivalentCourses</i>	Cursos canónicos equivalentes
<i>Courses</i>	Cursos canónicos que forman un grupo de cursos
<i>CourseOfferings</i>	Cursos ofrecidos asociados a un curso canónico
<i>CourseSections</i>	Aulas asociadas a un curso ofrecido
<i>Term</i>	Curso académico asociado a un curso ofrecido o aula
<i>CanonicalCourse</i>	Curso canónico correspondiente a un curso ofrecido
<i>CourseOffering</i>	Curso al que se asocia un aula, un registro de evaluación o una matrícula
<i>CourseSection</i>	Aula a la que se asocia a una matrícula

Assets	Materiales asociados a un curso ofrecido o un aula
Roster	Matrículas asociadas a un curso o aula
Schedule	Tareas planificadas asociadas a un aula
AgentId	Agente que define un registro de evaluación o que se matricula en un aula de un curso

Tabla 3: Tipos de relación de las clases del OSID CourseManagement

Existen cursos canónicos equivalentes (*EquivalentCourses*) y cada asignatura o curso canónico se puede ofrecer en varios cursos (*CourseOfferings*).

Cada curso ofrecido corresponde a un curso canónico (*CanonicalCourse*), a un curso académico (*Term*), utiliza materiales (*Assets*), tiene estudiantes matriculados en él (*Roster*) y se imparte en ciertas aulas (*CourseSections*).

Así mismo, las aulas o *CourseSections* están relacionadas con el curso que se imparte (*CourseOffering*), el período académico en cuestión (*Term*), la planificación del curso entendida como un conjunto de tareas (*Schedule*), los materiales utilizados (*assets*) y sus matrículas (*Roster*).

Un curso canónico o asignatura se relaciona con un grupo de cursos mediante la relación (*Courses*), lo que permite saber qué cursos que forman cada grupo de cursos. Sin embargo, no existen navegabilidad en sentido contrario y por tanto no se puede saber a qué grupo de cursos pertenece un curso canónico.

Un registro de evaluación pertenece a un curso ofrecido (*CourseOffering*) y es definido por un agente evaluador (*AgentId*). OKI-OSID no considera estas relaciones en sentido contrario y por tanto no ofrece interfaces para conocer qué cursos han sido definidos por un agente o cuales han sido las evaluaciones de un curso.

No existe tampoco una relación directa entre estudiante y cursos o aulas, sin embargo es posible saber los estudiantes asignados a un aula o curso y los estudiantes de un curso o aula a partir de un recorrido de las instancias de la clase *EnrollmentRecord*. Análogamente, sucede con los evaluadores del curso, que pueden obtenerse a través de un recorrido por las instancias de la clase *CourseGradeRecord*.

Algunos tipos de relación se pueden derivar mediante las reglas de derivación definidas en la ontología.

3.3.9. Métodos

Ofrece interfaces para métodos consultores que permitan obtener grupo de cursos, cursos canónicos, cursos ofrecidos, aulas o matrículas por tipo (*getCourseGroupByType*, *getCanonicalCourseByType*, *getCourseOfferingByType*, *getCourseSectionByType*, *getRosterByType*) o periodos académicos a partir de una fecha determinada (*getTermByDate*).

También existen métodos para la actualización de la descripción (updateDescripton), título (updateTitle) y número (updateNumber) de un courseOffering o courseSection.

3.4. OSID Grading

El OSID **Grading** tiene por objetivo caracterizar, almacenar y recuperar calificaciones.

Para cada actividad entregada existe un objeto de evaluación (GradableObject) que especifica el tipo de evaluación y un registro de evaluación (GradeRecord) para almacenar la evaluación concreta.

En la figura 5 se muestra el diagrama de clases UML correspondiente a este OSID y otras clases externas relacionadas.

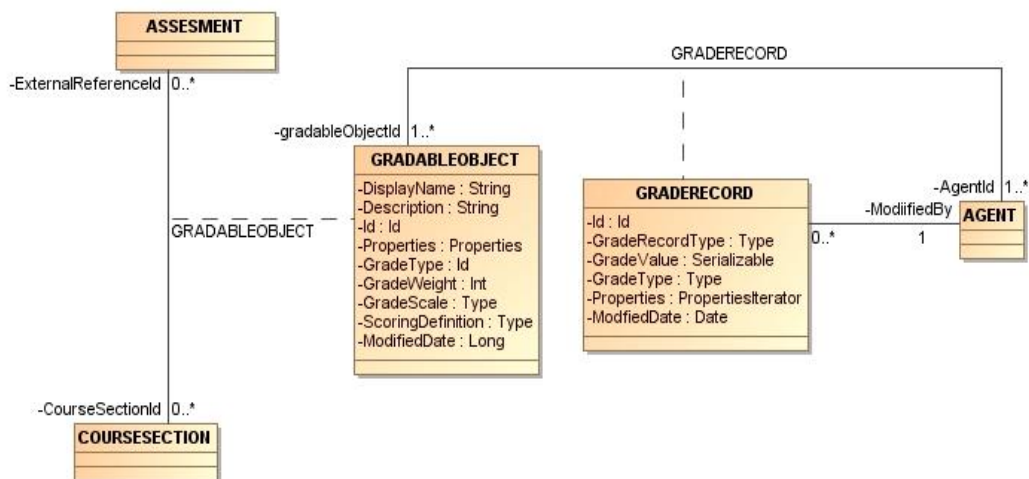


Figura 5: Diagrama de clases correspondiente al OSID Grading

3.4.1. Clase GradableObject

Un GradableObject u objeto de evaluación se caracteriza básicamente por cuatro elementos: 1) tipo de evaluación, 2) escala, 3) ponderación o peso respecto a otros objetos evaluables del curso y 4) descripción de la calificación establecida para intervalos de valores posibles.

