



UNIVERSITAT OBERTA DE CATALUNYA
ESTUDIOS DE INFORMÁTICA, MULTIMEDIA Y TELECOMUNICACIÓN

TESIS DOCTORAL

UN MARCO FORMAL PARA LA REPRESENTACIÓN DE
ESCENAS EDUCATIVAS REUTILIZABLES

Autor:

Àngels Rius Gavidia

Directores:

Dr. Jordi Conesa Caralt

Doctor Ingeniero en Informática

Dra. María Elena García Barriocanal

Doctora Ingeniera en Informática

Barcelona, May 2010

*A la meua filla, Ariadna, als meus pares i germana, els qui
més estimo i sempre em fan costat.*

Tabla de contenidos

Tabla de contenidos	I
Índice de Tablas	VI
Índice de Figuras	XI
Agradecimientos	XIV
Abstract	XVII
Resumen	XIX
1. Introducción	1
1.1. Problemática referente a las especificaciones de escenas educativas	1
1.2. Objetivos y aportaciones originales	5
1.3. Metodología de trabajo	8
1.4. Caso de ejemplo elegido	10
1.5. Estructura del documento	11
1.6. Resumen	12
2. Estado de la cuestión	13
2.1. Tecnología educativa	13
2.2. Escenas educativas	20
2.2.1. El concepto de escenario educativo	20
2.2.2. Especificación de escenas educativas	21
2.2.3. Especificaciones y estándares relevantes para el concepto de escena educativa	27
2.3. Automatización de escenas educativas	34
2.3.1. Arquitectura abstracta y tecnología de soporte	35
2.3.2. Formalización de escenas educativas	40
2.4. Limitaciones y carencias detectadas	44
2.5. Resumen	46

3. Planteamiento del problema	47
3.1. Motivación	47
3.2. Definición y alcance del problema	51
3.3. Estrategia de resolución	53
3.4. Caso de ejemplo	59
3.5. Resumen	64
4. Ontología para la especificación de patrones de escenas educativas	65
4.1. Especificación de patrones de escenas educativas	65
4.2. La ontología de procesos: OntoProc	69
4.2.1. El dominio de conocimiento de OntoProc	72
4.2.2. Formalización de OntoProc	75
4.2.3. Construcción de OntoProc en OWL	85
4.3. La ontología de escenarios educativos: OntoED	88
4.3.1. El dominio de conocimiento de OntoED	88
4.3.2. Formalización de OntoED	90
4.3.3. Construcción de OntoED en OWL	109
4.4. La ontología de patrones de escenas educativas: OntoProcED	112
4.4.1. Creación de la ontología de Patrones de escenas educativas	114
4.4.2. Construcción de la ontología de Patrones de escenas educativas en OWL	116
4.5. Resumen	116
5. Adaptación de patrones de escenas educativas a una organización concreta: el caso de la UOC	117
5.1. Cómo adaptar escenas educativas genéricas a una organización concreta	117
5.2. La ontología de Procesos concretos: OntoOrg	121
5.2.1. El dominio de conocimiento de OntoOrg	121
5.2.2. Formalización de OntoOrg	122
5.2.3. Construcción de OntoOrg en OWL	124
5.3. La ontología de escenarios educativos de la UOC: OntoUOC	124
5.3.1. El dominio de conocimiento de OntoUOC	124
5.3.2. Formalización de OntoUOC	130
5.3.3. Construcción de OntoUOC en OWL	151
5.4. La ontología de escenas en el contexto de una organización concreta: OntoEDUOC	157
5.4.1. Creación de la ontología de procesos concretos para la UOC	157
5.4.2. Creación de la ontología de escenarios educativos en el contexto de la UOC	159
5.4.3. Integración del nivel de patrones de escenas con el nivel de procesos concretos	161
5.5. Resumen	162

6. Ontología de Implementación para el caso de la especificación OKI-OSID	163
6.1. Formalización de OntoOKI	163
6.1.1. OSID Agent	166
6.1.2. OSID Assessment	168
6.1.3. OSID CourseManagement	178
6.1.4. OSID Grading	187
6.1.5. OSID Scheduling	190
6.1.6. OSID Repository	193
6.1.7. OSID User Messaging	201
6.1.8. OSID Workflow	204
6.2. Construcción de OntoOKI en OWL	209
6.3. Integración de OntoOKI en OntoProcEDUOC	215
6.3.1. Correspondencia entre OntoProcED y OntoOKI	216
6.3.2. Correspondencia entre OntoUOC y OntoOKI	222
6.4. Resumen	223
7. Validación del marco formal propuesto	224
7.1. Ámbitos a considerar en la validación	224
7.2. Correctitud del marco propuesto	225
7.3. Completitud del marco propuesto	226
7.4. Utilidad del marco propuesto	230
7.5. Resumen	231
8. Conclusiones y trabajo futuro	232
8.1. Satisfacción de los objetivos propuestos	232
8.1.1. Resumen del objetivo general	232
8.1.2. Satisfacción de los objetivos específicos	234
8.2. Aportaciones originales	237
8.3. Conclusiones	239
8.4. Líneas de trabajo futuras	240
8.4.1. Extensiones del marco propuesto	241
8.4.2. Posibles usos del marco presentado	241
A. Ontologías del marco formal propuesto	243
B. Guía práctica para la transformación de un diagrama UML a ontología OWL	247
C. Lenguaje gráfico para la representación de escenas educativas	249
D. Patrones de escenas utilizados en el caso de ejemplo	251
E. Diagramas del caso de ejemplo	255

F. Una herramienta CASE para especificar escenas educativas	263
F.1. Utilidad de la herramienta	264
F.2. Arquitectura del prototipo	264
F.3. La interfície del prototipo	265
G. Reglas SWRL para la traducción del marco propuesto a OKI	267
Bibliografía	275

Índice de cuadros

4.1. Tipos de conectores de secuencia	74
4.2. Propiedades de la clase ProcesoTipo	77
4.3. Propiedades de la clase ParticipanteTipo	77
4.4. Propiedades de la clase MensajeTipo	78
4.5. Propiedades de la clase RecursoTipo	79
4.6. Propiedades de la clase Secuencia	81
4.7. Tipos de conectores de secuencia	82
4.8. Propiedades de la clase Secuencia	83
4.9. Tipos de relación entre las clases de OntoProc (I)	84
4.10. Tipos de relación entre las clases de OntoProc (II)	85
4.11. Propiedades de la clase Titulacion	92
4.12. Propiedades de la clase Asignatura	92
4.13. Propiedades de la clase CursoAcademico	93
4.14. Propiedades de la clase Curso	94
4.15. Propiedades de la clase Evaluacion	94
4.16. Propiedades de la clase Planificacion	95
4.17. Propiedades de la clase Evento	96
4.18. Propiedades de la clase Material	96
4.19. Propiedades de la clase Actividad	97
4.20. Propiedades de la clase Clase	97
4.21. Propiedades de la clase Plan Docente	98
4.22. Propiedades de la clase Persona	100
4.23. Propiedades de la clase AsignacionTareaDocente	101

4.24. Propiedades de la clase AsignaturaTitulacion	102
4.25. Propiedades de la clase AsignacionEstudianteCurso	102
4.26. Propiedades de la clase EntregaActividad	103
4.27. Tipos de relación de la ontología OntoED (I)	104
4.28. Tipos de relación de la ontología OntoED (II)	105
4.29. Tipos de relación de la ontología OntoED (III)	106
4.30. Tipos de relación de la ontología OntoED (IV)	107
4.31. Tipos de relación de la ontología OntoED (V)	108
4.32. Tipos de participantes y recursos de OntoProcED	115
5.1. Propiedades de la clase Consultor en la UOC	132
5.2. Propiedades de la clase Curso en la UOC	135
5.3. Propiedades de la clase Aula en la UOC	137
5.4. Propiedades de la clase Aula en la UOC	138
5.5. Propiedades de la clase ActividadEC	143
5.6. Propiedades de la clase ActividadFinal	144
5.7. Propiedades de la clase AsignacionTareaDocente	144
5.8. Tipos de relación de la Ontología OntoUOC(I)	147
5.9. Tipos de relación de la clase AsignacionEncargoDocente	148
5.10. Tipos de relación de la Ontología OntoUOC(II)	148
5.11. Clases enumeradas de OntoUOC	154
5.12. Tipos de relación entre las clases de OntoOrg y OntoUOC	158
5.13. Clases de OntoProcED que se especializan en clases de OntoUOC	159
5.14. Clases enumeradas de OntoUOC que especializan clases de OntoProcED	160
5.15. Tipos de relación entre participantes y recursos de OntoProcED y OntoUOC	160
5.16. Propiedades de OntoUOC que especializan propiedades de OntoProcED	161
5.17. Tipos de relación entre clases de OntoOrg y OntoProcED	162
5.18. Tipos de relación entre clases de OntoOrg y OntoProcED	162
6.1. Relaciones entre OSIDs	164
6.2. Propiedades de la clase Agent	166

6.3. Propiedades de la clase Group	167
6.4. Tipos de relación de las clases del OSID Agent	168
6.5. Propiedades de la clase Assessment	170
6.6. Propiedades de la clase Assessment	171
6.7. Propiedades de la clase AssessmentPublished	172
6.8. Propiedades de la clase ObjectTaken	173
6.9. Propiedades de la clase Evaluation	174
6.10. Tipos de relación de las clases del OSID CourseManagement	175
6.11. Propiedades de la clase CanonicalCourse	179
6.12. Propiedades de la clase CourseOffering	180
6.13. Propiedades de la clase Term	181
6.14. Propiedades de la clase CourseSection	181
6.15. Propiedades de la clase CourseGroupRecord	182
6.16. Propiedades de la clase EnrollmentRecord	183
6.17. Propiedades de la clase GroupCourse	183
6.18. Tipos de relación de las clases del OSID CourseManagement	184
6.19. GradabeObjectProperties	188
6.20. Propiedades de la clase GradeRecord	188
6.21. Tipos de relación de las clases del OSID Grading	189
6.22. Propiedades de la clase ScheduleItem	191
6.23. Propiedades de la clase AgentCommitment	192
6.24. Tipos de relación de las clases del OSID Scheduling	192
6.25. Propiedades de la clase Asset	195
6.26. Propiedades de la clase Part	195
6.27. Propiedades de la clase PartStructure	196
6.28. Propiedades de la clase Record	197
6.29. Propiedades de la clase RecordStructure	197
6.30. Propiedades de la clase Repository	198
6.31. Tipos de relación del OSID Repository	199
6.32. Propiedades de la clase Message	202

6.33. Tipos de relación del OSID User Messaging	203
6.34. Propiedades de la clase Process	205
6.35. Propiedades de la clase Step	206
6.36. Propiedades de la clase Expression	206
6.37. Propiedades de la clase Work	207
6.38. Propiedades de la clase WorkEvent	207
6.39. Tipos de relación del OSID Workflow	208
7.1. Instancias para la representación de patrones de escenas	227
7.2. Instancias para la representación de escenas UOC	228
7.3. Ejemplo de aplicación de una regla	229

Índice de figuras

3.1. Delimitación del problema en base a sus tres dimensiones	53
3.2. Arquitectura ontología multinivel a nivel conceptual	55
4.1. OntoProcED como integración de OntoProc y OntoED	68
4.2. DistribuirTareasEntreProfesores	69
4.3. Secuencia que implementa DistribuirTareasEntreProfesores	71
4.4. Ontología de Procesos (OntoProc)	75
4.5. Asignar profesores a clases	76
4.6. Distribuir tareas entre profesores	80
4.7. Secuencia que implementa Distribuir tareas entre profesores	81
4.8. Jerarquía de clases y relaciones <i>is-a</i> de la ontología OntoProc	86
4.9. Vista de la clase ProcesoTipo con la herramienta Protégé	87
4.10. Ontología de Escenarios Educativos (OntoED)	91
4.11. Jerarquía de clases y relaciones <i>is-a</i> de la ontología OntoED	111
4.12. Taxonomías resultantes de la integración de OntoProc y OntoED	114
5.1. Asignar profesores a clases	118
5.2. Asignar consultores a aulas BDI	119
5.3. OntoOrgUOC como integración de OntoUOC y OntoOrg	120
5.4. Ontología de Procesos concretos (OntoOrg)	122
5.5. Ontología UOC (OntoUOC)	130
5.6. Jerarquía de clases y relaciones <i>is-a</i> de la ontología OntoUOC	151
5.7. Jerarquía de clases con raíz la clase <i>ParteCurso</i>	152
5.8. Jerarquía de clases con raíz la clase <i>MiembroCurso_uoc</i>	153

5.9.	Jerarquía de clases con raíz la clase <i>Miembro_uoc</i>	153
5.10.	Jerarquía de clases con raíz la clase <i>HerramientaDeSoporte</i>	154
6.1.	Diagrama de clases del OSID Agent	166
6.2.	Diagrama de clases correspondiente al OSID Assesment	169
6.3.	Diagrama de clases del OSID CourseManagement	178
6.4.	Diagrama de clases correspondiente al OSID Grading	187
6.5.	Diagrama de clases correspondiente al OSID Scheduding	190
6.6.	Diagrama de clases correspondiente al OSID Repository	194
6.7.	Diagrama de clases correspondiente al OSID UserMessaging	202
6.8.	Diagrama de clases correspondiente al OSID Workflow	204
6.9.	Jerarquía de clases y relaciones <i>is-a</i> de la ontología OntoOKI	210
6.10.	Composición de assessments	211
A.1.	Ontología de procesos	243
A.2.	Ontología de escenarios educativos	244
A.3.	Ontología de procesos concretos para la UOC	245
A.4.	Ontología de escenas educativas en el contexto de la UOC	246
C.1.	Componentes del lenguaje para describir escenas educativas reusables	249
C.2.	Componentes del lenguaje para describir escenas educativas complejas	250
D.1.	Escena PrepararCurso	251
D.2.	Secuencia que representa la escena compleja PrepararCurso	252
D.3.	Secuencia que representa la escena compleja PlanificarCurso	252
D.4.	Escena BuscarAsignatura	253
D.5.	Escena BuscarMatriculas	253
D.6.	Escena ObtenerTipoAsignatura	253
D.7.	Escena Contarelementoslista	253
D.8.	Escena Buscar tipos de clases	254
D.9.	Escena AsignarCodigoNoAperturaCurso	254
D.10.	Escena GenerarMensajeDeErrorAperturaCurso	254

E.1. Escena PrepararCursoBDI en el contexto de la UOC	255
E.2. Secuencia que representa la escena compleja PrepararCursoBDI en el contexto de la UOC	256
E.3. Secuencia que representa la escena compleja EstablecerAulasBDI en el contexto de la UOC	256
E.4. Secuencia que representa la escena compleja DeterminarAulasPorDefectoBDI.	257
E.5. Secuencia que representa la escena compleja SeleccionarConsultoresBDI. . . .	257
E.6. Escena AsignarConsultoresAAulasBDI.	258
E.7. Secuencia que representa la escena compleja ElaborarPlanDocenteBDI.	258
E.8. Secuencia que representa la escena compleja ActualizarPlanDocenteBDI. . . .	259
E.9. Secuencia que representa la escena compleja RevisarPlanDocenteCursoAnteriorBDI.	259
E.10. Secuencia que representa la escena compleja CrearPlanDocenteBDI.	260
E.11. Secuencia que representa la escena compleja AltaParametrosNuevoPlanDocenteBDI.	261
E.12. Secuencia que representa la escena compleja DistribuirEncargosDeConsultoriaBDI.	261
E.13. Escena PrepararAulasCursoBDI.	262
F.1. Componentes del DSL propuesto	263
F.2. Utilidad de la herramienta CASE	264

Agradecimientos

Es per mi una gran satisfacció escriure aquestes notes d'agraïment per tot el que implica, la finalització d'un treball acabat i considerat vàlid per col·legues, acadèmics i investigadors.

Més enllà d'això, a nivell personal diria que la satisfacció encara és més gran. La veritat es des de fa molt de temps he imaginat el dia en que escriuria aquests agraïments i en faria el dipòsit. Malgrat que el camí ha estat llarg i tortuós en algunes etapes, finalment sembla que ha arribat el moment desitjat i no puc dir altre cosa que em sento molt orgullosa d'haver-hi arribat.

No és fàcil fer un treball de tesi a partir de certa edat, quan no et dediques plenament a la recerca i tens obligacions familiars i professionals que no pots deixar d'atendre. Encara ho és menys quan ets cap de família monoparental i el sostén de la família, aleshores tot recau sobre una mateixa persona: l'educació dels fills, la logística del dia a dia de la casa i la feina, que cal cuidar com un tresor. Malgrat això, avui puc dir a tothom i sobre tot a aquelles dones que es trobin en una situació semblant, que la realització d'una tesi o qualsevol altre gran repte és assolible si es té la següent poció màgica: treball, sacrifici i esforç, tot això acompanyat d'un entorn adequat i estimulador. A totes les persones que formen part d'aquest entorn és a qui van dirigits els següents agraïments.

En primer lloc, la meva filla Ariadna, el millor que m'ha donat la vida i a qui dec molts moments robats. La seva personalitat tant madura, responsable i assenyada l'han convertit quasi bé en una companya. A ella agraeixo tots els ànims que sempre m'ha

donat i molt en especial la seva col·laboració en les feines de casa durant tant de temps.

Als meus pares i germana, que també han confiat en que tot arribaria a bon port i sempre m'han animat. Molt, molt en especial a la meva mare, que sempre ha estat al meu costat, ha suportat totes les tempestes d'aquest viatge i sempre m'ha fet veure el cantó bo de les coses. A la Kira, la nostra gossa, pel seu afecte i companyia incondicional. Per poder tenir-la sobre meu tantes hores i avisar-me quan havia de fer el menjar.

Pe què no dir-ho, al meu ex, per haver desaparegut de la meva vida, sense aquest fet mai hagués estat possible realitzar aquest treball. Com ell deia, no hagués suportat que fes una tesi i jo mai m'hagués decidit a emprendre aquest repte.

Un agraïment també als meus directors de tesi i especialment al Dr. Jordi Conesa, magnífic company que tantes estones m'ha dedicat, sobre tot en els moments en que semblava que no havia camí per endavant.

A la Universitat Oberta de Catalunya i en particular als Estudis d'Informàtica, Multimèdia i Telecomunicació per possibilitar la realització d'aquest treball de recerca, al IN-3 i l'eLearn Center pel seu suport. Als meus caps, per descarregar-me d'assignatures i permetre'm tenir dedicació en hores de feina, sense aquest fet no hagués pogut arrencar i per tant no m'hagués engrescat en aquesta aventura. Gràcies Josep, Rafael per confiar en mi i donar-me aquesta oportunitat. Al tutor d'aquesta tesi, en Julià, que també és el cap del grup de recerca, per liderar el grup i facilitar la recerca.

Als meus companys de feina, amb qui he comentat les ventures i desventures durant tant de temps, als que m'han ofert suport en LaTeX, en Toni, l'Àgata i en Robert; a les meves companyes i companys de despatx durant aquest anys, la Roser, l'Àgata i en Joan; als que m'han animat i s'han interessat durant els dinars i esmorzars, serien masses per anomenar-los, així que millor un agraïment conjunt a tots ells i elles. També als meus consultors, especialment, a en David Gañán que també hi ha posat el seu granet

de sorra, per ser tant responsables i donar-me pocs mal de cap. Gràcies per vostra feina ben feta, us en dec una.

A les meves amigues de tota la vida, na M.Àngels Junquera, na Susana Segura i na Montse Fernández per ser sempre a prop malgrat la distància, la feina i el temps que passa. Als bon amics i amigues de Vilanova que m'ha regalat la meva nova vida, en especial a na Judith Ventosa, na Lola Morales i na Montse Gudín que des de fa tant temps segueixen de tant a prop aquesta història, que tants ànims i suport m'han donat i que, malgrat la meva dedicació en la tesi, han seguit fent els possibles per a seguir trobant-nos i continuar la nostra amistat. Us prometo que a partir d'ara no em perdré ni una festa més i menys la comparsa;-)

Vilanova i la Geltrú, abril de 2010

Àngels Rius Gavidia

Abstract

In the last decade research in the field of e-Learning has evolved significantly, generating numerous specifications and standards. These standards, unfortunately, have not been widely applied due to their ambiguity and lack of interfaces that facilitate its use. Up to now, the research efforts in the e-Learning area have been mainly addressed to the concepts of learning object, its reusability and standardization, leaning design processes, and the development of learning management systems. In addition some specifications of the architecture of learning management systems and profiles oriented to the e-Learning systems' have been created lately in order to ease the development of e-learning applications.

The educational setting concept can be regarded as a set of activities which happens in an e-learning environment with the aim of preparing the educational environment, giving support to the learning processes and evaluating the results obtained and the competences reached during the learning process. Although the relevance of educational settings as integrator elements in e-learning, the research around it has not evolved enough. Due to the lack of a common and accepted definition of this concept and of standards to represent generic educational settings, the automation of educational settings is still in a preliminary phase.

The problem this thesis pretends to solve is the necessity of mechanisms to represent formally educational settings in order to automate, totally or partially, their implementation and promoting the reusability of the educational settings among different educational organizations. With the aim of doing a step towards the automation of educational settings, this dissertation proposes a formal framework for the specification of educational settings.

The formal framework proposed is based on an ontological structure with two conceptual levels. The first level is addressed to the representation of educational settings patterns, which can be seen as the plots that define what happens in an educational process and the actors and resources that contribute to such process. The second level focuses in the specifications of concrete educational settings. An educational setting is the concretion of an educational setting pattern in the context of a given institution and incorporates the operation mode and the normative about the educational institution considered. The patterns of educational settings defined in the first level of the framework define very generic educational processes that can be reused between organizations. These patterns can be reused in the context of a given organization to create a concrete educational setting. In that case, all the information and constraints defined in the pattern will be inherited by the new educational setting.

In order to specify easily educational settings and their patterns a graphical language based in BPMN is proposed. The reason of choosing BPMN as a basis is because of the similarities between learning and business processes. The thesis also demonstrates how to create a partial implementation of the specified educational settings automatically. In order to do so, a metamodel of the OKI-OSID specification and a set of rules that populate this metamodel from the educational settings have been created. As a result, a partial implementation of the specified educational settings in an OKI framework can be automatically created.

The presented framework has been proven to be correct, complete and useful. The correctness of the framework has been addressed formally demonstrating the consistency of the framework ontologies, its completeness has been evaluated empirically thru the instantiation of several real examples and its usefulness has been evaluated by demonstrating that is possible to generate the implementation of educational settings directly from its formal specifications.

The main contribution of this thesis is the creation of a formal framework that allows to specify patterns of educational settings, adapt these patterns to educational settings within a given educational institution and generate a skeleton of its implementation automatically.

Resumen

En la última década la investigación en el ámbito del *e-learning* ha evolucionado de forma importante, tal como demuestran los numerosos estándares y especificaciones desarrollados. Sin embargo, estos estándares no se han extendido suficientemente debido a la ambigüedad y la falta de interfaces que faciliten su uso. Principalmente, el estudio en este ámbito se ha centrado en los objetos didácticos, su reutilización y estandarización, así como en el diseño de los procesos de aprendizaje y el desarrollo de los sistemas de aprendizaje. También en los últimos años se han desarrollado algunas especificaciones sobre la arquitectura de este tipo de sistemas, así como algunos *profiles* orientados a la construcción de los mismos.

El concepto de escena educativa puede ser entendido como un conjunto de actividades que habitualmente tienen lugar en entornos educativos y que están relacionadas con el aprendizaje, ya sea para preparar dicho entorno, ofrecer soporte al proceso de aprendizaje y/o evaluar los resultados obtenidos y las competencias alcanzadas durante el proceso de aprendizaje. Este concepto, a pesar de ser un como elemento integrador del entorno de aprendizaje, constituye un terreno todavía poco explorado. Los motivos de ello son la falta de una definición consensuada sobre el mismo y la inexistencia de mecanismos que permitan la representación de escenas, limitando así las posibilidades de automatización de las mismas.

El problema planteado que este trabajo de tesis pretende resolver es la necesidad de mecanismos para representar formalmente escenas educativas orientadas a la automatización total o parcial de las mismas, promoviendo su reusabilidad entre organizaciones. Por consiguiente, este trabajo de investigación propone un marco formal para la especificación de escenas educativas que ofrezca soporte a la automatización de las mismas.

Dicho marco basado en una estructura ontológica de dos niveles desde un punto de vista conceptual. Un primer nivel para la representación formal de patrones de escenas educativas, el cual permite definir los tipos de procesos que se llevan a cabo en los entornos educativos, los actores que intervienen en ellos y los recursos que éstos utilizan. El segundo nivel en cambio está orientado a la especificación de escenas concretas. Las escenas concretas son una adaptación de los patrones de escenas al contexto de una organización de forma que se incorpora en ellas las normas y procedimientos de la organización. Los patrones de escenas del primer nivel del marco propuesto definen procesos genéricos que pueden ser reutilizados por distintas instituciones educativas y en el contexto de una misma organización para crear escenas educativas concretas. En este caso, toda la información y restricciones definidas en el patrón son heredadas por la nueva escena.

Para facilitar la especificación de escenas educativas y sus patrones se propone una notación gráfica basada en el lenguaje BPMN por las similitudes existentes entre los procesos de negocio y los procesos de aprendizaje. Por otro lado, este trabajo también demuestra que es posible la generación automática de implementaciones parciales de escenas educativas y con este fin se ha creado un metamodelo de la especificación OKI-OSID, así como un conjunto de reglas que permitan realizar la prueba. El resultado obtenido han sido implementaciones parciales de escenas educativas generadas automáticamente de acuerdo con la especificación OKI-OSID.

La validación del marco propuesto ha sido realizada según los criterios de correctitud, completitud y utilidad. La correctitud ha sido probada comprobando la consistencia de las ontologías que constituyen el marco, la completitud ha sido evaluada empíricamente mediante la instanciación de varios casos reales de ejemplo y su utilidad ha sido evaluada demostrando que es posible generar implementaciones parciales de escenas educativas a partir de sus especificaciones formales.

La contribución principal de este trabajo de tesis es la creación de un marco formal para la representación de escenas educativas que permite especificar patrones de escenas educativas, adaptar estos patrones para obtener escenas concretas dentro del contexto de las organizaciones y generar automáticamente un esbozo de su implementación.

Capítulo 1

Introducción

Dans la vie, il n'y a pas de solutions. Il y a des forces en marche: il faut les créer, et les solutions suivent.

Antoine de Saint-Exupéry

El objetivo del presente capítulo consiste en realizar una síntesis del problema a resolver en este trabajo de investigación.

Primero se centra el problema a resolver y se fijan los objetivos. Luego se enumeran las aportaciones resultantes que se obtienen de la propuesta de solución, así como la metodología seguida alcanzarla. Finalmente, se introduce la escena educativa seleccionada como caso de ejemplo, a la cual se hará referencia a lo largo de este trabajo de tesis, y se presenta su contexto.

1.1. Problemática referente a las especificaciones de escenas educativas

La obtención de especificaciones formales de escenas educativas en el contexto de una organización para ofrecer soporte a la implementación de los procesos que permiten su automatización es el problema que se plantea resolver en este trabajo de tesis.

Las escenas educativas como su nombre indica ocurren habitualmente en entornos educativos. En un entorno de aprendizaje tienen lugar todo tipo de procesos relacionados con el aprendizaje, ya sea antes de que se produzca el aprendizaje (p.e. la preparación de un curso), durante el propio proceso de aprendizaje (p.e. el envío/recepción de actividades

de aprendizaje para su evaluación) o después del mismo (p.e. valoración de resultados del proceso de aprendizaje en sí o del uso de los recursos empleados).

La investigación en tecnología educativa ha experimentado un notable avance en los últimos años. Los temas relacionados con estándares (Friesen 2005, Anido et al. 2002) y tecnologías de objetos de aprendizaje han evolucionado mucho en lo que se refiere a transportabilidad de contenidos entre plataformas (ADL 2004). También se ha fomentado la investigación en cuanto a reutilización de objetos de aprendizaje (Polsani 2003, Sicília & García-Barriocanal 2003, Quinn & Hobbs 2002) y el diseño de actividades de aprendizaje (IMS 2003*b*). Se puede decir que, en general, los esfuerzos se han concentrado en conseguir recursos de calidad reutilizables y en el desarrollo de herramientas de búsqueda de recursos automatizadas, más que en conseguir cierto grado de automatización en los sistemas de aprendizaje o *Learning Management Systems*(LMS).

La especificación IMS Digital Repository Interoperability (IMS 2003*a*) proporciona recomendaciones para la interoperabilidad de las funciones más comunes relacionadas con repositorios y propone una especificación de las mismas mediante Servicios Web. Esta implementación se centra en contenidos y, aunque reconoce un amplio rango de formatos de contenido, sistemas implementados, tecnologías y prácticas establecidas, no asume heterogeneidad de sistemas, ni de formatos, ni tampoco proporciona un lenguaje para especificar interacciones genéricas y complejas.

Tampoco existe a día de hoy una arquitectura a propósito que facilite la automatización de especificaciones de los principales escenarios educativos que ocurren en el seno de los LMS y sistemas interrelacionados, incluidos los repositorios.

No obstante, aunque en el número especial del boletín del grupo de interés LTTF de enero de 2004 (*Learning Technology newsletter* 2004) se comentaba que no existía una arquitectura para la automatización de escenarios educativos, sí que existe un modelo de arquitectura abstracta de referencia que permite definir el diseño de los LMS y de sus componentes (Farance & Tonkel 2001), así como otras propuestas de arquitecturas concretas basadas en servicios web (Eap et al. 2004, Macarro et al. 2006, Vogten et al. 2006) conformes a la especificación IMS-DRI, cuya intención es la resolución de problemas de heterogeneidad y distribución.

También existe una propuesta de arquitectura abierta y extensible para tecnología educativa, orientada específicamente a las necesidades de la comunidad de enseñanza superior, que contribuye a la reutilización de componentes de software. Se trata de la especificación OSID *Open Service Interfaces Definition* de OKI¹. Esta propuesta proporciona especificaciones para definir interfaces entre componentes de los LMS, facilitando la comunicación entre componentes y con otros sistemas con independencia de la tecnología que soporta los sistemas a integrar. Así pues, las especificaciones OSID de OKI permiten el desarrollo y la actualización de componentes de forma independiente, proporcionando interoperabilidad entre distintos sistemas y tipos de tecnología.

No obstante, esta última arquitectura de referencia propuesta no es suficiente para la especificación formal de escenas educativas cuyo fin es preparar, analizar o alcanzar un objetivo relacionado con el aprendizaje. Las interfaces propuestas por la especificación OKI-OSID describen la forma de invocar determinados servicios pero en ningún caso se especifica de qué manera interactúan los agentes que intervienen en la escena ni en qué orden deben realizarse dichas invocaciones para alcanzar un determinado objetivo. Por tanto puede considerarse que, aún siendo una especificación orientada a la reutilización de componentes de software, sólo puede contribuir parcialmente a la automatización de escenas educativas.

Por otro lado la Web Semántica (Beeners-Lee et al. 2003), propuesta por Beeners-Lee como extensión de la web actual, facilita la evaluación automática de la información. Dicha extensión está basada en la anotación semántica de la información disponible en la web, de manera que los metadatos describan el contenido y significado de la información, así como las interrelaciones entre ellos. Si además estas descripciones se formalizaran entonces se promovería el procesamiento automático de la información.

Concretamente, en la Web Semántica educativa, las anotaciones de objetos de aprendizaje reutilizables (LOR) se realizan mediante metadatos integrados en ontologías que facilitan la creación, mantenimiento y actualización de repositorios de LOR (Lytras et al. 2003). Sin embargo, algunos estudios (Friesen 2004, Najjar et al. 2002, Sicília &

¹<http://okicomunity.mit.edu/staticpages/index.php?page=OverOKI>

García-Barriocanal 2005) sostienen que la anotación semántica de los objetos de aprendizaje reusables no garantiza la automatización de escenarios educativos. El motivo es que los metadatos de los LOR han sido incorporados por sus creadores y éstos suelen ser incompletos y fragmentados, por tanto insuficientes para la construcción de elementos de software (Sicília, García-Barriocanal, Pagés, Martínez & Gutiérrez 2004).

En (Pagés et al. 2003) se menciona la necesidad de disponer de enfoques que faciliten la especificación de metadatos de LO de calidad suficiente para automatizar los procesos que ocurren en los sistemas de gestión de aprendizaje y en los repositorios de LO. A pesar de ello, aunque dispusiéramos de metadatos de calidad suficiente, la automatización de escenarios educativos posiblemente tampoco sería posible, puesto que también se requeriría un entorno adecuado que permitiera su descripción. En este sentido, algunos autores proponen la utilización de metadatos de LO como infraestructura de soporte a la implementación de funcionalidades de los LMS (Sicília & Lytras 2005).

En base a este enfoque y suponiendo metadatos de calidad suficiente, existen algunas aproximaciones para obtener especificaciones de escenas educativas orientadas a la automatización, como los perfiles de conformidad semánticos (Sicília, García-Barriocanal, Sánchez, Rius & Pagés 2004). Es una propuesta basada en el uso de metadatos provenientes de ontologías específicas para especificar contratos de procesos que implementan parte de las escenas educativas. Por tanto, se trata de definir precondiciones, postcondiciones y restricciones mediante metadatos con el fin de garantizar la correcta ejecución de los procesos que implementan tales escenas si se satisfacen los requisitos iniciales y las restricciones especificadas.

Aunque la propuesta de los perfiles de conformidad semánticos sugiere el uso de ontologías para definir formalmente los elementos que intervienen en las escenas educativas, ésta no tiene en cuenta las complejas interacciones entre participantes, ni la reusabilidad de procesos. Parece que más allá de la reusabilidad de componentes se requiere de lenguajes formales, no ambiguos que permitan describir procesos en tecnología educativa en términos sus metadatos y de las interacciones entre ellos.

Así pues, la necesidad de describir interacciones entre los participantes de escenas educativas sugiere el uso de algún lenguaje de especificación de procesos que permita

describir tales escenas. En tecnología educativa no existe ningún lenguaje para ello, sin embargo en el ámbito de la gestión de procesos de negocio existen técnicas y prácticas válidas para la especificación de procesos de negocio.

Entre los lenguaje de especificación de procesos gráficos que se utilizan previamente a la implementación de procesos destaca BPMN *Business Process Management Notation*, que es un lenguaje estándar y el más utilizado por la comunidad BPM *Business Process Management*. Este lenguaje permite describir de forma gráfica flujos de trabajo e indicar las interacciones entre los agentes que participan en él. Además, buena parte de las especificaciones tienen una correspondencia directa a código ejecutable BPEL *Business Process Execution Language*.

Incluso utilizando BPMN en el ámbito de la tecnología educativa, la automatización o semiautomatización de especificaciones de procesos en entornos educativos no sería posible. Antes, habría que resolver otros problemas previos como la inexistencia de un conjunto mínimo de procesos que implementan escenas en tecnología educativa, algún mecanismo para describir la composición y reutilización de dichas escenas y un entorno para la formalización de mismas que facilite su transcripción a código ejecutable. Solucionadas todas estas necesidades, se facilitará la automatización o semiautomatización de especificaciones de escenas en tecnología educativa a partir de sus descripciones formales.

En consecuencia, este trabajo de tesis pretende dar solución a **la necesidad de mecanismos que posibiliten la especificación formal de escenas educativas con el objetivo de ofrecer soporte a la implementación de las mismas y promover su reutilización.**

1.2. Objetivos y aportaciones originales

El principal objetivo de este trabajo de investigación es **la obtención de un marco formal que permita representar patrones de escenas educativas, adaptar dichos patrones a escenas educativas concretas en el contexto de una organización y generar una implementación parcial de las mismas.**

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- **Objetivo 1:** Proponer un lenguaje que facilite la representación gráfica de escenas educativas. Este lenguaje deberá permitir representar escenas de forma genérica (lo que llamaremos patrón) i escenas en el contexto de una organización.
- **Objetivo 2:** Obtener especificaciones formales de patrones de escenas en términos de participantes involucrados, recursos utilizados y la secuencias de procesos reutilizables para componer procesos más complejos.
- **Objetivo 3:** Obtener especificaciones formales de escenarios educativos que permitan contextualizar los procesos en entornos educativos.
- **Objetivo 4:** Obtener una integración de las especificaciones descritas en los objetivos 2 y 3 para representar patrones de escenas educativas.
- **Objetivo 5:** Obtener especificaciones formales de escenas en términos de secuencias de procesos educativos para organizaciones concretas.
- **Objetivo 6:** Obtener especificaciones formales de escenarios educativos que permitan contextualizar participantes y recursos de una organización concreta.
- **Objetivo 7:** Obtener especificaciones formales de escenas educativas adaptadas a alguna organización.
- **Objetivo 8:** Obtener un metamodelo de una parte de la especificación OKI-OSID para automatizar o semi-automatizar las especificaciones formales de escenas educativas.

La aportación fundamental de este trabajo de tesis es un **marco formal que permite representar patrones de escenas educativas, adaptar dichos patrones a escenas educativas concretas en el contexto de una organización y generar una implementación parcial de las mismas.**

Sin embargo, pueden detallarse otras aportaciones más específicas que conforman la aportación principal y que son las siguientes:

- **Aportación 1:** Una notación gráfica que permita la representación de escenas educativas tanto a nivel genérico como concreto.
- **Aportación 2:** Una ontología de procesos que permita formalizar especificaciones de procesos en términos de participantes involucrados, recursos utilizados y secuencias de procesos reutilizables para componer procesos más complejos.
- **Aportación 3:** Una ontología de escenarios educativos que facilite la especificación formal de escenarios educativos mediante la descripción de los tipos de participantes y recursos propios de un entorno educativo.
- **Aportación 4:** Un marco formal para especificar patrones de especificaciones de escenas educativas a partir de la integración de las ontologías anteriores.
- **Aportación 5:** Una ontología de definición de procesos de acuerdo a las necesidades funcionales de las distintas instituciones educativas.
- **Aportación 6:** Una ontología de una organización concreta, por ejemplo la UOC.
- **Aportación 7:** Un marco formal para especificar escenas educativas adaptadas a las necesidades de una organización concreta.
- **Aportación 8:** Una ontología de implementación basada en la especificación OKI-OSID extendida con un conjunto de reglas que permitan su instanciación de forma automática a partir del resto de ontologías del marco.
- **Aportación 9:** Una herramienta CASE que permita la representación gráfica de escenas educativas utilizando la notación referida en la aportación 1.

Algunas de estas aportaciones han sido objeto de diversas publicaciones científicas, tres en revista y cuatro en actas de congreso.

- 1) En (Rius et al. 2007*b*) se presenta una primera aproximación al catálogo de escenarios de escenarios primitivos como punto de partida para la generación de nuevos escenarios.
- 2) En (Rius et al. 2007*a*, Rius et al. 2008*a*) se justifica la necesidad de una ontología de escenarios educativos orientada a la automatización de especificaciones de escenarios en tecnología educativa y se esboza el dominio de conocimiento de la misma.
- 3) En (Sicília, García-Barriocanal, Sánchez, Rius & Pagés 2004, Rius et al. 2006, Rius et al. 2008*b*) se presentan los fundamentos de la técnica de especificación.
- 4) En (Rius, Santanach, Almirall, Conesa & García-Barriocanal 2010) se muestra una propuesta de arquitectura concreta basada en OKI que puede soportar la automatización de escenarios educativos.
- 5) En (Rius, Conesa & Gañán 2010) que presenta una herramienta CASE para asistir a la especificación de escenas educativas.

1.3. Metodología de trabajo

Con el objetivo de alcanzar los objetivos establecidos y realizar las aportaciones anteriormente mencionadas, se propuso la siguiente planificación del trabajo en etapas:

- **Estudio del estado de la cuestión:** Se estudia el estado de la cuestión acerca del concepto de escenario educativo y otros conceptos relacionados, así como recomendaciones y estándares aplicables a los diferentes elementos que intervienen en un escenario. También se tratan distintas técnicas de especificación de procesos y lenguajes de especificación ejecutables utilizados en entornos B2B, y más concretamente aquellos que son aplicables en el ámbito de tecnología educativa. Por último, se analizan las ontologías como herramienta de definición formal de procesos en tecnología educativa.