En el caso de ejemplo, el objeto evaluable de una PAC o práctica se crea en el momento en que se define el plan docente. Por tanto, existirán tantas instancias de esta clase como actividades evaluables (*PAC, práctica o exámenes*) determine el plan docente.

3.4.2. Clase GradeRecord

Un GradeRecord o registro de calificación almacena la calificación obtenida para cada estudiante y actividad entregada.

Para cada registro de evaluación, además del resultado de la evaluación de la actividad, se especifica el tipo de evaluación aplicado y la fecha en que ha sido evaluado.

En el caso de ejemplo, no existen instancias de esta clase puesto que los registros de calificaciones se generan una vez iniciado el curso. Suponiendo entregada y evaluada la PAC1, serían instancias de esta clase los registros de calificación de cada estudiante para esta actividad.

3.4.3. Tipos de relación

Las relaciones entre las clases de este OSID y otras clases externas son las que se detallan en la tabla 4.

Tipos de relación	Descripción
CourseSectionId	Aula asociada un objeto evaluable
ExternalReferencId	Actividad asociada un objeto evaluable
GradableObjectId	Objeto evaluable asociado a un registro de evaluación
AgentId	Estudiante al que corresponde un registro de evaluación
ModifiedBy	Agente que ha actualizado el registro de evaluación

Table 4: Tipos de relación de las clases del OSID Grading

Los objetos de evaluación establecen los criterios de evaluación para cada actividad asociada a un aula, siendo posible determinar el aula (CourseSectionId) y a actividad (ExternalReferencId) a la que se refieren.

Los registros de calificaciones en cambio existen para cada evaluación de actividad entregada y evaluada, lo que permite determinar qué estudiante ha realizado la actividad (AgentId), los criterios de evaluación utilizados (GradableObjectId) y el agente evaluador (ModifiedBy).

3.4.4. Métodos

No existen interfaces de métodos más allá de los que se derivan del metamodelo correspondiente a este OSID.

3.5. OSID Scheduling

El objetivo del OSID *Scheduling* es la asociación de agentes a tareas específicas durante un periodo determinado. Para cada tarea y agente involucrado se determina un estado que indica el nivel de compromiso adquirido por el agente en la realización de dicha tarea, tal como se muestra en la figura 6.

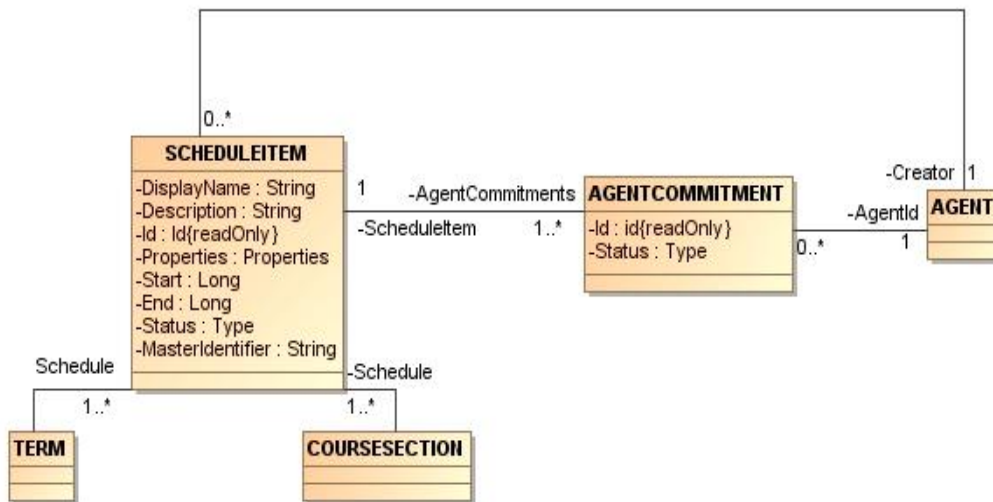


Figura 6: Diagrama de clases correspondiente al OSID Scheduling

3.5.1. Clase ScheduleItem

La clase *ScheduleItem* o ítem de planificación captura información sobre las tareas a llevar a cabo.

Un ítem de planificación o una tarea planificada se caracteriza por el periodo de realización de dicha tarea y los agentes que tienen el encargo de realizarla.

Una instancia de esta clase podría ser el encargo EBC que corresponde a la tarea de consultoría básica en la UOC.

3.5.2. Clase AgentCommitment

Un *AgentCommitment* o compromiso de realización almacena el grado de compromiso que adquiere cada agente en cada tarea que participa. Este OSID pretende mantener y gestionar las asignaciones de personas a tareas y el estado de la

realización de las mismas, independientemente de las características de la tarea y periodo de tiempo indicado en la clase *ScheduleItem*.

Las propiedades de esta clase son el identificador de asignación y el estado de la misma.

Una instancia de esta clase podría ser la asignación del encargo de consultoría básico de cada consultor del curso de bases de datos I.

3.5.3. Tipos de relación

Los tipos de relación para las clases de este OSID son los que se detallan en la tabla 5.

Tipo de relación	Descripción
AgentId	Agente a quien se encarga realizar una tarea asumiendo cierto compromiso
ScheduleItem	Tarea planificada
Creator	Agente que planifica la asignación de tarea
AgentCommitments	Encargos asignados en la planificación de una tarea

Tabla 5: Tipos de relación de las clases del OSID Scheduling

Para cada encargo, es decir tarea planificada y agente que participa, existe un compromiso de realización. Cada encargo asignado o compromiso de realización tienen asociado un agente (AgentId) y le corresponde un grado de compromiso, que puede ser el mismo para la realización de varias tareas. Sin embargo, una tarea planificada puede tener asociados varios compromisos de realización (AgentCommitments) según los agentes que intervengan en ella.

La especificación OKI-OSID no contempla las tareas en las que puede estar comprometido un agente y las tareas con un mismo nivel de compromiso.

3.5.4. Métodos

Se ofrecen interfaces para métodos consultores que permitan la consulta de: ítems de planificación por identificador (getScheduleItem), disponibilidades coincidentes entre los agentes que intervienen en una misma tarea (getAvailableTimes) y tareas asignadas a un grupo de agentes durante un periodo de tiempo (getScheduleItemsForAgents).

Para cada registro de evaluación, además del valor correspondiente al resultado de evaluar la actividad, debe de especificarse el tipo de evaluación aplicado y la fecha de evaluación.

3.6. OSID Repository

El OSID *Repository* tiene por objetivo proveer interfaces para el almacenamiento y recuperación de todo tipo de contenido digital, así como cualquier información relativa a dicho contenido y/o su composición.

Al contenido digital se le denomina *Asset* y su composición puede ser muy diversa. Así mismo, para describir un *Asset* pueden ser necesarias otras clases: *Part*, *PartStructure*, *Record*, y *RecordStructure*.

En la Figura 7 se muestra el diagrama de clases correspondiente a este OSID.

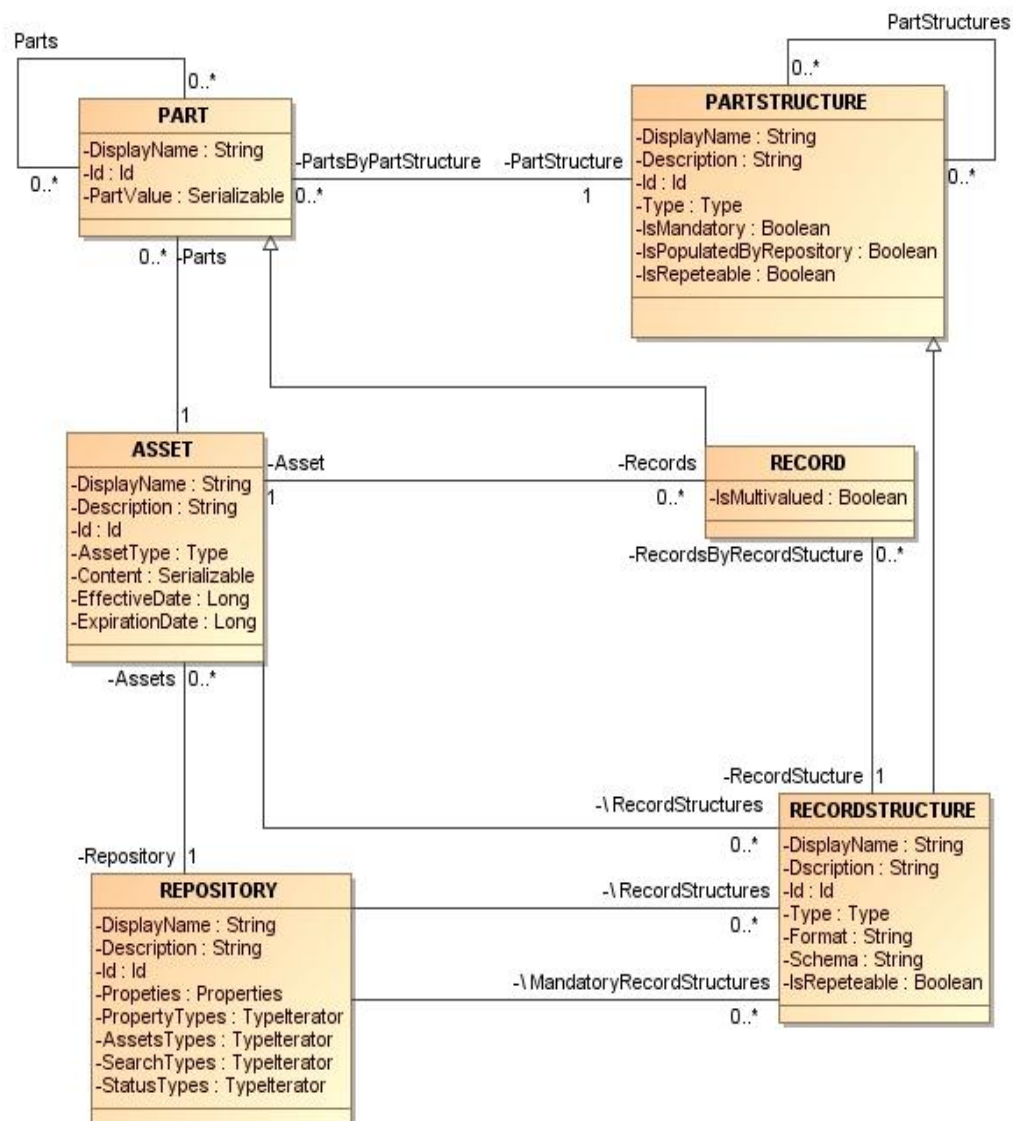


Figura 7: Diagrama de clases correspondiente al OSID Repository

3.6.1. Clase Asset

Un *Asset* es un recurso digital almacenado en algún repositorio con un periodo de validez.