- **Planteamiento del problema:** El problema se enmarca en el contexto de las especificaciones de escenas que habitualmente tienen lugar en los LMS y otros sistemas relacionados. Se parte de la consideración de las distintas carencias de la tecnología educativa para la automatización de especificaciones ejecutables de procesos de aprendizaje, haciendo énfasis en aquellas que impiden la obtención de especificaciones de escenarios educativos automatizables o semiautomatizables. A partir de éste análisis, se identifican los principales problemas a resolver y se determina la estrategia de resolución para obtener una solución a dicho problema. Además, se presenta la escena a utilizar como caso de ejemplo para la validación de la solución propuesta.
- **Definición de un marco formal:** Se define un marco formal para soportar la representación de patrones de escenas, su adaptación a escenas concretas en el contexto de una organización y la generación de implementaciones parciales de las mismas. Este marco es soportado por una arquitectura ontológica que se describe a tres niveles: 1) un primer nivel como metamodelo para la descripción de patrones de escenas, 2) un segundo nivel para obtener escenas concretas adaptadas a una organización a partir de los patrones de escenas proporcionados en el nivel anterior y 3) un tercer nivel para la transcripción de las especificaciones formales obtenidas en los anteriores niveles en términos de su implementación.
- **Validación de la propuesta:** La validación del marco formal propuesto se realiza desde distintos niveles. Se comprueba que es correcto, completo y útil. Para validar la correctitud de la propuesta se comprueba la consistencia de las especificaciones utilizando algún tipo de validación automática; la completitud se valida mediante la representación de diversos casos de ejemplo y la utilidad por medio de la generación de implementaciones parciales de las escenas educativas representadas mediante este marco a modo de prueba de concepto.
- **Formulación de conclusiones.** Se concluye con un análisis del grado de cobertura de los objetivos alcanzados, y se detallan las líneas futuras de investigación y ampliación del trabajo realizado.

1.4. Caso de ejemplo elegido

A lo largo de este trabajo de tesis se hace referencia a una escena educativa dentro de una organización concreta. La escena en cuestión es la preparación de un curso de Bases de Datos I en la UOC (Universitat Oberta de Catalunya) que se presenta en detalle en el apéndice D.

Dicha escena se desarrolla periódicamente en la UOC cada inicio de curso académico. El objetivo de la misma consiste en definir, organizar y preparar las aulas donde se va a impartir la asignatura Bases de Datos I el próximo semestre. El resultado tangible de esta escena es que el curso esté preparado para su impartición y que las aulas se encuentren a punto para recibir los estudiantes el primer día de curso.

Esta escena tiene lugar en cualquier institución educativa justo antes de que se inicie el curso. Sea cual sea la institución y las particularidades del contexto ésta se concreta de forma distinta.

En consecuencia se puede decir que existe un patrón de escena educativa que describe como es la preparación de un curso en general y que este se adapta de forma distinta para cada organización y contexto, puesto que el escenario educativo es diferente y se aplican normas y procedimientos específicos de cada institución.

Entre las muchas escenas que podían haberse seleccionado, se ha escogido la preparación de un curso de Bases de Datos I en la UOC por los siguientes motivos:

- 1) Es suficientemente compleja como para ser descompuesta en otras escenas, lo cual implica la existencia de procesos formados por otros procesos más sencillos que permiten probar la secuenciación y reutilización de especificaciones.
- 2) Es representativa puesto que se lleva a cabo en cualquier entorno de aprendizaje, siendo en el entorno de enseñanza virtual donde requiere mayor sistematización y planificación anticipada, lo que lo hace más interesante de mecanizar.
- 3) En ella intervienen variedad de participantes y gran número de recursos, llevándose

a cabo tareas muy variadas, lo cual permite realizar una prueba suficientemente amplia.

- 4) Se concreta para el caso de la UOC por ser una institución pionera en docencia virtual, que dispone de un conjunto de normas y procedimientos establecidos, bien definidos y probados. Mediante este contexto organizativo se probará la adaptabilidad de patrones de escenas educativas.
- 5) La autora/doctoranda está vinculada a la UOC desde 2001, lo que le confiere amplia experiencia en docencia virtual y profundo conocimiento sobre los procedimientos organizativos de esta institución en materia docente. Así mismo cuenta con más de diez años de experiencia en la misma asignatura en entorno de enseñanza presencial.

1.5. Estructura del documento

Tras esta introducción al trabajo de tesis, a continuación se presenta la estructura de la memoria con el fin de facilitar su lectura.

En el capítulo 1 se introduce este trabajo de investigación. Se plantea la problemática y se formula brevemente el problema a resolver. Se presentan también los objetivos y aportaciones del mismo, así como la metodología de trabajo empleada y el caso de ejemplo elegido al cual se va a hacer referencia a lo largo de este documento.

En el capítulo 2 se describe el estado de la cuestión. Se aborda el concepto de escenario educativo y otros conceptos relacionados, así como las recomendaciones de especificaciones y estándares aplicables a ellos. También se tratan diferentes técnicas de especificación de procesos en entornos B2B (*Business To Business*) que son aplicables en el ámbito de tecnología educativa y los lenguajes de especificación de escenas ejecutables existentes. Además, se revisan las arquitecturas de soporte y el uso de las ontologías como herramienta de definición formal de procesos en tecnología educativa.

El capítulo 3 describe el problema de forma detallada. Se presenta su motivación y se concreta el problema en cuestión delimitando su alcance. Se presenta la estrategia de resolución para abordarlo y se describe con más detalle el caso de ejemplo.

En los capítulos 4, 5 y 6 se presentan la mayor parte de las aportaciones de este trabajo de investigación.

Concretamente, en el capítulo 4 se describe el primer nivel del marco formal propuesto. Se trata de una ontología de procesos para entornos educativos, válida en cualquier entorno de enseñanza superior con independencia de la institución y de la tecnología del sistema de aprendizaje que ésta utilice. Su objetivo consiste en describir patrones de escenas educativas mediante secuencias de procesos reutilizables que se desarrollan en entornos educativos.

En el capítulo 5 se describe el segundo nivel del marco formal desarrollado. En él se presenta una ontología para adaptar patrones de escenas educativas a escenas concretas de una organización, en este caso la UOC, mediante participantes, recursos, normas y procedimientos propios de la organización.

Posteriormente en el capítulo 6, se presenta un metamodelo de parte de la especificación OKI-OSID. A partir de él se construye una ontología de implementación que permitirá traducir las descripciones formales obtenidas a partir de las ontologías de los niveles anteriormente descritos en términos de OKI-OSID.

En el capítulo 7 se presenta la verificación del marco formal propuesto para la obtención de descripciones formales de escenas educativas que puede resultar útil para la generación de implementaciones parciales de las mismas.

Finalmente, en el capítulo 8, se exponen las conclusiones del trabajo realizado y se esbozan algunas líneas futuras de investigación relacionadas.

1.6. Resumen

Se ha presentado una síntesis del trabajo de tesis descrito en esta memoria y la estructura de la misma.

Capítulo 2

Estado de la cuestión

Sólo sé que no sé nada.

(Sócrates)

En este capítulo se revisa el estado actual de las investigaciones y trabajos desarrollados acerca del concepto de escena educativa, las especificaciones de escenas y la automatización de dichas especificaciones. La finalidad del capítulo es ubicar el problema en contexto y obtener una relación de las limitaciones y carencias de la tecnología educativa que actualmente impiden la resolución del problema planteado en este trabajo de tesis.

2.1. Tecnología educativa

El nacimiento de Internet y la expansión de la World Wide Web han dado lugar a nuevas opciones de comunicación para la sociedad en red. Las Tecnologías de la Información y la Comunicación (TIC) constituyen el soporte para la realización de nuevas actividades que sin la red serían impensables. En el terreno educativo las TIC también han aportado una nueva forma de educación estrechamente ligada a Internet (Wagner 2000), conocida como *e-Learning*.

Tecnología educativa o *e-Learning* son términos que se utilizan frecuente e indistintamente a pesar de no existir una definición consensuada sobre dichos conceptos. Etimológicamente, el término *e-learning* se compone de dos partes: el prefijo 'e-' del inglés *electronic* y el término *learning* que representa el proceso de desarrollo de conocimientos o aprendizaje. Así pues, por el origen de este término no debe de extrañar que el ámbito

del *e-learning* se extienda a dos disciplinas bien diferenciadas: las referidas a los procesos de enseñanza propiamente dichos como la pedagogía, la sociología y la psicología, por una parte, y las relativas a los medios informáticos y de comunicación utilizados para la transmisión de contenidos en el proceso educativo como la informática y las telecomunicaciones, por otra.

Actualmente no existe una definición universalmente consensuada sobre el término *e-learning*, sino que existen tantas como campos de interés del autor o corporación que la propone. De forma general, el término *e-Learning* se emplea para hacer referencia a cualquier aplicación de las tecnologías de la información en los procesos de aprendizaje, pero existen otras definiciones:

“Se entiende por *e-Learning* el uso de las Tecnologías de la Información y de la Comunicación (TIC) para proveer un amplio abanico de soluciones destinadas a facilitar, compartir y mejorar el proceso de enseñanza-aprendizaje” (Bates & Poole 2003)

“Término que cubre un amplio rango de aplicaciones y procesos, tales como el aprendizaje basado en la Web, el aprendizaje basado en el ordenador, aulas virtuales y colaboraciones digitales. Incluye el envío de contenidos por Internet, Intranet/Extranet (LAN, WAN), audio y videocassette, vía satélite, TV interactiva y CD-ROM.”¹

“*e-Learning* es educación *just-in-time* integrada en cadenas de valor de alta velocidad. Consiste en el reparto de contenido de aprendizaje individualizado, comprensivo y dinámico en tiempo real, soportado por comunidades de desarrollo de conocimiento que establecen un vínculo entre estudiosos y practicantes con los expertos” (Drucker 2000)

Esta última definición de tecnología educativa resalta el valor del conocimiento en la sociedad actual, así como la existencia de cadenas de valor que generan conocimiento y facilitan el aprendizaje, adaptándose ágilmente a una realidad cambiante. Parte del hecho

¹<http://www.learningcircuits.org/glossary>

de que atrás queda ya la era industrial y que nos adentramos en la era del conocimiento, situando por tanto al mismo nivel de importancia la necesidad de producir productos que la de generar conocimiento o aprendizaje.

En una línea parecida, otros autores han establecido una comparación entre los ciclos de vida de los procesos de negocio y los procesos de aprendizaje sugiriendo la reutilización de experiencias y técnicas probadas dentro del área de negocio, conocida como BPM (Business Process Management), en el ámbito de la tecnología educativa (Helic et al. 2005).

En la definición de Drucker también se remarca la necesidad de vincular a quienes aprenden y a quienes instruyen, considerándolos respectivamente consumidores de conocimiento y productores del mismo, análogamente lo que ocurre en área de negocio entre consumidores y proveedores de productos o servicios. Este paralelismo no es de extrañar si tenemos en cuenta que la tecnología BPM es resultado de la convergencia de la integración de aplicaciones de empresa o *Enterprise Application Integration* (EAI) y la tecnología Web, del mismo modo que la tecnología educativa es resultado de la confluencia entre la integración de aplicaciones necesarias para soportar la educación on-line y la tecnología Web.

Por tanto, la tecnología educativa puede verse como un caso particular de la tecnología BPM y, en este sentido, podemos plantearnos tomar como punto de partida para nuestra investigación los trabajos realizados acerca de la automatización de procesos de negocio y considerar los LMS (*Learning Management Systems*) o plataformas de *e-learning* como sistemas basados en la Web que facilitan la adquisición de aprendizaje y la interacción entre proveedores de conocimiento y consumidores del mismo.

El IEEE Learning Technology Standards Committee (LTSC) define el LMS como:

“Un sistema especializado en tecnología educativa basado en Internet y en las tecnologías WWW, que proporciona educación y enseñanza siguiendo el paradigma de aprendizaje abierto a distancia” (Farance & Tonkel 2001).

Otra definición más concreta que la anterior y que va más allá de enunciar el objetivo final de este tipo de sistemas es la siguiente:

“Aplicación software basada en tecnología Web utilizada para planificar, implementar y asesorar en un entorno de aprendizaje. Típicamente un sistema de gestión de aprendizaje proporciona un instructor que puede crear o enviar contenido y un entorno donde los estudiantes participan y reciben asesoramiento con la finalidad de mejorar su rendimiento. Un LMS también puede proporcionar a los estudiantes la posibilidad de interactuar a través de zonas de debate, video-conferencia o foros de discusión²”.

A partir de esta definición resulta claro que el objetivo principal de este tipo de sistemas es favorecer el progreso académico de los estudiantes, y con esta finalidad los sistemas de aprendizaje *on-line* disponen de funcionalidades para el soporte de tareas de aprendizaje individuales, así como otras que afectan a grupos de estudiantes, por ejemplo la creación de aulas y asignación de instructores a cursos, la gestión de buzones para el envío de actividades de evaluación y de espacios virtuales que favorecen la interacción de los estudiantes a través del intercambio de opiniones y conocimientos.

En este sentido y destacando las funcionalidades del LMS, Kaplan-Leiderson define el LMS como:

“Software que automatiza la administración de eventos de aprendizaje. El LMS registra usuarios, cataloga cursos y almacena datos de estudiantes, a la vez que permite gestionarlos. Un LMS generalmente es capaz soportar cursos de múltiples publicadores y proveedores. Habitualmente no incluye herramientas de autoría, en lugar de ello se centra en la gestión de cursos creados por una amplia variedad de fuentes³”.

Así pues es fácil darse cuenta de que, aparte de las tareas propiamente dirigidas al aprendizaje y/o adquisición de competencias que son en las que se focaliza este tipo de sistemas, se precisa de otras funcionalidades para posibilitar el desarrollo de la actividad académica. Por ejemplo, funciones de gestión de estudiantes o usuarios del sistema, catalogación de cursos, gestión de aulas, de competencias, de conocimiento, gestión de

²<http://searchcio.techtarget.com/glossary>

³<http://www.learningcircuits.org/glossary.html>

servicios de comunicación (foros de discusión o videoconferencias), tutorización, generación de informes sobre la evolución del proceso de aprendizaje y evaluación de los estudiantes entre otros.

La gran diversidad de tareas que un LMS debe de llevar a cabo ha dado lugar a la creación de herramientas o sistemas complementarios que amplían las funcionalidades de la plataforma de *e-learning*. De acuerdo con (Paulsen 2002) un sistema de aprendizaje *on-line* es el conjunto de sistemas que soportan la educación *on-line*, ejemplos de ellos son: las herramientas de creación de contenidos, de autoría, de evaluación, los sistemas de gestión de contenidos de aprendizaje, proveedores de servicios de aprendizaje, sistemas de gestión de estudiantes, sistemas de planificación de recursos, sistemas de cuentas y por supuesto, el LMS.

Entre las plataformas de sistemas de *e-learning* más conocidas destacan Blackboard⁴, SAKAI⁵, MOODLE⁶ y Claroline⁷. Aunque los LMS actuales son sistemas concebidos para realizar tareas concretas relacionadas con el aprendizaje previamente determinadas, a día de hoy algunos autores hablan de una nueva generación de LMS basados no en el aprendizaje, sino en el propio proceso de aprendizaje con cierto carácter dinámico, lo que permite hablar de sistemas de aprendizaje adaptativos (Helic et al. 2005).

En cuanto al diseño e implementación de los LMS algunos autores sostienen que se trata de sistemas muy difíciles de desarrollar partiendo desde cero, ello es debido a que incorporan gran variedad de componentes organizacionales, administrativos, instruccionales y tecnológicos, por lo que aconsejan en la construcción de este tipo de software la reutilización de componentes y su composición (Carlson 1998, Moore & Kearsley 1996).

Puesto que los sistemas de aprendizaje deben gestionar recursos educativos como cualquier otro sistema de gestión e implementar procesos, concretamente destinados a facilitar la adquisición de conocimiento y competencias, no es de extrañar que para este tipo de sistemas el concepto de *learning objects (LO)* para referirnos a los datos que maneja el LMS resulte relevante, así como el concepto de escena educativa o *educational*

⁴<http://www.blackboard.com/us/index.Bb>

⁵<http://www.sakaiproject.org>

⁶<http://moodle.org>

⁷<http://www.claroline.net/>

setting (ES) para hacer referencia a las actividades que se llevan a cabo en entornos educativos y que pueden implementarse como funcionalidades del LMS.

Entre estos dos conceptos el más estudiado sin duda alguna ha sido el concepto de LO, prueba de ello son los abundantes resultados obtenidos como consecuencia de los esfuerzos de investigación acerca de los LO, su reusabilidad y estandarización (Polsani 2003, Sicilia & García-Barriocanal 2003, LTSC WG12 2002). En cambio en lo que respecta al concepto de escenario educativo o escena educativa y sobre todo desde su punto de vista más tecnológico, apenas existen trabajos de investigación acerca de él.

Dada la importancia del concepto de objeto de aprendizaje y la estrecha relación que mantiene con el concepto de escena educativa, a continuación se presentan algunas definiciones del concepto *learning object* y en la siguiente sección del concepto de escena educativa.

El concepto LO como acrónimo de *Learning Object* empezó a popularizarse en 1994 de la mano de Wayne Hodgins. Desde entonces ha acaecido un término clave en lo que se refiere a creación y agregación de contenidos en el ámbito de la tecnología educativa, sin embargo actualmente no hay unanimidad en lo que respecta a su definición. Algunas definiciones de LO son las siguientes:

“Una pieza de contenido de menor tamaño que un curso” (Friesen 2005)

“Objeto digital es cualquier recurso digital que puede ser reutilizado como soporte al aprendizaje” (Wiley 2002)

“Una entidad digital o no digital, que puede ser usada o referenciada en el aprendizaje, educación o enseñanza” (LTSC WG12 2002)

“Trazos independientes de contenido educativo que proporciona una experiencia educacional para un propósito pedagógico” (Quinn & Hobbs 2002)

“La experiencia estructural más pequeña e independiente que contiene un objetivo, una actividad de aprendizaje y una orientación” (L’Allier 1997)

“Unidad de aprendizaje independiente, autocontenida predispuesta para su reutilización en múltiples contextos educativos” (Polsani 2003)

Aunque son muchas y variadas las definiciones de LO, existe consenso en cuanto a requisitos mínimos. Polsani, en (Polsani 2003) establece que los LO deben ser etiquetados con metadatos para poder ser almacenados y referenciados en una base de datos, un vez creados los LO deben poderse utilizar en distintos contextos de aprendizaje y además deben de ser independientes del medio de transmisión y de los sistemas de gestión del conocimiento. Esto es, a los LO se les exige por lo menos cierta accesibilidad, reusabilidad e interoperabilidad.

También según Polsani (Polsani 2003) un LO se considera reusable y se denomina *Learning Object Reusable (LOR)* cuando puede utilizarse en tantas ocasiones como se precise habiéndolo creado una sola vez. Para ello se requiere que dicho objeto de aprendizaje sea creado definiéndolo en base a dos aspectos: la forma para su presentación en el medio y el interfaz como mecanismo de relación entre el usuario y la memoria en la que se aloja el recurso. Los objetos así diseñados con una clara orientación al aprendizaje son denominados objetos de aprendizaje reusables.

La reutilización de los LO por parte del LMS requiere en primer lugar de la definición del LO como unidad autocontenida y modular, reusable e interoperable. Una vez definido puede ser publicado y localizado para su uso. Para ello deberá de ser almacenado en algún repositorio de acuerdo con algún criterio que facilite su localización y posterior uso, por lo cual se se utilizan metadatos con el fin de anotar semánticamente dichos objetos de aprendizaje.

Los metadatos se definen como “datos sobre datos”. Según Nilson se caracterizan por: 1) ser datos objetivos sobre datos, 2) producirse una sola vez por recurso, 3) tener una semántica bien definida y 4) ser descritos por documentos de metadatos. Estas anotaciones, de acuerdo con Stojanovic, se utilizan para la descripción, definición de contenido y la estructura de materiales didácticos reutilizables (Stojanovic et al. 2001) y dotan al LMS de capacidad de razonamiento, lo promueve la inferencia de nuevo conocimiento.

Si bien existe abundante bibliografía sobre reusabilidad de objetos de aprendizaje (Sicília, García-Barriocanal, Pagés, Martínez & Gutiérrez 2004, Sicília & García-Barriocanal 2003, Najjar et al. 2002) como demuestran la variedad de especificaciones

y estándares desarrollados para su estandarización, no ocurre lo mismo con otros conceptos del ámbito de la tecnología educativa: el concepto de escenario educativo, el de sistema de aprendizaje (Avgeriou et al. 2003) o el de escena educativa que a menudo se confunde con el concepto de escenario educativo.

2.2. Escenas educativas

Esta sección se centra en los conceptos de escenario educativo y escena educativa y se estudian ambos desde el punto de vista de sus especificaciones y estándares relacionados.

En primer lugar, se revisan algunas definiciones sobre el concepto de escenario educativo, puesto que no se han encontrado definiciones para el concepto de escena educativa. En segundo lugar, se estudian las técnicas y lenguajes de especificación existentes, especialmente aquellas que están orientadas hacia la automatización y posteriormente se revisan las especificaciones y estándares relacionados con escenas educativas.

2.2.1. El concepto de escenario educativo

El diccionario define escenario como “un resultado esperado o natural del curso de eventos”, aunque existen otras definiciones más concretas en función de la disciplina que lo defina. Sin embargo, desde un punto educacional o pedagógico un escenario educativo puede definirse como:

“Un ejemplo de actividad de clase que refleja estrategias de reforma basadas en estándares y lenguajes de instrucción universales basados en el aprovechamiento de conocimientos y competencias adquiridas” (Tood et al. 1999)

Desde una perspectiva más tecnológica, centrada en la ingeniería del software, un escenario educativo puede considerarse como:

“Una herramienta flexible para el diseño de sistemas que no tiene una única forma de ser usada. Constituye un nexo entre la funcionalidad de los

LMS y los procesos de *e-learning* así como entre la arquitectura y la implementación de componentes, por lo que el concepto de interoperabilidad cobra relevancia desde este punto de vista.” (Toffolon 2006)

Sin embargo desde un punto de vista organizacional el concepto de escenario puede entenderse vinculado a la toma de decisiones o a la gestión estratégica de las organizaciones y puede definirse del siguiente modo:

“Son proyecciones de un potencial futuro por lo que describen narrativamente cómo ciertos aspectos pueden evolucionar. Un escenario está compuesto por los siguientes elementos: las fuerzas que lo motivan, las condiciones y circunstancias que prevalecen al finalizar el escenario, lo que debe de ocurrir para que se alcance el objetivo final del mismo y la explicación lógica o racional de lo que ocurre.” (Fahey & Randall 1997)

Así pues, no existe una definición comúnmente consensuada sobre este concepto, por lo que se detecta un problema de heterogeneidad semántica acerca del concepto de escenario educativo.

2.2.2. Especificación de escenas educativas

En este apartado interesa profundizar en la especificación de escenas educativas. Puesto que los escenas pueden considerarse una primera aproximación al comportamiento de un sistema, se considera interesante revisar los lenguajes y técnicas de especificación de escenas desde el punto de vista de la ingeniería del software. También, dada la estrecha relación existente entre procesos de *e-Business* y procesos de *e-Learning* y la experiencia acumulada en el ámbito de la automatización de sistemas de negocio basados en Internet, se presentan algunas técnicas y lenguajes de especificación de procesos de negocio.

La necesidad de documentar escenas, en el campo de la ingeniería del software, ha originado numerosas técnicas de descripción de escenas. Actualmente existen diversidad de técnicas, las cuales van desde las puramente narrativas hasta las más gráficas, siendo las mixtas las preferidas por la mayoría de autores (Uchitel 2003). Dado que para nuestro

propósito interesa la especificación de escenas educativas como punto de partida para su procesamiento automático, descartamos de entrada las notaciones textuales por ser poco rigurosas y nos interesamos por las mixtas y gráficas, así como las formales.

Según el diccionario el concepto de escena se define como una secuencia de eventos (Pearsall & Hanks 1998) o de actividades (Carroll 1995) que se refieren a la interacción entre entidades independientes, por lo que los lenguajes de especificación de escenas deben de proporcionar mecanismos que describan secuencias de eventos e interacciones entre los agentes que intervienen en ellos.

Una de las técnicas más usadas para describir escenas, la cual data de finales de los 80, ha sido los diagramas de secuencia o (*sequence charts*). Estos diagramas permiten modelar las interacciones entre agentes que intervienen en la escena (usuario o sistema) mediante el envío y recepción de mensajes, así como la secuencia de eventos que configuran la escena y su ordenamiento. La notación empleada en este tipo de diagramas identifica cada agente involucrado en la escena como una línea vertical y las interacciones entre ellos se dibujan mediante flechas dirigidas que van del emisor al receptor, siendo el flujo de ellas el que determina el orden de la secuencia de eventos.

Extensiones de esta notación permiten distinguir entre diferentes tipos de mensajes según si la comunicación es síncrona o asíncrona o emplear estructuras de datos de tipo cola para imponer restricciones o eventos de orden. Sin embargo, aún tratándose de una notación bastante sencilla, capaz de representar los aspectos fundamentales de las escenas, esta técnica tiene un gran inconveniente desde el punto de vista de la ingeniería de sistemas: únicamente ofrece una visión parcial del comportamiento del sistema.

Para solucionar este problema pueden utilizarse distintos mecanismos de combinación de diagramas de secuencia y obtener así una visión más general del comportamiento del sistema a especificar:

- a) Combinación: se basa en la descomposición de escenas complejas en otras más sencillas que pueden dibujarse sin conocer su implementación, simplemente conociendo su funcionalidad. La composición permite combinarlas en vertical (una escena empieza cuando otra acaba), horizontal (varias escenas pueden ocurrir en paralelo) o alternativamente (se descompone una escena en operaciones más sencillas que

pueden realizarse simultáneamente). El objetivo es que las escenas encajen de una manera u otra configurando la escena inicial.

- b) Identificación: se fundamenta en la identificación de componentes o estados del sistema comunes, representando el comportamiento de los componentes del sistema como una máquina de estados. Para identificar los distintos estados de una escena se pueden emplear dos métodos: solicitar a los usuarios que identifiquen estos estados en base a su experiencia o asignar a cada estado una regla que lo defina. Esta claro que el uso de reglas es mucho más útil cara a la automatización y además enriquece las descripciones de escenas con información adicional, pudiendo definir el comportamiento desde el punto de vista del dominio de conocimiento que más interese.
- c) Disparadores o precondiciones: se proporciona información acerca de las condiciones que deben darse para que una escena se desarrolle, en lugar de establecer relaciones entre escenas. La precondición puede venir expresada en lenguaje natural, en lógica temporal o en OCL (*Object Constraint Language*), por citar ejemplos. Los disparadores en cambio permiten expresar el contexto de las escenas independientemente de su existencia, lo que resulta útil como método de especificación de relaciones entre escenas y no para la especificación de escenas como un todo.

Los diagramas de secuencia han servido de base para otras técnicas de especificación gráficas más actuales como es el caso del diagrama de secuencia de UML (*Unified Model Language*) (OMG & BPMI.org 2006). UML es el lenguaje de especificación estandar definido mediante el metamodelo UML de OMG (*Object Management Group*) y como tal ofrece distintas técnicas y notaciones para modelar sistemas, tanto a nivel estructural (estático) como a nivel de comportamiento (dinámico). Por su relación con la especificación de escenas, nos interesan los siguientes:

- a) Diagrama de secuencia: basados en los *sequence charts* modelan la colaboración entre objetos en base a una secuencia temporal de eventos. Heredan su problemática: son útiles para modelar el comportamiento de sistemas sencillos o escenas individuales, pero no sirven para la modelización de sistemas complejos, cuyo

comportamiento depende de varias escenas a la vez, por ejemplo cuando la comunicación entre objetos es importante o muy variable, existen muchas excepciones que provocan saltos en la secuencia de mensajes, interacciones con problemas de sincronización, etc.

- b) Diagrama de casos de uso: permiten una visión funcional del sistema o escena en términos de agentes involucrados, sus objetivos (casos de uso) y dependencias entre casos de uso. Sirven pues para representar casos especiales que el diagrama de secuencia no recoge, por tanto constituyen una herramienta para particularizar diagramas de secuencia a nivel de instancias o de roles concretos.
- c) Diagrama de estados: son esencialmente los diagramas de Harel con notación estandarizada, capaces de modelar tanto procesos automatizados como procesos de negocio en base al concepto de máquina de estados. Su interés reside en la posibilidad de indicar que un estado individual, que describe una condición o evento parcial dentro un componente del sistema, puede repetirse y ser reutilizado dentro del sistema global. Además, utilizando ciertas extensiones, permite la representación de múltiples estados de forma simultánea dentro del sistema. Sin embargo, para este tipo de diagramas cómo la información relativa a estados afecta a la composición de distintas escenas es un tema aún no resuelto.
- d) Diagrama de colaboración: describen interacciones entre elementos de una escena o componentes del sistema. A diferencia de los diagramas de secuencia que describen el comportamiento del sistema enfatizando en la secuencia de eventos que ocurren, los diagramas de colaboración ponen el foco en la estructura del sistema/escena o de sus componentes.
- e) Diagrama de actividad: tiene su origen en el diagrama de flujo de tareas, conocido como *workflow*. Un *workflow* aplicado a una organización permite representar la operativa del negocio paso a paso, permitiendo la representación de patrones de actividades que se repiten por rol o escena. Este tipo de diagramas, por tanto,

facilita la documentación de las actividades que se llevan a cabo en una organización. La formalización de los diagramas de actividad ha dado lugar a las redes de Petri.

En el área de los negocios se define BPM *Business Process Management* como una metodología empresarial cuyo objetivo es mejorar la eficiencia de la gestión sistemática de procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. En éste ámbito, algunos autores han propuesto el uso de *workflows* para modelar y automatizar procesos de negocio.

Un *workflow* según *Workflow Management Coalition* es la automatización de procesos de negocio, completa o parcial, durante la cual se transmiten documentos, información y tareas entre participantes de acuerdo con un conjunto de reglas procedurales (Coalition 1997). Para definir workflows algunos autores proponen el uso de lenguajes de patrones (Van der Aalst, Hofstede & Weske 2003) y otros, incluso los proponen para la construcción de los LMS (Van der Aalst, Hofstede, Kiepuszewski & Barros 2003), bajo la hipótesis de similitud entre sistemas de negocio basados en la web y sistemas de aprendizaje basándose en la complejidad de este tipo de software y al gran número de usuarios que acceden a ellos para proveerse de servicios.

Otros autores como White, han estudiado la representación de *workflows* mediante diagramas de actividad de UML 2.0 y diagramas de proceso de negocio en BPMN (*Business Process Modelling Notation*).

BPMN es un notación gráfica estandarizada para describir procesos de negocio dentro de un *workflow*. BPMN ha sido desarrollado por *Business Process Management Initiative* y actualmente está siendo mantenido por OMG desde que las dos organizaciones se fusionaron en 2005 (OMG & BPML.org 2006). Existe una version 1.1 y a día de hoy existe ya una propuesta para la versión 2.0.

El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente comprensible para todos los usuarios del ámbito de los negocios, desde el analista que crea el esbozo de los procesos de negocio a los desarrolladores responsables de la implementación de la tecnología que soportará dichos procesos, así como los usuarios del negocio que gestionarán y monitorizarán dichos procesos. Además, BPMN es soportado

por un modelo interno que permite la generación de código ejecutable, por lo que una de las grandes aportaciones de BPMN es la reducción del *gap* existente entre el diseño de procesos de negocio y la implementación de los mismos, aspecto que no se daba en el resto de notaciones previamente mencionadas (White 2004).

La especificación BPMN (White 2004, OMG & BPMI.org 2006) incluye la definición de los diagramas de proceso de negocio (*Business Process Diagram*) como notación gráfica para el modelado de objetos y su interacción, la semántica de cada uno de los elementos que intervienen de esta notación y la posibilidad de formalizar tales descripciones utilizando BPEL4WS (*Business Process Execution For Web Services*) como lenguaje de especificación ejecutable. Con el fin de ser usada por el máximo número de usuarios y facilitar la definición de procesos mejorando aspectos relacionados con la colaboración y las transacciones de negocio, recopila y consolida las mejores ideas provenientes de otras notaciones (p.e. UML, RosettaNet, ebXML, EPCs, etc) en una única y simple notación que, además, debe de ser capaz de representar la complejidad de los procesos de negocio. BPMN puede utilizarse para especificar tres tipos de submodelos BPMN: procesos privados o internos, procesos públicos o abstractos y procesos globales o colaborativos.

- Procesos privados (internos): representan flujos de tareas o *workflows* dentro de la organización y por tanto no existe interacción con elementos externos. Pueden ser mapeados en un simple proceso BPEL.
- Procesos públicos (abstractos): representan las interacciones entre un proceso interno y otro proceso, mostrando de este último únicamente los puntos de conexión con el proceso interno. Pueden mapearse en un proceso BPEL abstracto.
- Procesos colaborativos (globales): definen las interacciones entre dos o más entidades de negocio. Estas interacciones se definen como una secuencia de actividades que representa el intercambio de patrones de mensajes entre las entidades involucradas. Un proceso de colaboración puede ser mapeado en varios lenguajes de colaboración (p.e. ebXML, RosettaNet) o bien considerado como varios procesos abstractos que se comunican con otro proceso.

A nivel de especificación de procesos los diagramas BPMN permiten representar distintas categorías de elementos: flujos de objetos (eventos, actividades y conectores para la convergencia o divergencia de múltiples flujos de secuencia), conectores de objetos (flujo de secuencia, flujo de mensajes, asociaciones), agrupación de elementos (por actividades de un mismo participante o categorización de actividades) y elementos para aportar información adicional acerca de los procesos (objetos de datos, grupos, anotaciones).

Esta notación resulta muy adecuada como herramienta de especificación de procesos de negocio a diferentes niveles de abstracción y desde distintos puntos de vista, facilitando a su vez su interpretación semántica, así como su mapeo a un lenguaje ejecutable. Además, por si fuera poco aporta otras ventajas que fomentan la interoperabilidad entre LMS: 1) soporta el intercambio de formatos entre aplicaciones por ser XML serializable lo que fomenta la interoperabilidad y 2) reduce la heterogeneidad entre LMS (Mendling & Neumann 2005). Por tanto, en el área del BPMN existen mecanismos para la descripción de procesos de negocio orientadas a la automatización de los mismos (OMG & BPML.org 2006), en cambio en tecnología educativa no existe ningún mecanismo con este mismo fin.

Existe pues se detecta la carencia de mecanismos que permitan describir escenas educativas, no sólo a nivel estático (componentes, interrelaciones entre ellos y/o restricciones), sino también a nivel dinámico (comportamiento del sistema y de otros agentes involucrados durante la ejecución del mismo) de forma que puedan definirse patrones de comportamiento para los escenas educativas cuyas especificaciones se desea automatizar.

2.2.3. Especificaciones y estándares relevantes para el concepto de escena educativa

Puesto que este trabajo pretende contribuir a la automatización de especificaciones de escenas educativas y para alcanzar dicho objetivo es fundamental la reutilización de las mismas, así como la interoperabilidad entre el LMS y otros sistemas relacionados.

A continuación se presentan aquellos estándares y especificaciones que se consideran relevantes para el concepto de escena educativa y sus especificaciones.

OKI OSID⁸- Esta especificación es una iniciativa de Mellon Foundation que nació en 2001 para ofrecer soporte al desarrollo de los LMS y actualmente está siendo soportada por el MIT y otras instituciones colaboradoras. La especificación OSID de OKI define una arquitectura abierta y extensible para sistemas de *e-learning*, especialmente dirigida a la comunidad educativa de estudios superiores.

La principal aportación de esta especificación es la definición de un conjunto de interfaces conceptuales para la comunicación entre componentes del entorno de gestión del aprendizaje, ya sea dentro de la propia plataforma o entre distintas plataformas, diseñados de tal forma que faciliten la adaptación a nuevas tecnologías y la integración con el resto de infraestructura tecnológica del sistema de aprendizaje.

Algunas de las ventajas que aporta esta solución de código abierto son la estabilidad, durabilidad y escalabilidad de los sistemas de aprendizaje desarrollados utilizando interfaces OSID, además de incrementar la productividad en el desarrollo de recursos a integrar en el entorno de aprendizaje, ya que proporciona una infraestructura tecnológica que libera a los desarrolladores de esta tarea, permitiéndoles centrarse únicamente en el desarrollo de dichos recursos. Por tanto, permite solucionar problemas de interoperabilidad entre el LMS y sistemas relacionados, facilitando al mismo tiempo la reusabilidad de servicios con independencia de la infraestructura tecnológica, lo que la hace especialmente interesante para automatizar los aspectos de comunicación entre participantes de escenas.

IMS DRI⁹ (IMS 2003a) - Esta especificación proporciona recomendaciones para la interoperación de las principales funciones de repositorios y como su nombre indica es ha sido propuesta por IMS.

Su finalidad es facilitar el descubrimiento, acceso y uso de los recursos almacenados en los repositorios, por lo que establece como principales funciones la búsqueda, recuperación, almacenamiento, petición y envío de recursos. Aunque esta especificación reconoce un amplio rango de formatos de contenido, sistemas implementados, tecnologías

⁸<http://okicomunity.mit.edu/staticpages/index.php?page=OverOKI>

⁹<http://www.imsglobal.org/digitalrepositories/index.html>

y prácticas establecidas existentes, recomienda dos tipos de repositorios, los que utilizan la recomendación Z39.50 para conseguir interoperabilidad de repositorios y los que son capaces de implementar las recomendaciones basadas en XQuery y SOAP.

Por otro lado, dicha especificación propone una implementación basada en servicios web y un *middleware* como componente intermedio entre proveedores de servicios y utilizadores de los mismos. Así pues, IMS DRI establece una arquitectura funcional de referencia entre repositorios (objetos de aprendizaje y metadatos asociados) y utilizadores de los mismos. La aportación de esta especificación contribuye pues a la interoperación entre funciones de repositorios y facilita su integración con otras especificaciones (*IMS-Metadata* e *IMS-Content and Packaging*).

LTSC-LTSA¹⁰ (Farance & Tonkel 2001)- Es el estándar propuesto por *IEEE Learning Technology Standards Committee* para definir una arquitectura abstracta de referencia para los sistemas de aprendizaje con el fin de conseguir los siguientes objetivos: 1) proporcionar un marco para entender los futuros sistemas, 2) promocionar la interoperabilidad y portabilidad identificando interfaces de sistema a nivel abstracto, y 3) establecer un horizonte técnico de aplicabilidad de 5-10 años que asegure la adaptabilidad a nuevas tecnologías y LMS.

Dicho estándar propone un marco abstracto con cinco niveles de refinamiento o perspectivas a tener en cuenta en el diseño de la arquitectura. Estos cinco niveles de refinamiento se corresponden con descripciones de la arquitectura a distintos niveles de abstracción. Estos niveles hacen referencia a los siguientes aspectos: 1) interacciones del estudiante con el entorno, 2) características de diseño relacionadas con el estudiante, 3) componentes del sistema, 4) prioridades y perspectivas de implementación y 5) componentes opcionales e interoperabilidad.

Todos ellos niveles son descriptivos, excepto el tercero que es normativo. El tercer nivel especifica el sistema desde el punto de vista de sus componentes, proponiendo una organización de ellos en base a tres tipos de componentes: procesos (*learner entity, evaluation, coach, delivery*, almacenes de datos *learning resources, learner records* y flujos

¹⁰http://www.edutool.com/ltsa/09/IEEE_1484_01_D09_LTSA.pdf

de datos *learning parameters, behaviour, assesment information, learner information, query, catalog information, locator, learning content, multimedia, interaction context*. Los procesos vienen especificados por sus precondiciones y postcondiciones, los almacenes a partir de la información que contienen y de los procesos que interactúan con ellos y los flujos de información por el tipo de información que se transmite, así como los procesos que la emiten y los reciben.

Esta propuesta de definición de la arquitectura es neutral tanto a nivel pedagógico, de contenidos, de implementación o de plataforma, pero en cualquier caso ayuda a entender cómo son los LMS, a procurar interoperabilidad y transportabilidad entre componentes y es adaptable a nuevas tecnologías a medio plazo. En consecuencia es relevante para la automatización de escenas en tanto en cuanto ayuda a entender cuáles son las interacciones entre los componentes del sistema.

IEEE Learning Objects Metadata¹¹ (LTSC WG12 2002) - Es el estándar de metadatos para objetos de aprendizaje basado en Dublin Core y llevado a cabo con el esfuerzo conjunto de destacadas organizaciones como IMS, ARIADNE o AICC de IEEE. Su nombre oficial es *IEEE 1484.12.1-2002 Standard for Information Technology-Education and Training Systems-Learning Objects and Metadata* y se conoce con el nombre de LOM.

LOM determina un esquema de datos conceptual que define la estructura de una instancia de metadatos para objetos didácticos. Dicha instancia describe características del objeto agrupadas en categorías: general, ciclo de vida, meta-metadatos, educativas, técnicas, derechos, relación, anotación y clasificación.

Su propósito es facilitar la búsqueda, evaluación, adquisición y uso de objetos didácticos por parte de los estudiantes, instructores o sistemas automatizados, así como el intercambio de los mismos y su uso compartido, permitiendo el desarrollo de catálogos e inventarios.

En consecuencia, IEEE-LOM resulta especialmente útil para la localización y consulta de objetos de aprendizaje, ya que el hecho de permitir la definición de objetos

¹¹<http://ltsc.ieee.org/wg12/index.html>

de aprendizaje independientemente de su uso facilita la reutilización de los mismos en distintas escenas educativas.

ADL SCORM¹² *Shareable Content Object Reference Model* (ADL 2004) - Es un conjunto de especificaciones y estándares (elaborados por distintos organismos) que se postula como el modelo común para los objetos de aprendizaje. SCORM aspira a convertirse en el modelo final único para el diseño y manejo de objetos de aprendizaje hacia el que converjan el resto de iniciativas, y así, engloba varios estándares y especificaciones como IEEE-LOM o las especificaciones de IMS sobre secuenciación y diseño de contenidos.

Su objetivo principal es la compartición de objetos de aprendizaje entre diferentes sistemas de *e-learning*, por lo que centra su atención en el empaquetamiento de dichos recursos didácticos facilitando la transportabilidad de los mismos y la interoperabilidad entre sistemas de aprendizaje, además de potencia su uso.

En el plano técnico, ADSL-SCORM lo integran dos ejes principales. Por un lado, un modelo de agregación de contenidos (CAM - *Content Agregation Model*) que es una especificación de contenidos de cursos con el fin de crear cursos estandarizados y transportables entre distintos LMS. Por otro, dos especificaciones relacionadas con el ambiente de ejecución (RTE - *Run Time Environment*): una API que define cómo llevar a cabo la interacción entre los contenidos y el LMS, y un modelo de datos que define la estructura de los datos a intercambiar en una interacción.

Por tanto, se considera esta especificación especialmente interesante para el propósito de este trabajo en tanto en cuanto posibilita el transporte de contenidos educativos entre distintas plataformas y permite integrar en él otras especificaciones como IEEE-LOM.

A parte de las especificaciones y estándares anteriormente mencionados, existen otras especificaciones y estándares en tecnología educativa también relacionados pero más indirectamente.

¹²<http://www.adlnet.gov/Scorm/index.aspx>

Otras especificaciones y estándares en tecnología educativa

IMS CP¹³ (*Content Packaging*)- Es una especificación, comercialmente conocida como LRN (*Learning Resource Interchange*), que provee la funcionalidades para describir, y empaquetar material de aprendizaje en paquetes portables e interoperables.

IMS QTI¹⁴ (*Question and Test Interoperability*)- Es una especificación que describe un modelo de datos para la representación de ítems de preguntas y respuestas en *e-Learning*. Su objetivo es facilitar el intercambio de estos tests y los correspondientes resultados de evaluación entre distintas herramientas des soporte al aprendizaje o LMS.

IMS LD¹⁵ (*Learning Design*) (IMS 2003b)- Esta especificación se centra en los procesos de aprendizaje. Soporta un amplio rango de pedagogías en tecnología educativa, para ello utiliza un lenguaje genérico y flexible que permite la descripción y codificación de las distintas metodologías de aprendizaje. Los conceptos fundamentales utilizados por este lenguaje o modelo son: actividades de aprendizaje, objetivos, roles y recursos.

CEN-LORI¹⁶ (*Learning Object Repository Interoperability*) - Es un conjunto de especificaciones que resultaron del *CEN/ISSS Learning Technologies Workshop* con el objetivo de conseguir interoperabilidad entre objetos de aprendizaje.

Esta especificación la constituyen tres documentos: 1) *LORI Framework* que presenta el marco propuesto para alcanzar la interoperabilidad entre LO basado en la distinción de servicios e interoperabilidad, 2) *A Simple Query Interface* que ofrece recomendaciones para un *Application Program Interface (API)* que facilita el acceso a los objetos almacenados en repositorios (Simon et al. 2005) y 3) *Authentication and Session Management* con recomendaciones para la implementación de estas funcionalidades.

UN/CEFACT & OASIS-ebXML - Esta especificación (UN/CEFACT & OASIS 2001) es el resultado de los esfuerzos de dos organismos, UN/CEFACT (*United Nations for Trade Facilitation and Electronic Business*) y OASIS (*Advancement of Structured Information Standards*) con el objetivo de proporcionar un marco estándar mediante el

¹³<http://www.imsglobal.org/content/packaging/>

¹⁴<http://www.imsglobal.org/question/>

¹⁵<http://www.imsglobal.org/learningdesign/>

¹⁶<http://ariadne.cs.kuleuven.be/lomi/index.php/LorInteroperability>

cual pueden configurarse sistemas de negocio con la finalidad de soportar la ejecución de transacciones basadas en colaboraciones de procesos de negocio.

La notación que propone va dirigida a diseñadores de herramientas de definición de procesos de negocio que precisan traducir las correspondientes especificaciones de procesos de negocio en términos de XML. En consecuencia utiliza un lenguaje muy adecuado para la descripción de interacciones entre elementos que participan en una escena, fomentando la interoperabilidad entre aplicaciones de negocio.

RosettaNet (RosettaNet 2001)- Es un estándar no propietario para estandarización de aplicaciones de negocio basadas en Internet que proporciona una plataforma de comunicación o lenguaje común para la automatización de procesos de la cadena de producción de un determinado sector industrial (Badakchani 2004).

El objetivo principal de este estándar es mejorar la eficiencia y rendimiento de los procesos de producción, optimizando la integración de las distintas aplicaciones del negocio. Para conseguir este propósito RosettaNet ofrece distintos estándares para la comunicación y colaboración entre procesos, un marco de implementación y diccionarios técnicos y de negocio (RosettaNet 2001). Puede integrarse en ebXML, ya que pueden utilizarse las recomendaciones de RosettaNet para definir el interface de comunicación que encaja en el protocolo de comunicación de ebXML.

OAGi & OAGIS ¹⁷ (*Open Applications Group Integration Specification*) - Es una especificación resultado de los esfuerzos del grupo *Open Applications Group* que nace de la necesidad de un lenguaje de negocio común para el establecimiento de la comunicación entre aplicaciones de negocio. La primera versión de la misma data de 1995.

El objetivo de la especificación OAGIS es proporcionar un lenguaje de negocios canónico para la integración de la información. Se caracteriza por el uso del lenguaje XML como único alfabeto para la definición de mensajes de negocio y por identificar un conjunto de procesos de negocio (escenas) capaces de establecer la comunicación entre aplicaciones empresariales. OAGIS, además tiene la ventaja de adaptarse a los requerimientos de industrias específicas, lo que permite construir grupos de industrias verticales entre distintos socios.

¹⁷<http://www.ibm.com/developerworks/xml/library/x-oagis/>

De toda esta revisión de especificaciones y estándares relacionados con el tema objeto de estudio, se detecta que en BPM existe una especificación, OAGi-OAGIS¹⁸, que proporciona un catálogo de escenas de negocio básicas que facilita su integración con independencia de la plataforma tecnológica sobre la que se ejecuten las aplicaciones empresariales.

En tecnología educativa no existe ninguna catalogación de escenas básicas, ni siquiera una relación de ellos. Únicamente se dispone de la especificación OKI-OSID¹⁹ que ofrece recomendaciones para el diseño de sistemas de aprendizaje dirigidos a entornos virtuales de enseñanza superior. Esta especificación proporciona un conjunto de interfaces de servicios web para el acceso a las principales funcionalidades de dichos sistemas de aprendizaje, sin embargo no describe tales funciones en términos de interacciones entre componentes de escenas educativas, que en definitiva es lo que interesa para la automatización de escenas en un entorno de *e-learning*.

2.3. Automatización de escenas educativas

Después de este breve repaso a los estándares y especificaciones relacionados, más directa e indirectamente con el concepto de escena educativa, esta sección del estado del arte se focaliza en la automatización de escenas educativas.

Si bien es cierto que recientemente se ha investigado mucho sobre objetos de aprendizaje, su reusabilidad y estandarización, no ha ocurrido lo mismo en lo que se refiere a escenas educativas. La automatización de escenas educativas, entendidas éstos como unidades lógicas de funcionalidad susceptibles de ser usadas para la construcción de sistemas de aprendizaje o como herramientas de diseño facilitadoras de la automatización de sistemas, ha sido un campo prácticamente inexplorado.

En primer lugar, se revisan las arquitecturas existentes a día de hoy capaces soportar la automatización de escenas educativas. Es decir, aquellas que promueven la reusabilidad de escenas y propician la interoperabilidad entre LMS y otros sistemas relacionados. Posteriormente, las distintas técnicas y lenguajes de formalización con el fin obtener

¹⁸<http://www.ibm.com/developerworks/xml/library/x-oagis/>

¹⁹<http://okicomunity.mit.edu/staticpages/index.php?page=OverOKI>

especificaciones de escenas educativas que faciliten su implementación y garanticen el correcto funcionamiento del sistema.

2.3.1. Arquitectura abstracta y tecnología de soporte

En este apartado se ampliarán detalles sobre las especificaciones y estándares relacionados con arquitecturas de sistemas de aprendizaje e infraestructura tecnológicas recomendadas. También se comentarán algunos de los trabajos realizados de acuerdo con estas propuestas.

Learning Technology Commitee propone la especificación EEE P1484.1/D9, oficialmente conocida por *LTSC- Learning Technology System Architecture* (Farance & Tonkel 2001), como arquitectura abstracta de sistemas de aprendizaje basados en la red.

El objetivo de esta especificación es proporcionar un marco que facilite la comprensión de ciertas clases de sistemas, subsistemas y sus interacciones con otros sistemas relacionados con el aprendizaje, independientemente de la tecnología de implementación, por lo que admite distintas arquitecturas concretas. Dado que se trata de un marco neutral en todos los sentidos (a nivel pedagógico, de contenidos, cultural o incluso de plataforma) se considera aplicable cualquier plataforma de LMS. En el apartado Estándares y especificaciones relevantes para el concepto de escena educativa ya se ha comentado su organización en cinco niveles o prioridades.

El primer nivel, que es el más prioritario, describe la arquitectura del sistema desde la perspectiva de las interacciones entre el sistema y las entidades que interactúan con él, el segundo nivel tiene en cuenta el efecto que ejerce el estudiante sobre el diseño del propio sistema, el tercero se centra en los componentes del sistema y sus funcionalidades, el cuarto define los sistemas relacionados con el aprendizaje desde diferentes perspectivas de implementación y el quinto presenta la arquitectura desde el punto de vista de componentes operacionales e interoperabilidad (codificación, protocolos, APIs, etc.). Como se ha comentado anteriormente de todos estos niveles sólo el tercero es normativo, el resto simplemente son informativos.

Dado que para este trabajo interesan las arquitecturas generales y abstractas capaces de soportar la automatización de las principales funcionalidades del LMS, en este

apartado se detallará un poco más el tercer nivel de la propuesta de LTSC-LTSA. En él se presenta la arquitectura del sistema a partir de tres componentes-tipo:

- Procesos: A nivel de procesos se identifican el estudiante, la evaluación, el instructor y la entrega de actividades y todos ellos se especifican en términos de *input-funcionalidad-output*.
- Repositorios: Los repositorios se describen en función de la información que almacenan (registros de estudiantes o recursos de aprendizaje) y las operaciones elementales que se realizan sobre ellos (búsqueda, recuperación y modificación).
- Flujos de información: Los flujos de información son descritos desde el punto de vista de la conectividad de componentes que relacionan y el tipo de información que se envía.

Además, esta arquitectura admite la posibilidad que un mismo componente pueda desempeñar varios roles en una escena educativa y que simultáneamente puedan estar ejecutándose múltiples escenas educativas. A pesar de ello, no ofrece recomendaciones acerca de la infraestructura tecnológica que puede ser más adecuada para la implementación.

Existen otras dos especificaciones relativas a arquitecturas de sistemas de aprendizaje que sí hacen alguna referencia a la infraestructura tecnológica subyacente. Se trata de la especificación IMS-DRI y la propuesta del Workshop CEN-LORI. Ambas se centran en arquitecturas de repositorios focalizando cada una de ellas en un aspecto distinto, parten de la necesidad de una arquitectura basada en servicios y comparten un mismo objetivo general, conseguir interoperabilidad de sistemas en lo que se refiere a repositorios.

La especificación LORI (Simon et al. 2005) propone recomendaciones para la definición de funciones necesarias para la construcción de un API que permita interoperabilidad a nivel de consultas en repositorios. Así pues, ofrece recomendaciones para la configuración y la realización de consultas contra repositorios, ya sea síncrona o asíncronamente. El *middleware* propuesto para conseguir estos objetivos establece dos tipos de servicios:

- Los servicios básicos o *Core Services* tienen por objetivo facilitar la identificación unívoca de recursos de aprendizaje que se hallan en repositorios.
- Los servicios de aplicación o *Application Services* se centran en la autenticación de usuarios y la creación y gestión de sesiones que posibilitan la interacción de aplicaciones.

A nivel de infraestructura tecnológica CEN-LORI hace referencia a las tecnologías SOAP, XML, RPCs y JRI para soporte a servicios de mensajería y HTTP, SMTP, TCP/IP como protocolos de comunicación para servicios de red. Así mismo se menciona WSDL *Web services Description Language* como posible lenguaje de descripción de servicios web.

La especificación IMS-DRI (IMS 2003a), por su parte, ofrece recomendaciones para conseguir interoperabilidad a nivel de las principales funciones relacionadas con repositorios, no sólo centrándose en la consulta como LORI, sino extendiendo su alcance a otras funcionalidades básicas relativas a repositorios, como pueden ser la recuperación, envío o el almacenamiento de recursos de aprendizaje.

La arquitectura general que propone IMS-DRI distingue entre tres niveles:

- Nivel de acceso a servicios: A él acceden las distintas aplicaciones desempeñando distintos roles. Ofrece recomendaciones sobre las distintas funcionalidades que facilitan el acceso a dichos servicios y destaca los siguientes roles: estudiante, instructor, buscador y agente.
- Nivel de provisión de servicios: Lo constituyen diferentes tipos de repositorios y especifica funciones de almacenamiento, gestión y presentación de directorios y registros.
- Nivel intermedio o motor de búsqueda: Relaciona los otros dos niveles anteriormente citados. Propone indicaciones para la especificación de ciertos componentes funcionales como son los utilizadores de recursos, repositorios, gestión de accesos y provisión de servicios.

Aunque esta especificación reconoce un amplio rango de formatos de contenido, sistemas implementados, tecnologías y prácticas establecidas en el área en lo relativo a repositorios, a nivel de infraestructura tecnológica recomienda sistemas que sigan la norma Z39.50 y repositorios que sean capaces de implementar recomendaciones basadas en XQuery y SOAP.

Se observa pues que existe cierta complementariedad entre los tres marcos de de arquitectura general propuestos, denotándose una clara apuesta por la tecnología de servicios web en la concreción de dichas arquitecturas. En (Stojanovic et al. 2001) se argumenta la necesidad de una infraestructura tecnológica basada en servicios para la construcción de sistemas de aprendizaje basados en la red, tal como proponen las especificaciones CEN-LORI o IMS-DRI, argumentado los requerimientos de este tipo de sistemas y las propias características del entorno.

La tecnología de servicios web puede considerarse una instanciación del paradigma SOC (*Service Oriented Computing*). Este paradigma tiene por objetivo facilitar la integración de aplicaciones, sobre una o distintas plataformas, evitando problemas de heterogeneidad de todo tipo (de lenguajes, de formatos, etc.) y proporcionando interoperabilidad entre aplicaciones. Esta tecnología como su nombre indica, está basada en servicios web, que se definen como unidades modulares de lógica de aplicación autocontenidas que proporcionan funcionalidades a otras aplicaciones a través de Internet (Bucchiarone & Gnesi 2006).

Los servicios web proporcionan un interfaz común, para cualquier entidad computacional que lo implemente, sin importar lo autónoma o heterogénea que sea, ya que dicho interfaz facilita el acceso al servicio desde cualquier plataforma que se solicite (McIlraith & Zeng 2001). Así pues, facilitan reusabilidad de funcionalidades y su composición, además de la interoperabilidad entre sistemas.

Posibilitar la reusabilidad de servicios web implica utilizar lenguajes de definición para describir las funcionalidades de cada servicio y combinarlos para crear nuevos servicios web (Peltz 2003). La composición de servicios web precisa de una infraestructura adicional que permita la definición de reglas de composición.

Existen dos modelos de composición de servicios web, estática o dinámica. La composición estática se produce cuando, conociendo *a priori* los servicios web a utilizar y las condiciones bajo las cuales se puede invocar, se define el orden de invocación de los servicios. Para ello existen dos mecanismos de composición de servicios web estática: la orquestación (un coordinador central que dirige la composición de servicios) y la coreografía (se definen las interacciones posibles entre cada par de servicios y no existe la figura del coordinador central). Como ejemplos de lenguajes de servicios web que soportan la combinación de servicios web a nivel estático hay que destacar BPEL o BPML como lenguajes estándar de orquestación y WS-CDL como propuesta de lenguaje estándar de coreografía.

La composición dinámica en cambio consiste en la composición de servicios en tiempo de ejecución, por lo que se trata de reconocer los servicios web que interesan y pueden acoplarse en tiempo real, ello supone la anotación semántica de dichos servicios con el objetivo de ser publicados, descubiertos, compuestos y ejecutados de forma automática (Bucchiarone & Gnesi 2006).

Una vez tratados los estándares y especificaciones relacionados con la arquitectura de los sistemas de aprendizaje en red, se comentan algunos trabajos relacionados con arquitecturas de sistemas de aprendizaje y arquitecturas de repositorios en los que prima la interoperabilidad entre sistemas y ,en algún caso, orientados a la automatización de escenas:

- El proyecto eduSource de Canadá (Eap et al. 2004) presenta el diseño e implementación de un protocolo de comunicación estándar llamado ECL cuyo objetivo es facilitar a los repositorios, nuevos y los existentes, la comunicación y compartición de recursos a través de la red. Está diseñado conforme a la especificación IMS-DRI e implementa las principales funciones de repositorios recomendadas, a la vez que incorpora definiciones basadas en el protocolo OAI.
- El proyecto PAWSEL, también conforme a la especificación IMS-DRI, propone una arquitectura para facilitar patrones de conversación heterogéneos entre los agentes participantes de una escena basándose en servicios web (Macarro et al. 2006).

- El proyecto SleD2 presenta una arquitectura que facilita la integración de servicios de aprendizaje en lugar de servicios web y utiliza el motor CooperCore como servicio conforme a IMS LD integrándolo en un *workflow* que automatiza el envío de actividades (Vogten et al. 2006).

Se ha visto que existen diversas propuestas de arquitecturas basadas en Servicios Web conformes a la especificación IMS Digital Repository Interoperability (Eap et al. 2004, Macarro et al. 2006, Vogten et al. 2006) con la intención de resolver problemas de heterogeneidad y distribución. En cuanto a infraestructura tecnológica, la tecnología de servicios web es actualmente la que más se ajusta a las necesidades de los LMS, aunque es probable que el día que los computadores puedan entender los mismos conceptos que el ser humano, los servicios web semánticos sean más indicados para ello.

A pesar de todo, tal como se comentaba en el número especial del boletín del grupo de interés LTTF de enero de 2004 (*Learning Technology newsletter* 2004), no existe una arquitectura de referencia para la automatización de escenas educativas. A día de hoy, aún no existe tal arquitectura. Existen algunas recomendaciones en este sentido, IMS-DRI (IMS 2003a) o LTSC-LTSA (Farance & Tonkel 2001), así como algunas propuestas concretas de arquitecturas para sistemas de aprendizaje en red basadas en servicios web (Eap et al. 2004, Macarro et al. 2006, Vogten et al. 2006) cuyo objetivo es solucionar ciertos problemas de heterogeneidad y distribución en los LMS. Sin embargo, éstas propuestas están estrechamente ligadas a un entorno tecnológico concreto, utilizando APIs específicas para acceder a los servicios web que implementan las funcionalidades del LMS correspondientes a escenas educativas, lejos de lo que sería una arquitectura abstracta.

2.3.2. Formalización de escenas educativas

Las especificaciones de escenas educativas automatizables total o parcialmente requieren de descripciones de escenas educativas consistentes y no ambiguas, que garanticen una única interpretación por parte del sistema que las va a procesar automáticamente. Es por ello que es necesario formalizar las especificaciones de escenas educativas. En este apartado se comentaran algunas técnicas de especificación formal existentes para este propósito.

Las redes de Petri constituyen una de las técnicas de formalización más conocidas. Se trata de representaciones matemáticas de un sistema distribuido discreto y constituyen una generalización de la teoría de autómatas que permite expresar eventos concurrentes. Se basa en estados más que en eventos y describe, a partir de los eventos que ocurren y las condiciones que determinan el paso de un estado a otro, los distintos estados por los que va evolucionando el sistema. Dado que permiten modelar sistemas formados por distintos procesos o servicios que cooperan para la realización de un objetivo común, las redes de Petri han resultado ser herramientas muy útiles para capturar interacciones entre procesos o servicios y, en consecuencia, para la formalización de flujos de trabajo o *workflow*, tal como ya se comentó anteriormente en la sección Especificaciones de escenas educativas. Sin embargo, presentan limitaciones para expresar dependencias entre flujos de control, lo que constituye un problema para representar escenas educativas mediante la composición de servicios web (Van der Aalst, Hofstede & Weske 2003).

El procesamiento automático de escenas educativas requiere información semántica que posibilite el razonamiento del sistema. En el caso de las escenas educativas una de las dificultades con las que hay que enfrentarse, al margen de la complejidad de interacciones que se dan dentro del propio sistema, es la captura de información informal que se deriva de dichas experiencias educativas, lo cual implica a menudo recopilar información inconsistente e incompleta, no apta para su procesamiento automático (Helic 2007). Sin embargo, por otro lado, parte de esta información viene dada por los metadatos de los objetos de aprendizaje que se ven involucrados en dichas escenas.

Algunos estudios (Friesen 2004, Najjar et al. 2002, Sicília & García-Barriocanal 2005) sostienen que la anotación semántica de los objetos de aprendizaje reusables no garantiza la automatización de escenas educativas, puesto que los metadatos de los LOR son incorporados por sus creadores y suelen ser incompletos y estar fragmentados, resultando por tanto insuficientes para la construcción de elementos de software (Sicília, García-Barriocanal, Pagés, Martínez & Gutiérrez 2004). Sin embargo, en (Pagés et al. 2003) se argumenta que si se dispusiera de algún mecanismo que facilitara la especificación de metadatos de LO de calidad suficiente, sí que sería posible automatizar los procesos que ocurren en los sistemas de gestión de aprendizaje y en los repositorios de LO.

Por otro lado, la compartición de recursos dispersos en la red y el procesamiento automático de escenas educativas precisa de información semántica que garantice la localización de estos recursos y una única interpretación de los mismas. En este sentido las ontologías pueden considerarse mejor técnica que las redes de Petri.

Una de las definiciones de ontología más comúnmente aceptadas en el campo de las ciencias de la computación es la de Gruber (Gruber 1993): “una ontología es una especificación formal explícita de una conceptualización”, entendiendo por conceptualización una visión abstracta y simplificada del mundo a representar. No obstante, no debe de caerse en el error de considerar la ontología como el esquema conceptual de un sistema de información, puesto que una ontología puede contener instancias y además, aunque la mayoría de ontologías representen información estática, pueden utilizarse para representar conocimiento dinámico gracias a su capacidad de inferencia y a la posibilidad de ser extendidas mediante reglas.

Actualmente las ontologías constituyen un tópico de interés por parte de la comunidad científica, sea cual sea su campo de interés. Aunque su uso más extendido es como herramienta de categorización de la información en sitios web, también son muy útiles para la definición de restricciones semánticas asociadas a taxonomías de conceptos, tanto en tiempo de diseño como en tiempo de ejecución.

Independientemente del dominio de conocimiento de cada ontología y su finalidad, las ontologías reportan una serie de ventajas. Entre los beneficios del uso de una ontología para el desarrollo de sistemas de información cabe destacar:

- a) Promueve el consenso entre los distintos participantes en la creación del sistema.
- b) Facilita la reutilización de conocimiento en el modelo conceptual de actividades.
- c) Mejora la comprensión del dominio de conocimiento a nivel de tareas y de funciones.
- d) Valida esquemas conceptuales creados previamente.

En tiempo de ejecución las ontologías también aportan beneficios que redundan en el incremento de la eficacia de los sistemas de información:

- a) Mejoran la comunicación entre distintos agentes debido a que ofrecen soporte a los lenguajes de comunicación y facilitan el consenso entre colectivos.
- b) Soportan la integración de distintas fuentes de datos.
- c) Fomentan la interoperabilidad entre distintas aplicaciones.
- d) Soportan la interpretación del lenguaje natural,
- e) Permiten modelar el contenido semántico de las páginas web.
- f) Soportan las aplicaciones de comercio electrónico.

En el ámbito de la tecnología educativa se han desarrollado diversas ontologías. Algunas de ellas han sido utilizadas en importantes estándares y especificaciones dentro del área del e-Learning. En el estándar LOM (LTSC WG12 2002) la ontología es clave para la categorización, localización y recuperación de objetos de aprendizaje anotados semánticamente. En la especificación IMS LD (IMS 2003*b*), la semántica de la propia especificación es descrita por una ontología (Amorim et al. 2006), lo cual implica representar tanto la parte estructural o estática del proceso de aprendizaje, como la funcional o dinámica del mismo.

También existen ontologías para la descripción de contenidos de aprendizaje de documentos técnicos (Kabel et al. 1999) para describir interacciones entre estudiante y sistema de aprendizaje dentro de entornos colaborativos (Ikeda et al. 1995), para la descripción de tareas de aprendizaje (Mizoguchi et al. 1996) y para describir objetivos de aprendizaje y trabajo en grupo (Inaba et al. 2001) entre otros. Además también existe alguna propuesta para describir mediante ontologías escenas educativas en entornos colaborativos (Barros et al. 2002). Esta ontología se utiliza como meta-modelo para representar los conceptos básicos sobre aprendizaje colaborativo y las interrelaciones y dependencias entre ellos con la finalidad de facilitar la definición de nuevos escenarios colaborativos, así como el análisis y asesoramiento de colaboraciones en grupo.

Sin embargo, no existe ninguna ontología para la especificación de escenas educativas automatizables o semi-automatizables, aunque sí se argumenta su necesidad en (Rius et al. 2007*a*, Rius et al. 2008*b*).

Algunos autores ven las ontologías como una técnica de formalización capaz de integrar en los contenidos educativos sus correspondientes metadatos, de forma que puede ser útil para la definición de escenas educativas relacionados con la creación, mantenimiento y reutilización de repositorios de LOR (Lytras et al. 2003). Otros, las consideran como un mecanismo para capturar distintas vistas que faciliten su reutilización (Sicília 2007).

De lo anteriormente dicho, se deduce que aunque existen ontologías que permiten formalizar escenas colaborativas en entornos virtuales (Helic 2007), éstas no contemplan el aspecto dinámico de las mismas, totalmente indispensable para obtener código ejecutable a partir de especificaciones de escenas. Se detecta pues, la inexistencia de algún mecanismo de formalización de especificaciones de escenas orientado a la automatización.

2.4. Limitaciones y carencias detectadas

En esta sección se presenta una relación de los principales problemas detectados en tecnología educativa que, en la actualidad, impiden la obtención de alguna técnica que permita la obtención de especificaciones de escenas automatizables, ya sea de forma total o parcial.

Las principales limitaciones y carencias detectadas son las siguientes:

a) **Heterogeneidad semántica del término escena educativa**

La carencia de una definición comúnmente aceptada sobre los términos LO y escenario educativo denotan la existencia de un problema de heterogeneidad semántica en ambos casos. De acuerdo con (Sheth & Larson 1990) y a partir de las distintas definiciones consideradas, se ha observado que las definiciones de LO varían según el nivel de granularidad de las mismas, mientras que en lo que respecta al término escenario educativo, la variedad de las mismas atiende al nivel de abstracción que se considere o mejor dicho la disciplina que defina el término. Así pues, existe un problema de heterogeneidad semántica para el término escenario educativo y faltan definiciones de escenas educativas para poder resolver el problema de la automatización de especificaciones de escenas educativas.

b) **Carencia de mecanismos para capturar la semántica de las escenas educativas y facilitar su automatización**

En el área del BPMN existen mecanismos para la descripción de procesos de negocio orientadas a la automatización de los mismos (OMG & BPML.org 2006), en cambio en tecnología educativa no existe ningún mecanismo con este mismo fin.

Se requieren técnicas de especificación de escenas educativas orientadas a la automatización de tales especificaciones, lo cual implica la necesidad de mecanismos que faciliten además descripciones estáticas de escenas (objetivos, componentes, interrelaciones entre componentes, estructura entre otros), descripciones dinámicas (patrones de comportamiento del sistema y de otros agentes involucrados durante la ejecución del mismo) de los mismos.

Además, puesto que dicha técnica debe de facilitar la automatización de las especificaciones de escenas en tecnología educativa, interesa algún mecanismo que permita reutilizar tales especificaciones en la construcción o personalización de sistemas de aprendizaje en red.

c) **Falta de un conjunto mínimo de especificaciones de escenas educativas**

En tecnología educativa no existe ninguna catalogación de escenas básicas, tal como ocurre en el ámbito del BPM con la especificación OAGIS. Ni siquiera una relación de ellos.

Para la automatización de procesos de *e-learning* igual que ocurre para la automatización de procesos de negocio, debe de existir un conjunto mínimo a partir del cual formar nuevas especificaciones de escenas o adaptar las ya existentes.

d) **Carencia de una arquitectura abstracta que soporte especificaciones ejecutables de escenas**

Actualmente, no existe una arquitectura abstracta capaz de soportar especificaciones ejecutables de escenarios, tal como ya se comentó en el número especial del boletín del grupo de interés LTTF de enero de 2004 (*Learning Technology newsletter* 2004).

Por consiguiente, falta de un marco conceptual que ofrezca soporte al procesamiento automático de especificaciones de escenas educativas, válido para entornos distribuidos e independiente de la plataforma del sistema de aprendizaje. Esto es, un marco que promueva la interoperabilidad y portabilidad de especificaciones de escenarios y escenas educativas.

e) **Inexistencia de un mecanismo de formalización de escenas orientado a la automatización**

Aunque existen ontologías para la formalización de aspectos concretos en tecnología educativa e incluso para la formalización de escenarios colaborativos en entornos virtuales (Helic 2007), no existe ningún mecanismo de formalización de especificaciones de escenas educativas que facilite su transcripción a lenguaje ejecutable.

Dada la complejidad de las escenas que se desea automatizar, se requiere de algún mecanismo capaz de formalizar un gran número de interacciones entre participantes, es decir que tenga en cuenta el comportamiento individual de cada participante y la cooperación y colaboración entre ellos para determinar el flujo de ejecución de los procesos que implementan un escenario.

2.5. Resumen

En este capítulo se ha analizado el estado actual de las investigaciones y trabajos desarrollados en el ámbito de la tecnología educativa relacionados con la automatización de escenas educativas enfatizando en la reusabilidad e interoperabilidad.

Relacionados con estos conceptos, se han presentado diversos estándares y especificaciones destacados en la comunidad de *e-learning* (OKI-OSID, IMD-DRI, ADL-SCORM, LTSC-LTSA), así como ciertos aspectos relativos a la automatización de procesos de negocio por la similitud existente entre escenarios educativos y escenarios de negocio. Finalmente, como resultado de la revisión del estado del arte, se presenta una relación de las limitaciones y carencias de la tecnología educativa para la resolución del problema planteado con la finalidad de formular una propuesta que de solución al mismo.

Capítulo 3

Planteamiento del problema

Historically, the success of automation has relied on two factors. The first is an understanding of the assembly process, first discovered and performed by hand, then captured, studied, and expressed in a technology. The second factor is the standardization of component properties [...] that creates a marketplace in which many vendors can create components competitively. The broad availability of inexpensive standardized components makes assembly processes easier to express in concrete terms.

David Wiley

En este capítulo se presenta el planteamiento del problema que se pretende resolver en este trabajo de investigación. Aunque dicho problema ha sido introducido en el primer capítulo en éste se procede a profundizar en él con el objetivo de trazar una estrategia de resolución.

Primeramente, se presenta su motivación, luego se define el problema en cuestión y se delimita su alcance. Posteriormente, se describe una estrategia de resolución para el mismo y finalmente, se presenta brevemente la escena utilizada como caso de ejemplo y el contexto organizativo en que tiene lugar.

3.1. Motivación

La motivación principal de este trabajo de investigación es **contribuir a la automatización de sistemas de aprendizaje mediante la automatización de especificaciones de escenas que ocurren habitualmente en entornos educativos concretos.**

Existen escenas educativas que se repiten cíclicamente en las instituciones educativas y que son susceptibles de ser automatizadas o semiautomatizadas. En algunas de estas escenas el sistema podría asumir el rol de alguno de los participantes, por ejemplo el del profesor coordinador, liberando a éste de ellas y permitiendo que se dedique a otras que aporten mayor valor añadido, con el consiguiente beneficio que ello conlleva, tanto a nivel personal como organizacional.

La idea surge de la confluencia de varios factores. Por un lado, la experiencia de la doctoranda como profesora en el entorno virtual de la UOC y por otro, el resultado del estudio del estado del arte en tecnología educativa, el cual ha permitido la detección de limitaciones y carencias que impiden la automatización de especificaciones de procesos en entornos educativos.

Cada vez que se inicia un periodo académico se repiten una serie de tareas para la puesta en marcha de los nuevos cursos que se ofrecen, durante el periodo académico se van repitiendo una serie de eventos que van marcando el ritmo del curso e implican la realización de tareas concretas. Luego, al finalizar el curso con el periodo de evaluación final y cierre del mismo, cada vez se repiten las tareas de introducción de notas y revisión de pruebas de evaluación, así como la valoración del curso que termina. Dentro de esta rueda son necesarios algunos procedimientos que podrían automatizarse para simplificar la tarea del profesor, sobre todo los relacionados con temas de gestión docente.

Por tanto, la evidencia sugiere que al igual que en el área de negocios la automatización de procesos ha reportado importantes ventajas para las empresas, la automatización de procesos en entornos educativos, y especialmente en aquellos que requieren mayor planificación y sistematización como es el caso de la enseñanza en entornos virtuales, puede ser de gran interés. Sin embargo, se observa que la investigación en el ámbito del *e-learning* en lo que respecta a escenas educativas ha sido un campo todavía poco explorado y apenas existe contexto en la definición de escenario educativo.

Así pues, la justificación del interés por el tema de investigación escogido puede concretarse en los siguientes motivos:

1) La clarificación de los conceptos de escenario y escena educativa, promoverá la creación de un conjunto mínimo de procesos relacionados con el aprendizaje, el cual podría servir de base para definir nuevos procesos y escenas educativas potenciando su reutilización.

En el capítulo 2 se ha explicado lo mucho que ha avanzado recientemente la investigación en tecnología educativa en lo que se refiere a objetos de aprendizaje y su estandarización. Sin embargo, los conceptos de escenario educativo y escena educativa han quedado un tanto relegado a la vista de los resultados de investigación obtenidos hasta el momento.

Dada la importancia que en tecnología educativa tienen estos conceptos como elementos integradores de los entornos de aprendizaje, resulta claro que profundizar en este concepto y promover la investigación en este sentido supondría un paso adelante en la investigación dentro del ámbito del *e-learning*.

Sólo disponiendo de una definición de los mismos y eliminando la ambigüedad respecto a estos conceptos podrá crearse un conjunto mínimo de procesos que sean útiles para automatizar las escenas que ocurren en entornos educativos fomentando su reutilización.

2) Los beneficios que pueden reportar las descripciones formales de escenas educativas orientadas a su automatización en entornos de aprendizaje y en las organizaciones que centran su actividad en la enseñanza mediante sistemas de aprendizaje.

Por un lado, la especificación de procesos implica la descripción detallada de éstos y por tanto un análisis de los procedimientos que se implementan en base a ellos. Si además, dicho estudio se realiza con la finalidad de automatizar tareas para poder realizar otras de mayor valor añadido aún pueden conseguirse más ventajas.

Por otro lado, si consideramos la similitud entre los procesos de negocio y los procesos de aprendizaje, así como las ventajas que ha reportado la automatización de los procesos de negocio en el área del BPMN, cabe pensar que las ventajas de la automatización de los procesos de negocio podrían trasladarse al ámbito del e-learning para los procesos

relacionados con el aprendizaje obteniéndose las siguientes ventajas o beneficios:

a) **Agilización de los procedimientos en entornos de enseñanza virtual**

La especificación de procesos en entornos educativos permite describir patrones de comportamiento que conducen a un determinado objetivo. Dicha especificación facilita la identificación de tareas implicadas y el establecimiento de un orden de precedencia entre ellas, así como la identificación de los distintos roles que intervienen y los tipos de interacción que se requieren.

Este conocimiento detallado que se deriva de la especificación facilita simplificación del mismo, promueve la optimización de recursos y evita numerosos errores de implementación que retardan la consecución del objetivo perseguido. Por tanto, la automatización de procesos que implementan ciertos procedimientos permite maximizar el número de procesos que pueden llevarse a cabo paralelamente y clarificar las funciones que cada participante asume al verse involucrado en él.

b) **La asunción de algún rol por parte del sistema**

El hecho que el sistema asuma el rol de uno alguno de los participantes de una escena educativa, permite a éste liberarse de tareas que pueden mecanizarse y por tanto, centrarse en otras de mayor valor añadido, interviniendo sólo cuando sea realmente imprescindible. Por ejemplo, en ciertos procedimientos relacionados con el aprendizaje el LMS puede reemplazar al profesor al realizar ciertas tareas mecanizables y requerir sólo su intervención cuando deba de tomar decisiones.

c) **Reducción del coste de desarrollo y mantenimiento del LMS**

La especificación de una escena contextualizada en un escenario educativo concreto, no es más que la adaptación de un patrón de escena que permite alcanzar un objetivo relacionado con el aprendizaje. Tal patrón tiene identificados los procesos que se llevan a cabo, algunos de los cuales forman parte de otras escenas. Identificar los procesos necesarios y localizarlos para componer nuevas escenas facilita la reutilización de componentes y por tanto el coste de desarrollo y de mantenimiento de cualquier sistema de aprendizaje.

d) Mejora del diseño del sistema para conseguir un LMS adaptativo

La modularidad del sistema promueve la reutilización e integración de componentes. Disponer de un conocimiento profundo del comportamiento del sistema frente a determinados estímulos, facilita su adaptación ante posibles cambios.

e) Promocionar la certificación de procedimientos que lleva a cabo el LMS

El análisis documentado de los procedimientos que se llevan a cabo y la forma como éstos se llevan a cabo, permite conocer el funcionamiento de la organización o sistema y verificar el cumplimiento de las normas y criterios empleados en ellos, de manera que pueda garantizarse la calidad de los procesos y comparar unos sistemas con otros.

f) Reducción de costes por eliminación de errores

La realización de ciertas tareas fuera de plazo o incorrectamente, pueden llevar a la organización a incurrir en una serie de costes que la automatización podría evitar.

De lo expuesto hasta el momento resulta claro que la obtención de especificaciones formales de procesos para ofrecer soporte a la implementación puede reportar numerosas ventajas en cuanto a desarrollo y adaptación de sistemas de aprendizaje, lo cual redundará en el funcionamiento de las organizaciones que centran su actividad en este tipo de sistemas y en una futura estandarización de procesos en entornos educativos.

3.2. Definición y alcance del problema

La automatización de especificaciones de procesos en entornos educativos es un problema no resuelto a día de hoy al que puede darse luz mediante la obtención de especificaciones formales de escenas educativas que faciliten su posterior implementación.

En un entorno de aprendizaje se dan situaciones relacionadas con el aprendizaje que se repiten periódicamente y en el caso de la enseñanza virtual la mayoría de ellas son soportadas por una plataforma de sistema de aprendizaje (LMS).

Estas escenas se llevan a cabo en el seno de organizaciones educativas concretas que determinan un contexto específico al que denominaremos escenario educativo. Las escenas educativas como su nombre indica ocurren habitualmente en entornos educativos. En un entorno de aprendizaje tienen lugar todo tipo de procesos relacionados con el aprendizaje, ya sea antes de que se produzca el aprendizaje (p.e. la preparación de un curso), durante el propio proceso de aprendizaje (p.e. el envío/recepción de actividades de aprendizaje para su evaluación) o después del mismo (p.e. valoración de resultados del proceso de aprendizaje en sí o del uso de los recursos empleados). Y precisamente en el caso de la enseñanza virtual la implementación de parte de estas escenas puede entenderse como la ampliación de funcionalidades del LMS.

Es bien sabido que dentro de cualquier organización, al margen de la actividad principal que ésta lleve a cabo, existe un conjunto de normas y procedimientos que adaptan procesos genéricos, comunes a organizaciones similares, de acuerdo con la idiosincrasia de la organización. En este trabajo nos referiremos a organización como sinónimo de institución educativa y por tanto estaremos considerando aquellas organizaciones centran su actividad en la enseñanza.

Como ejemplo de institución educativa se ha escogido la Universitat Oberta de Catalunya (UOC). El motivo principal es que es pionera dentro del estado español en entornos de enseñanza virtual, por lo tiene probada experiencia en este tipo de entornos y por tanto dispone de un conjunto de normas y procedimientos suficientemente establecidos. Por otro lado, la doctoranda, aunque ejerce de profesora en la universidad desde 1996, está vinculada a la UOC desde principios de 2001, motivo por el cual conoce en profundidad las principales escenas educativas y sobretodo en el contexto de la UOC.