Todo asset tiene un tipo, un contenido y un periodo de validez, además del nombre, descripción e identificador.

El contenido de un asset puede estar formado por: 1) únicamente contenido (p.e. un documento resumen de un tema), 2) registros de contenido y estructuras de registros que describen dicho contenido (p.e. una PAC anotada semánticamente) y 3) sólo metadatos descritos mediante estructuras de registros (p.e. metadatos que describen una plantilla de examen).

En el caso de ejemplo, los exámenes del semestre anterior, las PAC o prácticas de otros semestres o material de soporte a la instalación del software para las aulas de laboratorio serían instancias de la clase *Asset*.

3.6.2. Clase Part

Un *Part* o parte de un recurso digital se define de acuerdo con una estructura. El contenido del asset se almacena en la propiedad *PartValue* y las partes de asset forma parte de un asset. Además, una parte de asset puede ser estructurada y estar organizada de acuerdo con una estructura.

Así pues, un asset se caracteriza por: 1) almacenar un valor o contenido de la parte, 2) estar estructurado de acuerdo con alguna estructura y 3) ser parte de un asset.

Las propiedades de esta clase son el identificador, el nombre que lo describe y su contenido.

Aunque en el caso de ejemplo, los materiales del curso son vistos como un todo, en el caso de los exámenes se consideran sus partes, los ejercicios. Por tanto, una instancia de parte de asset podría ser el primer ejercicio del primer turno de examen.

3.6.3. Clase PartStructure

Un *PartStructure* o estructura de una parte de recurso digital permite definir la estructuración de partes de asset. Dich estructuración tiene en cuenta si las estructuras de registro son obligatorias u opcionales, repetibles o no y si son pobladas por algún repositorio.

En el caso de ejemplo, los ejercicios de examen se consideran assets formados sólo por contenido, no existen instancias de estructuras de partes de asset. Si los

ejercicios de examen además de contenido tuvieran asociados metadatos, éstos formarían parte de estructuras de registros y serían instancias de esta clase.

3.6.4. Clase Record

Un *Record* o registro es una parte de asset que se define de acuerdo a una estructura de registro. Se caracteriza por la estructura que lo define y el asset al que pertenece.

Los registros pueden ser estructurados y estar formados por otros assets, que se denominan partes y éstos pueden contener otras partes.

A parte de las propiedades que hereda de la clase *Part*, tiene otras propiedades para indicar si es posible la existencia de múltiples instancias de registro para una misma estructura de registro o no.

En el caso de ejemplo, por el mismo motivo que no se consideran partes o estructuras de partes, tampoco se consideran registros. Considerando el ejercicio de examen como una parte del asset examen, una instancia de registro podría ser un tipo concreto de ejercicio de examen, ya que es una especialización de una parte de asset.

3.6.5. Clase RecordStructure

Un *RecordStructure* o estructura de registro define la estructuración de un registro o parte de asset. Concretamente, permite definir el formato y el esquema de representación del registro asociado y determinar si la estructura es repetible o no.

En el caso de ejemplo, tampoco se contemplan estructuras de registro, por no existir registros propiamente. Sin embargo, en un proceso de preparación de materiales de curso, en el que se diseñen las actividades en base a unidades modulares, la definición de cada una de ellas sería una instancia de estructura de registro.

3.6.6. Clase Repository

Un *Repository* o repositorio es un almacén de recursos digitales o partes de ellos que permite almacenar y gestionar cierto tipo de assets.

Además de las propiedades básicas de un repositorio (identificador, descripción y nombre) tienen un tipo, el tipo de contenido digital que acepta, criterios de localización de assets y un estado.

En el caso de ejemplo, como instancia tenemos el RepositorioUOC, que representa el repositorio institucional de la UOC, el cual almacena todo tipo de recursos digitales de la organización.

3.6.7. Tipos de relación

Los tipos de relación que se establecen entre las clases del OSID Repository y otras clases externas son las que se describen en la tabla 6. Aunque debieran de figurar en la tabla las restricciones asociadas a los tipos de relación, no se incluyen para mejorar la legibilidad de la información contenida ella. Sin embargo, interesa mencionar los tipos de relación Parts y PartStructures tienen la propiedad transitiva.

Tipo de relación	Descripción
Repository	Repositorio donde se almacena el recurso digital
Assets	Recursos digitales almacenados en un repositorio
Parts	Recursos digitales componentes de un asset o parte de asset
Records	Registros de un asset
Asset	Asset que corresponde a un registro
RecordStructure	Estructura de registro que determinado por la organización
RecordsByPartStructure	Registros que tienen una estructura de registro determinado
PartStructure	Estructura de una parte de asset
PartsByPartStructure	Partes que tienen una estructura de parte de asset determinada
PartStructures	Estructura de una parte de asset que contiene otras estructuras de partes de asset
RecordStructure	Estructura de registro que corresponde a un registro
RecordStructures	Estructuras de registros que reconoce un repositorio o definen un asset
MandatoryRecordStructures	Estructuras de registro obligatorias para los assets de un repositorio

Tabla 6: Tipos de relación del OSID Repository

Un asset, se almacena en un repositorio (Repository) y puede estar compuesto por partes (Parts), que a su vez pueden estar formadas por otras partes (Parts). Un asset que no esté formado sólo por contenido, tendrá al menos algún registro (Record) y cada uno de estos registros tendrá una estructura de registro (RecordStructures) que describirá como se organizan los registros que componen el asset.