Dado que el problema que se presenta se centra en la obtención de especificaciones formales que faciliten la implementación de procesos que forman parte de las escenas educativas fomentando su reutilización, es necesario considerar una vertiente tecnológica en el problema a resolver. Concretamente, aquella que involucra los lenguajes, especificaciones y estándares capaces de ofrecer soporte a dichas especificaciones orientándolas hacia la automatización, ya sea de forma total o parcial, así como en los mecanismos de descripción formal de escenas que garanticen un comportamiento del sistema correcto,

consistente y no ambiguo.

A partir de lo dicho anteriormente, se deduce que el problema puede contextualizarse en la intersección de tres ejes o dimensiones. Por un lado, la vertiente del *e-learning*, centrándose en el concepto de escenas educativa y demás conceptos relacionados. Por otro, el contexto organizativo concretándose mediante sus escenarios, normas y procedimientos y la tercera dimensión, la tecnológica, focalizándose en los lenguajes de representación y su formalización mediante ontologías. Por lo que gráficamente, puede delimitarse el problema tal como sugiere la figura 3.1.

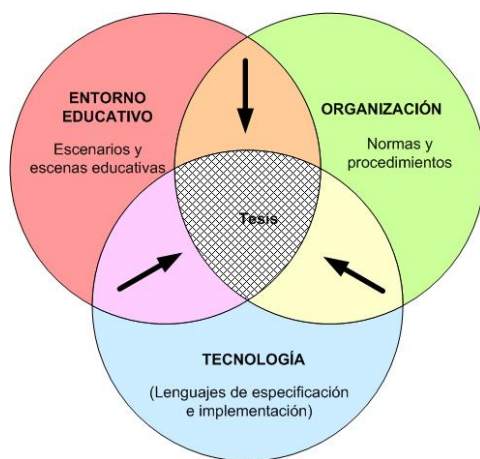


Figura 3.1: Delimitación del problema en base a sus tres dimensiones

A modo de resumen y para centrar el problema a abordar en este trabajo de investigación, se define éste como sigue:

La necesidad de mecanismos que permitan la representación formal de escenas educativas con el fin de ofrecer soporte a la automatización total o parcial de las mismas y promoviendo la reusabilidad de los procesos que se llevan a cabo en ellas.

3.3. Estrategia de resolución

Encontrar una propuesta de solución al problema planteado en este trabajo de investigación implica trazar una estrategia que permita formular una propuesta de solución

que satisfaga los objetivos descritos en el capítulo 1.

Dado que el objetivo principal de este trabajo es la obtención de un marco que permita representar patrones de escenas educativas, adaptar dichos patrones a escenas educativas concretas, contextualizándolas en alguna organización, y generar una implementación parcial de las mismas interesa abordar:

- La especificación formal de escenas educativas genéricas reutilizables.
- La especificación de escenas concretas adaptadas a alguna organización de manera que incorpore el contexto organizativo, tanto a nivel de normas como de procedimientos.
- La traducción de las anteriores especificaciones en términos de implementación procurando interoperabilidad de las especificaciones.

Se propone por tanto la formalización de patrones de escenas y escenas concretas con una clara orientación hacia la automatización. Es decir, interesa describir escenas en términos de los procesos que se llevan a cabo en ellas, los participantes que interactúan en ella y los recursos que se utilizan, así como la secuenciación de escenas y su reusabilidad para constituir escenas más complejas reutilizables.

Tal como se ha descrito en el capítulo 2, no existe una técnica de especificación de procesos en *e-Learning* para describir procesos en términos de interacciones entre participantes. Por este motivo y dada la existencia de un estándar para las especificaciones de procesos de negocio, cuyo éxito ya ha sido probado, se plantea el uso de una representación gráfica para especificar de procesos en entornos educativos.

También, para el propósito descrito en este trabajo es fundamental la obtención de descripciones formales correctas, consistentes y no ambiguas que puedan ser interpretadas por una máquina. Por tanto, se requiere de algún mecanismo de representación de conocimiento que permita explicitar toda la información (estructural y funcional) escenas, genéricas o concretas, partir de su descripción gráfica. Dicha representación debe de permitir la definición consistente y no ambigua de patrones de escenas y escenas concretas adaptadas a una organización, de manera que puedan componerse nuevos patrones

de escenas y escenas concretas, así como la adaptación de estos patrones a escenas en el contexto de una organización.

Además, para traducir las descripciones formales de escenas educativas en términos de su implementación promoviendo la interoperabilidad, se propone el uso de la especificación OKI-OSID. Por tanto, para alcanzar el objetivo propuesto, se establece la siguiente estrategia de resolución:

1) **La definición de un lenguaje gráfico para la representación de patrones de escenas y escenas concretas.**

La notación propuesta para la descripción gráfica de procesos de entornos educativos se encuentra en el apéndice C. Se inspira en BPMN pero incorpora la posibilidad de asociar flujos de datos a los flujos de control y permitir así generación de nuevos flujos de datos que aseguren la alineación de precondiciones y postcondiciones de los procesos componentes.

2) **La creación de una ontología multinivel que permita la representación formal de patrones de escenas y escenas concretas a partir de los diagramas gráficos correspondientes.**

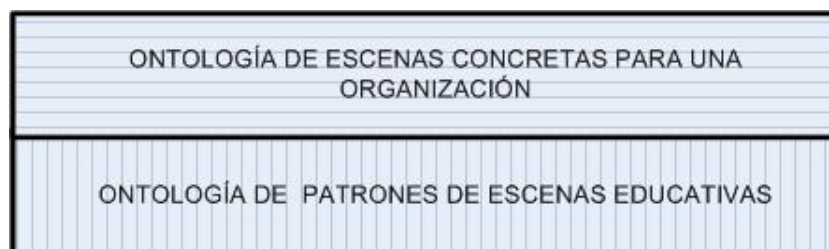


Figura 3.2: Arquitectura ontología multinivel a nivel conceptual

La ontología multinivel que se propone, conceptualmente, está organizada en dos niveles tal como se muestra en la figura 3.2.

Un primer nivel para describir escenas genéricas o patrones de escenas y un segundo nivel para las concreciones de estos patrones, llamadas escenas, fijado un contexto organizativo. Sin embargo, a nivel de diseño y/o implementación la propuesta se implementa con más de una ontología por nivel e incluso se le añade un nuevo nivel

para poder probar que, a partir de tales descripciones formales, pueden generarse implementaciones parciales de las mismas.

A continuación, se describe cada una de las ontologías que forman la arquitectura ontológica propuesta y el proceso seguido para su integración de manera que conforme el marco propuesto.

2.1 **OntoProc**: Ontología de procesos para la secuenciación y composición de patrones de escenas en entornos educativos en general. Se define en base a secuencias de procesos que intercambian mensajes, utilizan recursos e interactúan con participantes. La secuenciación describe la alineación de procesos a nivel de flujo de control y su conexión mediante de flujos de datos.

2.2 **OntoED**- Ontología que define escenas educativas es decir el contexto en que se desarrollan las escenas, lo cual implica la tipología de recursos, participantes y tipos enumerados que caracterizan el entorno educativo.

Ambas ontologías se integrarán para formar la ontología de procesos para entornos educativos (**OntoProcED**) que permite la definición de patrones de escenas educativas. La integración permitirá crear una taxonomía de recursos y de participantes, así como de tipos enumerados que permitirán contextualizar procesos en entornos educativos para describir patrones de escenas educativas.

2.3 **OntoProcED** Ontología para la representación de patrones de escenas educativas que constituye el primer nivel de la arquitectura ontológica propuesta. El objetivo de esta ontología es facilitar la representación de escenas educativas genéricas independientemente de la organización y la tecnología del sistema de aprendizaje que ésta utilice.

Para concretar escenas educativas dentro del contexto de una organización, adaptando patrones de escenas a escenas, es necesario crear dos ontologías más que forman parte del segundo nivel de la arquitectura ontológica propuesta:

2.4 **OntoOrg**- Ontología que define procesos concretos para organizaciones específicas con independencia del contexto de éstas y a partir de patrones de escenas educativas. Se define en términos de procesos, secuencias de procesos, conectores y mensaje, omitiendo participantes y recursos. Permite el refinamiento y restricción de procesos genéricos y representa la parte dinámica de las escenas.

2.5 **OntoUOC**- Ontología que define escenarios educativos en la UOC mediante la definición de figuras, recursos y normativa UOC. La definición de participantes UOC especializa la taxonomía de participantes de OntoProcED, la definición de recursos UOC especializa la taxonomía de recursos de OntoProc, la normativa UOC se materializa mediante reglas SWRL y una taxonomía de tipos enumerados. Estos escenarios son válidos sólo para la organización UOC y todas las escenas que se representen en ella puesto que representan la parte estática o de contextualización de dichas escenas.

Estas dos ontologías son complementarias y se integran para dar lugar a una ontología que permita la concreción de escenas a nivel de una organización concreta, en este caso la UOC (OntoEDUOC). Sin embargo, esta ontología no tiene sentido por si sola si no es acompañada de la ontología de primer nivel (OntoProcED). La integración de ambas reproduce la ontología de procesos pero a menor nivel de abstracción, concretando escenarios específicos y procesos concretos.

2.6 **OntoEDUOC**- Es la ontología de escenas educativas para el caso de la UOC como ejemplo de ontología de escenas educativas para una organización concreta. Su finalidad es la representación de escenas que habitualmente ocurren en una organización específica mediante la adaptación de patrones de escenas establecidos y la incorporación de particularidades de la organización.

Finalmente, para verificar la utilidad del marco propuesto se creará la ontología de implementación u ontología OKI, que a partir de las descripciones las formales de escenas obtenidas generará implementaciones parciales de

dichas escenas en términos de los interfaces de servicios web propuestos por OKI.

2.7 OntoOKI- Ontología define una parte de la especificación OKI-OSID, concretamente aquella que es relevante para la especificación de patrones de escenas educativas y la implementación de escenas concretas. Su objetivo es traducir las especificaciones formales obtenidas a partir de la arquitectura ontológica propuesta en términos de OKI-OSID para garantizar la interoperabilidad.

3) **La instanciación de la ontología multinivel a partir de la representación gráfica de escenas y patrones de escenas.**

Para facilitar parte de la instanciación de esta ontología se propone el uso de una herramienta CASE (*Computer Aided Software Engineering*) que automatice parte de la ontología multinivel a partir de la representación gráfica de escenas que permite generar. La instanciación automática es sólo de una parte de la ontología, concretamente la de procesos, porque el objetivo es realizar una prueba de concepto con el fin de probar la posibilidad de instanciar la estructura ontológica de forma automática. En el apéndice G presenta dicha herramienta CASE.

Finalmente, definida e implementada la propuesta de solución, sólo quedará pendiente la validación del marco formal propuesto. Esta validación se realizará mediante un caso de ejemplo suficientemente representativo que permita probar la representación de escenas educativas y la obtención de descripciones formales válidas para obtener una implementación parcial de las mismas, por lo que como paso previo a la implementación se decide la infraestructura tecnológica a utilizar.

En lo que se refiere a lenguaje de descripción de ontologías se propone el uso de OWL (Ontology Web Language) por ser la tercera propuesta del World Wide Web Consortium para la creación de aplicaciones de web semántica. En concreto, se plantea el uso de OWL DL debido a la necesidad de garantía computacional y al hecho de que está construido sobre RDF y RDFS, por lo que define tipos de relaciones expresables en RDF, utilizando XML.

Para trabajar con OWL la forma más óptima consiste en disponer de un editor gráfico que facilite la edición y creación de clases, propiedades, restricciones e instancias de la ontología. Entre los editores de ontologías posibles se ha escogido Protégé por ser un editor de ontologías de código abierto y multiplataforma, además de ser el más utilizado en la actualidad. Dicha herramienta permite crear de manera automática el código de cualquier ontología en formato RDF/OWL, además de ofrecer otras ventajas (Noy & McGuinness 2001) como la instalación de determinados *plugins*, por ejemplo para visualizar la estructura ontológica gráficamente y para extender OWL con el lenguaje de reglas SWRL.

SWRL (Semantic Rule Web Language) es una extensión natural de OWL, que combina OWL DL y OWL Lite (la parte más básica de OWL), la cual incorpora un sublenguaje de RuleML (Rule Makeup Language) que permite expresar cláusulas de Horn mediante un lenguaje de alto nivel. Sin embargo, tiene el inconveniente de que no permite la negación. Puesto que en este trabajo se requiere un lenguaje de reglas para traducir las especificaciones formales de escenas educativas en términos del metamodelo OKI-OSID, SWRL es una buena opción.

También se propone el uso de un DSL (*Domain Specification Language*) para facilitar la representación gráfica de escenas educativas, posibilitar el reconocimiento automático de los elementos de las especificaciones e instanciar la ontología de procesos a partir de éstos. Sin embargo, deberá de considerarse como prueba de concepto para probar la posibilidad de instanciar automáticamente toda la estructura ontológica.

3.4. Caso de ejemplo

En esta sección se introduce la escena que se va a utilizar como caso de ejemplo a lo largo de esta tesis y concretamente, en la validación de la solución propuesta.

Como se ha comentado en el capítulo 1 la escena educativa escogida es la preparación de un curso de Bases de Datos I en la UOC. Para aquellos lectores que no conozcan el entorno educativo de esta organización, su funcionamiento y organización docente, se describe su contexto. Para más información se remite al lector a las siguientes fuentes (Rodríguez et al. 2006, Serra et al. 2005, Duart et al. 2006).

La docencia en un entorno de enseñanza virtual se realiza de distinta manera que en un entorno presencial, ello es debido a que la infraestructura sobre la que se lleva a cabo condiciona la forma de aprender y enseñar. Este hecho afecta, tanto al modelo educativo como al sistema de organización docente que deben ser diferentes a los convencionales.

El modelo educativo de la UOC se centra en el estudiante. El estudiante puede elegir su propio ritmo de estudio, por lo que es necesario que el curso esté totalmente planificado antes del inicio de curso y las actividades del curso preparadas. En este modelo es fundamental la planificación anticipada. Del mismo modo juegan un papel relevante los materiales del curso, puesto que en cierta forma sustituyen las explicaciones del profesor. Así mismo, cuenta con la evaluación continua como herramienta que garantiza el progreso académico de los estudiantes. La evaluación continua consiste en la realización de actividades evaluables durante el curso para un mejor aprovechamiento de la asignatura.

En lo que respecta al sistema de organización docente, en la UOC está formado por las siguientes figuras:

- 1) El profesor responsable de asignatura (PRA) que es una persona a tiempo completo en la UOC, experta en el ámbito de conocimiento de las asignaturas de las cuales es responsable y que tiene la máxima responsabilidad en lo que se refiere a tareas docentes i académicas.

El PRA tiene libertad de cátedra para determinar el currículum de sus asignaturas, selecciona los materiales de la asignatura y en caso de no existir los idóneos, realiza un encargo de autoría para su creación, selecciona también el equipo de colaboradores que redactaran dichos materiales y establece el enfoque y estructura de los mismos, determina la estructura de aulas necesaria para la impartición de la asignatura y diseña las actividades de evaluación del curso, además de establecer una planificación del curso para asegurar el progreso académico de los estudiantes que cursan la asignatura siguiendo las recomendaciones de estudio pautadas.

Aunque no tiene importancia para el caso de ejemplo seleccionado, mencionar que a parte de las múltiples tareas docentes que realiza el profesor también es investigador y participa en proyectos de investigación e innovación docente.

- 2) El consultor es un colaborador docente seleccionado por el profesor para impartir la asignatura. Es decir una persona que conoce los contenidos de la asignatura y que en caso de no haber impartido docencia en el entorno virtual de aprendizaje es formada previamente para poder impartir docencia en este entorno.
El consultor interactúa directamente con el estudiante, guía su proceso de aprendizaje, dinamiza el aula, resuelve dudas, prepara y corrige las actividades del curso. Sigue las indicaciones del PRA y es supervisado por este. En caso de formar parte de un equipo de varios consultores de una misma asignatura, trabaja asincrónicamente en grupo y debe seguir el calendario de trabajo internamente establecido por el PRA.
- 3) El tutor es también un colaborador docente conocedor del ámbito disciplinar y del entorno virtual, en caso contrario es formado para ello. Su principal misión es acompañar al estudiante a lo largo de sus estudios, asesorarle y orientarle en su trayectoria académica. No está relacionado con una asignatura completa, sino que su relación es con una o más titulaciones en general. Por consiguiente, esta figura no se tendrá en cuenta en el caso de ejemplo que se ha seleccionado.
- 4) El autor de los materiales de aprendizaje que se crean para una asignatura dada es un experto en el ámbito de conocimiento de esta materia. Es seleccionado por el profesor y recibe el encargo de preparación de los mismos. Puesto que la asignatura Bases de Datos I es una asignatura consolidada con materiales ya creados y probados, esta figura tampoco interviene en el caso de ejemplo.

El equipo docente de la asignatura está formado por el profesor y el equipo de consultores a quienes lidera, coordina y supervisa en la impartición de la docencia. Puesto que la asignatura que se desea preparar, Bases de datos I, es una asignatura obligatoria y troncal de los Estudios de Informática, el número de estudiantes matriculados en ella es importante y deben distribuirse en varias aulas atendiendo a los ratios que establece la organización. Además, dado el carácter práctico de esta asignatura, existe la necesidad de más de un tipo de aula, en este caso aulas de teoría y aulas de laboratorio.

Por otro lado, la virtualidad conlleva a extender el concepto de clase como lugar de encuentro y de comunicación entre los estudiantes y el profesor de una asignatura. El aula virtual de la UOC no sólo tiene que dar respuesta a la necesidad de comunicación entre los miembros del aula, sino que también debe de ofrecer algún espacio para la planificación, evaluación y disposición de recursos del aula. Por tanto, la estructura del aula virtual va más allá de lo que sería la estructura de una clase convencional.

La comunicación en el aula se lleva a cabo mediante unos espacios virtuales. Uno de ellos, el tablón, hace las veces de pizarra y en él sólo publica mensajes el consultor o el profesor de la asignatura. En el caso que nos ocupa, la preparación del curso implica preparar el aula para recibir a los estudiantes el primer día de curso y por tanto, asegurarse que en él se encuentra como mínimo el mensaje de bienvenida al curso y unas primeras indicaciones sobre el funcionamiento del mismo. El otro espacio de comunicación es el foro a través del cual pueden intercambiar mensajes relativos a la asignatura y el curso todos los estudiantes del curso.

La descripción de cada curso y los compromisos que se establecen entre el profesorado y los estudiantes matriculados en el curso se recogen en el plan docente. En el caso de la UOC, el plan docente de curso consta de dos partes, una parte estática que se mantiene semestre a semestre y una parte variable, ésta última sujeta a revisión cada curso académico. Como parte estática del plan docente incluye aquellas secciones que hacen referencia a la asignatura con independencia de la edición: 1) descripción de la asignatura, 2) objetivos de aprendizaje de la misma, 3) ubicación de la asignatura dentro del plan de estudios y 4) metodología de la asignatura. En cambio, como parte variable del plan docente se detallan los aspectos que son específicos del curso o edición de la asignatura: 1) evaluación del curso, 2) recursos docentes recomendados para el mismo y 3) el calendario de curso.

El calendario de curso establece de forma gráfica el conjunto de fechas clave para el estudiante. En él se indica en distinto color el día de publicación de enunciados, ya sean actividades a realizar e o soluciones de las mismas, el día límite para la de entrega de actividades evaluables y/o el día en que se van a publicar las notas. Mencionar que dicho calendario de curso está sujeto al calendario académico institucional y las normas que

internamente pueden establecerse dentro de cada estudio, como por ejemplo el número máximo que debe de transcurrir desde la entrega de una actividad por parte de los estudiantes y la publicación de la solución oficial de la misma.

Existen varios modelos de evaluación en la UOC pero todos ellos incorporan la evaluación continua. En el caso que nos ocupa se trata de una evaluación convencional con examen final y actividades de curso evaluables, entre las cuales están las prácticas que forman parte de la nota final de curso, por lo que se exige para la superación del curso la obtención de una nota mínima en este tipo de actividad.

Más adelante, en el capítulo 5 cuando al describir el dominio de conocimiento de la UOC se entra en más detalle sobre los distintos tipos de actividades de evaluación continua. No obstante, mencionar que tanto el tipo de evaluación como la ponderación de las distintas actividades de evaluación del curso deben de quedar explicitadas en el plan docente y que en el aula virtual existe un espacio para la introducción y gestión de notas de los estudiantes matriculados en el curso.

En el plan docente también se incluye la referencia de los recursos docentes disponibles en el aula. En el aula hay un espacio dedicado a la publicación de los materiales de soporte del curso que se actualiza cada curso académico. Posteriormente en el capítulo 5, al describir el dominio de conocimiento de la UOC, se detallan algunos de los distintos tipos de materiales que utiliza esta institución y en concreto los que se utilizan en la asignatura de Bases de Datos I.

Finalmente, mencionar que la preparación de un curso implica un trabajo en equipo que suele llevarse a cabo asincrónicamente entre los miembros del equipo docente de la asignatura con el apoyo del personal de gestión y que es liderado por el profesor responsable de asignatura. Este proceso implica el establecimiento de un calendario de trabajo interno para la preparación de las actividades del curso y de los exámenes. La distribución de estas tareas la realiza el PRA y asigna encargos al equipo de colaboradores que tienen una fecha de inicio y de finalización. Algunos de estos encargos son propios de la asignatura y otros son más generales, establecidos por la organización. El detalle de los encargos docentes de la UOC se encuentra más adelante en el capítulo 5.

Conocido el contexto organizativo de la UOC y la terminología empleada para definir los elementos que componen un escenario educativo, se procede a describir en pocas líneas en qué consiste la escena de preparación de curso de Bases de Datos I.

Preparar el curso de Bases de Datos I significa organizar y planificar el curso al detalle para que el primer día de curso los estudiantes matriculados en el curso encuentren en el aula los recursos que necesitan para iniciar y seguir correctamente el curso, conozcan al consultor que les guiará y asesorará durante el proceso de aprendizaje, así como definir el plan docente que determina todos los aspectos del curso a modo de contrato entre estudiante y docentes del curso.

El plan docente es el resultado más significativo de la preparación de un curso. Éste contiene contenidos, objetivos del curso y competencias a desarrollar, el sistema de evaluación que se aplicará, la planificación del curso a nivel de actividades (PACs) y pruebas de evaluación final (EXA) y los recursos docentes recomendados.

Además, detrás de la parte visible al estudiante que descubre el primer día de curso, existe una parte que es transparente a él. Se trata de la organización docente del grupo de consultores, la preparación del plan docente, la asignación de aulas y el reparto de tareas docentes entre los miembros del equipo docente de Bases de Datos I. Se trata pues de un escenario donde el protagonista principal es el equipo docente liderado por el profesor responsable de asignatura (PRA).

3.5. Resumen

En este capítulo se ha presentado la motivación de este trabajo y se ha profundizado en el problema que pretende resolver con el objetivo de trazar una estrategia de resolución para el mismo. También se ha introducido el contexto de la UOC para comprender el caso de ejemplo que se utilizará a lo largo del presente trabajo y para la validación de la solución propuesta.

Capítulo 4

Ontología para la especificación de patrones de escenas educativas

If I have seen further, it is by standing on the shoulders of giants.

Isaac Newton

En este capítulo se presenta el primer nivel de la arquitectura ontológica propuesta en este trabajo de tesis: la ontología de patrones de escenas educativas (***OntoProcED***).

Esta ontología es el resultado de la integración de dos ontologías: la ontología de procesos (***OntoProc***) y la ontología de escenarios educativos (***OntoED***). Para cada una de ellas se presenta: dominio de conocimiento, formalización mediante diagrama UML y construcción en OWL.

4.1. Especificación de patrones de escenas educativas

En el segundo capítulo se ha visto que en la actualidad no existe ninguna técnica que permita la especificación de escenas educativas capaz de soportar la automatización total o parcial de dichas especificaciones. Tampoco existe en el ámbito de la tecnología educativa ningún lenguaje para describir los procesos que habitualmente acontecen en entornos educativos. Por tanto, para resolver el problema planteado en este trabajo de tesis, se requiere de algún mecanismo para la especificación de escenas educativas en general.

A continuación, se presentan algunos conceptos relacionados con esta temática como

punto de partida para la obtención de una definición del concepto de escena educativa que sea válida para el propósito de este trabajo:

Proceso (*processus* del latín):

- 1) Acción de ir adelante.
- 2) Transcurso del tiempo.
- 3) Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial.

Escenario:

- 1) Parte del teatro construida y dispuesta convenientemente para que en ella se puedan colocar las decoraciones y representar las obras dramáticas o cualquier otro espectáculo teatral.
- 2) Lugar en que ocurre o se desarrolla un suceso.
- 3) Conjunto de circunstancias que rodean a una persona o un suceso.

Escena:

- 1) Sitio o parte del teatro en que se representa o ejecuta la obra dramática o cualquier otro espectáculo teatral. Comprende el espacio en que se figura el lugar de la acción a la vista del público.
- 2) Aquello que se representa en el escenario.
- 3) Cada una de las partes en que se divide el acto de la obra dramática, y en que están presentes unos mismos personajes.
- 4) Suceso o manifestación de la vida real que se considera como espectáculo digno de atención.

De acuerdo con estas definiciones entenderemos que una escena educativa es *un conjunto de actividades que ocurren en un entorno de aprendizaje con el objetivo de preparar el entorno de aprendizaje, ofrecer soporte durante el proceso de aprendizaje y evaluar los resultados obtenidos y las competencias adquiridas durante el proceso de aprendizaje.*

Además, entenderemos por patrón de escena educativa la descripción de una escena genérica propia de cualquier entorno de aprendizaje, es decir un conjunto de actividades relacionadas con el aprendizaje que se repiten en distintos entornos educativos, ya sea antes, durante o después del proceso de aprendizaje. Por tanto, en la especificación de patrones de escenas educativas deberán de tenerse en cuenta las actividades comunes que se realizan en distintos entornos educativos, las figuras características del entorno de aprendizaje y los recursos de este tipo de entornos que se utilizan para alcanzar el objetivo que se perseguido.

Así pues, para obtener especificaciones de patrones de escenas educativas se ha optado por la utilización de ontologías como mecanismo de representación y formalización de especificaciones. Éstas resultan idóneas para formalizar y compartir conocimiento, puesto que permiten generar descripciones de precisas, concisas y no ambiguas, interpretables por una máquina. Además, una ontología para la especificación de patrones de escenas en entornos educativos facilitará la transcripción de especificaciones a código ejecutable, la compartición y reutilización de dichas especificaciones, promoverá la creación de un catálogo de especificaciones de procesos genéricos para entornos educativos y podrá utilizarse como herramienta para la validación del comportamiento del sistema previa implementación de sus funcionalidades.

La necesidad de una ontología para la especificación de procesos contextualizados en escenarios educativos se presentó en (Rius et al. 2008a, Rius et al. 2007a) y se justificó a partir de las respuestas a las preguntas que según Noy & Mc.Guinness (Noy & McGuinness 2001) deben formularse antes de crear una ontología: 1)¿qué uso va a tener la ontología?, 2)¿a qué tipo de preguntas debe de dar respuesta? y 3)¿quién va a utilizarla?.

De acuerdo con las definiciones del RAE mencionadas anteriormente, se deduce que una escena está formada por procesos y escenarios. Concretamente, en nuestro caso se trata de describir los patrones de escenas educativas en términos de secuencias de procesos reutilizables y luego contextualizarlos en escenarios educativos. Atendiendo a esta doble necesidad y procurando una arquitectura ontológica modular que facilite la reusabilidad de las especificaciones de escenas educativas se propone la creación de las siguientes dos ontologías:

- 1) **OntoProc**: Ontología de procesos para la secuenciación y composición de patrones de escenas en entornos educativos con el fin de especificar y formalizar de secuencias de procesos a partir de la composición de procesos reutilizables.
- 2) **OntoED**: Ontología de escenarios educativos para la descripción del contexto en que se llevan a cabo los procesos en un entorno educativo. Su objetivo es especificar y formalizar escenarios educativos, incluyendo participantes y recursos del entorno educativo.

La integración de estas ontologías dará origen a la ontología de patrones de escenas educativas u ontología de procesos para entornos educativos que denominaremos **OntoProcED**.

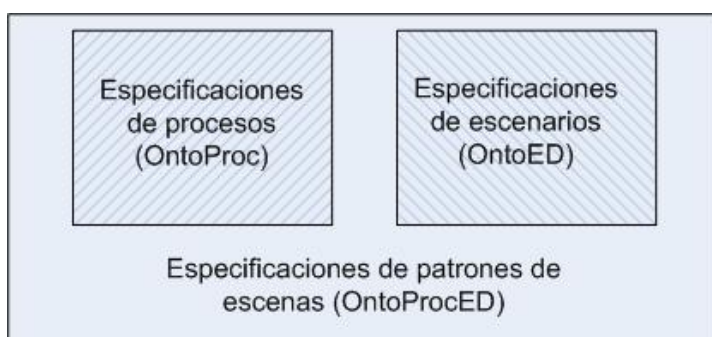


Figura 4.1: OntoProcED como integración de OntoProc y OntoED

La ontología de procesos para entornos educativos, constituye el primer nivel de la ontología multinivel propuesta en este trabajo de tesis y se obtendrá a partir de la integración de dos ontologías tal como indica la figura 4.1. Esta ontología permite la definición de especificaciones de procesos genéricos propios de entornos educativos, adaptables a cualquier institución de enseñanza superior independientemente de la plataforma de sistema de aprendizaje.

4.2. La ontología de procesos: OntoProc

La ontología de procesos (OntoProc) es una ontología de dominio (Guarino 1998) cuyo objetivo es la especificación formal de procesos en términos de secuencias de procesos reutilizables.

Esta ontología describe conceptos relacionados con los procesos y su composición, desde los elementos que intervienen o se utilizan en cada proceso hasta los elementos requeridos para poder componer nuevas especificaciones de procesos.

Para comprender mejor a qué nivel deben de describirse los procesos a continuación se presenta a modo de ejemplo el proceso *DistribuirTareasEntreProfesores* cuyo objetivo es la distribución de tareas entre los miembros del equipo docente de una asignatura.

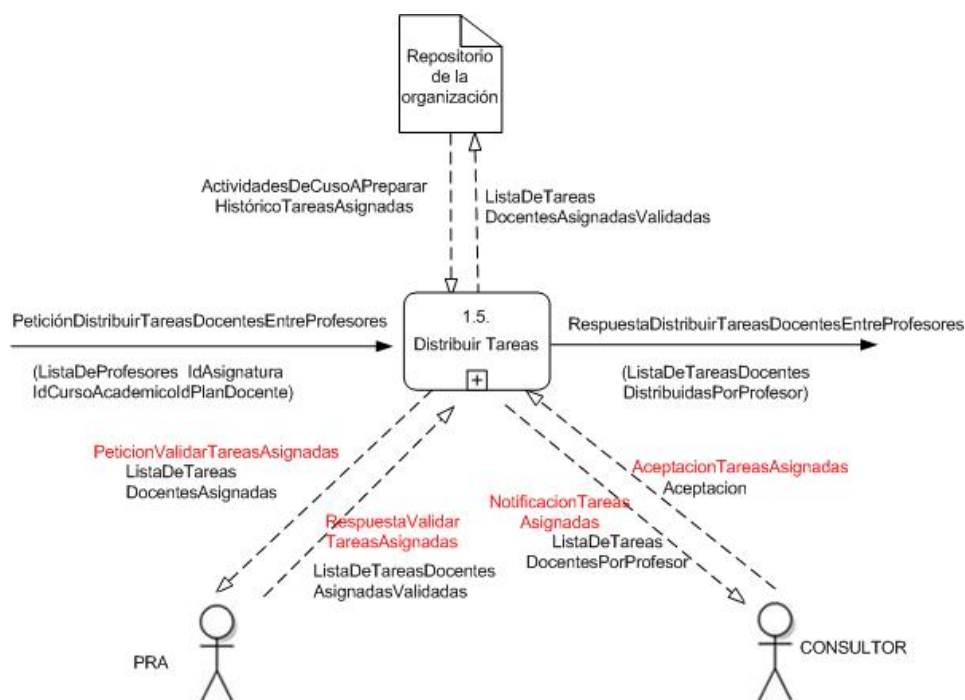


Figura 4.2: DistribuirTareasEntreProfesores

En la figura 4.2 se visualiza gráficamente el proceso *DistribuirTareasEntreProfesores*. Se observa que, como fuente de datos, se utiliza el repositorio UOC y que intervienen como agentes participantes: el profesor responsable de asignatura y el profesor que imparte la docencia.

El proceso *DistribuirTareasEntreProfesores* se inicia tras la recepción del mensaje de petición (*PeticiónDistribuirTareasAProfesores*) y finaliza con el mensaje de respuesta (*RespuestaDistribuirTareasAProfesores*). Además, se observa que, como fuente de datos, se utiliza el repositorio de la organización y como participantes intervienen: el profesor coordinador de asignatura y el profesor que imparte la asignatura.

La interacción de los participantes con el proceso se realiza mediante el envío de mensajes. Tanto el profesor coordinador de asignatura como el profesor interactúan con el proceso. En el caso del profesor coordinador éste recibe la petición de validación de tareas asignadas (*peticiónValidarTareasAsignadas*) con la lista de tareas asignadas pendientes de validar y envía la correspondiente respuesta (*RespuestaValidarTareasAsignadas*) con la lista de tareas asignadas validadas, mientras que el profesor recibe la notificación de tareas asignadas (*NotificaciónTareasAsignadas*) con la lista de tareas asignadas por profesor y envía como respuesta un mensaje de confirmación (*RespuestaAceptacionTareasAsignadas*) con la correspondiente aceptación.

Además, se observa que el proceso requiere dos tipos de recursos: 1) recursos que constituyen una fuente de datos como es el caso del repositorio de la organización y 2) recursos contenidos en mensajes como pueden ser la lista de tareas docentes asignadas (*ListaDeTareasAsignadas*).

A partir de la información extraída del repositorio (*HistóricoTareasAsignadas* y *ActividadesDeCursoAPreparar*) y de la información contenida en el mensaje que desencadena el proceso (*ListaDeProfesores*, *IdAsignatura*, *IdCursoAcademico* e *IdPlanDocente*) se elabora automáticamente una propuesta de asignación de tareas que el profesor coordinador debe de validar. Realizada la validación se notifica a cada profesor la asignación de tareas que le corresponde y éste acepta o rechaza dicha asignación. Como resultado se genera el mensaje de finalización de dicho proceso, el cual contiene la lista de tareas docentes distribuidas por profesor (*ListaDeTareasDocentesDistribuidasPorProfesor*).

Sin embargo, este proceso también puede representarse teniendo en cuenta los procesos que lo componen y como se organizan para describir las secuencias de procesos que reutiliza. En la figura 4.7 se representa el proceso *DistribuirTareasEntreProfesores* como una secuencia de dos procesos: *IdentificarTareas* y *AsignarTareas*.

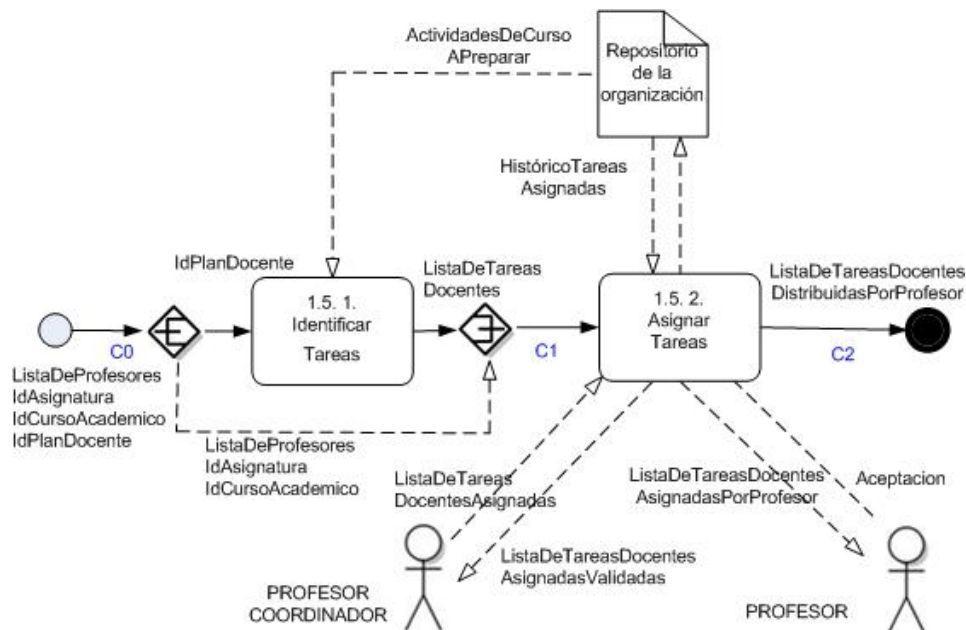


Figura 4.3: Secuencia que implementa DistribuirTareasEntreProfesores

Tal como puede observarse en la figura 4.7, dentro de la secuencia que implementa el proceso complejo *DistribuirTareasEntreProfesores* se establece un orden de precedencia entre los procesos componentes, determinando que la identificación de tareas es previa a su asignación de las mismas entre los profesores del curso.

También puede observarse que ambos procesos componentes se conectan mediante flujos de control que llevan asociados los correspondientes flujos de datos. Para conseguir que a la entrada de cada proceso figuren los datos que se espera, se introducen unos elementos que permiten crear nuevos flujos de datos, los conectores.

El conector C0 es un conector que permite desagregar un flujo de datos en varios separando los recursos que éste contiene. En este caso, el flujo de entrada proviene de la salida del proceso inicial, que no es más que el mensaje *peticionDistribuirTareasEntreProfesores* del proceso complejo. Como salidas tiene dos flujos, uno formado por el identificador de plan docente del curso y el otro por la lista de profesores del curso entre los que tienen que repartirse las tareas, el identificador de asignatura y de curso académico del curso.

El conector C1 a la inversa que C0, agrega varios flujos de entrada en un único flujo

de salida. Así pues, a partir de la lista de tareas de docentes retornada por el proceso *IdentificarTareas* y uno de los flujos disgregados por C0 se constituye un único flujo de salida que contiene la lista de profesores del curso, el identificador de asignatura, el identificador de curso académico y la lista de tareas docentes que se alineará con la entrada del proceso *AsignarTareas*.

El conector C2 es el conector más simple, permite avanzar el flujo de entrada del conector a la salida del mismo con el mismo contenido, de manera que el mensaje generado por el proceso *AsignarTareas* llega al proceso final y por tanto coincide con el flujo de salida del proceso compuesto *DistribuirTareasEntreProfesores*.

4.2.1. El dominio de conocimiento de OntoProc

El dominio de conocimiento de la ontología de procesos se centra en el concepto de especificación de proceso y su reutilización para formar nuevas especificaciones de procesos.

Tal como se ha visto al describir el proceso *DistribuirTareasEntreProfesores*, todo proceso viene definido por el mensaje que lo desencadena y el mensaje o mensajes que genera, los recursos que utiliza y los participantes que interaccionan con él. En consecuencia, los principales conceptos de este dominio son:

- 1) **Participante:** usuario o sistema que colabora en la consecución de un determinado objetivo relacionado con el aprendizaje. Por tanto, si se considera el LMS como el elemento que orquesta un proceso, un participante es una entidad externa al LMS con la cual éste interacciona, por ejemplo un profesor o instructor, un estudiante o un sistema de autoría.
- 2) **Mensaje:** flujo de información que intercambia un proceso. Si el sentido es entrante, se trata de un flujo de información que inicia un proceso o que es recibido por algún participante que colabora en él. Si el flujo es saliente, se trata de un mensaje generado por el proceso como resultado del mismo o de un mensaje enviado a algún participante de dicho proceso.
- 3) **Recurso:** elemento del entorno de aprendizaje que el proceso utiliza para conseguir

su cometido. Así pues, un recurso podría ser un repositorio de actividades de aprendizaje, una lista de actividades a preparar o el código de una asignatura por citar algunos ejemplos de distinta granularidad.






- 4) **Proceso:** acción que se define en términos de mensaje entrante que lo desencadena, mensajes salientes (más de uno en caso de elevarse alguna excepción) que genera, recursos que utiliza y participantes que interactúan con él.
- 5) **Proceso Complejo:** acción definida por una secuencia de procesos que se combinan mediante conectores para formar un nuevo proceso y alcanzar un determinado objetivo que no es alcanzable por medio de un sólo proceso.

Puesto que un proceso complejo puede representarse en términos de sus componentes interesa definir los conceptos relacionados con la secuencialización de los procesos que reutiliza, por lo que es necesario introducir otros dos conceptos:

- 1) **Secuencia:** forma de representar la implementación de un proceso complejo. Toda secuencia empieza con un proceso genérico inicial y finaliza con un proceso genérico final y además, contiene una serie ordenada de conectores que relaciona los procesos que componen.
- 2) **Conector:** elemento dentro de una secuencia que se caracteriza por enlazar o conectar uno o más elementos de entrada con uno o más elementos de salida. El objetivo de un conector es alinear flujos entrantes y salientes de los procesos que se secuencializan de acuerdo con el flujo de control establecido.

El conector suele tener como entrada y salida un proceso, no obstante en ciertas ocasiones, por ejemplo cuando hay que combinar varios flujos de datos antes de proceder a paralelizar procesos componentes o para sincronizar secuencias dentro de una misma secuencia, se crea una secuencia de conectores que no intercala ningún proceso. Para contemplar estos casos la entrada y salida de un conector puede ser tanto un conector como un proceso.

La semántica del conjunto de conectores creado para este trabajo se presenta en la tabla 4.1.

Tipo de conector	Símbolo	Descripción
Desagregación		Establece un orden de precedencia entre un elemento y distintos elementos siguientes. Existen tantos flujos salientes como elementos siguientes y el contenido de éstos siempre es un subconjunto del flujo entrante.
Agregación		Establece un orden de precedencia entre varios elementos y el siguiente. El flujo de control saliente es el que resulta de la unión de todos los flujos entrantes de los elementos precedentes.
Or		Establece un orden de precedencia entre un elemento y varios elementos siguientes en función del cumplimiento o no de una condición. Existen tantos flujos salientes como elementos siguientes, cuya condición asociada ha evaluado cierto y el contenido de estos flujos es el mismo que el del flujo entrante.
Xor		Establece un orden de precedencia entre un elemento y varios elementos siguientes. El flujo de datos entrante sólo fluye por el único elemento cuya condición asociada evalúa cierto
Discriminación		Establece un orden de precedencia entre varios elementos y un elemento siguiente. El flujo saliente coincide con el del flujo entrante correspondiente al flujo que fluye primero

Cuadro 4.1: Tipos de conectores de secuencia

Aunque podrían definirse gran cantidad de conectores, en este trabajo se ha optado por un conjunto mínimo de ellos, cinco concretamente. El objetivo de este trabajo de tesis no es obtener una clasificación exhaustiva de conectores, sino ofrecer aquellos conectores que permitan representar la mayoría de especificaciones de procesos para demostrar la factibilidad de la técnica propuesta.

Por ejemplo, obsérvese que no existe ningún conector para representar la elección de una alternativa según múltiples condiciones, sin embargo dicha elección múltiple podría simularse mediante una secuencia de condicionales que incluya todas las posibles combinaciones de condiciones que pudieran satisfacerse a la vez, desde el caso en que se satisfagan todas ellas hasta el caso en que se satisfaga sólo una de ellas.

4.2.2. Formalización de OntoProc

El diagrama de clases UML obtenido a partir de la descripción del dominio de conocimiento de la ontología de procesos descrita anteriormente es el que se presenta en la figura 4.4.

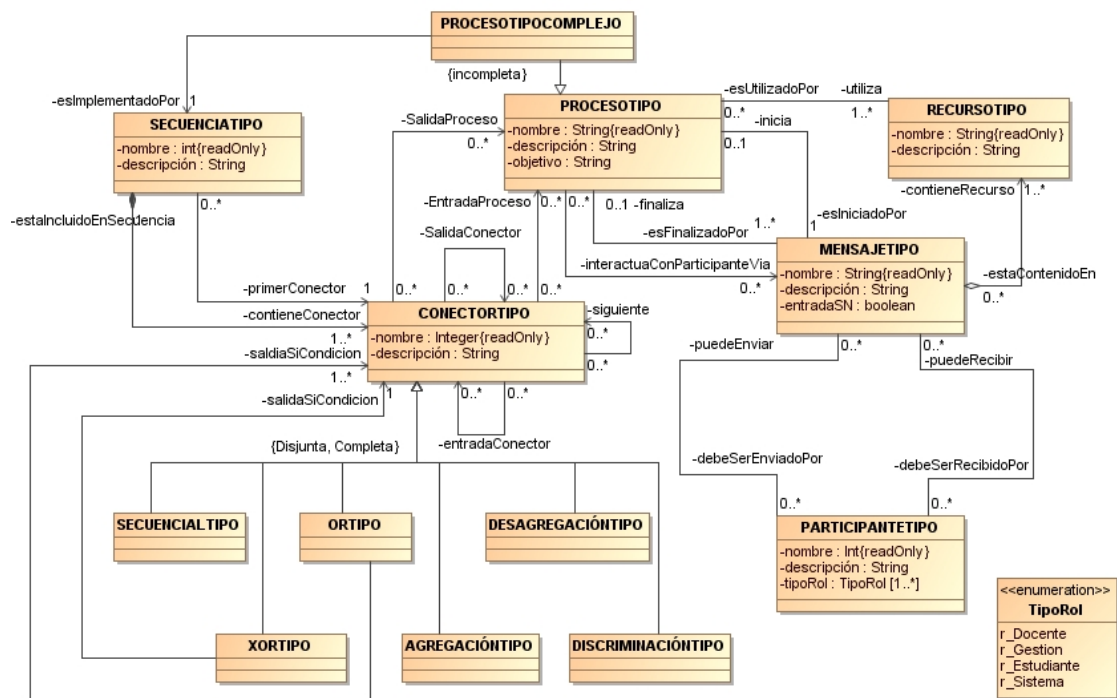


Figura 4.4: Ontología de Procesos (OntoProc)

Este diagrama UML también puede encontrarse ampliado en el apéndice A, concretamente en la figura A1.

A continuación, se describe las clases y posteriormente los tipos de relación establecidos entre ellas. Para cada clase se presenta una breve descripción, se enumeran sus propiedades indicando el tipo de dato que le corresponde y sus restricciones. Análogamente, para cada tipo de relación entre clases se describe el tipo de relación, se indica el dominio y rango de la misma y las posibles restricciones de cardinalidad o simetría en el caso de existir.

■ Clase *ProcesoTipo*

La clase *ProcesoTipo* especifica una acción que se lleva a cabo con el fin de alcanzar un objetivo. Su finalidad es la especificación de procesos reutilizables en entornos educativos. Es el principal concepto de esta ontología.

Para comprender mejor este concepto y los demás conceptos relacionados que se describirán a continuación, supóngase el proceso de asignación de profesores a las clases del curso (*Asignar profesores a clases*).

En la figura 5.1 puede observarse una representación gráfica del proceso *Asignar profesores a clases* en términos de sus interacciones con participantes externos (Profesor coordinador, Profesor) y los flujos de datos entrantes (*PeticionAsignarProfesoresClases*) que lo desencadenan y salientes (*RespuestaAsignarProfesoresAClases*) que genera.

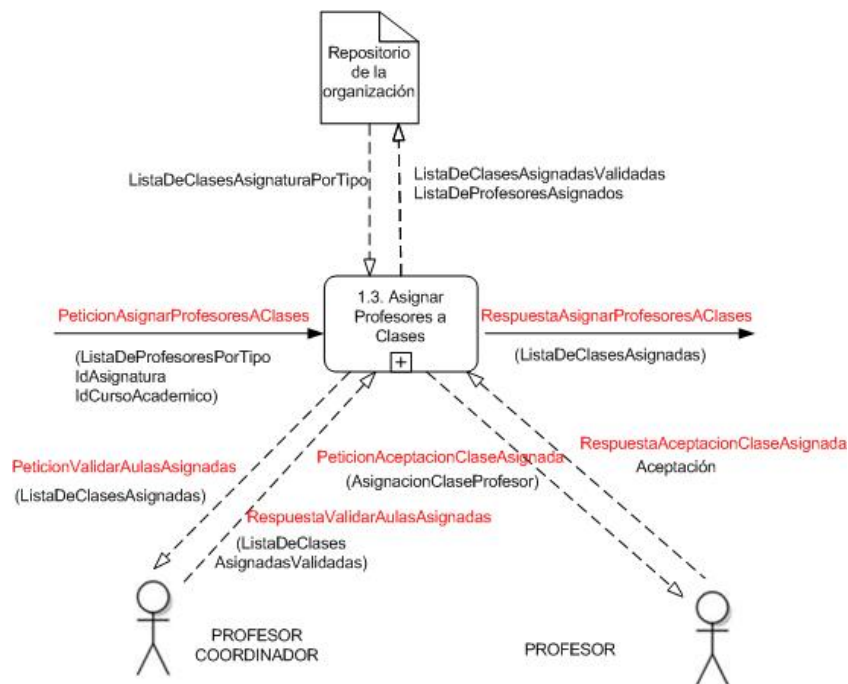


Figura 4.5: Asignar profesores a clases

Todo proceso tiene como propiedades un nombre que lo identifica unívocamente, una descripción y un objetivo tal como se describe en la tabla 4.2.

El proceso de la figura 5.1 tiene por nombre 'Asignar profesores a clases', descripción

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de proceso	Texto	Cardinalidad=1 Es identificador
descripcion	Descripción del tipo de proceso	Texto	Cardinalidad=1
objetivo	Descripción del objetivo de un proceso	Texto	Cardinalidad=1

Cuadro 4.2: Propiedades de la clase ProcesoTipo

'Asignación de los profesores de un curso a sus correspondientes clases' y objetivo 'Asignar a cada clase del curso un profesor en activo del equipo docente de la asignatura'. °Se observa que dicho proceso se inicia con el mensaje *PeticionAsignarProfesoresAClases* y finaliza con el mensaje *RespuestaAsignarProfesoresAClases*. Más adelante se detallarán las interrelaciones entre clases.

■ Clase ParticipanteTipo

La clase ***ParticipanteTipo*** permite especificar tipos de participantes o figuras que tienen asignados un conjunto de permisos para el envío y recepción de mensajes dependiendo del rol o roles que se asumen.

En la tabla 4.3 se muestran las propiedades de la clase ParticipanteTipo.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre del tipo de participante	Numérico	Cardinalidad=1 Es identificador
descripcion	Descripción del tipo de participante	Texto	Cardinalidad=1
tipoRol	Tipo de rol que asume el tipo de participante	TipoRol	Card. Mín.1

Cuadro 4.3: Propiedades de la clase ParticipanteTipo

Cada participante tipo puede desempeñar uno o más roles, los valores de la propiedad tipoRol son los establecidos por la clase enumeración *TipoRol*: 'r_Docente', 'r_Gestion', 'r_Estudiante' o 'r_Sistema'.

Algunos ejemplos de participantes del entorno educativo podrían ser el profesor que asume rol docente, el técnico de soporte docente que asume rol de gestión, el estudiante

de una titulación concreta que asume rol de Estudiante o el profesor coordinador que asume dos roles, el docente y el de gestión.

▪ Clase MensajeTipo

La clase *MensajeTipo* especifica los flujos de información que inician o finalizan un proceso o que intercambian información entre participantes y proceso. Cada flujo de información contiene al menos un recurso.

Como propiedades cada mensaje tipo tiene un nombre, una descripción y un tipo que distingue entre mensajes entrantes o salientes tal como indica la tabla 4.4.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de mensaje	Texto	Cardinalidad=1 Es identificador
descripción	Descripción del tipo de mensaje	Texto	Cardinalidad=1
entradaSN	Tipo del mensaje. Es entrante si su valor es cierto y saliente en caso contrario	Booleano	Cardinalidad=1

Cuadro 4.4: Propiedades de la clase MensajeTipo

Supongamos el proceso (*Asignar profesores a clases*) al que nos venimos refiriendo. Dicho proceso es iniciado por el mensaje *PeticiónAsignarProfesoresAClases* que contiene la lista de profesores, la lista de clases, el identificador de asignatura y del curso académico. El sistema realiza una asignación automática que el profesor coordinador del curso debe de validar. Validada la asignación de clases por el profesor coordinador, el proceso envía una notificación a cada profesor indicando la clase que le ha sido asignada y éste responde con su aceptación. Aceptadas todas las asignaciones de aulas, el proceso finaliza y se envía el mensaje *RespuestaAsignarProfesoresAClases*.

Por tanto, este proceso requiere de la interacción de dos tipos de participantes: los profesores y el profesor coordinador de asignatura.

El Profesor coordinador interactúa con el proceso mediante la recepción del mensaje *PeticiónValidarClasesAsignadas* para validar la lista de clases asignadas y responde con el mensaje *RespuestaValidarClasesAsignadas* que devuelve al proceso la lista asignaciones de clases validadas.

El profesor por su parte interactúa con el proceso al recibir el mensaje *PeticionAceptacionClaseAsignada* que contiene la notificación de su asignación a una clase y como respuesta envía el mensaje *RespuestaAceptacionClasesAsignadas*.

■ Clase RecursoTipo

La clase **RecursoTipo** permite especificar los recursos de un entorno de aprendizaje. Todo recurso tiene un nombre que lo identifica y una descripción.

En la tabla 4.5 se muestran las propiedades que caracterizan la clase RecursoTipo.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de recurso	Texto	Cardinalidad=1 Es identificador
descripción	Descripción del tipo de recurso	Texto	Cardinalidad=1

Cuadro 4.5: Propiedades de la clase RecursoTipo

La definición de recursos a este nivel es muy básica, pero será refinada en un segundo nivel de la ontología multinivel ya sea a nivel de cada institución o de acuerdo con algún estándar como puede ser IEEE-LOM.

Para comprender mejor esta clase supongamos de nuevo el proceso Asignar profesores a clases. En él se utiliza el repositorio de la organización como fuente de información, así como otros recursos que están contenidos en mensajes. Por ejemplo, los mensajes *PeticionValidarAulasAsignada* y *RespuestaValidarAulasAsignadas* contienen el recurso lista de asignaciones de profesores a clases (*ListaDeAsignacionesProfesoresAClases*), el mensaje *NotificacionAceptacionClaseAsignada* que contiene la asignación de clase de cada profesor (*AsignacionClaseProfesor*) y el mensaje *AceptacionClaseAsignada* con la aceptación de la asignación.

Además, el mensaje de inicio del proceso *PeticionAsignarProfesoresAClases* contiene los recursos: lista de profesores (*ListaDeProfesores*), el identificador de asignatura (*IdAsignatura*), el identificador de curso académico (*IdCursoAcademico*) y la lista de clases de la asignatura pendientes de asignar (*ListaClasesAsgnatura*). El mensaje de

finalización de proceso *RespuestaAAsignarProfesoresAClases* por su parte contiene la lista de clases asignadas(ListaDeClasesAsignadas).

■ Clase *ProcesoTipoComplejo*

La clase *ProcesoTipoComplejo* permite especificar procesos complejos a partir de secuencias de procesos más simples. Es una especialización de la clase *ProcesoTipo*.

Supóngase el proceso *Distribuir tareas entre profesores* presentado en la figura 4.2 cuyo objetivo es el reparto de tareas docentes entre todos los profesores que forman el equipo docente del curso.

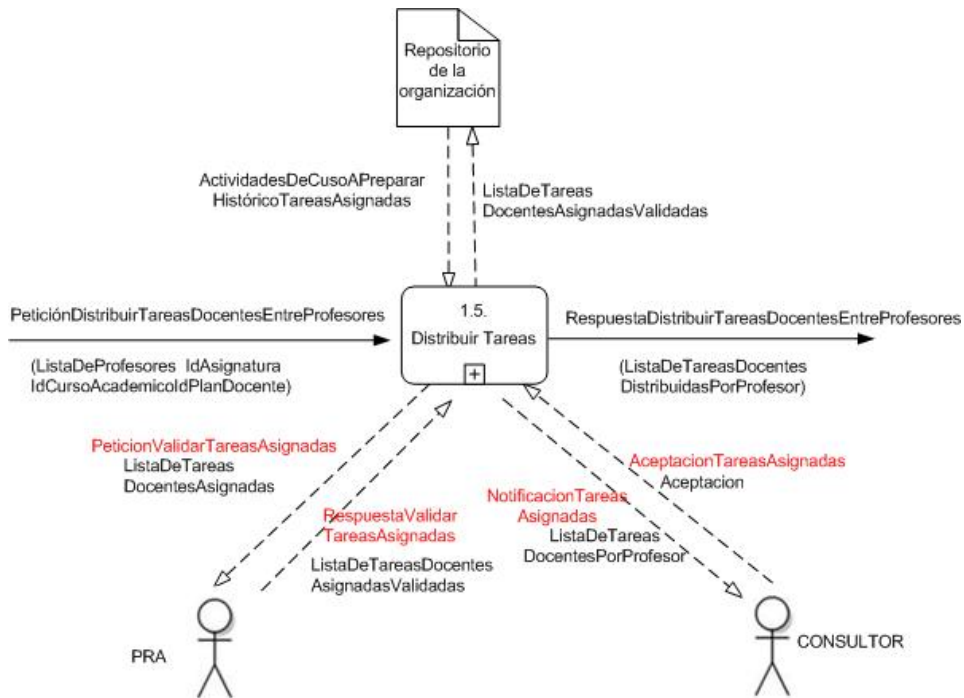


Figura 4.6: Distribuir tareas entre profesores

Todo proceso complejo puede representarse como una secuencia de procesos reutilizables que se inicia mediante un proceso genérico Inicial y finaliza con el proceso genérico Final. Sin embargo, no añade nuevas propiedades a las de la clase *Proceso*.

En la figura 4.7) se muestra la secuencia que implementa el proceso complejo *Distribuir tareas entre profesores*.

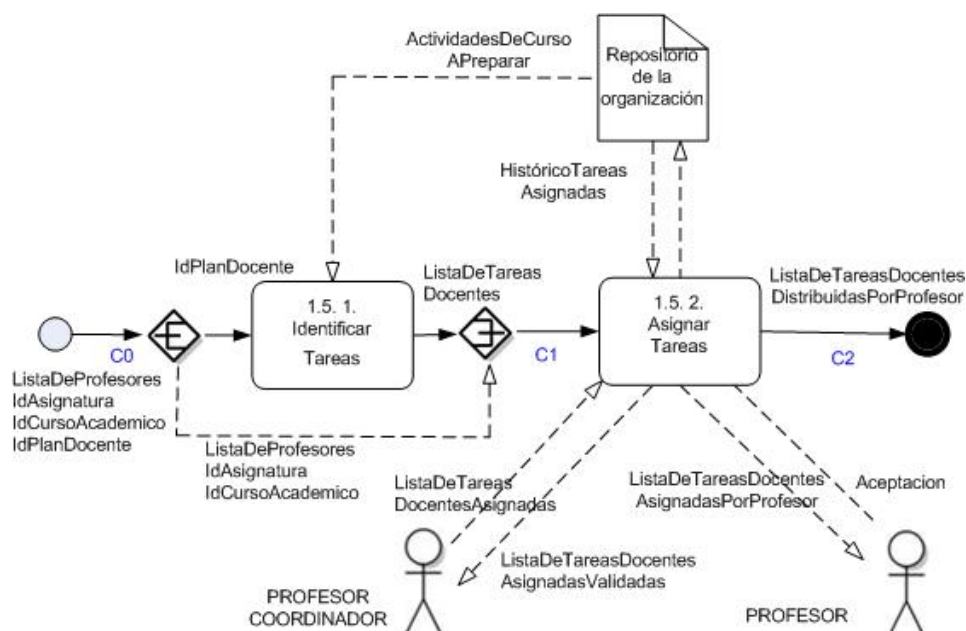


Figura 4.7: Secuencia que implementa Distribuir tareas entre profesores

Otro ejemplo sería el proceso del caso de ejemplo de este trabajo de tesis que entre otros componentes tendría el proceso *Distribuir tareas entre profesores*. Su descomposición en términos de los procesos que lo componen a varios niveles puede encontrarse en el apéndice A.

■ Clase *SecuenciaTipo*

La clase *SecuenciaTipo* describe una serie ordenada de conectores que relacionan los procesos para representar la implementación de un proceso complejo. Toda secuencia tiene un nombre y una descripción tal como se muestra en la tabla 4.6.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de secuencia	Texto	Cardinalidad=1 Es identificador
descripcion	Descripción del tipo de secuencia	Texto	Cardinalidad=1






Cuadro 4.6: Propiedades de la clase Secuencia

La secuencia correspondiente al proceso *Distribuir tareas entre profesores* está formada por los conectores C0, C1 y C2, siendo el conector C0 el que inicia la secuencia y por tanto relaciona el proceso genérico de inicio con el proceso *Identificar tareas*.

■ Clase ConectorTipo

La clase **ConectorTipo** describe aquellos elementos que forman parte de una secuencia relacionando procesos para establecer un orden entre ellos y alinear flujos de datos entrantes y salientes.

El conjunto de conectores seleccionado para este trabajo de tesis se presenta en la tabla 4.7. Para cada uno de ellos se indica el número de flujos de entrantes y salientes que relaciona y la restricción que determina los datos contenidos en el flujo o flujos salientes en función de los datos contenidos en el flujo o flujos entrantes.

Tipo de conector	Símbolo	Núm. entrantes	Núm. Salientes	Restricciones
DesagregaciónTipo		1	2 o más	El contenido de los flujos salientes es un subconjunto de todos los flujos entrantes
AgregaciónTipo		2 o más	1	El contenido del flujo de salida es la unión del contenido de todos los flujos entrantes
OrTipo		2 o más	1	El contenido del flujo saliente coincide con el contenido de uno o más flujos entrantes
XorTipo		1	1 o 2	El contenido de los flujos salientes coincide con el contenido de algunos de los flujos entrante
DiscriminaciónTipo		2 o más	1	El contenido del flujo saliente es el de alguno de los flujos entrantes

Cuadro 4.7: Tipos de conectores de secuencia

Las propiedades de esta clase son las descritas en la tabla 4.8.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre del conector	Texto	Cardinalidad=1 Es identificador
descripcion	Descripción del conector	Texto	Cardinalidad=1

Cuadro 4.8: Propiedades de la clase Secuencia

Como ejemplos de conectores pueden citarse aquellos que relacionan los procesos *Identificar tareas* y *Asignar tareas* en la secuencia que representa el proceso complejo *Distribuir tareas entre profesores*. Tal como se observa en la figura 4.7 el conector C0 distribuye el flujo de datos entrante de dicho proceso en otros dos flujos de datos, uno de los cuales alimenta al proceso *Identificar tareas*; el conector C1 que forma un flujo de datos a partir de otros dos flujos con el fin de iniciar el proceso *Asignar tareas* y el conector C2 que traslada el flujo de datos saliente del proceso *Asignar tareas* a la salida del proceso *Distribuir tareas entre profesores*.

■ Tipos de relación de OntoProc

A continuación, se describen los tipos de relación que se establecen entre las clases de OntoProc. Concretamente, en la tabla 4.9 se presentan los tipos de relación relacionados con la clase **ProcesoTipo** y en la tabla 4.10 los relativos a la clase **ProcesoComplejoTipo**.

En cuanto a procesos mencionar que todo proceso tiene un único mensaje de entrada que lo desencadena (*esIniciadoPor*) y uno o más mensajes de salida (*esFinalizadoPor*) que se generan como resultado. Y análogamente, un mensaje puede iniciar (*inicia*) o finalizar (*finaliza*) un proceso.

Además, un proceso para alcanzar un objetivo determinado debe de utilizar ciertos recursos (*utiliza*) o requerir la interacción de ciertos participantes que se comunican con el proceso a través de mensajes (*interactúaConParticipanteVia*).

Todo participante en el proceso asume al menos un rol. Este rol le otorga ciertos permisos para el envío (*puedeEnviar*) y recepción (*puedeRecibir*) de mensajes dentro del

Tipo de relación	Descripción	Dominio	Rango	Restricción
esIniciadoPor	Mensaje entrante que desencadena el proceso	ProcesoTipo	MensajeTipo	Cardinalidad=1
esFinalizadoPor	Mensaje saliente que genera como resultado un proceso	ProcesoTipo	MensajeTipo	Cardinalidad Mín.1
inicia	Proceso que incia	MensajeTipo	ProcesoTipo	Cardinalidad Max.1
finaliza	Proceso que finaliza	MensajeTipo	ProcesoTipo	Cardinalidad Max.1
utiliza	Recurso que utiliza un proceso	ProcesoTipo	RecursoTipo	Cardinalidad Mín.1
interactuaConParticipanteVia	Interacción entre participantes y procesos	ProcesoTipo	ParticipanteTipo	
puedeEnviar	Mensaje tipo que puede enviar un participante tipo	ParticipanteTipo	MensajeTipo	
puedeRecibir	Mensaje tipo que puede recibir un participante tipo	ParticipanteTipo	MensajeTipo	
debeSerRecibidoPor	Participante tipo que puede recibir este tipo de mensaje	MensajeTipo	ParticipanteTipo	
debeSerEnviadoPor	Participante tipo que puede enviar este tipo de mensaje	MensajeTipo	ParticipanteTipo	
contieneRecurso	Recurso contenido en un mensaje	MensajeTipo	RecursoTipo	Cardinalidad Mín.1
estaContenidoEn	Recurso contenido en un mensaje	RecursoTipo	MensajeTipo	
esUtilizadoPor	Recurso utilizado por un proceso	RecursoTipo	ProcesoTipo	

Cuadro 4.9: Tipos de relación entre las clases de OntoProc (I)

proceso. Visto desde el punto de vista del proceso para determinar la secuencia de interacciones interesa saber qué participantes tienen permiso de envío (*debeSerEnviadoPor*) o de recepción (*debeSerRecibidoPor*) en cada momento.

En cualquier caso, se dice que existe interacción entre procesos y participantes o participantes y procesos via mensajes (*interactuaConParticipanteVia*), independientemente de si son entrantes o salientes respecto al proceso.

Un recurso puede ser utilizado por algún proceso (*esUtilizadoPor*) y/o puede estar contenido en algún mensaje (*estaContenidoEnMensaje*). Además, los mensajes contienen recursos por lo que se define la relación (*contieneRecurso*).

En lo que se refiere a procesos complejos existen otros tipos de relación como se muestra en la tabla 4.10.

La implementación de un proceso complejo se representa mediante una secuencia (*estaImplementadoPor*). Los procesos para poder ser reutilizados como parte de alguna secuencia deben ser vistos como una caja negra. En este sentido interesa saber qué procesos

Tipo de relación	Descripción	Dominio	Rango	Restricción
<i>estaImplementadoPor</i>	Secuencia que implementa un proceso complejo	ProcesoComplejoTipo	SecuenciaTipo	Cardinalidad=1
<i>primerConector</i>	Primer conector de la secuencia	SecuenciaTipo	ConectorTipo	Cardinalidad=1
<i>siguienteConector</i>	Siguiente conector en la secuencia	SecuenciaTipo	ConectorTipo	Transitiva
<i>contieneConector</i>	Conectores contenidos en una secuencia	SecuenciaTipo	ConectorTipo	Cardinalidad Mfn.1
<i>estaIncluidoEnSecuencia</i>	Secuencia a la que pertenece el conector	ConectorTipo	SecuenciaTipo	Cardinalidad=1
<i>entrada</i>	Conectores o procesos a la entrada de un conector	ConectorTipo	ConectorTipo o ProcesoTipo	Cardinalidad Mfn.1
<i>salida</i>	Conectores o procesos a la salida de un conector	ConectorTipo	ConectorTipo o ProcesoTipo	Cardinalidad Mfn.1
<i>salidaSiCondicion</i>	Conectores o procesos a la salida de un conector	OrTipo XorTipo	ConectorTipo o ProcesoTipo	Cardinalidad=1
<i>salidaSiNoCondicion</i>	Conectores o procesos a la salida de un conector	OrTipo XorTipo	Conector o ProcesoTipo	

Cuadro 4.10: Tipos de relación entre las clases de OntoProc (II)

forman parte de una secuencia pero no en que secuencias se utilizan ciertos procesos, por lo que este tipo de relación no tiene navegabilidad en sentido contrario.

Las secuencias están formadas por procesos y conectores. Toda secuencia tiene al menos un conector (*contieneConector*) y uno de ellos es el primero (*primerConector*). Si la secuencia contiene más de un conector, entonces a excepción del último conector, existe un conector siguiente (*siguiente*) dentro de la secuencia.

Todo conector se caracteriza por tener como mínimo una entrada (*entrada*) y una salida (*salida*) y por estar incluido en una secuencia (*estaIncluidoEnSecuencia*).

El conector Or y Xor determinan el flujo o flujos salientes según el resultado de la evaluación de una condición, por lo que existe una relación entre conectores de tipo Or y Xor y la salida de éste, la cual depende de si la condición ha evaluado cierto (*salidaSiCondicion*) o si falso (*salidaSiNoCondicion*).

4.2.3. Construcción de OntoProc en OWL

De acuerdo con las indicaciones dadas en el apéndice B para obtener una ontología OWL a partir de un diagrama UML se ha construido en OWL la ontología OntoProc utilizando Protégé.

La ontología resultante puede resumirse mediante la jerarquía de clases que se presenta en la figura 4.8.

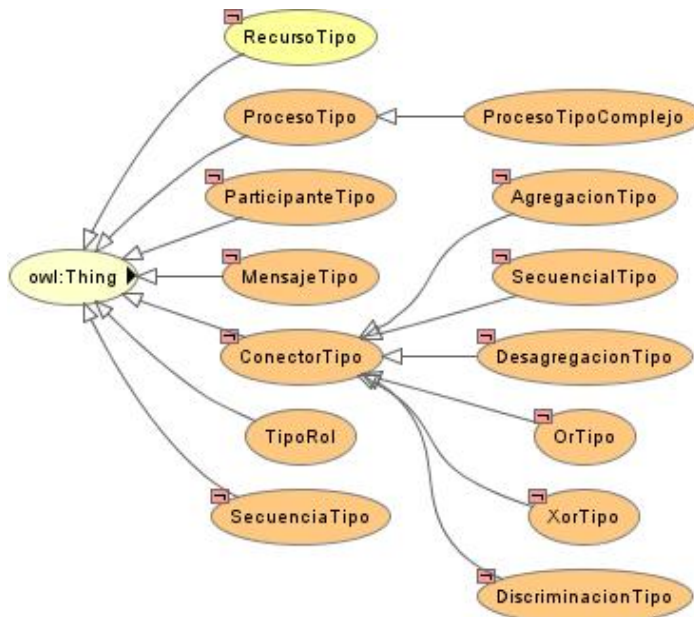


Figura 4.8: Jerarquía de clases y relaciones *is-a* de la ontología OntoProc

Tal como puede observarse en la jerarquía de clases de OntoProc el primer nivel de la ontología está formado por las clases: *ProcesoTipo*, *MensajeTipo*, *ParticipanteTipo*, *RecursoTipo*, *SecuenciaTipo* y *ConectorTipo* y la clase enumeración *TipoRol*. Las seis primeras clases son disjuntas entre sí. La clase enumeración *TipoRol* se relaciona con la clase *ParticipanteTipo* mediante la propiedad de objeto (*Object Property*) *tipRol* que establece los distintos roles que puede adoptar cada instancia de la clase *ParticipanteTipo*.

Los demás tipos de relación entre las clases del primer nivel son propiedades de objetos (*Object Property*) que se corresponden con los tipos de relación descritos en el apartado de la formalización de la ontología. Se ha decidido no mostrarlos en la figura anterior dado que no aportan nueva información y complican la legibilidad de la estructura ontológica.

En un segundo nivel se encuentran las especializaciones de las clases *ProcesoTipo*

y *ConectorTipo*. La clase *ProcesoTipo* tiene como subclase la clase *ProcesoComplejoTipo* a la que pertenecen todos los procesos que son implementados por alguna secuencia. La clase *ConectorTipo* en cambio se especializa en los distintos tipos de conectores. Los tipos de conectores seleccionados dan lugar a las clases: *SecuencialTipo*, *DiscriminaciónTipo*, *OrTipo*, *XorTipo*, *AgregacionTipo*, *DesagregacionTipo*.

En lo que respecta a propiedades *Datatype property* y *Annotation*, así como a las restricciones de cardinalidad de cada propiedad de la clase podría obtenerse una vista de propiedades de la ontología para cada clase. En la figura 4.9 se muestran la vista de la clase *ProcesoTipo*.

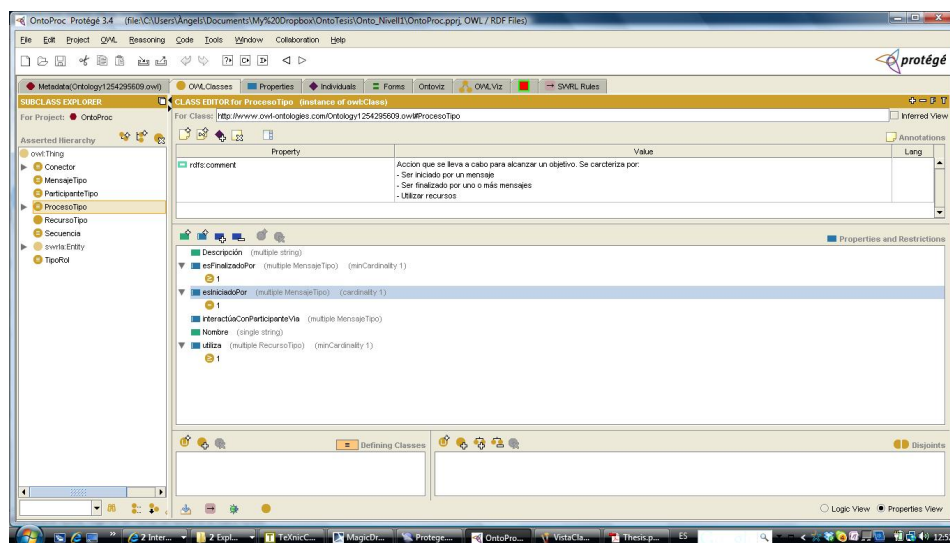


Figura 4.9: Vista de la clase *ProcesoTipo* con la herramienta Protégé

En el *frame* derecho se observa, en la parte superior, la definición de la clase como valor de la propiedad *Annotation* y debajo las propiedades de la clase. En azul las *Object Property* y en verde las *Datatype Property*, junto a las propiedades de objetos se visualizan las correspondientes restricciones de cardinalidad. Análogamente, podrían mostrarse todas las vistas de los restantes conceptos de la ontología pero no aportaría información adicional a la especificada en la formalización de la ontología.

Para el lector que desee profundizar en la ontología de procesos creada en OWL puede encontrarla en: <http://personal.uoc.edu/personalonto/ontologies>

4.3. La ontología de escenarios educativos: OntoED

El objetivo de la ontología de escenarios educativos es la especificación formal del contexto en que se llevan a cabo los procesos en un entorno educativo.

Crear una ontología válida para describir cualquier escenario educativo que pueda tener lugar en cualquier comunidad educativa de estudios superiores sería muy complejo. Por ello se propone la definición de una parte de esta ontología, concretamente aquella que permite especificar el escenario en que se representa mediante el patrón de escena preparación de un curso.

A continuación, se describe y formaliza el dominio de conocimiento de esta ontología desarrollada para el caso de ejemplo presentado en este trabajo.

4.3.1. El dominio de conocimiento de OntoED

Puesto que el objetivo de la ontología de escenarios educativos es la especificación formal del contexto en que se llevan a cabo los procesos en un entorno educativo, el dominio de conocimiento de la misma incluye todos aquellos conceptos que permiten contextualizar los procesos en el entorno en que se desarrollan, principalmente los participantes y los recursos que conforman el escenario.

El concepto fundamental de esta ontología obviamente es el curso. Un curso es una edición de una asignatura fijado un periodo académico específico, ya sea semestre o cuatrimestre o cualquier otro periodo académico que se quiera definir.

Las asignaturas forman parte de alguna titulación y cada titulación está formada por un conjunto de asignaturas. Cada asignatura dentro de una titulación tiene una ubicación en el plan de estudios de dicha titulación y puede ser obligatoria o no. Además, una misma asignatura puede ser compartida por más de una titulación, si los contenidos y objetivos de aprendizaje son los mismos para ambas titulaciones, en este caso se trata de asignaturas equivalentes.

Todo curso tiene un profesor coordinador que define y se responsabiliza del curso. Sin embargo, el curso puede ser impartido por uno o más profesores que constituyen el equipo docente para un curso dado.

La docencia exige la realización de un conjunto de tareas docentes que se planifican para ser realizadas dentro de un intervalo de tiempo y que se distribuyen entre los profesores encargados de impartir el curso. Algunas de estas tareas pueden estar relacionadas con las actividades asociadas al curso.

Cada curso tiene su propia planificación. La planificación del curso establece los eventos del curso. Cada evento es una fecha clave en el curso que tienen una descripción asociada. Existen eventos de distinto tipo y algunos de ellos están relacionados con las actividades que los estudiantes deben de realizar a lo largo del curso. Todo curso tiene siempre una planificación aunque inicialmente sólo existan las fechas de inicio y fin de curso.

Los estudiantes durante el proceso de matrícula se asocian a un curso y como resultado se les asigna un número de matrícula. Cuando hay un número mínimo de estudiantes que permite la apertura del curso, se establecen las clases. Una clase es un grupo de estudiantes que comparte docencia de un curso, en una misma ubicación y por parte del mismo equipo docente. Por tanto, existe una asignación docente de profesores a clases de un mismo curso.

Además, cada curso tiene definido un tipo de evaluación. La evaluación del curso se define en base a la evaluación de algunas de las actividades de curso. Los estudiantes pueden realizar las actividades propuestas y entregarlas para su evaluación. A cada actividad entregada le corresponde una fecha de entrega, comentarios al respecto de la evaluación y una calificación.

Opcionalmente, un curso puede tener asociados unos materiales, de igual manera que una asignatura. Y pueden existir materiales que no se correspondan con ningún curso o asignatura en un momento concreto.

Como resultado de la preparación del curso, se obtiene un plan docente. El plan docente contiene información sobre el curso en todos los aspectos del curso, considerándose como un contrato entre estudiantes matriculados en el curso y el profesorado lo imparte y coordina. Así pues, el plan docente contiene información textual sobre la asignatura.

El plan docente contiene información que se reutiliza en distintas ediciones de la asignatura e información que varía cada curso. Como ejemplo de información estable

tenemos la descripción de la asignatura o su ubicación dentro del plan de estudios, mientras que como información variable suele haber la planificación del curso, los detalles sobre las actividades a realizar y su evaluación, así como la metodología que se utilizará y los materiales recomendados.

La preparación de un curso, además de comportar la creación del plan docente, supone determinar los miembros del curso y/o de las clases.

Los miembros de un curso son el profesor coordinador del mismo, el profesor o profesores que imparten la docencia y por tanto tienen una asignación y/o tarea docente asignada y los estudiantes que se han matriculado en el curso. Todas estas personas que tienen acceso a recursos del curso forman el grupo de miembros del curso.

Análogamente sucede con el profesor coordinador del curso, los profesores que imparten docencia en la clase y los estudiantes asignados a dicha clase que tienen acceso a los distintos recursos relacionados con las clases. Todos ellos forman el grupo de miembros de una clase y la unión de todos los miembros de clase constituyen el grupo de miembros del curso.

4.3.2. Formalización de OntoED

A partir de la descripción del dominio de conocimiento de la ontología de escenarios descrita en la sección anterior, se obtiene su formalización mediante el correspondiente diagrama de clases UML. A continuación, en la figura 4.10, se presenta dicho diagrama de clases, el cual se encuentra también en el apéndice A de este trabajo, concretamente en la figura A2.