Las partes de recursos digitales o partes de asset tienen asociado un registro de estructura, es decir información sobre la organización de las partes de asset (PartStructure). Estos registros de estructura de partes de asset pueden a su vez estar formados por otros registros de estructura (PartStructures).

Los registros corresponden a un asset (Asset) y tienen una estructura de registro (RecordStructure). Puesto que los registros son partes de asset que tienen asociada alguna estructura de parte (PartStructure), ésta a su vez puede estar formada por otras estructuras de partes (PartStructures) y es posible derivar todas las estructuras

de registro (RecordStructures) correspondientes a un asset por transitividad de las mismas.

Los repositorios son de un determinado tipo y reconocen estructuras de registro (RecordStructures), algunas de las cuales son obligatorias (MandatoryRecordStructures).

Dado que cualquier parte de asset o registro tiene una estructura (PartStructure, RecordStructure), también existen relaciones simétricas para determinar las partes de asset (PartsByPartStructure) o registros (RecordsByRecordStructure) que tienen una estructura.

Las estructuras de registro reconocidas por cada repositorio (RecordStructures) se pueden derivar a partir de los assets que almacena (Assets) y las estructuras de registro que tiene asociadas cada asset (RecordStructures).

3.6.8. Métodos

Se ofrecen interfaces para la consulta de assets, registros de assets y repositorios por tipo (getAssetByType, getRecordBType, getRepositoryByType).

También es posible la búsqueda de assets por fecha (getAssetByDate) o por algún criterio de búsqueda (getAssetsBySearch) específico. Así mismo, se puede consultar el valor de las partes de asset que tengan una determinada estructura (getPartValuesByPartStructure).

3.7. OSID User Messaging

El OSID *User Messaging* proporciona interfaces para gestionar el envío y recepción de mensajes, así como las suscripciones para la recepción de mensajes asociados a un tema o a mensajes de cierto tipo.

Un mensaje siempre es enviado por un único agente y puede ser recibido por varios agentes. Debe notarse que clase *Agent* ha sido ampliada con dos nuevas propiedades para permitir la suscripción. En la figura 8 se presenta el diagrama de clases de este OSID.

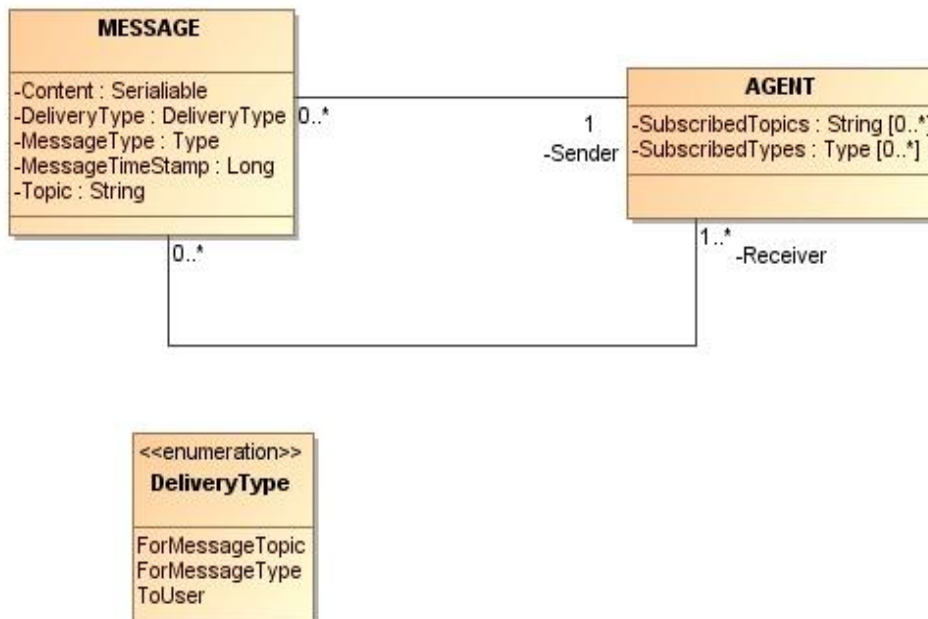


Figura 8: Diagrama de clases correspondiente al OSID UserMessaging

3.7.1. Clase Message

Un *Message* o mensaje es todo aquello que envía algún agente en un momento determinado.

Todo mensaje tiene asociado un tópic, un tipo de mensaje y un tipo de envío. El tipo de envío puede ser por tópic (se envía a todos los suscritores del tópic), por tipo de mensaje o para uno o más usuarios.

En el caso de estudio, los mensajes están representados por los flujos de información que se transmiten entre el PRA y los consultores a través del sistema. Un ejemplo de mensaje sería aquel que el PRA envía al grupo de consutores activos de la asignatura Bases de Datos I para asignar el encargo EBC. Puesto que el grupo de consultores es reducido se puede enviar directamente al grupo de consultores, sin suscripción.

3.7.2. Tipos de relación

La clase *Message* se relaciona únicamente con la clase *Agent* el *OSID Agent*. Todo mensaje es enviado por un único agente (*Sender*) y puede ser recibido por varios agentes (*Receiver*).