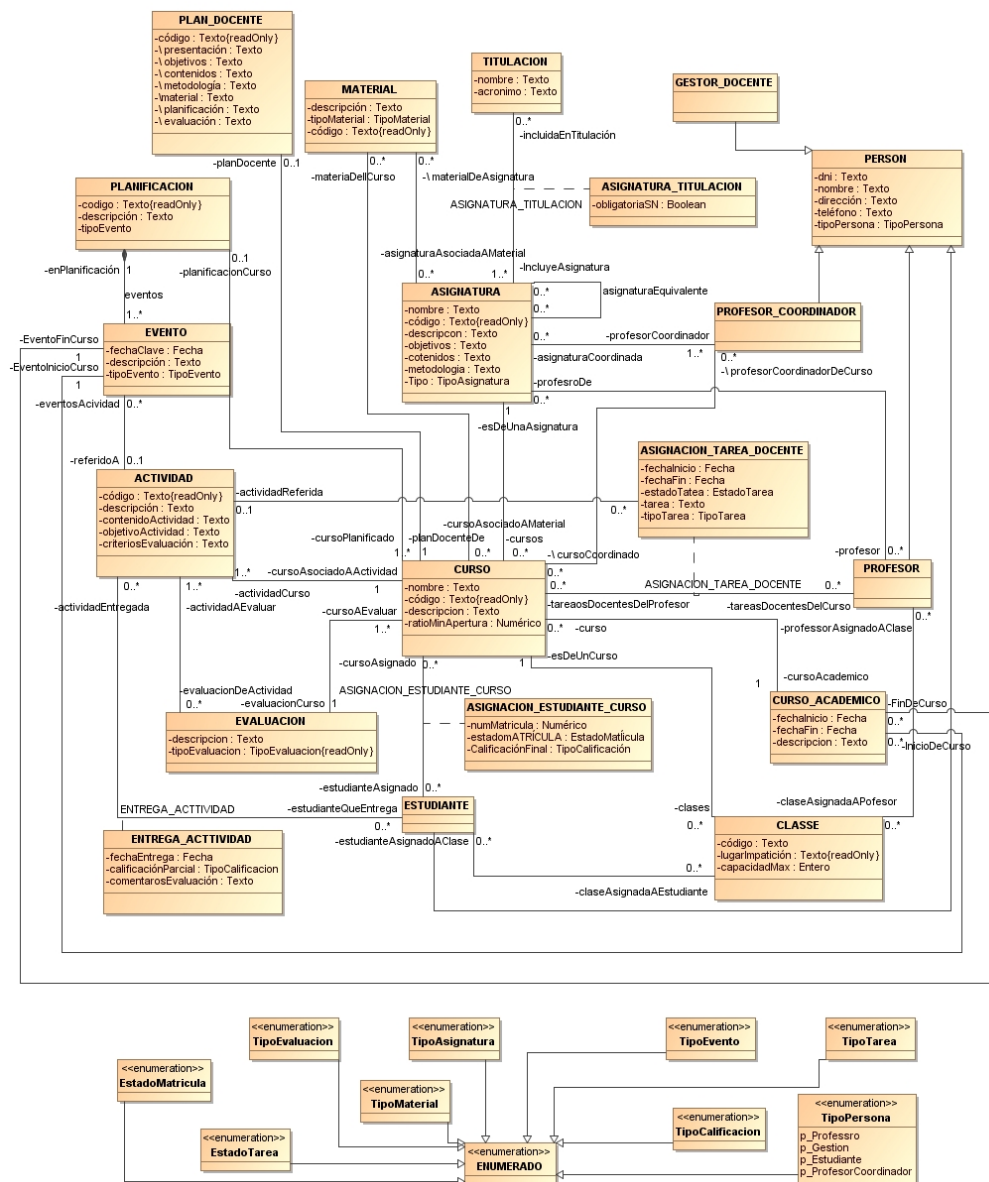


Figura 4.10: Ontología de Escenarios Educativos (OntoED)

A continuación, se describe cada una de las clases de este diagrama, empezando por aquellas que no son resultado de ninguna asociación. Luego se describen los distintos tipos de relación, incluidos los que pueden derivarse de otros tipos.

■ Clase Titulación

La clase *Titulación* describe el concepto de titulación como un conjunto de asignaturas que constituyen un plan de estudios.

Las propiedades de esta clase son un nombre, una descripción y un acrónimo tal como se muestra en la tabla 4.11.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de titulación	Texto	Cardinalidad=1
descripción	Descripción de la titulación	Texto	Cardinalidad=1
acrónimo	Abreviación del nombre de la titulación	Texto	Cardinalidad=1 Es identificador

Cuadro 4.11: Propiedades de la clase Titulacion

Las titulaciones de Ingeniería Informática de Gestión (ITIG) o Ingeniería Informática de Gestión de Sistemas (ITIS) o Ingeniería Informática (II) son ejemplos de instancias de esta clase.

■ Clase Asignatura

La clase *Asignatura* describe el concepto de asignatura entendido como materia dentro de un plan de estudios.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de asignatura	Texto	Cardinalidad=1
código	Código con el que se identifica	Texto	Cardinalidad=1 Es identificador
descripción	Descripción de la asignatura	Texto	Cardinalidad=1
objetivos	Descripción de los objetivos de aprendizaje	Texto	Card.Mín.1
contenidos	Descripción de los contenidos	Texto	Card.Mín.1
metodología	Descripción de la metodología recomendada	Texto	Cardinalidad=1
tipoAsignatura	Descripción del tipo de asignatura	TipoAsignatura	

Cuadro 4.12: Propiedades de la clase Asignatura

Toda asignatura tiene como propiedades un nombre, un código que la identifica, una descripción, un tipo, una lista de objetivos de aprendizaje, una lista de contenidos

y una metodología tal como se muestra en la tabla 4.12. Los valores de la propiedad `TipoAsignatura` vienen determinados por la clase enumeración `TipoAsignatura`.

La asignatura Bases de Datos o Sistemas Operativos serían ejemplos de asignaturas con un código, objetivos, contenidos y metodología asociados que se enmarcarían dentro de alguna titulación de Informática.

■ Clase `CursoAcademico`

La clase *`CursoAcademico`* describe el concepto de periodo académico como el intervalo durante el cual transcurre un curso.

Un curso académico se caracteriza por tener una fecha de inicio y una fecha de finalización que determinan el intervalo temporal del curso, además de un nombre y una descripción.

En la tabla 4.13 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
<code>nombre</code>	Denominación del curso académico	Texto	Cardinalidad=1
<code>descripcion</code>	Descripción del periodo académico	Texto	Cardinalidad=1
<code>fechaInicio</code>	Fecha de inicio del curso	Fecha	Cardinalidad=1
<code>fechaFin</code>	Fecha de finalización del curso	Fecha	Cardinalidad=1

Cuadro 4.13: Propiedades de la clase `CursoAcademico`

Una instancia de esta clase sería el curso académico correspondiente al Semestre de Otoño 2009 o al Cuadrimestre de Primavera 2010.

■ Clase `Curso`

La clase *`Curso`* describe el concepto de curso, entendido éste como la impartición de una asignatura en un curso académico específico. Un curso es la impartición de una asignatura en un período académico dado, con una planificación y tipo de evaluación concretos.

Todo curso se caracteriza por el curso académico y la asignatura que se imparte, le corresponde un sistema de evaluación y una planificación. En cuanto a propiedades tiene un código que lo identifica, un nombre, una descripción y un ratio mínimo de estudiantes para su apertura.

En la tabla 4.14 se presentan las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de curso	Texto	Cardinalidad=1
código	Código con el que se identifica	Texto	Cardinalidad=1 Es identificador
descripción	Descripción del curso	Texto	Cardinalidad=1
RatioMinApertura	Número mínimo de estudiantes para que se imparta el curso	Numérico	

Cuadro 4.14: Propiedades de la clase Curso

Una instancia de esta clase sería el curso Bases de Datos del semestre Otoño 2009 o el curso de Sistemas operativos del cuadrimestre Primavera 2010. El curso Bases de Datos podría tener código 11705, su descripción ser 'Introducción a las Bases de Datos y el ratio mínimo para la apertura del curso ser 10.

■ Clase Evaluacion

La clase *Evaluacion* describe el tipo de evaluación asociado al curso.

Toda evaluación de un curso tiene asociado un nombre, una descripción y un tipo de evaluación cuyos valores vienen determinados por la clase enumeración TipoEvaluacion.

En la tabla 4.15 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de evaluación	Texto	Cardinalidad=1 Es identificador
descripción	Descripción del tipo de evaluación	Texto	Cardinalidad=1
tipoEvaluación	Tipo de evaluación	TipoEvaluación	

Cuadro 4.15: Propiedades de la clase Evaluacion

Una instancia de esta clase podría ser la evaluación convencional que se describiría por 'Evaluación de curso mediante examen final' o evaluación por curso que se describiría por 'Evaluación continua mediante actividades evaluables del curso'.

■ Clase Planificacion

La clase *Planificacion* describe el conjunto de eventos que determinan la planificación de un curso.

Toda planificación se caracteriza por tener asociado un código y una descripción.

En la tabla 4.16 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
código	Código que identifica la planificación	Texto	Cardinalidad=1 Es identificador
nombre	Nombre de la planificación	Texto	Cardinalidad=1
descripción	Descripción de la planificación de algun curso	Texto	Card.Mín=1

Cuadro 4.16: Propiedades de la clase Planificacion

Por ejemplo una instancia de esta clase podría ser la planificación de un curso determinado, por ejemplo la planificación del curso de Bases de Datos del semestre de Otoño 2009 cuyo código fuera `Planificacion.BD.Otoño09`.

■ Clase Evento

La clase *Evento* describe el concepto de fecha clave dentro de la planificación de un curso.

Todo evento se caracteriza por ser de un determinado tipo, tener una descripción, un nombre y una fecha clave asociada. En la tabla 4.17 se muestran las propiedades de esta clase.

Una instancia de evento podría ser el inicio de curso, la entrega de una actividad concreta o la realización del examen final.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de la planificación	Texto	Cardinalidad=1
descripción	Descripción del evento	Texto	Cardinalidad=1
fechaClave	Fecha en que ocurre el evento	Fecha	Cardinalidad=1
tipoEvento	Tipo de evento	TipoEvento	

Cuadro 4.17: Propiedades de la clase Evento

■ Clase Material

La clase *Material* describe el concepto de material de soporte asociado a una asignatura. Un material tiene un código, una descripción, un nombre y se clasifica como material de un cierto tipo. Los valores posibles de tipo de material vienen determinados por la clase enumeración TipoMaterial.

En la tabla 4.18 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
código	Codigo que identifica un material	Texto	Cardinalidad=1 Es identificador
nombre	Nombre de material	Texto	Cardinalidad=1
descripción	Descripción de material	Texto	Cardinalidad=1
tipoMaterial	Tipo de material	TipoMaterial	

Cuadro 4.18: Propiedades de la clase Material

Suponiendo que existe un material que tiene por descripción Material didáctico del curso de Bases de datos cuyo tipo es material de soporte de una asignatura, este material sería una instancia de esta clase.

■ Clase Actividad

La clase *Actividad* describe el concepto de actividad de curso. Una actividad de curso se caracteriza por un código, una descripción, unos objetivos de aprendizaje, un contenido y unos criterios de evaluación.

En la tabla 4.19 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
Código	Código que identifica la actividad	Texto	Cardinalidad=1 Es identificador
nombre	Nombre de actividad	Texto	Cardinalidad=1
Descripción	Descripción de actividad	Texto	Cardinalidad=1
ObjetivoActividad	Objetivos de aprendizaje	Texto	Cardinalidad=1
ContenidoActividad	Contenidos de la actividad	Texto	Cardinalidad=1
CriteriosEvaluación	Criterios de evaluación	Texto	Cardinalidad=1

Cuadro 4.19: Propiedades de la clase Actividad

Las instancias de actividad pueden ser ejercicios de curso evaluables o no y/o las pruebas de evaluación final del curso.

■ Clase Clase

La clase *Clase* describe una concreción de un curso que se asocia al lugar concreto donde éste se imparte para un grupo de estudiantes. Se caracteriza por tener un código, un lugar de impartición y un ratio máximo de estudiantes por aula.

Las propiedades de esta clase son: un nombre, un código y una descripción, además de las anteriormente mencionadas para la caracterización de dicha clase. Se presentan en la tabla 4.20.

Propiedad	Descripción	Tipo de dato	Restricción
código	Código con el que se identifica	Texto	Cardinalidad=1 Es identificador
nombre	Nombre de clase	Texto	Cardinalidad=1
Descripción	Descripción de clase	Texto	Cardinalidad=1
lugarDeImpartición	Lugar donde se imparte	Texto	Cardinalidad=1
capacidadMax	Ratio máximo de inscripciones	Entero	Cardinalidad=1

Cuadro 4.20: Propiedades de la clase Clase

Por ejemplo, una instancia de clase podría ser la clase A del curso Bases de Datos del semestre de otoño 2009 que se imparte en el edificio Nexus y cuya capacidad máxima es de 50 estudiantes.

■ Clase PlanDocente

La clase *PlanDocente* describe el curso, tanto a nivel de la asignatura que se imparte como de la edición concreta de la asignatura. Se trata de un documento que establece un compromiso entre estudiantes y profesores durante el período académico en que se imparta la asignatura.

Todo plan docente tiene una parte estable que se refiere a la asignatura y otra que varía según el curso académico.

La parte fija del plan docente contiene una descripción de la asignatura a modo de presentación, los objetivos de aprendizaje y los contenidos de la asignatura, así como la metodología que seguirá. La parte variable, que forzosamente debe de actualizarse cada curso, está formada por el material del curso, la planificación o temporización del curso con el detalle de actividades a realizar y las fechas asociadas y el detalle de la evaluación del curso.

En la tabla 4.21 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
código	Código con el que se identifica	Texto	Cardinalidad=1 Es identificador
nombre	Nombre del plan docente	Texto	Cardinalidad=1
Descripción	Descripción del plan docente	Texto	Cardinalidad=1
presentación	Descripción de la asignatura	Texto	Cardinalidad=1
objetivos	Objetivos de aprendizaje	Texto	Cardinalidad=1
contenidos	Contenidos de la asignatura	Texto	Cardinalidad=1
metodología	Metología que utiliza la asignatura	Texto	Cardinalidad=1
material	Material de soporte del curso	Texto	Cardinalidad=1
planificación	Planificación de las actividades del curso	Texto	Cardinalidad=1
evaluación	Evaluación del curso	Texto	Cardinalidad=1

Cuadro 4.21: Propiedades de la clase Plan Docente

A excepción del código, el resto de atributos son derivados y se deducen a partir de las propiedades de la clase *Asignatura*, *Material*, *Planificación* y *Evaluación*. Se describen mediante las siguientes reglas de derivación (1)-(7).

Regla 1 (Rule-PresentacionPD):

La presentación del plan docente de un curso es la descripción de la asignatura correspondiente.

Regla 2 (Rule-ObjetivosPD):

Los objetivos del plan docente de un curso son los objetivos de la asignatura correspondiente.

Regla 3 (Rule-ContenidosPD):

Los contenidos del plan docente de un curso son los contenidos de la asignatura correspondiente.

Regla 4 (Rule-MetodologiaPD):

La metodología del plan docente de un curso es la descripción de la metodología de la asignatura correspondiente.

Regla 5 (Rule-MaterialPD):

Los materiales de soporte del plan docente de un curso son los descritos como materiales del curso.

Regla 6 (Rule-PlanificacionPD):

La planificación del plan docente de un curso es la descripción de la planificación del curso.

Regla 7 (Rule-EvaluacionPD):

La evaluación del plan docente de un curso es la descripción de la evaluación del curso.

Un ejemplo podría ser el plan docente de la asignatura Bases de Datos para el curso académico Otoño 2009 o el plan docente de la asignatura Sistemas Operativos que se impartió en la Primavera 2009.

- **Clase Persona**

La clase **Persona** describe todos los individuos que forman parte de la comunidad educativa.

Toda persona se caracteriza por tener un número de identificación nacional, un nombre, una dirección y un teléfono, así como un tipo. El tipo de persona viene determinado por la propiedad *tipoPersona* que toma valores de la clase enumeración **TipoPersona**. Los valores posibles son los siguientes: 'p_Profesor', 'p_ProfesorCoordinador', 'p_Estudiante' y 'p_Gestion'. Debe de tenerse en cuenta que una persona puede actuar en calidad de distintos tipos de persona según el papel que desempeñe en cada escena.

Así pues, las propiedades de esta clase son las que se muestran en la tabla 4.22.

Propiedad	Descripción	Tipo de dato	Restricción
Dni	Código con el que se identifica	Texto	Cardinalidad=1 Es identificador alternativo
nombre	Nombre/apellidos de la persona	Texto	Es identificador
dirección	Dirección	Texto	
teléfono	Teléfono	Texto	
tipoPersona	Rol que desempeña una persona en la comunidad educativa	TipoPersona	

Cuadro 4.22: Propiedades de la clase Persona

De acuerdo con los distintos valores de la clase ***TipoPersona*** las personas pueden clasificarse en:

- 1) La clase ***Estudiante*** es aquella cuyas instancias tienen como tipo de persona 'p_Estudiante'.
- 2) La clase ***Profesor*** cuya propiedad tipo de persona tiene el valor 'p_Profesor' y se caracteriza por tener la relación *ProfesorDe*.
- 3) La clase ***ProfesorCoordinador*** es aquella cuyas instancias tienen como tipo de persona el valor 'p_ProfesorCoordinador' y se caracteriza por la relación *asignaturaCoordinada*.
- 4) La clase ***GestorDocente*** es aquella cuyas instancias tienen como tipo persona el valor 'p_PersonalDeGestion'.

A partir de este punto se describen clases que son resultado de asociaciones por lo que en su definición se incluyen algunos tipos de relación.

■ Clase ***AsignacionTareaDocente***

La clase ***AsignacionTareaDocente*** describe el estado de las tareas docentes que tiene asignadas cada profesor en un curso dado.

Toda asignación de tarea docente asignada es una planificación de tarea, por lo que tiene asociada una fecha de inicio y fin, una descripción de la tarea, un tipo de tarea y el estado en que se encuentra.

En la tabla 4.23 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de la asignación docente	Texto	Cardinalidad=1
descripción	Descripción de la asignación docente	Texto	Cardinalidad=1
tarea	Descripción de la tarea asignada	Texto	Cardinalidad=1
fechaInicio	Fecha en que está planificado el inicio de la tarea	Fecha	Cardinalidad=1
fechaFin	Fecha en que está planificada la finalización de la tarea	Fecha	Cardinalidad=1
tipoTarea	Tipo de la tarea asignada	TipoTarea	Cardinalidad=1
estadoTarea	Estado de realización de la tarea	EstadoTarea	Cardinalidad=1

Cuadro 4.23: Propiedades de la clase AsignacionTareaDocente

Los valores de las propiedades tipo y estado que describen respectivamente el tipo de tarea y el estado de la tarea son las clases enumeradas TipoTarea y EstadoTarea respectivamente.

Una instancia de esta clase sería la asignación de tarea docente de un curso a un profesor, por ejemplo la preparación de exámenes de un curso que se asigna a un profesor concreto o la corrección de actividades del curso por parte de cada uno de los profesores que imparte al curso, cada una de ellas con un periodo de realización determinado.

■ Clase AsignaturaTitulacion

La clase *AsignaturaTitulación* permite concretar cada materia dentro de cada plan de estudios, por ejemplo indicando si una asignatura es obligatoria o no en un plan de estudios.

Toda asignatura en el contexto de una titulación tiene como propiedades un nombre, una descripción y un indicador de obligatoriedad o no de la asignatura en el plan de estudios, tal com se muestra la tabla 4.24.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de la asignatura en titulación	Texto	Cardinalidad=1
Descripción	Descripción de la asignatura en titulación	Texto	Cardinalidad=1
obligatoriaSN	Indicador de obligatoriedad de la asignatura en la titulación	Boolean	Cardinalidad=1

Cuadro 4.24: Propiedades de la clase AsignaturaTitulacion

Por ejemplo, la asignatura Bases de Datos podría formar parte del plan de estudios de Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas y por ejemplo ser obligatoria en la primera pero no en la segunda.

▪ Clase AsignacionEstudianteCurso

La clase *AsignacionEstudianteCurso* describe la asignación de estudiantes a cursos con el fin de poder obtener una evaluación a fin de curso y saber qué estudiantes están matriculados en un curso y/o el estado de su matrícula.

Así pues, cada asignación de estudiante a un curso tiene asociadas las propiedades número de matrícula, estado de la matrícula y calificación final, además de las propiedades nombre y descripción. La calificación final puede tener alguno de los valores que determina la clase enumeración TipoCalificacion, Análogamente, el estado de la matrícula se determina mediante la clase enumeración EstadoMatricula.

En la tabla 4.25 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de la asignación de un estudiante a un curso	Texto	Cardinalidad=1
Descripción	Descripción de la asignación de un estudiante a un curso	Texto	Cardinalidad=1
numMatricula	Número de matrícula	Entero	Cardinalidad=1
estadoMatricula	Estado en que se encuentra una matrícula	EstadoMatricula	Cardinalidad=1
CalificaciónFinal	Calificación del estudiante a final de curso	TipoCalificacion	Cardinalidad Máx.1

Cuadro 4.25: Propiedades de la clase AsignacionEstudianteCurso

Una instancia de esta clase sería la información relativa a la matriculación de un estudiante concreto, llámese Juan, a un curso concreto, por ejemplo el de Bases de Datos I con un número de matrícula y un estado. Además al finalizar el curso tendrá una calificación final.

■ Clase **EntregaActividad**

La clase **EntregaActividad** describe las entregas de actividades de un curso por parte de los estudiantes matriculados en él.

Toda entrega de actividad tiene asociada una fecha de entrega y una vez ésta ha sido evaluada, se e asocia una calificación y un comentario. Dicha calificación sólo puede tomar como valores dentro de la escala de valores posibles, cuyos valores vienen determinados por la clase enumeración TipoCalificación.

En la tabla 4.26 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
nombre	Nombre de una entrega de actividad	Texto	Cardinalidad=1
descripcion	Descripción de una entrega de actividad	Texto	Cardinalidad=1
fechaEntrega	Fecha en que se ha entregado la actividad	Fecha	Cardinalidad=1
comentariosEvaluacion	Comentarios sobre la evaluación	Texto	Cardinalidad Máx.1
calificacionParcial	Calificacion obtenida como resultado de la evaluación	TipoCalificación	Cardinalidad Máx.1

Cuadro 4.26: Propiedades de la clase EntregaActividad

Una instancia de esta clase sería cualquier actividad de curso entregada por algún estudiante matriculado en el curso Bases de Datos I.

■ Clase **Enumerado**

Esta clase generaliza todos las clases enumeración necesarias para definir el rango de los tipos que caracterizan el entorno educativo. Las clases enumeradas que especializan están clase són: EstadoMatricula, EstadoTarea, TipoAsignatura, TipoEvaluación,

TipoTarea, TipoCalificación, TipoRecurso y TipoPersona. Únicamente se concretan los valores de la clase persona a nivel de participante genérico del entorno educativo tal como se ha mencionado anteriormente.

A continuación se detallan los tipos de relación que se establecen entre clases que no se generan como resultado de asociaciones entre otras clases.

■ Tipos de relación

Puesto que existen numerosos tipos de relación se han dividido en tres grupos según establezcan una relación con las clases *Asignatura* (tabla 4.27) , *Curso* (tablas 4.28 4.29 y 4.30) y *Clase* tabla 4.31).

Tipo de relación	Descripción	Dominio	Rango	Restricción
courses	Cursos de una asignatura	Asignatura	Curso	
asignaturaEquivalente	Asignatura equivalente	Asignatura	Asignatura	Simétrica, Transitiva
incluidaEnTitulación	Titulaciones de las que forma parte la asignatura	Asignatura	AsignaturaEnTitulacion	Card.Mín=1
incluyeAsignatura	Asignatura incluida en una o más titulaciones	Titulación	AsignaturaEnTitulacion	Card.Mín=1
TitulaciónDeLaAsignatura	Titulación en la que se incluye una asignatura	AsignaturaEnTitulacion	Titulación	Cardinalidad=1
AsignaturaDeLaTitulación	Asignatura que pertenece una titulación	AsignaturaEnTitulación	Asignatura	Cardinalidad=1
profesorCoordinador	Profesor coordinador de una asignatura	Asignatura	ProfesorCoordinador	Cardinalidad Mín.1
asignaturaCoordinada	Asignatura que coordina un profesor coordinador	ProfesorCoordinador	Asignatura	
profesorDe	Asignatura de la cual se asocia un profesor	Profesor	Asignatura	
profesor	Profesores asociados a una asignatura	Asignatura	Profesor	
materialDeAsignatura	Material asociado a una asignatura	Asignatura	Material	
asignaturaAsociadaAMaterial	Asignatura a la que pertenece un material	Material	Asignatura	

Cuadro 4.27: Tipos de relación de la ontología OntoED (I)

Una asignatura se imparte en varios cursos (*courses*). Toda asignatura forma parte de alguna titulación (*incluidaEn*) y una titulación está formada por un grupo de asignaturas (*incluye*). Además, existen asignaturas de varias titulaciones que pueden considerarse equivalentes por contenidos y objetivos de aprendizaje (*asignaturaEquivalente*).

Cada asignatura tiene al menos un profesor coordinador (*profesorCoordinador*) y un profesor coordinador lo es de al menos de alguna asignatura (*asignaturaCoordinada*). Además, una asignatura puede tener un conjunto de profesores que pueden impartir la asignatura (*profesor*) y un profesor puede impartir al menos una asignatura (*profesor-De*).

Una asignatura puede tener asociado un material de soporte (*materialDeAsignatura*) y los materiales pueden estar asociados o no a una asignatura (*asignaturaAsociadaAMaterial*).

Tipo de relación	Descripción	Dominio	Rango	Restricción
esDeUnaAsignatura	Asignatura a la que pertenece un curso	Curso	Asignatura	Cardinalidad=1
cursoAcademico	Periodo academico asociado a un curso	Curso	CursoAcademico	Cardinalidad=1
curso	Cursos asociados a un periodo académico	CursoAcademico	Curso	
clases	Clases de un curso	Curso	Clases	
profesorCoordinadorDeCurso	Profesor que coordina algún curso	Curso	ProfesorCoordinador	
cursoCoordinado	Curso que coordina un profesor coordinador	ProfesorCoordinador	Curso	
tareasDocentesDelCurso	Curso al que corresponde una tarea docente asignada	Curso	AsignacionTareaDocente	
tareasDocentesDelProfessor	Profesor al que corresponde una tarea docente asignada	Profesor	AsignacionTareaDocente	
profesorAsignadoAunaTarea	Profesor a quien se refiere la asignación	AsignacionTareaDocente	Profesor	
cursoAsignadoAunaTarea	Curso al que se refiere la asignación	AsignacionTareaDocente	Curso	
actividadReferida	Actividad a la que se refiere una asignacion de tarea docente	AsignacionDocente	Actividad	Cardinalidad Max.1
estudianteAsignado	Estudiante asignado a un curso	Curso	AsignacionEstudianteCurso	
cursoAsignado	Curso asignado a un estudiante	Estudiante	AsignacionEstudianteCurso	
cursoEnAsignacion	Asignaciones de cursos a los que está asignado un estudiante	AsignacionEstudianteCurso	Curso	Cardinalidad=1
estudianteEnAsignacion	Asignaciones de estudiantes a un curso	AsignacionEstudianteCurso	Estudiante	Cardinalidad=1

Cuadro 4.28: Tipos de relación de la ontología OntoED (II)

Respecto al curso mencionar que todo curso es la edición de una asignatura (*esDeUnaAsignatura*) y se imparte en un curso académico dado (*cursoAcademico*) en distintas

clases (*clases*). En un mismo curso académico se imparten varios cursos (*curso*).

Todo curso tiene al menos un profesor coordinador (*profesorCoordinador*) y un curso siempre es coordinado por lo menos por un profesor coordinador (*cursoCoordinado*). El profesor coordinador es quien asigna las tareas docentes del curso (*tareasDocentesDelCurso*) y por tanto los profesores pueden tener asignadas tareas docentes (*tareasDocentesProfessor*, lo cual significa que imparten docencia en ese curso. Algunas tareas están relacionadas con la preparación de alguna actividad del curso (*actividadReferida*). El profesor coordinador de un curso (*profesorCoordinadorDeUnCurso*) y el curso coordinado por un profesor (*cursoCoordinado*) son tipos de relación derivados, cuyas reglas de derivación se exponen más adelante.

Un curso puede tener estudiantes asignados al curso (*estudianteAsignado*) tras el periodo de matriculación y un estudiante puede estar asignado a algún curso (*cursoAsignado*) durante el periodo académico si realiza algún curso.

Tipo de relación	Descripción	Dominio	Rango	Restricción
materialDelCurso	Material asociado a un curso	Curso	Material	
cursoAsociadoAMaterial	Curso al que se asocia un material	Material	Curso	
planificacionCurso	Planificación de un curso	Curso	Planificacion	Cardinalidad=1
cursoPlanificado	Curso al que corresponde una planificación de curso	Planificacion	Curso	Cardinalidad Mín.1
eventos	Evento que determina una fecha clave en la planificación de un curso	Planificacion	Eventos	Cardinalidad Mín.1
enPlanificacion	Planificación a la que pertenece un evento de curso determinado	Evento	Planificacion	Cardinalidad=1
inicioDeCurso	Curso que se inicia en la fecha del evento inicio de curso	Evento	Curso	
finDeCurso	Curso que se finaliza en la fecha del evento fin de curso	Evento	Curso	
eventoInicioDeCurso	Evento que indica la fecha de inicio de curso académico	CursoAcadémico	Evento	
eventoFinDeCurso	Evento que indica la fecha de fin de curso académico	CursoAcademico	Evento	

Cuadro 4.29: Tipos de relación de la ontología OntoED (III)

Además, todo curso puede tener materiales asociados (*materialDelCurso*) y los materiales se pueden asociar a cursos (*cursoAsociadoAMaterial*), así como una planificación del mismo (*planificacionCurso*), siendo ésta válida para al menos un curso (*cursoPlanificado*). Una planificación está formada por eventos y un evento determina una fecha clave

dentro de la planificación de un curso. Los eventos siempre pertenecen a alguna planificación de curso (*enPlanificacion*). Como eventos destacados de un curso académico están los eventos de inicio (*inicioDeCurso*) y fin de curso (*finDeCurso*).

Un curso tiene asociadas actividades de curso (*actividadCurso*) y cada actividad de curso tiene al menos asociado un curso (*cursoAsociadoAAktividad*). Algunas actividades forman parte de algún evento del curso (*eventosActividad*) y un evento a veces puede referirse a alguna actividad (*referidoA*). También algunas actividades pueden formar parte de la evaluación de curso (*evaluaciónDeActividad*).

Todo curso tiene asociada un tipo de evaluación (*evaluacionCurso*) y un mismo tipo de evaluación puede ser utilizado en más de un curso (*cursoEvaluado*). En la evaluación de un curso se tiene en cuenta como mínimo alguna actividad a evaluar (*actividadAEvaluar*).

El curso una vez preparado tiene un plan docente (*planDocente*) y todo plan docente lo es de un curso (*planDocenteDe*).

actividadCurso	Actividades de un curso	Curso	Actividad	Cardinalidad Mín.1
cursoAsociadoAAktividad	Curso al que se asocia una actividad	Actividad	Curso	Cardinalidad=1
eventosActividad	Eventos correspondientes a una actividad de curso	Actividad	Eventos	
referidoA	Actividad a la que se refiere un evento dentro de la planificación del curso	Evento	Actividad	Cardinalidad Max.1
evaluacionCurso	Evaluación de un curso	Curso	Evaluacion	Cardinalidad=1
cursoEvaluado	Curso que tiene una determinada tipo evaluación	Evaluacion	Curso	Cardinalidad Mín.1
evaluacionDeActividad	Evaluacion que puede considerar alguna actividad	Actividad	Evaluacion	
actividadAEvaluar	Actividad que se tiene en cuenta en la evaluación	Evaluacion	Actividad	Cardinalidad Mín.1
actividadDeLaEntrega	Actividad a la que corresponde la entrega	EntregaActividad	Actividad	
estudianteDeLaEntrega	Estudiante al que corresponde la entrega	EntregaActividad	Estudiante	
planDocente	Plan docente de un curso	Curso	PlanDocente	Cardinalidad Max.1
planDocenteDe	Curso al que pertenece un plan docente	Plan Docente	Curso	Cardinalidad=1

Cuadro 4.30: Tipos de relación de la ontología OntoED (IV)

La clase es el lugar donde se imparte un curso (*esDeUnCurso*) y un curso puede tener varias clases asociadas (*classes*).

Tipo de relación	Descripción	Dominio	Rango	Restricción
esDeUnCurso	Clase a la que pertenece un curso	Clase	Curso	Cardinalidad=1
profesorAsignadoAClase	Profesor al que se refiere una asignación docente	Clase	Profesor	
claseAsignadaAProfesor	Clase a la que se refiere una asignación docente	Profesor	Clase	
actividadEntregada	Actividad entregada por algún estudiante	Estudiante	EntregaActividad	
estudianteQueEntrega	Estudiante que ha entregado alguna actividad	Actividad	EntregaActividad	
estudianteAsignadoAClase	Estudiante asignado a una clase	Clase		
claseAsignadaAEstudiante	Clase asignada a un estudiante	Estudiante		

Cuadro 4.31: Tipos de relación de la ontología OntoED (V)

Cuando se ha asignado la docencia a una clase, la impartición de la clase corre a cargo de un profesor (*profesorAsignadoAClase*) y un profesor puede impartir docencia en alguna clase (*claseAsignadaAProfesor*).

Respecto a la clase decir que a una clase se asignan estudiantes (*estudianteAsignadoAClase*) y que un estudiante puede estar asignado a varias clases (*claseAsignadaAEstudiante*) dependiendo del número de cursos en que se haya matriculado.

Durante el curso los estudiantes pueden entregar actividades de curso (*actividadEntregada*) y una actividad puede ser entregada por varios estudiantes (*estudianteQueEntrega*).

Hasta aquí se han presentado los tipos de relación de la ontología de Escenarios educativos que no son derivables. Sin embargo, existen otros tipos de relación que pueden obtenerse mediante reglas de derivación (Reglas 8 - 11). Por ejemplo, el profesor coordinador de un curso y su inversa el curso coordinado por un profesor o los materiales de la asignatura que pueden ser materiales asociados a la asignatura o materiales que se emplean en algún curso en que se imparte la asignatura.

Regla 8 (Rule-ProfesorCoordinador):

El profesor coordinador de un curso es el profesor coordinador de la asignatura que se imparte en el curso.

Regla 9 (Rule-CursoCoordinado):

Todo curso es coordinado por el profesor que coordina la asignatura que se imparte en el curso.

Regla 10 (Rule-MaterialDeUnaAsignatura):

El material de una asignatura es material asociado a la asignatura.

Regla 11 (Rule-MaterialDeUnaAsignatura):

El material de una asignatura es material de algún curso en que se imparte la asignatura.

4.3.3. Construcción de OntoED en OWL

Al igual que para la construcción de OntoProc se han seguido las indicaciones del apéndice B para construir la ontología de escenarios educativos (OntoED) en OWL. Sin embargo, esta ontología ha sido extendida mediante el lenguaje de reglas SWRL.

Las reglas (1-7) permiten derivar el plan docente a partir de la información textual descrita en las clases relacionadas.

1) **Rule-PresentacionPD**

$$\text{Asignatura} (?x) \wedge \text{descripcion} (?x, ?y) \wedge \text{PlanDocente} (?p) \wedge \text{planDocenteDe} (?p, ?c) \wedge \text{esDeUnaAsignatura} (?c, ?a) \rightarrow \text{presentacion} (?p, ?x)$$

2) **Rule-ObjetivosPD**

$$\text{Asignatura} (?x) \wedge \text{objetivos} (?x, ?o) \wedge \text{PlanDocente} (?p) \wedge \text{planDocenteDe} (?p, ?c) \wedge \text{esDeUnaAsignatura} (?c, ?a) \rightarrow \text{objetivos} (?p, ?o)$$

3) **Rule-ContenidosPD**

$$\text{Asignatura} (?x) \wedge \text{contenidos} (?x, ?ct) \wedge \text{PlanDocente} (?p) \wedge \text{planDocenteDe} (?p, ?c) \wedge \text{esDeUnaAsignatura} (?c, ?a) \rightarrow \text{contenidos} (?p, ?ct)$$

4) **Rule-MetodologiaPD**

$$\text{Asignatura} (?x) \wedge \text{metodologia} (?x, ?m) \wedge \text{PlanDocente} (?p) \wedge \text{planDocenteDe} (?p, ?c) \wedge \text{esDeUnaAsignatura} (?c, ?a) \rightarrow \text{metodologia} (?p, ?m)$$

5) Rule-MaterialPD

materialDelCurso(?c, ?m) \wedge PlanDocente(?p) \wedge planDocenteDe(?p, ?c) \wedge
 descripcion (?m, ?d) \rightarrow material (?p, ?d)

6) Rule-PlanificacionPD

planificacionCurso (?c, ?pf) \wedge PlanDocente (?p) \wedge planDocenteDe (?p, ?c) \wedge
 descripcion (?pf, ?d) \rightarrow planificacion (?p, ?d)

7) Rule-EvaluacionPD

evaluacionCurso (?c, ?e) \wedge PlanDocente (?p) \wedge planDocenteDe (?p, ?c) \wedge
 descripcion (?e, ?d) \rightarrow evaluacion (?p, ?d)

Las reglas 8-11 en cambio formalizan algunas de las propiedades derivadas que surgen de la relación entre curso y asignatura:

8) Rule-ProfesorCoordinadorDeCurso

esDeUnaAsignatura(?c, ?a) \wedge profesorCoordinadorDeAsignatura(?a, ?p) \rightarrow
 profesorCoordinadorDeCurso(?c, ?p)

9) Rule-CursoCoordinado

asignaturaCoordinada(?p, ?a) \wedge cursos(?a, ?c) \rightarrow cursoCoordinado(?p, ?c)

10) Rule-MaterialDeAsignatura_1

asignaturaAsociadaAMaterial(?m, ?a) \rightarrow materialDeAsignatura(?a, ?m)

11) Rule-MaterialDeAsignatura_2

materialDelCurso(?c, ?m) \wedge esDeUnaAsignatura(?c, ?a) \rightarrow
 materialDeAsignatura(?a, ?m)

En en <http://personal.uoc.edu/personalonto/ontologies> puede encontrarse la ontología de escenarios educativos creada utilizando Protégé. En la figura fig:JerarquiaOntoED se presenta la jerarquía de clases hasta segundo nivel para ofrecer una visión gráfica de conjunto de la misma.

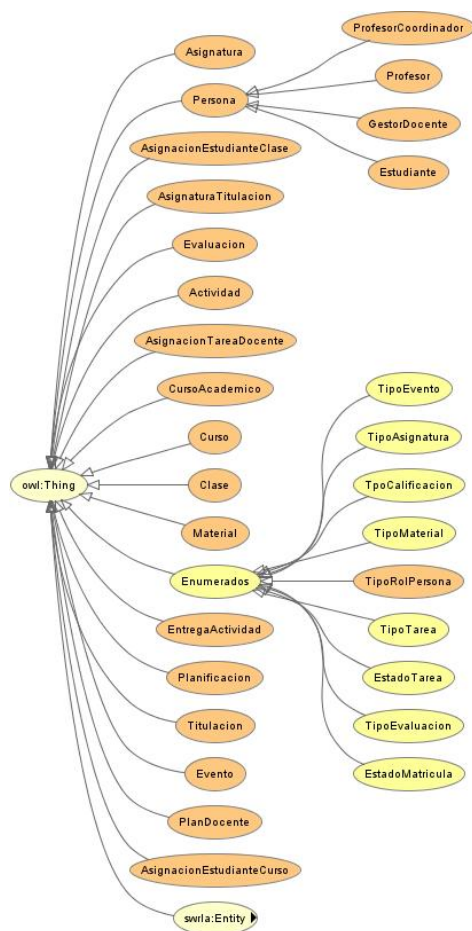


Figura 4.11: Jerarquía de clases y relaciones *is-a* de la ontología OntoED

Las propiedades de cada clase y las restricciones de cardinalidad podrían mostrarse visualizando una vista de cada clase pero no aportaría información nueva. Lo mismo ocurre por los *Object Types* y *DataProperties*.

4.4. La ontología de patrones de escenas educativas: OntoProcED

OntoProcED es una ontología de dominio (Guarino 1998) cuyo objetivo es la descripción formal de especificaciones de procesos en entornos educativos con el fin de ser utilizadas posteriormente para la especificación de escenas concretas, es decir en el seno de una organización y traducidas, total o parcialmente, en términos de su implementación.

En consecuencia se trata de una ontología para especificar patrones de escenas que ocurren en los entornos educativos, útil para cualquier institución de enseñanza superior y para cualquier plataforma de sistema de aprendizaje.

Algunas de las ventajas de esta ontología son las siguientes:

a) **Facilita la obtención de especificaciones formales de escenas educativas orientadas a su automatización**

La ontología va a permitir obtener especificaciones formales de escenas educativas consistentes y no ambiguas, capaces de ser interpretadas por una máquina mediante algún lenguaje de especificación ejecutable. El hecho de que tales especificaciones estén orientadas a la automatización significa que la descripción de dichos procesos va a tener muy presente la finalidad de éstas, por lo que los conceptos que puedan representarse en esta ontología y sus relaciones serán aquellos que realmente sean útiles para la posterior ejecución de las mismas.

b) **Promueve la creación de un catálogo de especificaciones de procesos en entornos educativos y su reutilización**

La definición de escenas educativas implica la representación de las actividades que se llevan a cabo y por tanto de todos aquellos procesos que automatizar. La reutilización de escenas supone definir procesos, básicos o complejos, estos últimos constituidos por alguna secuenciación de procesos reutilizables propios de entornos educativos. La existencia de un catálogo de especificaciones suficientemente amplio propiciará la creación de una taxonomía de procesos que facilite la localización de dichas especificaciones y su reutilización. En (Rius et al. 2007b) se propone una primera taxonomía.

c) **Facilita la compartición de especificaciones de escenas comunes a varias organizaciones.**

Como ontología permite la compartición de conocimiento común entre instituciones educativas y por tanto la posibilidad de utilizar los mismos patrones de escenas educativas, lo que a su vez reporta ventaja cara a una futura estandarización de procesos en entornos educativos.

c) **Permite la adaptación de patrones de escenas educativas a escenas concretas en un contexto organizativo específico**

Los patrones de escenas que permite representar pueden ser compartidos por distintas instituciones educativas y adaptados de acuerdo con la normativa y procedimientos de cada organización concretándose así para su posterior implementación.

c) **Validación del comportamiento del sistema previa implementación de las especificaciones de escenas**

La creación de nuevas escenas mediante la secuenciación de especificaciones de procesos existentes, supone definir secuencias de especificaciones que representan la implementación de procesos reutilizables. La instanciación de esta ontología permite comprobar si una secuencia de procesos es válida y si la alineación de éstos conduce al objetivo perseguido o lo que es lo mismo, si satisface la postcondición de los procesos que requiere automatizar. Por tanto, servirá para validar el comportamiento consistente y no ambiguo del sistema en caso de ejecutar tales especificaciones.

Puesto que esta ontología permite especificar patrones de escenas educativas, de acuerdo con lo dicho al inicio del capítulo, se procede a crearla por integración de las dos ontologías presentadas anteriormente. Luego se creará en OWL y se instanciará con parte del caso de ejemplo.

4.4.1. Creación de la ontología de Patrones de escenas educativas

En esta sección se describe el proceso de integración entre la ontología de procesos (OntoProc) y la ontología de escenarios educativos (OntoED) para constituir la ontología de Patrones de escenas educativas o procesos en entornos educativos (OntoProcED).

El proceso de integración consiste en la especialización de participantes tipo, recursos tipo y tipos enumerados que caracterizan un entorno educativo tal como se muestra gráficamente en la figura 4.12.

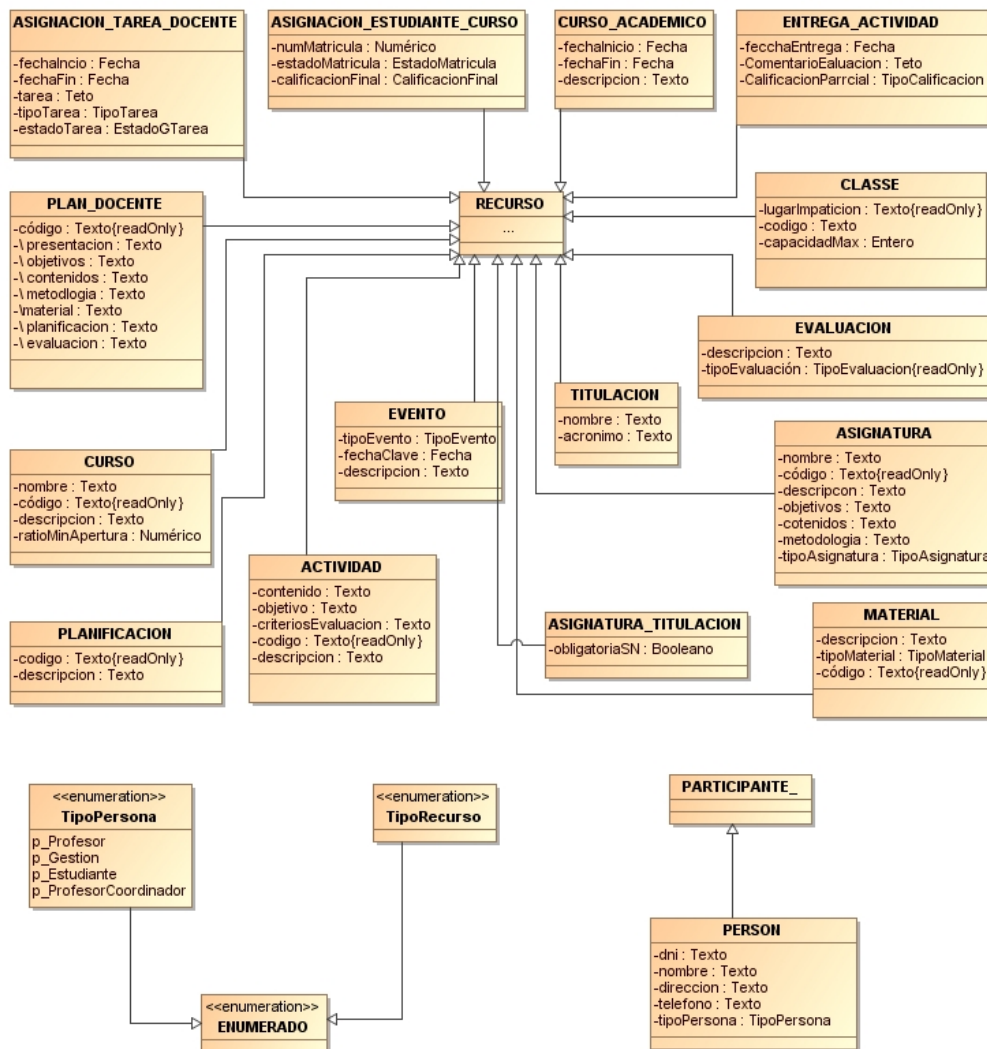


Figura 4.12: Taxonomías resultantes de la integración de OntoProc y OntoED

El proceso de integración se describe mediante la siguiente secuencia de pasos:

- 1) Creación de la clase ***Participante*** para categorizar los participantes de un entorno educativo. Se especializa en la clase ***Persona*** de OntoED para distinguir así los participantes humanos o sistema.
- 2) Creación de la clase ***Recurso*** para categorizar los recursos de un entorno educativo. Como especialización de esta clase se encuentran todas las clases que no representan un participante, ni un tipo enumerado y que son propias de un entorno educativo.
- 3) Creación de las clases enumeradas ***TipoRecurso*** y ***TipoParticipante*** para determinar los tipos de recursos y de participantes que puede tener asociados un ***RecursoTipo*** y un ***ParticipanteTipo*** respectivamente y su inclusión en la jerarquía de la clase ***Enumerado*** de OntoProcED.
- 4) Creación de los tipos de relación entre las clases ***RecursoTipo*** y ***ParticipanteTipo*** y las clases ***Recurso*** y un ***Participante*** con el fin de relacionar tipos de participantes y tipos de recursos del entorno educativo con instancias de participantes y de recursos. En la tabla 4.32 se detallan estas relaciones.

Tipo de relación	Descripción	Dominio	Rango	Restricción
tipoDeParticipante	Tipo de participante de un participante dado	Participante	ParticipanteTipo	Card.Mín=1
instanciaDeParticipante	Participantes que son instancia de un tipo de participante	ParticipanteTipo	Participante	
tipoDeRecurso	Tipo de recurso de un recurso dado	Recurso	RecursoTipo	Card.Máx=1
instanciaDeRecurso	Recursos que son instancia de un tipo de recurso	RecursoTipo	Recurso	

Cuadro 4.32: Tipos de participantes y recursos de OntoProcED

El resultado que se obtiene es la ontología de primer nivel de la arquitectura ontológica propuesta, OntoProcED para especificar procesos en entornos educativos o patrones de escenas educativas.

4.4.2. Construcción de la ontología de Patrones de escenas educativas en OWL

De acuerdo con las indicaciones del apéndice B se construye la ontología OntoProcED, la ontología de Patrones de escenas educativas o Procesos en entornos educativos.

La construcción de OntoProcED en OWL consiste en la importación de OntoProc en OntoED.

Disponiendo de ambas ontologías se procede a la creación de las clases anteriormente indicadas (*ParticipanteED* y *RecursoED*) y a la creación de las correspondientes taxonomías de acuerdo con la figura 4.12, así como a la creación de los object types que representan las propiedades de objetos de la tabla 4.32.

4.5. Resumen

En este capítulo se ha descrito el primer nivel de la ontología multinivel propuesta en este trabajo de tesis: la ontología de Procesos para entornos educativos (OntoProcED), llamada también ontología de Patrones de escenas educativas.

La creación en OWL de esta ontología y su extensión mediante el lenguaje de reglas SWRL se ha llevado a cabo a partir de la creación e integración de las dos ontologías que la forman: la ontología de Procesos (OntoProc) y la ontología de Escenarios educativos (OntoED) que permite contextualizar los procesos en entornos educativos.

Capítulo 5

Adaptación de patrones de escenas educativas a una organización concreta: el caso de la UOC

C'est le temps que tu as perdu pour ta rose qui fait ta rose si importante.

Antoine de Saint-Exupéry

En este capítulo se describe el segundo nivel de la arquitectura ontológica propuesta y su integración con el primer nivel para formar la ontología de escenas educativas para el caso concreto de la UOC (*OntoEDUOC*).

Primeramente, se ofrecen algunas indicaciones para la adaptación de patrones de escenas educativas al contexto específico de una organización. En segundo lugar, se detallan cada una de las ontologías que forman parte del segundo nivel ontológico: la ontología de procesos concretos (OntoOrg) y la ontología de escenarios educativos de la UOC (OntoUOC). Finalmente se procede a su integración y a la integración del segundo nivel con el primero para generar la ontología de escenas educativas para el caso de la UOC (OntoEDUOC).

5.1. Cómo adaptar escenas educativas genéricas a una organización concreta

La automatización de escenas educativas requiere de la formalización de escenas que habitualmente tienen lugar en organizaciones educativas específicas. Es por ello que

se pretende la adaptación de patrones de escenas educativas genéricos de manera que incorporen las particularidades de cada institución (incluidas normas y procedimientos).

En la figura 5.1 se presenta el diagrama correspondiente a la escena educativa que permite la asignación de profesores a clases. En ella intervienen participantes del entorno educativo (el profesor coordinador y el profesor) así como recursos propios de cualquier organización educativa (el repositorio de la organización, la lista de profesores, la lista de clases, etc.). Esta escena permite la asignación de profesores a clases previa validación del profesor coordinador de la asignatura y la notificación de asignaciones docentes para cada profesor del curso.

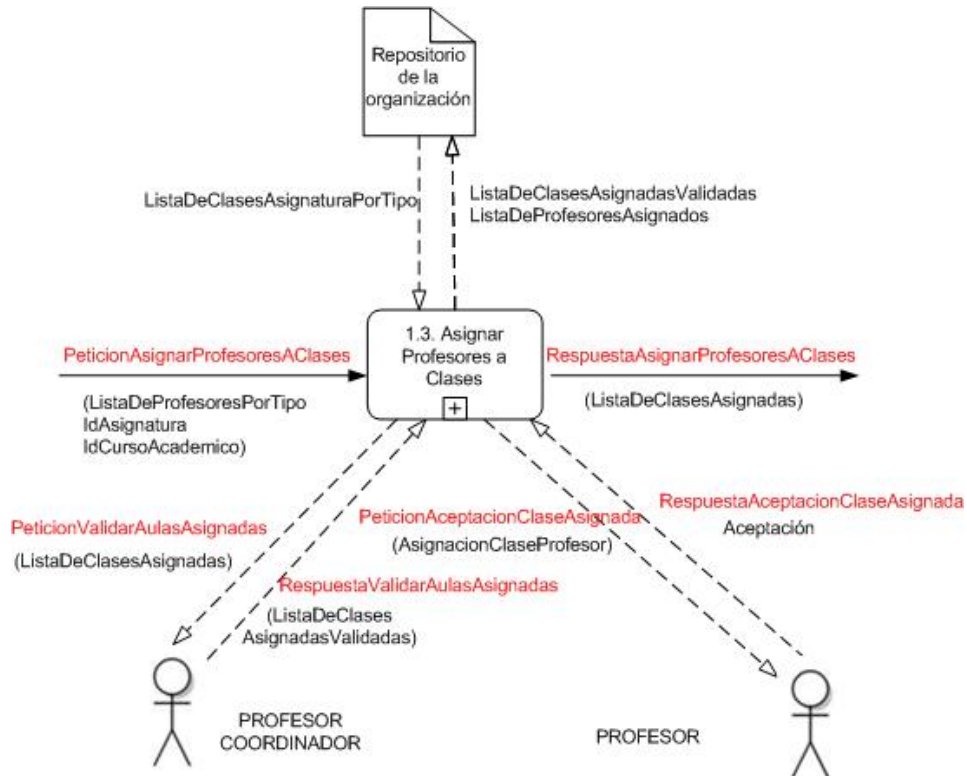


Figura 5.1: Asignar profesores a clases

Utilizando la misma notación gráfica que se ha utilizado para la representación de escenas educativas genéricas se pueden representar escenas adaptadas a una organización concreta, en este caso la UOC, tal como se muestra en la figura 5.2 para el curso de Bases de Datos I (BDI).

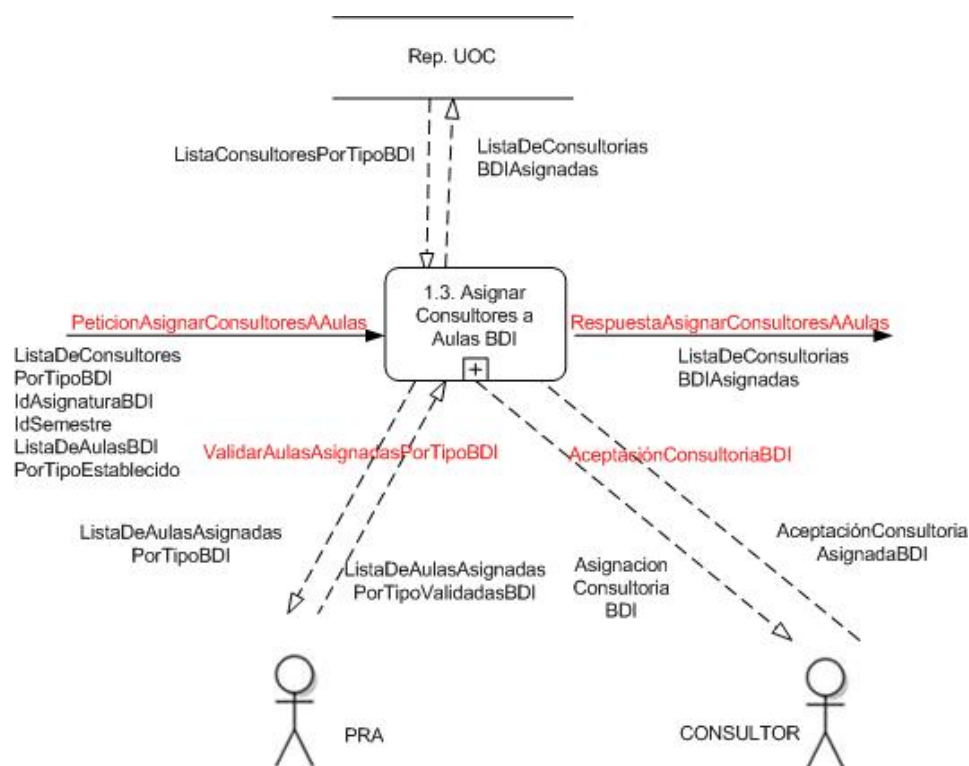


Figura 5.2: Asignar consultores a aulas BDI

Se observa que los participantes genéricos de la figura 5.1 han sido substituidos por las figuras de la institución, lo cual significa que en lugar del profesor coordinador interviene el PRA y en lugar del profesor, el consultor. Además, los recursos utilizados también son específicos de la UOC, es decir el repositorio de la Organización es substituido por el repositorio de la UOC, así como los demás recursos que han sido adaptados a la organización. Por ejemplo, la *ListaDeProfesoresPorTipo* pasa a denominarse *ListaDeConsultoresPorTipo_BDI*, el *idCursoAcadémico* es el *idSemestre*, puesto que en la UC los cursos son semestrales, la *ListaDeClasesAsignaturaPorTipo* es substituida por la *ListaDeAulasAsignaturaPorTipo_BDI* dado que el concepto de clase se asimila al concepto de aula utilizado por la UOC y así sucesivamente.

Por otro lado, el proceso *AsignarProfesoresAClases* se ha concretado con el proceso *AsignarConsultoresAAulas_BDI* para el caso concreto de la asignación de aulas del curso Bases de Datos I en la UOC con lo cual sería posible incorporar alguna restricción específica para el curso en cuestión o refinar incluso algún proceso componente, si fuera

el caso.

En consecuencia se observa que por un lado, los participantes y recursos se concretan para cada organización y escena que acontezca en ella y por otro, los procesos o secuencias de procesos se concretan para cada escena mediante los recursos y participantes que más convengan, además de poder ser modificados de acuerdo a las normas y procedimientos de la institución en concreto.

La adaptación en estos dos sentidos puede traducirse a nivel ontológico, en la creación de dos ontologías complementarias:

- **OntoUOC**: ontología de la UOC que describe la parte estática (contexto) de las escenas educativas que tienen lugar en la UOC, esto es especializando participantes y recursos de cada institución.
- **OntoOrg**: ontología de la organización para describir la parte dinámica de las escenas de forma que se concreten procesos, secuencias de procesos y flujos de mensajes.

La integración de ambas ontologías constituye la ontología de segundo nivel en la arquitectura ontológica que se propone en este trabajo de tesis (*OntoOrgUOC*) tal como se representa gráficamente en la figura 5.3.

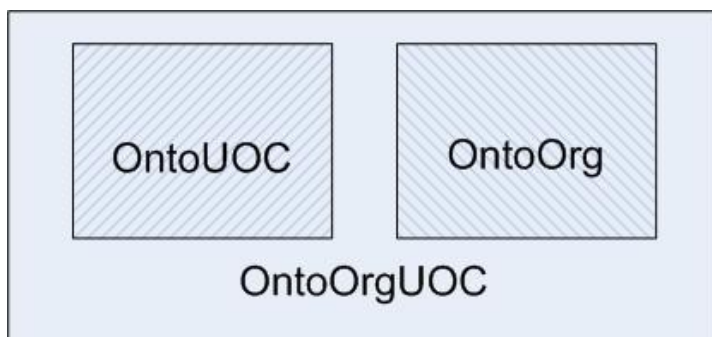


Figura 5.3: OntoOrgUOC como integración de OntoUOC y OntoOrg

OntoOrgUOC es la ontología creada para adaptar la ontología de procesos para entornos educativos (OntoProcED) a las particularidades de una organización concreta, la

UOC, y una escena concreta, la Preparación del curso de Bases de Datos I. Sin embargo, la creación de otra ontología con la misma finalidad adaptada a otra organización y escena podría llevarse a cabo de manera similar teniendo en cuenta las siguientes indicaciones:

- 1) Especialización de conceptos que extienden la jerarquía de clases.
- 2) Especialización de relaciones que originan subpropiedades de *Object Types*.
- 3) Especialización de la imagen de relaciones que restringen el rango de *Object Types* de las clases enumeradas constituyendo jerarquías de clases enumeradas.
- 4) Adición de restricciones mediante las especializaciones anteriores y definición de reglas SWRL que relacionan clases, propiedades y rangos.

5.2. La ontología de Procesos concretos: OntoOrg

OntoOrg es una ontología de dominio (Guarino 1998) cuyo objetivo es la formalización de los procesos concretos que se desean automatizar, parcial o totalmente, en las organizaciones con independencia del escenario en que ocurran.

5.2.1. El dominio de conocimiento de OntoOrg

Esta ontología tiene por objetivo la concreción de las escenas que ocurren en cada organización. Podría decirse que concreta parte del dominio de conocimiento de la ontología de procesos, puesto que permite describir secuencias de procesos concretos que intercambian mensajes concretos para alcanzar un objetivo específico.

Si bien la ontología de procesos (OntoProc) describe especificaciones de procesos, esta ontología concreta dichas especificaciones mediante procesos, mensajes, secuencias y conectores específicos que tiene lugar en una escena y organización concreta. Sin embargo, tal como se ha comentado anteriormente no considera participantes, recursos ni normas de la institución, dado que estos son descritos como parte del escenario educativo.

una organización y tiene las mismas propiedades que la clase *ProcesoTipo*.

La clase ***Mensaje*** también es instanciada por los mensajes concretos que se intercambian los anteriores procesos mencionados y tiene propiedades análogas a las de la clase *MensajeTipo*.

Por su parte, la clase ***ProcesoComplejo*** se instancia con procesos complejos concretos y sus propiedades coinciden con las de la clase *ProcesoTipoComplejo*.

La formación de procesos complejos requiere de la construcción de secuencias de procesos, por lo que las instancias de secuencias específicas son instanciadas en la clase ***Secuencia***. Análogamente tiene las mismas propiedades que la clase *SecuenciaTipo*.

Puesto que los conectores se caracterizan por sus entradas y salidas que se corresponden con procesos concretos y/o conectores, también existe la clase *Conector* que instancia conectores concretos de procesos específicos. Las propiedades de esta clase son análogas a las de la clase ***ConectorTipo***. Las clases ***Secuencial***, ***Or***, ***Xor***, ***Agregación***, ***Desagregación*** y ***Discriminación*** son especializaciones de la clase *Conector* y se comportan a como las correspondientes en la ontología de procesos aunque a otro nivel de abstracción.

A nivel de relaciones entre clases se establecen prácticamente las mismas relaciones a excepción de *Utiliza/UtilizadoPor*, *estaContenidoEn*, *puedeEnviar/debeDeSerEnviadoPor*, *puedeRecibir/debeDeSerRecibidoPor* de *OntoProc*. Simplemente hay que considerar que hay que cambiar el dominio y el rango de las relaciones por las correspondientes clases de *OntoOrg*.

En este dominio no se consideran clases enumeración debido a que la clase *TipoRol* no procede puesto que no intervienen los participantes en el dominio de conocimiento.

Igualmente que en el caso de la ontología de procesos no existen reglas para definir atributos derivados ni restricciones, por tanto tampoco procede en el caso de la ontología de la organización, dado que es un reflejo de parte de la de procesos a menor nivel de abstracción.

5.2.3. Construcción de OntoOrg en OWL

Análogamente, aunque a distinto nivel de abstracción, la ontología de procesos (OntoProc) y la de Procesos concretos (OntoOrg) requieren procedimientos muy similares para la creación de las mismas dada su enorme similitud en cuanto a secuencias de procesos y de flujos de mensajes. Por este motivo se omite el detalle de la misma en esta sección.

5.3. La ontología de escenarios educativos de la UOC: OntoUOC

OntoUOC es una ontología de dominio (Guarino 1998) cuyo objetivo es la formalización de los escenarios educativos donde habitualmente se desarrollan los procesos la Universitat Oberta de Catalunya (UOC).

Recoger el *Know-How* no es tarea de este trabajo de tesis, sin embargo se propone la creación de una parte de esta ontología con el fin poder dar solución al problema planteado fijada la UOC como organización concreta a modo de ejemplo y para el caso concreto de la escena de preparación del curso Bases de Datos I en esta institución.

Tal como se ha comentado anteriormente, esta ontología pretende describir escenarios educativos en la UOC, es decir, el contexto en el que van a representarse las escenas y se define a partir de los participantes y recursos específicos de la institución que se organizan en taxonomías, así como aquellos otros aspectos normativos relativos a la institución.

5.3.1. El dominio de conocimiento de OntoUOC

Dentro de los Estudios de Informática, Multimedia y Telecomunicaciones (EIMT) de la UOC se ofrece el curso Bases de Datos I (BDI) en las titulaciones de Ingeniería Informática de Gestión (ITIG) e Ingeniería Informática de Sistemas (ITIS).

Las asignaturas que no son de nueva creación, es decir aquellas que ya han tenido alguna edición anterior como es el caso de Bases de Datos I, disponen de un equipo de consultores que fue seleccionado a propósito para impartir la asignatura. Puesto que

Bases de Datos I es una asignatura teórico-práctica, el equipo de consultores de esta asignatura está formado tanto por consultores de teoría como de prácticas, todos ellos expertos en el ámbito de conocimiento de las Bases de Datos.

Cada asignatura es responsabilidad en última instancia del PRA o profesor responsable de asignatura (en algunos casos puede ser más de uno) que desempeña el rol del profesor coordinador durante el curso académico. La docencia es impartida por personal externo colaborador y experto en la materia. Tanto el PRA como los consultores constituyen el equipo docente de la asignatura durante el curso académico. El PRA es quien se encarga de la contratación y supervisión del equipo de consultores, así como del liderazgo y la coordinación del equipo.

En lo que respecta a evaluación, la asignatura Bases de Datos I sigue el modelo de evaluación convencional, es decir aquel en que es necesario superar un examen final para poder aprobar el curso. No obstante, tal como establece el modelo educativo de la institución se recomienda al estudiante que se acoja a la evaluación continua.

Uno de los rasgos característicos de la UOC es el hecho de facilitar la evaluación continua. La evaluación continua consiste en la realización de un conjunto de actividades propuestas durante el curso que deben de entregarse en unas fechas establecidas que forman parte del calendario de curso. Estas actividades de evaluación continua, en adelante les llamaremos actividades de EC, pueden ser de dos tipos:

- 1) Pruebas de evaluación continua (PEC)
- 2) Prácticas (PRAC)

Las PEC tienen un componente teórico más marcado que las PRAC. Ambas tienen una ponderación dentro de la evaluación continua del curso, de manera que tienen incidencia en la evaluación final de la asignatura. Sin embargo, la principal diferencia reside en el hecho que las prácticas son de realización obligatoria para la superación del curso.

El examen final es uno de los tipos de actividades de evaluación final que ofrece la UOC. Las actividades de evaluación final se caracterizan por ser realizadas síncronamente por todos los estudiantes del curso, casi siempre de forma presencial y en una fecha y momento concreto. Existen dos tipos de pruebas de evaluación continua:

- 1) Examen (EXA)
- 2) Prueba de validación (PV)

El examen pretende medir el nivel de conocimientos adquiridos por el estudiante y la PV comprobar la autenticación del estudiante como autor de las actividades de evaluación entregadas. A parte del objetivo que es distinto en cada una de ellas, también es distinta la duración de cada prueba, la PV tiene menor duración que el examen, 30 minutos para la PV y un mínimo de dos horas para el examen. Además, el resultado de la evaluación de estas pruebas también es distinto.

Como resultado de una PV se pueden obtener sólo dos valores (Valida/No valida) y en caso de validación, se confirma la nota de evaluación continua del estudiante como nota final. En cambio las actividades de evaluación continua (PEC i PRAC) pueden tomar más valores de acuerdo con la siguiente escala: A, B+, B-, C+, C- y D. Las calificaciones C- y D corresponden a la no superación de la asignatura en distinto grado.

El examen tiene mayor peso que la evaluación continua, pero su nota puede ser complementada por la nota de evaluación continua. Para determinar la nota final de curso se emplean dos tablas de cruce que han sido seleccionadas para la asignatura a partir de un conjunto de tablas de cruce definidas por la institución.

Una de las tablas determina la nota que se obtiene entre la prueba de evaluación final y la práctica (TablaCruceFinal). Luego, esta nota debe de ajustarse con la la nota de evaluación continúa, aplicando la segunda tabla de cruce (tablaCruceEC).

Además, para preparar un curso deben de crearse las aulas y asignar consultores y estudiantes a ellas.

El número de aulas se determina automáticamente a partir del tipo de asignatura y los ratios máximo y mínimo por aula. Si el número de estudiantes matriculados en el curso es inferior a 8 que es el ratio mínimo de estudiantes por aula, el curso no se abre. Si en cambio el número de estudiantes por aula es superior a 75, éstos se distribuyen entre las aulas que sean necesarias para garantizar el cumplimiento del ratio máximo de estudiantes establecido por la UOC.

No obstante, el PRA puede ajustar el número de aulas asignadas por el sistema, de

acuerdo con las características y necesidades específicas de la asignatura. Así pues, se determinan el número de aulas de cada tipo, en este caso aulas de teoría y de laboratorio.

La asignación de estudiantes a aulas concretas de la asignatura la realiza el técnico de apoyo docente a partir de la lista de estudiantes matriculados en el curso y el número de aulas de cada tipo establecidas para la asignatura.

El número de aulas determina el número de consultores necesarios de cada tipo. En el caso de Bases de Datos I el equipo de consultoría está formado por dos tipos de consultores: consultores de teoría y consultores de laboratorio. Esta distinción va a permitir asignar convenientemente a las aulas del curso los consultores que más convengan de cada tipo según el criterio del PRA de la asignatura.

Una vez asignados los consultores a las aulas, el PRA debe de asignar los encargos docentes. A nivel UOC existen unos encargos genéricos sea cual sea la consultoría de las titulaciones homologadas:

- 1) Encargo EBC o encargo básico de consultoría: consiste en la asignación de un consultor a un aula, lo que implica dirigir el aprendizaje en ella y corregir las actividades que entreguen los estudiantes asignados a ella.
- 2) Encargo ACTPD: consiste en tener asignado el encargo de actualización del plan docente.
- 3) Encargo CORREXA: consiste en la asignación de correcciones de actividades de evaluación final.
- 3) Encargo TROBI: corresponde con la asistencia a eventos presenciales como el acto de presentación del curso.

Además de estos encargos, existen otros propios de la asignatura relativos a la preparación de actividades del curso y otras tareas de consultoría que determina el PRA de la asignatura.

Las aulas de un curso, aunque no tienen una ubicación física, tienen una ubicación virtual. Las aulas virtuales de la UOC están diseñadas de acuerdo con cuatro tipos de espacios:

- 1) Espacio de Comunicación: Está formado como mínimo por un tablón y un foro como herramientas que facilitan la comunicación entre el consultor y los estudiantes y entre los estudiantes. Adicionalmente pueden existir foros de debate o *blogs*.
- 2) Espacio de planificación: contiene el calendario de actividades del curso y por tanto las fechas clave relativas al curso y específicamente aquellas que están relacionadas con las actividades de evaluación continua.
- 3) Espacio de recursos: tiene asociado todos los recursos del aula que pueden ser de interés para los estudiantes matriculados en el curso. También es un espacio transversal a todas las aulas del curso. El PRA, conjuntamente con el equipo docente, determina el material que va a estar disponible en este espacio del aula. Concretamente, se establecen tres subapartados para tipos de material distintos: a) Material de la asignatura, b) Fuentes de información y 3) Herramientas de soporte.

En el primer apartado están los materiales didácticos del curso creados a propósito para el curso en cuestión. En el segundo, materiales complementarios de otras asignaturas y enlaces a materiales de bases de datos externas. En el tercer apartado se encuentran el software necesario para la realización de las prácticas, los materiales complementarios que se consideran de interés, como puede ser documentación extra para aclaración de dudas, complemento de materiales y actualización de los mismos y sobre todo, actividades realizadas en cursos anteriores como referencia para los estudiantes. En concreto en la UOC, se establece que lo ideal es ofrecer material de dos cursos anteriores, a poder ser de las dos últimas ediciones.
- 4) Espacio de evaluación: indispensable para las aulas en las que se procede a la evaluación del curso. Estos espacios están formados por buzones de entrega para actividades evaluables (PECs Prácticas) y por otras herramientas que ofrecen soporte en la introducción y mantenimiento de las notas de los estudiantes del aula.

De hecho, más que los espacios en sí, interesan los componentes que configuran cada espacio y el tipo de acceso que se tiene a ellos. A continuación se comentan los más destacados.

El tablón hace las veces de pizarra en un entorno presencial y a él sólo tienen acceso de escritura el equipo docente (PRA y consultor), los estudiantes sólo pueden leer de él. El foro en cambio admite cualquier tipo de acceso (lectura y escritura) por parte de los miembros del aula.

El calendario de curso es transversal a todas las aulas del curso y a él pueden acceder en modo lectura tanto los miembros del curso como del aula. El permiso de escritura o actualización en cambio sólo lo tiene el PRA y el consultor de la asignatura que tiene el encargo de la actualización del plan docente.

Al espacio recursos del aula tienen acceso de lectura los estudiantes, el PRA y el consultor. Sin embargo, el PRA es quien tiene permiso de actualización. Sin embargo, la actualización física la gestiona el técnico de apoyo docente a través del PRA.

En lo que respecta a evaluación, el acceso a los buzones de entrega para todos los estudiantes asignados al aula y se les permite la escritura, en cambio para el PRA y consultor del aula, sólo está permitido el acceso de lectura en ellos. En cambio, al espacio de introducción de notas sólo puede acceder el consultor y el PRA.

Puesto que el acceso a cada uno de estos componentes se determina en función del papel que desempeña cada persona relacionada con el curso y/o aula es conveniente establecer algunos grupos de personas atendiendo a su rol dentro del escenario.

El PRA y el consultor, constituyen el equipo docente del curso. Tanto los estudiantes asignados a un aula del curso como el equipo docente del mismo son considerados miembros del aula. Como miembros del aula tienen acceso de lectura los distintos componentes que configuran los espacios del aula, sin embargo el acceso de escritura se restringe a cada componente específico según sea su rol.

Todos los miembros del aula de un curso dado, constituyen el grupo de miembros del curso. Los miembros del curso tienen acceso a todas las partes del éste, es decir partes del curso comunes a todas las aulas del mismo. En concreto el PRA y consultor que tiene el encargo de actualización del plan docente tienen permiso de actualización, así como el personal de apoyo docente que gestiona estos espacios de acuerdo con las indicaciones del PRA. Finalmente, cualquier persona de la comunidad educativa UOC es un miembro UOC, incluidos estudiantes que no se han matriculado en el presente curso académico.

docente en el curso.

- 4) El buzón de entrega de actividades de un aula sólo es actualizable por los estudiantes miembros del aula.
- 5) El tablón y las notas de un aula sólo son actualizables por el equipo docente miembros del aula.
- 6) Toda aula tiene al menos un espacio de comunicación constituido por un tablón y un foro.
- 7) Los consultores que tienen encargo EBC tienen aula asignada.

No debe perderse de vista que esta ontología restringe la ontología escenarios educativos. En ningún caso pretende ser una ontología que describa por si sola escenarios educativos en el contexto de la UOC, sino que debe de entenderse como una extensión de la ontología OntoED. Por este motivo la mayoría de las clases de esta ontología son clases equivalentes o especializaciones de la ontología de escenarios educativos o clases enumeración con el fin de adaptar tipos de datos genéricos al contexto de la UOC.

Así pues, esta ontología describe aquellos conceptos que son particulares de la UOC. Por ejemplo, el aula virtual además de ser un espacio físico incluye algunas herramientas que posibilitan la docencia en un entorno virtual, y por tanto habría que añadir estos elementos a la descripción de clase que ofrece la ontología de escenarios educativos que representa una clase en un entorno de enseñanza presencial. Consiguientemente, la ontología de la UOC también incorporaría restricciones de acceso a los elementos que son propios de la UOC.

A continuación se describen primero las clases del modelo de datos que formalizan los conceptos de la ontología, incluidas las clases de tipo enumeración, luego los tipos de relación entre clases y finalmente las restricciones del modelo.

Las clases del modelo de datos propuesto son las siguientes:

- **Clase PRA**

La clase **PRA** describe la figura del profesor responsable de asignatura.

En el caso de Bases de Datos I el PRA será María.

- **Clase Consultor**

La clase **Consultor** describe la figura del consultor de la UOC. Todo consultor tiene un tipo que permite discriminar el tipo de aula que puede conducir. Sus propiedades son las que figuran en la tabla 5.1.

Propiedad	Descripción	Tipo de dato	Restricción
tipoConsultor	Descripción del tipo de consultor	TipoConsultor_uoc	

Cuadro 5.1: Propiedades de la clase Consultor en la UOC

Los valores de la clase tipo consultor son las instancias de la clase enumeración TipoConsultor_uoc que se describe a continuación. Los valores de esta enumeración permiten especializar esta clase en: la clase **Consultor_Teo**, la clase **Consultor_Lab** y la clase **Consultor_Final**.

En el caso del curso Bases de Datos I del semestre Otoño 2009 los consultores de teoría son: Juan, Ana, Jaime, Pilar, Óscar de teoría y los consultores de laboratorio: Manuel y Antonio. Permanecen inactivos dicho semestre Ángel y Marta, que son respectivamente consultores de teoría y laboratorio de la asignatura.

- **Clase TipoConsultor_uoc**

La clase enumeración **TipoConsultor_UOC** admite los valores: 'ConsTeoria', 'ConsPractica', 'ConsTeoriaPractica' y 'ConsFinal'.

El valor 'ConsTeoria' es para designar a los consultores que pueden asignarse a un aula de teoría, 'ConsPractica' para aquellos consultores con perfil más técnico que se

asignan a las aulas de laboratorio y 'ConsFinal' para los que pueden dirigir trabajos o proyectos fin de carrera.

En el caso de la asignatura de Bases de Datos I se requieren tanto consultores de teoría como de práctica, por lo que los consultores de la asignatura deben de tener tipo ConsTeoria, ConsPráctica o ConsTeoriaPractica.

■ Clase **Estudiante_uoc**

La clase *Estudiante_uoc* describe del estudiante en la UOC.

En el caso de ejemplo serían instancias de esta clase cualquier estudiante asignado al curso de Bases de Datos I.

■ Clase **TecnicoSoporteDocente**

La clase *TecnicoSoporteDocente* describe la figura del técnico de gestión docente en la UOC. Aunque existe un importante equipo de gestión detrás de un entorno virtual de aprendizaje y se especializan en distintos tipos, en este trabajo no se considera la taxonomía en cuestión.

En el caso de ejemplo, Lidia sería una instancia de esta clase.

■ Clase **MiembroDocente_uoc**

Esta clase describe todas las figuras UOC que tienen una asignación docente en un curso: el PRA de la asignatura o los consultores que tienen algún aula del curso asignada.

Es una generalización de las clases *PRA* y *consultor* con la restricción de ser miembros del aula por tener alguna asignación docente.

En el caso de ejemplo serían miembros del equipo docente María como PRA y todos los consultores anteriormente citados que tienen asignación docente (Juan, Ana, Jaime, Pilar, Óscar, Manuel y Antonio).

- **Clase *MiembroAula_uoc***

Esta clase describe todas las personas de la comunidad educativa UOC que son participantes de un aula y por tanto tienen acceso a ella. Por tanto, aquellos que tienen una asignación a un aula, ya sean estudiantes o miembros del equipo docente.

Es una generalización de las clases *MiembroDocenteUOC* y *Estudiante_uoc* con la restricción de estar asignado al aula.

- **Clase *MiembroCurso_uoc***

Esta clase es una generalización de la clase *MiembroAulaUOC*, la cual describe todas las personas que son participantes de un curso, es decir la unión de todos los miembros de las aulas asociadas a un curso. Por tanto, el equipo docente y los estudiantes matriculados en un curso.

- **Clase *Miembro_uoc***

Esta clase describe todas las figuras de la comunidad educativa UOC con independencia de si actualmente son miembros de algún curso o no.

Es una generalización de las clases *PRA*, *Consultor*, *Estudiante_uoc*, *TecnicoSoporteDocente* y *MiembroCurso_uoc*.

En el caso de ejemplo serían miembros de esta clase, María como PRA, Juan como consultor de teoría, Antonio como consultor de laboratorio y Lúcia como técnico de apoyo docente que ofrece soporte en la publicación de materiales en el aula.

- **Clase *Curso_uoc***

La clase *Curso_uoc* describe las características específicas de un curso en la UOC que no se contemplan en la ontología de escenarios educativos genérica.

Propiedad	Descripción	Tipo de dato	Restricción
tablaExaPrac	Identificador de la tabla de cruce utilizada para calcular la nota del curso que determinan el examen y práctica	TablaExaPrac_eimt	Card.Máx=1
tablaFinalEC	Identificador de la tabla de cruce utilizada para incorporar la nota de evaluación continua la nota de cruce entre examen y práctica	TablaFinalEC_eimt	Cardinalidad=1

Cuadro 5.2: Propiedades de la clase Curso en la UOC

En la tabla 5.2 se muestran las propiedades de esta clase:

Puesto que la UOC utiliza calificaciones cualitativas y determina las notas de cada curso en función de las tablas de cruce, esta clase se extiende con dos propiedades: una para la tabla de cruce entre el examen y de la práctica (TablaExaPrac) y otra para la tabla de cruce entre la calificación obtenida a partir de la tabla anterior y la nota de evaluación continua (TablaFinalEC).

En el caso de ejemplo que nos ocupa el curso de Bases de Datos I del semestre Otoño 2009 utiliza el modelo 2a como tabla de cruce entre la nota de examen y la práctica y el modelo 1 para la tabla de cálculo de la nota final considerando la nota de evaluación continua.

■ Clase TipoTablaExaPra_eimt

La clase *TipoTablaExaPrac_eimt* permite describir cada una de las posibles tablas que describen la fórmula de cálculo entre la nota de examen y la nota de prácticas utilizadas por los Estudios de Informática, Multimedia y Telecomunicaciones de la institución.

En general estas tablas tienen en cuenta si la nota del examen y la práctica tienen el mismo peso, si el examen tiene mayor peso que la práctica o a la inversa, pero presentan variantes según si el examen está apovado o no, si la práctica está aprobada o no y las posibles combinaciones.

En la actualidad los valores con que se identifican dentro de la institución estas tablas son los siguientes: '1a', '1b', '1c', '1d', '2a', '2b', '3a', '3b', '4', '4a' y '4b'.

- **Clase TipoTablaFinaEC_eimt**

La clase *TipoTablaCruceFinalEC_eimt* permite describir cada una de las tablas que describen la fórmula de cálculo entre la nota de cruce entre el examen y las prácticas y la nota de evaluación continua utilizadas por los Estudios de Informática, Multimedia y Telecomunicaciones de la institución.