La recepción de mensajes por tema o por tipo de mensaje requiere de suscripción. Con esta finalidad, tal como se ha comentado anteriormente, la clase *Agent*, externa a este OSID, ha sido extendida con dos nuevas propiedades que permiten respectivamente la suscripción de un agente a diferentes temas (*SubscribedTopics*) y tipos de envío *SubscribedTypes*. La suscripción es opcional, por lo que estas propiedades no tienen ningún tipo de restricción de cardinalidad.

3.7.3. Métodos

Este OSID ofrece interfaces de métodos para la gestión de suscripciones (*Subscribe/Unsubscribe*) de agentes o la eliminación de todas las suscripciones (*UnsubscribeAll*). Así como interfaces de métodos consultores que permiten conocer los agentes subscriptores (*getSubscribers*), los agentes suscritos a un tópico (*getSubscribersByTopic*) y los mensajes recibidos por tópico (*getReceiveByTopic*) o por tipo de mensaje (*getReceiveForMesageType*).

También proporciona interfaces para el determinar los mensajes enviados por un agente (*Send*) o por un grupo de agentes (*SendToAll*) y para purgar un determinado mensaje (*purgeMessage*).

3.8. OSID Workflow

El OSID **Workflow** proporciona interfaces para definir procesos. Un proceso se realiza en una o más etapas y cada etapa lleva a cabo una o más tareas. La ejecución de las tareas por parte de algún agente permite avanzar el proceso desde la etapa inicial a la final.

Este OSID requiere para su definición las clases *Function* y *Qualifier* pertenecientes al OSID *Authorization*, ya que la realización de una tarea por parte de un agente sólo podrá ser realizada si éste está autorizado para ello.

En la figura 9 se muestra el diagrama de clases UML correspondiente al OSID Workflow.

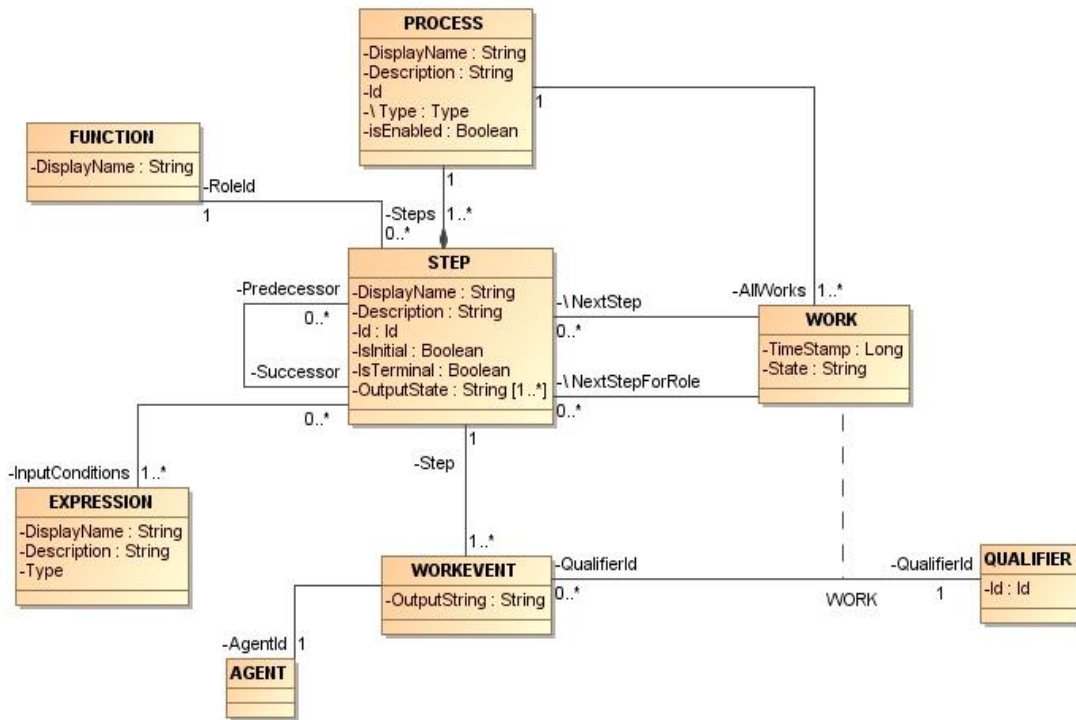


Figura 9: Diagrama de clases correspondiente al OSID UserMessaging

3.8.1. Clase Process

Un *Process* o proceso define un workflow. Se compone de una o más etapas (*Step*) y tiene asociadas una o más tareas o trabajos (*Work*). Todo proceso tiene un estado cuyo valor depende de los trabajos realizados y puede ser habilitar o no.

Una instancia de la clase proceso puede ser el proceso *EstablecerAulas* del caso de ejemplo.

3.8.2. Clase Step

Un *Step* o paso o etapa está constituido por un conjunto de eventos de trabajo y tiene asociados uno o más estados de salida y una o más condiciones de entrada. Se define dentro del contexto de un proceso y se ordena respecto a las demás etapas del mismo proceso. Además, cada etapa tiene una correspondencia con un rol o función reconocido por el sistema y que se corresponde con una de las clases del OSID Authorization.

En el caso de ejemplo no existen instancias para esta clase dado que no se presenta una descomposición del proceso en secuencia de procesos.

3.8.3. Clase Expression

Un Expression o expresión no es más que una descripción textual de las condiciones iniciales de una etapa que tiene un tipo determinado.

Dentro del caso de ejemplo la condición inicial para llevar a cabo la asignación de consultores a aulas es disponer del identificador de asignatura, del identificador de semestre y del número de aulas establecido.

3.8.4. Clase Work

Un *Work* o tarea es la realización de un trabajo sobre un objeto determinado, el cual se corresponde con la clase Qualifier del OSID Authorization. De cada tarea se conserva el instante de tiempo en que se llevará a cabo el trabajo y el estado en el que se encuentra.

Por los mismos motivos que se ha comentado anteriormente, no existen instancias en el caso de ejemplo para ilustrar esta clase.

3.8.5. Clase WorkEvent

Un *WorkEvent* o evento de trabajo es cada una de las tareas que realiza un agente como parte del trabajo de una etapa y que por tanto se describe con un estado de salida. Esta información no es modificable.

Considerando como trabajo la localización de los consultores de la asignatura, el sistema realizaría la búsqueda y se describiría un estado de salida que avisara de la existencia de la lista de consultores disponibles para el curso de BDI.

3.8.6. Tipos de relación

En la tabla 7 se muestran los tipos de relación entre las clases del OSID Workflow y otras clases externas.

Tipo de relación	Descripción
Steps	Etapas que componen un proceso
Predecessor	Etapa anterior
Successor	Etapa posterior

RoleId	Identificador de rol o función reconocida por el sistema
InputConditions	Condiciones de entrada de una etapa
AllWorks	Tareas que se realizan en un proceso
NextSteps	Etapas que proceden a una trabajo o tarea
NextStepForRole	Etapas que proceden a una trabajo o tarea que tienen un mismo rol
AgentId	Agente que realiza un evento de trabajo
QualifierId	Calificador sobre el que se realiza el trabajo

Tabla 7: Tipos de relación del OSID Workflow

Un Proceso se compone de un conjunto de etapas o pasos (Steps), en caso de una única etapa, coincide la etapa inicial con la final. Las etapas están ordenadas dentro del proceso por lo que existen relaciones para establecer la etapa anterior (Predecesor) y la etapa posterior (Sucesor).

Toda etapa tiene asociada una función dentro del proceso que relaciona dicha etapa con las funciones que reconoce el sistema (RoleId). Además toda etapa tiene una o más condiciones de entrada (InputConditions) y se define por un conjunto de estados de salida (OutputStates) que se derivan a partir de los estados de salida de cada tarea o trabajo que se realiza, pudiendo ser completado, disponible o inacabado.

Un proceso lleva a cabo uno o más trabajos o tareas (AllWorks) de forma ordenada. El orden lo establece la etapa o etapas siguientes (NextSteps) o incluso etapas siguientes de un mismo rol (NextStepForRole). Una tarea se compone de uno o más eventos de trabajo. Cada evento de trabajo es una actividad que realiza un agente (AgentId) en un momento determinado sobre un objeto concreto (QualifierId).

Debe de observarse que a parte de la clase Agent correspondiente al OSID Agent, existen relaciones con la clase Function y Qualifier del OSID Authorization.

4. Limitaciones

Como consecuencia del estudio de la especificación OKI-OSID se han detectado una serie de limitaciones que deben de tenerse en cuenta al utilizar esta especificación y que pueden servir para su mejora en una versión posterior. Son las siguientes:

- No contempla la posibilidad de que un agente pueda desempeñar más de un rol, lo cual resulta necesario para la descripción de procesos complejos en los que un mismo actor puede jugar dos papeles. Por ejemplo, en el caso de estudio el PRA podría desempeñar a la vez el rol de PRA y consultor de la asignatura. En este caso la especificación OKI-OSID exigiría crear dos agentes para el mismo

participante, de forma que la misma persona pudiera actuar tanto en calidad de PRA como de consultor.

- No permite la compartición de actividades publicadas entre aulas de un mismo curso. La especificación OKI-OSID asocia cada actividad publicada a un aula, no a un curso concreto y por consiguientemente a todas sus aulas.
- Carece de algunas interfaces para la gestión de cursos.
 - No es posible saber qué actividades ha entregado un estudiante y cuales ha corregido un evaluador. No obstante, esta información se puede obtener mediante un recorrido por las actividades entregadas o las evaluaciones realizadas.
 - No permite especificar más de un evaluador por curso, impidiendo que distintas actividades de curso puedan ser corregidas por evaluadores diferentes o distintas preguntas de examen puedan ser corregidas por distintos evaluadores.
 - Tampoco es posible especificar actividades asociadas a un curso y por tanto asociar actividades a todas las aulas en que se imparte el curso.
 - No es posible saber los cursos o aulas de un periodo académico determinado, las evaluaciones de un curso o las correcciones realizadas por un determinado evaluador.
 - Aunque no existen interfaces para saber los cursos en los que está matriculado un estudiante o las aulas a las que ha sido asignado, esta información es derivable a partir del registro de matriculaciones. Análogamente ocurre con los resultados de las evaluaciones de un curso o las correcciones realizadas por un evaluador.
- Permite definir objetos de evaluación a nivel de aula, pero no a nivel de curso. En consecuencia, las normas/criterios de evaluación no pueden compartirse entre aulas de un mismo curso o actividades similares de distintos cursos. En este sentido sería deseable vincular los objetos de evaluación a cursos y no a aulas.
- No soporta la planificación de tareas en varias etapas, a no ser que se cree el correspondiente flujo de trabajo. Por tanto, el OSID Scheduling no permitiría la especificación de tareas llevadas a cabo en varias etapas, como podría ser la preparación independiente de distintas preguntas de examen, si no es conjuntamente con el OSID workflow.
- No contempla distintas formas de composición de assets como por ejemplo, la composición jerárquica o secuencial. Por tanto, podría considerarse interesante la extensión de interfaces teniendo en cuenta aspectos como éste del estándar IEEE-LOM.
- El envío/recepción de mensajes por tema y tipo de mensaje exige la extensión de la clase Agent con nuevas propiedades.
- Se han detectado varias limitaciones en el OSID Workflow que impiden su utilización para la automatización de los flujos de trabajo que componen los procesos.

- La descripción textual de condiciones iniciales y estados de salida no es útil para la alineación de precondiciones y postcondiciones, ni el procesamiento automático.
- Las tareas y etapas no son reutilizables.
- No existe ningún tipo de relación entre procesos que permita su composición y promueva la modularidad.
- No se permite la representación de procesos complejos en los que un mismo participante asume más de un rol, ya que cada etapa tiene asociado un sólo rol.
- Requiere el uso del OSID Authorization, que se considera de otro nivel.

5. Conclusiones

La creación de nuevas funcionalidades para sistemas de aprendizaje o la creación de nuevas aplicaciones para entornos de e-learning requiere de un conjunto de servicios a los cuales se pueda acceder independientemente de la plataforma y del lenguaje de programación que se utilice para su implementación.

La especificación OKI-OSID con el fin de conseguir interoperabilidad entre aplicaciones propone varios paquetes (OSID), que organizan un conjunto de interfaces de servicios web para entornos de enseñanza superior. De esta especificación se han analizado en profundidad aquellos OSID que permiten describir escenas educativas a alto nivel, se han enumerado propiedades y métodos asociados a cada OSID, así como sus interrelaciones con otros OSIDs a efectos de tener una visión de conjunto que facilite la comprensión de esta especificación y promueva su uso. El resultado ha sido un metamodelo en UML que describe la especificación OKI-OSID.

Durante la construcción de este modelo se han detectado una serie de carencias o limitaciones de esta especificación. Éstas han sido identificadas y recopiladas para ayudar a otros usuarios que pretendan utilizar esta especificación. También se han propuesto algunas mejoras para superar tales limitaciones.

El trabajo realizado al formalizar la especificación OSID de OKI constituye un punto de partida para trabajos posteriores como pueden ser la ampliación y actualización de dicha especificación o la creación de otras especificaciones similares. Además, tal metamodelo puede ser utilizado para la construcción de una ontología que describa la semántica de la especificación, de manera que pueda ser utilizado para describir formalmente escenas educativas en términos de OKI-OSID y ofrecer soporte a la implementación. Dado que las ontologías permiten la compartición y descripción formal de conocimiento, quizás a más a largo plazo ésta pudiera servir para conseguir cierto grado de automatización de las especificaciones de escenas educativas.

Bibliographic references

- IEEE Learning Technology Standards Committee [LTSC WG12].(2002). Draft for Learning object metadata. Retrieved 04/05/2011 from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
- IMS Global Learning Consortium. (2007). Public Draft 2 Content Packaging Specification v1.2. Retrieved from <http://www.imsglobal.org/content/packaging/>.
- IMS Global Learning Consortium. (2006). *IMS Tools Interoperability Guidelines v1.0 Final*. Retrieved from <http://www.imsglobal.org/ti/index.html>.
- IMS Global Learning Consortium. (2008). Public Draft 2 IMS Question Test & Interoperability v2.1. Retrieved from <http://www.imsglobal.org/question/>.
- Inaba, A., Tamura, T., Ohkubo, R., Ikeda, M., Mizoguchi, R., & Toyoda, J. (2001). Design and Analysis of Learners' Interaction based on Collaborative Learning Ontology. *Proceedings of the 1st European conference on computer-supported collaborative learning (Euro-CSCL'2001)* (p. 308–315). Maastricht, The Netherlands.
- Kabel, S. C., Wielinga, B. J., & Hoog, R. de. (1999). Ontologies for Indexing Technical Manuals for Instruction. *AI-ED (Artificial Intelligence in Education) conference. Proceedings workshop on ontologies for intelligent educational systems* (pp. 44-53).
- Advanced Distributed Learning. (2004). Sharable Content Object Reference Model Version 1.3.1. Retrieved from <http://www.adlnet.org>.
- Mizoguchi, R., Sinita, K., & Ikeda, M. (1996). Task Ontology Design for Intelligent Educational/Training Systems. *Position Paper for ITS 96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs* (Vol. 96, pp. 1-21).
- Open Knowledge Initiative. (2004). Open Service Interface Definitions. v 2.0.0. Retrieved from: http://heanet.dl.sourceforge.net/project/okiproject/Doc%20%28previous%20OSID%20versions%29/Full%20Documentation%20Set/OSID_Documentation_rc6.1.pdf
- Object Management Group & BPML.org. (2006). Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification. dtc/06-02-01.
- Rius, A. (2010). Un marco formal para la especificación de escenas educativas. IN3-Internet Interdisciplinary Institute. Barcelona: UOC.

Rius, A., Conesa, J., & Gañan, D. (2010). A DSL Tool to assist specifications of educational settings. In Solo A.; Marsh A; Arabnia H (Eds.), *The 2010 International Conference on Semantic Web and Web Services (SWWS 2010)* (pp. 112-118). CSREA Press.