En general estas tablas tienen en cuenta si la nota final de EC es igual, superior o inferior a la nota final del curso sin tener en cuenta la evaluación continua y en función de ello cada tabla determina si la EC mantiene dicha nota, la mejora o la penaliza teniendo en cuenta algunas variantes.

En la actualidad los valores con que se designan estas tablas dentro de la institución estas tablas son los siguientes: '1' '2', 'GMMD', '3', '3b', '4', '4b', '4c' y '5'.

- **Clase ParteCurso**

La clase *ParteCurso* describe el concepto de elementos transversales a las aulas de un curso, que son accesibles por todos los miembros del curso. Por ejemplo, un mismo calendario académico, un plan docente o un material asociados al curso.

En particular, todos los miembros del curso tienen permiso de lectura sobre las partes de un curso. Sin embargo, el permiso de actualización se restringe a una parte del equipo docente (PRA y consultor con encargo de actualización del plan docente). Por otro lado, la gestión de las partes del curso es llevada a cabo por técnicos de apoyo docente.

Esta clase constituye una generalización de otras clases de OntoED como son: la clase *Calendario*, la clase *PlanDocente* y la clase *Material* con restricciones de acceso según el perfil de los miembros del curso.

En el caso de ejemplo, el calendario del curso de Bases de Datos I de Otoño 2009, los materiales asociados a dicho curso o la planificación concreta de actividades del curso serían partes del aula que se comparten entre las distintas aulas donde se imparte el curso.

■ Clase Aula

La clase **Aula** describe el concepto de clase en la UOC, que es bastante más amplio de lo que es la clase en un entorno tradicional. Por tanto, es una especialización de la clase *Clase*.

Las aulas tienen como propiedad un tipo de aula que determina en parte su configuración. En la tabla 5.3 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
tipoAula	Descripción del tipo de aula	TipoAula_uoc	Cardinalidad=1

Cuadro 5.3: Propiedades de la clase Aula en la UOC

Los valores de la propiedad TipoAula son las instancias de la clase enumeración TipoAula_uoc que se describe más adelante. Según sea el valor de esta propiedad las instancias de esta clase se especializan en: la clase **Aula_Teo**, la clase **Aula_Lab** y la clase **Aula_Final**.

Cada tipo de aula tiene su finalidad docente. En las aulas de teoría se imparten los conocimientos más teóricos de la asignatura y se evalúa a los estudiantes del curso. En el aula de laboratorio se ofrece soporte a la instalación y configuración de software utilizado para la realización de prácticas de curso y en las aulas de tipo final se llevan a cabo los trabajos y proyectos fin de carrera.

Además, cada tipo de aula tiene establecidos unos ratios mínimo y máximo de estudiantes. En el caso de las aulas de teoría el ratio máximo de estudiantes es 75 mientras que el mínimo es de 10. Las aulas de laboratorio oscilan entre 10 y 200 estudiantes y las aulas de trabajos o proyectos fin de carrera entre 1 y 10 estudiantes.

En el caso de Bases de Datos I, si en el campus catalán se matriculan 280 estudiantes en el curso de Bases de Datos I, se necesitarían cuatro aulas de teoría y dos de laboratorio, por lo que tendríamos como instancias la clase AulaTeo: Aula_BDI_Teo_0910_1, Aula_BDI_Teo_0910_2, Aula_BDI_Teo_0910_3 y Aula_BDI_Teo_0910_4 y como instancias de la clase AulaLab: Aula_BDI_Lab_0910_1 y Aula_BDI_Lab_0910_2.

■ Clase **TipoAula_uoc**

La clase enumeración **TipoAula_uoc** establece los siguientes valores: 'Aula_Teo', 'Aula_Lab' y 'Aula_Final'.

El valor 'AulaTeo' se utiliza para designar aulas de teoría, el valor 'Aula_Lab' para las aulas de laboratorio y el valor 'Aula_Final' para la dirección de trabajos o proyectos fin de carrera. Estructuralmente todos los tipos de aula disponen de los mismos espacios aunque pueden tener partes inhabilitadas, por ejemplo las aulas de laboratorio no permiten la gestión de notas.

En el caso de la asignatura que nos ocupa, el tipo de aula que se le asocia es 'Aula_Teo' y 'Aula_Lab'.

■ Clase **ParteAula**

La clase **ParteAula** describe el concepto herramienta integrada en el aula en el Campus de la UOC con el fin de hacer posible la docencia en un entorno virtual.

Toda aula tiene como mínimo dos partes: un tablón y un foro. Ambas partes de aula posibilitan la comunicación entre los miembros del aula. Sin embargo, la mayoría de aulas disponen también de un buzón para la entrega de actividades de evaluación y de un espacio para la gestión de notas.

Las propiedades de las partes del aula son las que se muestran en la tabla 5.4.

Propiedad	Descripción	Tipo de dato	Restricción
código	Código de la parte del aula	Texto	Cardinalidad=1
descripción	Descripción de la parte del aula	Texto	Cardinalidad=1
tipoParteAula	Tipo de parte del aula	TipoParteAula_uoc	Card. Mín.2

Cuadro 5.4: Propiedades de la clase Aula en la UOC

Los valores de la propiedad Tipo pertenecen a la clase enumeración *TipoParteAula_uoc*.

En particular, todos los miembros del aula tienen permiso de lectura a las partes del aula. Sin embargo, existen restricciones en cuanto al permiso de actualización de las mismas según se trate de un rol u otro.

El equipo docente tiene permiso de actualización sobre el tablón, las notas y demás partes del curso por ser transversales a todas las aulas del mismo. El estudiante tiene permiso de escritura sobre el buzón de entrega de actividades del aula.

Esta clase se especializa en las clases: **Tablón**, **Foro**, **Notas** y **BuzonEntrega** que se describen a continuación a través de la propiedad tipo cuyos valores vienen definidos por la clase enumeración *TipoParteAula*.

- **Clase TipoParteAula_uoc**

La clase enumeración **TipoParteAula_uoc** establece los siguientes valores: 'Tablon', 'Foro', 'Notas' y 'BuzonEntrega'.

En el caso de ejemplo, el aula de Bases de Datos I consta de las cuatro partes.

- **Clase Tablon**

La clase **Tablon** describe el concepto equivalente al de pizarra en un entorno presencial. Todos los miembros del aula tienen acceso de lectura y únicamente el consultor o consultores asignados al aula tienen acceso de escritura.

En el caso de ejemplo, si consideramos el aula1 de teoría tendríamos como instancias de esta clase el `Tablon_Aula_BDI_Teo_0910_1`.

- **Clase Foro**

La clase **Foro** describe el concepto de espacio para compartir conocimientos en un aula, ya sea entre estudiantes y profesor o entre los estudiantes del aula. Todos los miembros del aula tienen acceso de lectura y escritura sobre este espacio.

En el caso de ejemplo, si consideramos el aula1 de teoría tendríamos como instancia de esta clase el Foro_Aula_BDI_Teo_0910_1.

■ Clase Notas

La clase **Notas** describe el registro de calificaciones del aula. Sólo tienen acceso de lectura y escritura sobre este espacio el consultor o PRA; el estudiante se limita únicamente a acceder en modo lectura.

En el caso de ejemplo, si consideramos el aula1 de teoría tendríamos como instancias de esta clase NotasEC_Aula_BDI_Teo_0910_1 o NotasFinales_Aula_BDI_Teo_0910_1 por poner algún ejemplo.

■ Clase BuzonEntrega

La clase **BuzonEntrega** describe el concepto de lugar del aula en el que se entregan las actividades de evaluación continua. Es decir un lugar al que acceden los estudiantes del aula para entregar sus PEC o prácticas y el consultor para proceder a su corrección. El PRA también puede acceder a ellas si es necesario.

En el caso de ejemplo, en el aula1 de teoría tendríamos como instancia de esta clase el BuzonEntrega_Aula_BDI_Teo_0910_1 para entregar las PEC y prácticas los estudiantes del curso Bases de Datos de dicha aula.

■ Clase TipoMaterial_uoc

La clase enumeración **TipoMaterial_uoc** admite los siguientes valores: 'm_PEC', 'm_Practica', 'm_Examen', 'm_PV', 'm_ActividadEC', 'm_ActividadFinal', 'm_Actividad', 'm_FuenteDeInformacion', 'm_ModuloDidactico', 'm_HerramientaDeSoporte'y 'm_Software'.

En el caso de ejemplo, existen instancias de todo tipo de material excepto de m_PV por no existir la posibilidad de evaluación mediante prueba de validación.

■ Clase Material

Esta clase es de vital importancia en un entorno de enseñanza virtual dado que no existe la figura del profesor como transmisor de conocimiento.

De acuerdo con su clasificación en las aulas, esta clase se especializa en la UOC en una jerarquía de materiales, cuyo primer nivel está formado por las clases: ***ModuloDidáctico***, ***FuentesDeInformación*** y ***HerramientaDeSoporte***.

Los módulos didácticos son un tipo de material que se utiliza para designar todo aquel material que ha sido creado de acuerdo con las normas y estilo de la UOC a propósito para una asignatura y que puede ser utilizado por otra. El tipo de material que le corresponde es 'm_ModuloDidactico'.

El material designado como fuente de información es aquel que designa aquellos recursos, generalmente externos, que puedan resultar de interés para el curso. Se caracteriza por tener como tipo de material 'm_FuenteDeInformacion'.

Las herramientas de soporte son de tipo de material 'm_HerramientaDeSoporte' e incluyen todas aquellas herramientas de soporte al curso como su nombre indica. Es decir materiales como las actividades de curso, sean del tipo que sean, y el software correspondiente para su realización, así como la documentación relativa al uso, instalación y configuración del mismo. De acuerdo con esto, la clase ***Material*** se especializa en las clases: ***m_Actividad*** y ***m_Software***. La clase ***m_Actividad*** se detallará más adelante.

Para el propósito de este trabajo no se pretende formalizar los materiales UOC, sin embargo sólo interesa formalizar aquellos que se requieren para el caso de ejemplo que son básicamente las actividades.

Un ejemplo de material didáctico podría ser el material de la asignatura Bases de Datos y en concreto el módulo 1 cuyo nombre es 'Los Datos: Conceptos introductorios'. Un ejemplo de software podría ser Informix como sistema de gestión de Bases de datos.

Y un ejemplo de material complementario o de soporte podría ser el manual de instalación de Informix o las prácticas, PECs y exámenes del curso anterior al actual.

▪ Clase Actividad

La clase *Actividad* de OntoED se especializa en el contexto de la UOC en las clases *ActividadFinal* y *ActividadEC*.

El concepto de actividad debe de entenderse como material de soporte que utilizan los estudiantes en el entorno de la UOC para asentar los conocimientos adquiridos y ponerlos en práctica. Algunas actividades son evaluables y otras no. Las especializaciones de la clase actividad son ambas evaluables.

La primera subclase se corresponde con aquellas actividades de evaluación cuyo valor en la propiedad tipo de material es 'ActividadFinal' y se caracterizan por ser realizadas de forma presencial cuando finaliza el curso. La segunda subclase se refiere a las actividades de evaluación continua, es decir que se realizan durante el curso y pueden ser evaluadas.

Dada la importancia de los dos primeros tipos de actividad a la hora de preparar un curso en la UOC, se describen ambas clases de forma separada.

En el caso de ejemplo, las PEC y prácticas que se preparan para el curso de Bases de Datos I de Otoño 2009 son actividades de evaluación continua y los exámenes un ejemplo de actividades de evaluación final.

▪ Clase ActividadEC

La clase *ActividadEC* es una especialización de la clase *Actividad*. Esta clase hace referencia a las actividades de evaluación continua que se proponen realizar durante el curso y que pueden ser evaluables.

En la UOC existen dos tipos de actividades de evaluación continua: las PECs y las prácticas. Cada una de ellas se formaliza mediante una especialización de la clase *ActividadEC*, dando lugar a dos subclases: las clase *PEC* y la clase *Practica*.

Las actividades de evaluación continua se caracterizan por tener asignada una ponderación dentro de la nota de evaluación continua. Y se diferencian en la obligatoriedad o no de las mismas dentro de la evaluación continua.

Propiedad	Descripción	Tipo de dato	Restricción
ponderación	Nombre de asignatura	Entero	Cardinalidad=1
obligatoria	Indicador de obligatoriedad de la actividad de EC	Boolean	Cardinalidad=1

Cuadro 5.5: Propiedades de la clase ActividadEC

Las propiedades de esta clase son las mostradas en la tabla 5.5.

La clase **PEC** describe una actividad de evaluación continua recomendable pero no obligatoria, por tanto se caracteriza por tener el valor False en la propiedad Obligatoria. En cambio la clase **Práctica** es de obligatoria realización para superar la asignatura, por lo que la propiedad Obligatoria tiene para las instancias de esta clase el valor True.

■ Clase ActividadFinal

La clase **ActividadFinal** es una especialización de la clase **Actividad** para designar aquellas actividades que se realizan presencialmente al finalizar el curso con el fin de proceder a la evaluación de dicho curso.

En la UOC existen dos tipos de actividades finales: los exámenes y las pruebas de validación. Cada uno de estos tipos de actividad se formaliza mediante una subclase de la clase **ActividadFinal** y da origen a dos subclases: la clase **Examen** y la clase **PV**.

Cualquier actividad de evaluación final se realiza un día concreto, se inicia a una hora específica y tiene una duración determinada. Al margen del objetivo de cada prueba, la PV tiene una duración de 30 minutos y el examen de dos horas o más. Se distinguen por la propiedad Duracion30min cuyo valor booleano permite discriminar entre un tipo de actividad final u otra.

Sus propiedades se presentan en la tabla 5.6.

Propiedad	Descripción	Tipo de dato	Restricción
fechaRealizacion	Fecha en que se realiza el examen final	Fecha	Cardinalidad=1
horaInicio	Hora de inicio de la prueba	Hora	Cardinalidad=1
duracion30min	Duración de la prueba	Boolean	Cardinalidad=1

Cuadro 5.6: Propiedades de la clase ActividadFinal

■ Clase AsignacionEncargoDocente

La clase *AsignacionEncargoDocente* describe el estado de cada uno de los encargos docentes que tiene un consultor respecto a un curso dado. Puesto que es una especialización de la clase *AsignacionTareaDocente* de OntoED, en la tabla 5.7 únicamente se presentan las propiedades que difieren respecto de clase que especializa.

Propiedad	Descripción	Tipo de dato	Restricción
Tipo	Tipo de encargo asignado	TipoEncargo	Cardinalidad=1

Cuadro 5.7: Propiedades de la clase AsignacionTareaDocente

Los valores de las propiedad tipo provienen de la clase enumerada TipoEncargo_uoc.

En el ejemplo, una instancias de esta clase sería el encargo básico de consultoría que se asigna a cada uno de los consultores del curso de Bases de Datos I o el encargo específico que puede tener un consultor específico como Ana para la preparación del primer examen del curso.

■ Clase TipoEncargo_uoc

La clase *TipoEncargo_uoc* permite describir los encargos docentes que se asignan a los consultores. Los valores posibles para describir tipos de tareas son los siguientes: 'EBC', 'ACTPD', 'CORREXA', 'TROBI' y 'OtroEncargo'.

La descripción de los cuatro primeros encargos se encuentra en la sección dedicada al dominio de conocimiento de OntoUOC. Mencionar que el encargo EBC es lo que caracteriza al grupo de consultores activos de una asignatura, que el encargo ACTPD

debe de tenerlo al menos un consultor dentro del grupo de consultores activos. Y que el encargo TROBI no tiene sentido en el entorno iberoamericano, ya que no se producen encuentros presenciales.

En cuanto al encargo 'OtroEncargo' éste va a ser utilizado para referirse a aquellas tareas docentes que se asignan dentro del contexto de la coordinación de una asignatura. Instancias de ejemplo podrían ser: la propuesta de una PEC, práctica o examen, la elaboración de la solución correspondiente de cada propuesta de actividad, la preparación de la versión final de una PEC, práctica o examen.

■ Clase **TipoEvaluacion_uoc**

La clase *TipoEvaluacion_uoc* permite describir la tipología de modelos de evaluación en la UOC.

El rango de valores posibles es el siguiente: 'EC' (evaluación continua), 'EXA_Prac_EC' (Examen, práctica y evaluación continua), 'EXA_EC' (Examen y evaluación continua), 'PV_EC' (Prueba de validación y evaluación continua).

En el caso de la asignatura de Bases de Datos I el tipo de evaluación es EXA_EC.

■ Clase **TipoCalificacion_uoc**

La clase *TipoCalificacion_uoc* permite describir la escala de calificaciones de las actividades de curso y de las actividades finales. Son calificaciones cualitativas y el rango de valores posibles es el siguiente: 'A+', 'B+', 'B-', 'C+', 'C-', 'D' y 'N'.

En el caso de la asignatura de Bases de Datos I puesto que existen PECs, prácticas y exámenes, esta escala de valores se aplica a todas las calificaciones del curso.

■ Clase **TipoCalificacionPV_uoc**

La clase *TipoCalificacionPV_uoc* permite describir la escala de calificaciones de las actividades finales de tipo prueba de validación.

El rango de valores posibles es el siguiente: 'Valida', 'NoValida'.

En el caso de ejemplo no se utilizan las instancias de esta clase dado que la asignatura de Bases de Datos I no utiliza las pruebas de validación como mecanismo de evaluación.

A continuación se describen los tipos de relación entre clases.

▪ Tipos de relación

En las tablas 5.8, 5.9 y 5.10 se presentan los tipos de relación de la ontología OntoUOC que se describen a continuación.

Un aula pertenece por lo menos a un curso (*perteneceACurso*) y un curso puede tener varias aulas (*aulasDelCurso*). Puesto que las aulas en la UOC son de tres tipos, existen relaciones análogas para cada tipo de aula con el correspondiente curso, existiendo por tanto las relaciones correspondientes (*perteneceATeoCurso*, *perteneceALabCurso*, *perteneceAFinalCurso*, *aulasTeoDelCurso*, *aulasLabDelCurso* y *aulasFinalDelCurso*) que se omiten en la tabla.

Todo curso tiene partes comunes a todas las aulas del mismo (*partesCurso*) y una parte del curso lo es de un curso (*perteneceAcurso*). Un ejemplo de parte del curso son los materiales del mismo.

Las distintas partes del curso son accesibles por los miembros del curso (*accesiblePor*) y todo miembro de un curso tiene acceso a las partes de un curso (*permisoAcceso*). Sin embargo, las partes del curso sólo son actualizables por el equipo docente (*parteCursoActualizablePor*) y éste es por tanto quien tiene permiso de actualización de las parte del curso (*permisoActualizacionParteCurso*).

La gestión de cada parte del curso, en especial los materiales, son gestionados por técnicos de apoyoDocente (*gestionadaPor*) y por la relación inversa dado un técnico de apoyo docente puede saberse qué partes de curso gestiona (*gestionaParteCurso*).

Las aulas también se pueden considerar formadas por partes (*PartesAula*) y toda parte del aula es de una única aula (*perteneceApartesAula*). Las partes del aula pueden ser un tablón, un foro, un buzón de entrega de actividades y un apartado para la gestión

Tipo de relación	Descripción	Dominio	Rango	Restricción
aulasDelCurso	Aulas de un curso	Curso_uoc	Aula	
perteneceACurso	Curso al que pertenece un aula	Aula	Curso_uoc	Card.Mín=1
partesCurso	Partes de un curso transversales a todas las aulas	Curso_uoc	ParteCurso	
partePerteneceACurso	Curso al cual pertenece una parte del curso	ParteCurso	Curso_uoc	Card.Mín=1
accesiblePor	Miembros del curso que tienen acceso a las partes del curso	ParteCurso	MiembroCurso_uoc	Card.Mín=1
permisoAcceso	Partes del curso accesibles por los miembros del curso	MiembroCurso_uoc	ParteCurso	Card.Mín=1
parteCursoActualizablePor	Parte del curso actualizable por los miembros del equipo docente	MiembroDocenteUOC	ParteCurso	Card.Mín=1
permisoActualizacionParteCurso	Figura que tiene permiso de actualización de una parte del curso	ParteCurso	MiembroDocente_uoc	Card.Mín=1
gestionaParteCurso	Parte del curso que gestiona algún técnico de apoyo docente	TecnicoApoyoDocente	ParteCurso	
gestionadaPor	Técnico de apoyo docente que gestiona alguna parte del curso	ParteCurso	TecnicoApoyoDocente	Card.Mín=1
partesAula	Partes del aula	Aula	PartesAula	Card.Mín=2
partePerteneceAaula	Aula a la cual pertenece una parte del aula	ParteAula	Aula	Cardinalidad=1
parteAulaActualizable	Parte del aula que puede ser actualizada por algún miembro del aula	MiembroAula_uoc	ParteAula	Card.Mín=1
permisoActualizacionAula	Miembro del aula que tiene permiso de actualización de alguna parte del aula	ParteAula	MiembroAula_uoc	Card.Mín=1

Cuadro 5.8: Tipos de relación de la Ontología OntoUOC(I)

de notas. Como mínimo un aula se compone de dos partes: un tablón y un foro.

Los miembros del aula pueden modificar ciertas partes del aula (*ParteAulaActualizable*) y una parte del aula es actualizable por algún miembro del aula (*permisoActualizacionParteAula*). Más adelante, al definir las restricciones, se mencionaran qué figuras UOC como miembros del aula tiene permiso de modificación de las partes del aula.

Los encargos docentes asociados a un curso se distribuyen entre los consultores que imparten docencia. Cada consultor que imparte docencia tiene asignados unos encargos docentes (*encargosDocentesDelCurso*). Y cada curso tiene asociados los encargos docentes asignados a consultores (*encargosDocentesDelConsultor*). Estas relaciones se muestran en la tabla 5.9.

Y en la tabla 5.10 se presentan otros tipos de relación que son derivables relativos a la coordinación y supervisión de aulas.

Tipo de relación	Descripción	Dominio	Rango	Restricción
consultorDelEncargoDocente	Consultor a quien se asigna el encargo docente	AsignacionEncargoDocente	Consultor	Card.Mín=1
encargosDocentesDelCurso	Encargos docentes asociados a un curso	Curso_uoc	AsignacionEncargoDocente	
cursoDelEncargoDocente	Curso al que se asigna el encargo docente	AsignacionEncargoDocente	Curso_uoc	Card.Mín=1
encargosDocentesDelConsultor	Encargos docentes asociados a un consultor	Consultor	AsignacionEncargoDocente	

Cuadro 5.9: Tipos de relación de la clase AsignacionEncargoDocente

Tipo de relación	Descripción	Dominio	Rango	Restricción
consultor	Consultor asignado a un aula	Aula	Consultor	
aulaAsignada	Aula asignada a un consultor	Consultor	Aula	
coordinadoPor	PRA que coordina un consultor	Consultor	PRA	
coordina	consultores coordinado por un PRA	PRA	consultor	
supervisa	Aulas supervisada por un PRA	PRA	Aula	
supervisadaPor	PRA que supervisa un aula	Aula	PRA	

Cuadro 5.10: Tipos de relación de la Ontología OntoUOC(II)

La asignación de consultores a aulas (*aulaAsignada*) puede derivarse a partir del conjunto de consultores del curso que tienen encargo EBC y por tanto es posible saber los consultores asignados a un aula (*ConsultorAsignado*) y las aulas que tiene asignado algún consultor (*AulaAssignada*) mediante las reglas 12, 13 y 14.

Regla 12 (Rule-AsignacionConsultoriaTeo):

Las aulas de teoría de un curso se asignan a los consultores de teoría de la asignatura que tienen asignación docente en el curso.

Regla 13 (Rule-AsignacionConsultoriaLab):

Las aulas de laboratorio de un curso se asignan a los consultores de laboratorio de la asignatura que tienen asignación docente en el curso.

Regla 14 (Rule-AsignacionConsultoriaFinal):

Las aulas de de proyectos o trabajos fin de carrera de un curso se asignan a los consultores de de proyectos o trabajos fin de carrera de la titulación que tienen asignación docente en el curso.

Puesto que las aulas de un curso son la unión de todos los tipos de aulas, la propiedad *aulaAsignada* se obtiene a partir de la unión de las subpropiedades *aulaTeoAsignada*,

aulaLabAsignada y *aulaFinalAsignada* cuyos valores son determinados por las reglas anteriores. De igual manera, se obtienen la propiedad *ConsultorAsignado* que es la inversa de la propiedad *aulaAsignada*.

La coordinación del equipo de consultoría por parte del PRA, reflejada por la propiedad *coordina*, también se puede derivar mediante la regla (15). Análogamente, la propiedad inversa (*coordinadoPor*). De la misma manera, la supervisión de las aulas de un curso por parte del PRA (*supervisa*) puede deducirse a partir de todas las aulas del curso de las cuales es miembro el PRA (16). Y análogamente su propiedad inversa (*supervisadoPor*).

Regla 15 (Rule-CordinacionEquipoDocente):

Los consultores miembros del aula son coordinados por el PRA o PRAs miembro del aula.

Regla 16 (Rule-SupervisiónAulasDelCurso):

El PRA supervisa las aulas del curso de las cuales es miembro y que pertenecen al curso.

Además de las clases y tipos de relación entre clases existen algunas restricciones asociadas al modelo de datos correspondiente que no han sido recogidas en el diagrama de clases UML y que por tanto no han sido traducidas en OWL.

■ **Restricciones**

Las restricciones asociadas al modelo de datos son aquellas que establecen los permisos de actualización de las partes del aula en función de cada rol, tal como se ha comentado en el apartado de formalización de la ontología. Se detallan a continuación mediante las reglas (17, 18, 19 y 20).

Regla 17 (Rule-BuzonEntregaPermisoActualizacion):

El buzón de entrega de actividades de un aula sólo es actualizable por los estudiantes miembros del aula.

Regla 18 (Rule-TablonPermisoActualizacion):

El tablon de un aula sólo es actualizable por el equipo docente miembro del aula.

Regla 19 (Rule-NotasPermisoActualizacion):

Las notas asociadas a un aula sólo son actualizables por el equipo docente miembro del aula.

Regla 20 (Rule-ForoPermisoActualizacion):

El foro de un aula es actualizable por todos los miembros del aula.

Regla 21 (Rule-PartesObligatoriasDeUnAula):

Toda aula tiene al menos un espacio de comunicación constituido por un tablón y un foro.

5.3.3. Construcción de OntoUOC en OWL

Aplicando las correspondencias citadas en el apéndice B, igual que para OntoProc y OntoED, se ha creado la ontología UOC.

Una posible visión de la estructura ontológica la muestra la jerarquía de clases de la figura 5.6.

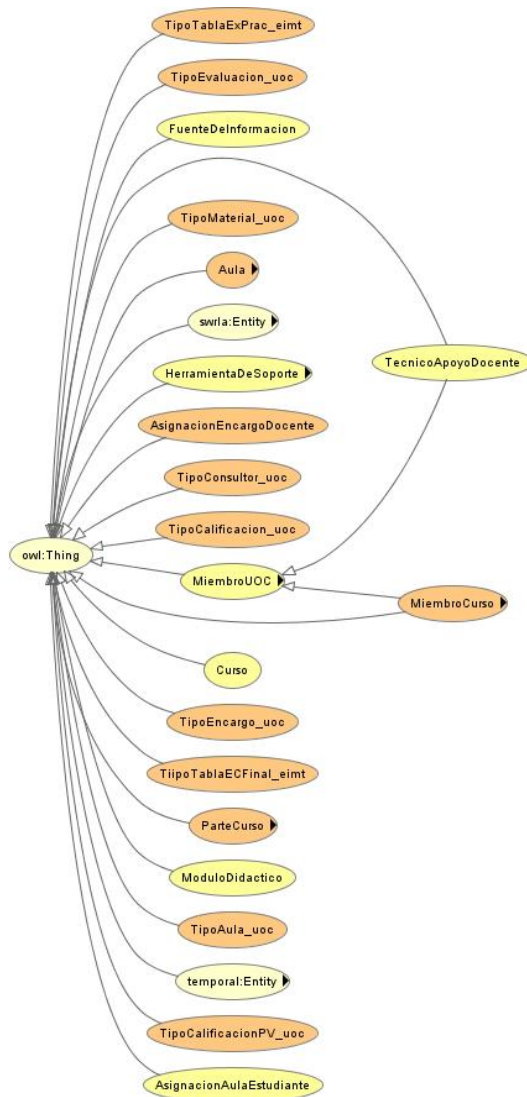


Figura 5.6: Jerarquía de clases y relaciones *is-a* de la ontología OntoUOC

Se puede observar que en el primer nivel de la jerarquía existen un gran número de clases enumeración cuyos nombres finalizan en `_uoc` y `_eimt`. También se observa herencia múltiple en lo que respecta a la clase *MiembroCurso_uoc*. El resto de clases del primer nivel son o bien clases existentes en la jerarquía de OntoED o bien clases nuevas propias del dominio de conocimiento de la UOC.

Entre las clases existentes en la ontología de escenarios educativos está la clase ***Curso***, que incorpora como propiedades las tablas de cruce para determinar la calificación final del curso a partir de calificaciones cualitativas; la clase ***Material***, que se especializa y relaciona los distintos tipos de recursos que forman parte del aula virtual. Como tipo específico de material están las actividades de curso que se corresponden con la clase ***Actividad***. Y como parte del curso está la clase ***PlanDocente***.

Como clases nuevas destacan la clase ***Aula***, que como se ha comentado anteriormente se refiere al concepto de clase en sentido más amplio; las figuras docentes (***PRA***, ***Consultor***) y de gestión (***TecnicoSoporteDocente***) de la UOC; los tipos de materiales establecidos por la institución como partes del aula (***MaterialDidactico***, ***FuentesDeInformación*** y ***HerramientaDeSoporte***) y otros materiales (***Software***, ***ActividadEC***, ***ActividadFinal***, ***PEC***, ***Pracctica***, ***Examen***, ***PV***), así como los grupos de personas que tienen determinados permisos para acceder y/o actualizar distintas partes de aula y/o curso (***MiembroCurso***, ***MiembroAula_uoc***, ***MiembroDocente_uoc***.

A partir del primer nivel de la ontología UOC aparecen cinco nuevas jerarquías:

- La clase ***ParteCurso*** se especializa en los recursos asociados al curso, una parte de los cuales son recursos del aula que pueden ser el foro, el tablón, el espacio de notas o el buzón de entrega de actividades tal como se aprecia en la figura 5.7.

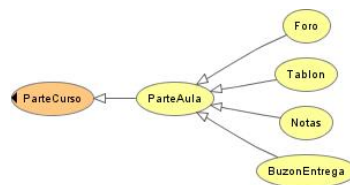


Figura 5.7: Jerarquía de clases con raíz la clase *ParteCurso*

- La clase **Aula** que se especializa en los distintos tipos de aulas (**AulaTeo**, **AulaLab** y **AulaFinal**).
- La clase **MiembroCurso_uoc** que se especializa en las distintas figuras que tienen permiso de acceso al curso como puede observarse en la figura 5.8.

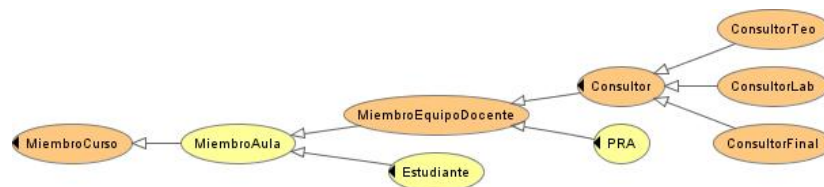


Figura 5.8: Jerarquía de clases con raíz la clase *MiembroCurso_uoc*

Los miembros del curso son todos los miembros del aula (**MiembroAula_uoc**). Los miembros del aula pueden ser estudiantes (**Estudiante_uoc**) o profesorado (**MiembroDocente_uoc**) que de acuerdo con el sistema organizativo docente de la UOC puede ser el profesor responsable de la asignatura (**PRA**) o el consultor (**Consultor**). Este último se especializa en distintos tipos de consultores (**ConsultorTeo**, **ConsultorLab** o **ConsultorFinal**).

- La clase **Miembro_uoc** que se especializa en las clases: **PRA**, **Consultor**, **Estudiante_uoc**, **TecnicoApoyoDocente** y **MiembroCurso_uoc** con el fin de reconocer como miembros de la comunidad educativa aquellos que no están en activo, es decir que en la actualidad forman parte de un curso bien sea como profesorado o como estudiante. En la figura 5.9 se muestra la jerarquía de clases.

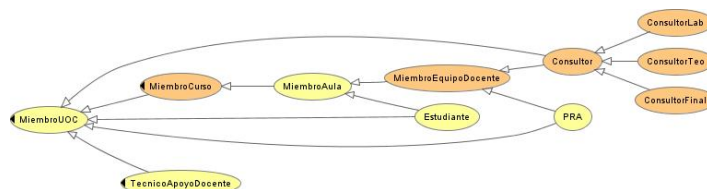


Figura 5.9: Jerarquía de clases con raíz la clase *Miembro_uoc*

- La clase *HerramientaDeSoporte* se especializa en las clases *Software* y *Actividad*. Las actividades de curso en la UOC se especializan en: *ActividadED*, *ActividadFinal*, *PEC*, *Practica*, *Examen* y *PV*.

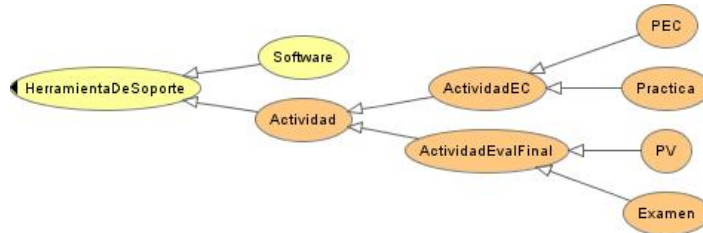


Figura 5.10: Jerarquía de clases con raíz la clase *HerramientaDeSoporte*

En la figura 5.10 puede observarse como la clase *Actividad* se integra dentro de la jerarquía de materiales de soporte y además, se especializa en los distintos tipos de actividades.

Y en el primer nivel de la jerarquía, como especialización de la clase *Thing*, también se encuentran el conjunto de clases enumeradas creadas para definir rangos de valores de ciertas propiedades de clases o relaciones. Dichas clases se detallan en la tabla 5.11.

Nueva clase enumerada	Clase Origen	Propiedad	Valores
TipoConsultor_uoc	Consultor	Tipo	ConsTeoria, ConsLab, ConsFinal
TipoAula_uoc	Aula	Tipo	AulaTeo, Aulalab, AulaFinal
TipoParteAula_uoc	ParteAula	Tipo	Tablon, Foro, Notas, BuzonEntrega
TipoParteCurso_uoc	ParteCurso	Tipo	PlanificacionDocente, PlanDocente, RecursosDocente
TipoTablaExaPrac_eimt	Curso	TablaCruceExaPrac	1a, 1b, 1c, 1d, 2a, 2b, 3a, 3b, 4, 4a, 4b
TipoTablaFinalEC_eimt	Curso	TablaCruceFinalEC	1, 2, 3a, 3b, 4, 4b, 4c, 5
TipoMaterial_uoc	Material	Tipo	m_Actividad, m_PV, m_ActividadEC, m_PEC, m_ActividadFinal, m_Examen, m_Practica, Software, m_ModuloDidactico, m_HerramientaDeSoporte, m_FuenteDeInformacion
TipoEncargo_uoc	Tarea	Tipo	EBC, ACTPD, CORREXA, TROBI, OtroEncargo
TipoEvaluacion_uoc	Evaluación	Tipo	EC EXA_EC PV_EC EXA_PV_EC
TipoCalificacion_uoc	Calificación	Tipo	A Bmas Bmenos Cmas Cmenos D NP
TipoCalificacionPV_uoc	Calificación	Tipo	Valida NoValida

Cuadro 5.11: Clases enumeradas de OntoUOC

Además, se han creado *Object Types* para definir los tipos de relación indicados en las tablas 5.8 y 5.10 y los *Data Types* correspondientes para definir las propiedades de las clases formalizadas en la sección anterior.

Así mismo, como extensión del lenguaje OWL, se han creado las reglas SWRL que permiten obtener las propiedades derivadas a partir del conjunto de reglas de derivación enumeradas en el apartado de formalización de OntoUOC:

12) *Rule-AsignacionConsultoriaTeo*

$$\begin{aligned} & \text{ConsultorTeo}(?ct) \wedge \text{encargosDocentesDeCurso}(?ct, ?ae) \wedge \\ & \text{tipoEncargoDocente}(?ae, ?te) \wedge \text{swrlb:equal}(?te, \text{EBC}) \wedge \\ & \text{cursoDelEncargoDocente}(?ae, ?c) \wedge \text{aulasTeoDelCurso}(?c, ?at) \\ & \rightarrow \text{aulaTeoAsignada}(?at, ?ct) \end{aligned}$$

13) *Rule-AsignacionConsultoriaLab*

$$\begin{aligned} & \text{ConsultorLab}(?cl) \wedge \text{encargosDocentesDeCurso}(?cl, ?ae) \wedge \\ & \text{tipoEncargo}(?ae, ?te) \wedge \text{swrlb:equal}(?te, \text{EBC}) \wedge \\ & \text{cursoDelEncargoDocente}(?ae, ?c) \wedge \text{aulasFinalDelCurso}(?c, ?at) \\ & \rightarrow \text{aulaLabAsignada}(?at, ?cl) \end{aligned}$$

14) *Rule-AsignacionConsultoriaFinal*

$$\begin{aligned} & \text{ConsultorFinal}(?cf) \wedge \text{encargosDocentesDeCurso}(?cf, ?ae) \wedge \\ & \text{tipoEncargo}(?ae, ?te) \wedge \text{swrlb:equal}(?te, \text{EBC}) \wedge \\ & \text{cursoDelEncargoDocente}(?ae, ?c) \wedge \text{aulasFinalDelCurso}(?c, ?at) \\ & \rightarrow \text{aulaFinalAsignada}(?at, ?cf) \end{aligned}$$

15) *Rule-CoordinacionEquipoDocentes*

$$\begin{aligned} & \text{PRA}(?p) \wedge \text{Consultor}(?c) \wedge \text{permisoAcceso}(?p, ?pc) \\ & \wedge \text{permisoAcceso}(?c, ?pc) \wedge \text{partePerteneceAcurso}(?pc, ?cu) \\ & \rightarrow \text{coordina}(?p, ?c) \end{aligned}$$

16) Rule-SupervisiónAulasDelCurso

$$\text{PRA}(?p) \wedge \text{MiembroAula_uoc}(?p) \wedge \text{parteAulaActualizable}(?p, ?pa) \wedge \\ \text{partePertenceAaula}(?pa, ?a) \rightarrow \text{supervisa}(?p, ?a)$$

En cuanto a las restricciones de integridad previamente mencionadas también han sido formuladas en SWRL del siguiente modo:

17) Rule-BuzonEntregaPermisoActualizacion

$$\text{MiembroAula_uoc}(?e) \wedge \text{Estudiante_uoc}(?e) \wedge \text{BuzonEntrega}(?b) \wedge \text{ParteAula}(?b) \wedge \\ \text{partePertenceAaula}(?b, ?a) \rightarrow \text{parteAulaActualizable}(?e, ?b)$$
18) Rule-TablonPermisoActualizacion

$$\text{MiembroAula_uoc}(?d) \wedge \text{MiembroDocente_uoc}(?d) \wedge \text{Tablon}(?t) \wedge \text{ParteAula}(?t) \wedge \\ \text{partePertenceAaula}(?t, ?a) \rightarrow \text{parteAulaActualizable}(?d, ?t)$$
19) Rule-NotasPermisoActualizacion

$$\text{MiembroAula_uoc}(?d) \wedge \text{MiembroDocente_uoc}(?d) \wedge \text{Notas}(?n) \wedge \text{ParteAula}(?n) \wedge \\ \text{partePertenceAaula}(?n, ?a) \rightarrow \text{parteAulaActualizable}(?d, ?n)$$
20) Rule-ForoPermisoActualizacion

$$\text{MiembroAula_uoc}(?d) \wedge \text{MiembroDocente_uoc}(?d) \wedge \text{Foro}(?f) \wedge \text{ParteAula}(?f) \wedge \\ \text{partePertenceAaula}(?f, ?a) \rightarrow \text{parteAulaActualizable}(?d, ?f)$$
21) Rule-PartesObligatoriasDeUnAula

$$\text{Tablon}(?t) \wedge \text{Foro}(?f) \wedge \text{partePertenceAaula}(?t, ?a) \wedge \text{partePertenceAaula}(?f, ?a) \rightarrow \\ \text{Aula}(?a)$$

5.4. La ontología de escenas en el contexto de una organización concreta: OntoEDUOC

Esta ontología es el resultado de la integración del primer y segundo nivel de la arquitectura ontológica propuesta. A continuación, se presentará la integración de OntoOrg y OntoUOC para formar la ontología de segundo nivel y posteriormente, la integración de cada una de estas ontologías componentes con OntoProcED, la ontología de procesos en entornos educativos que constituye el primer nivel del marco propuesto.

5.4.1. Creación de la ontología de procesos concretos para la UOC

OntoOrgUOC es la ontología resultante de la integración de la ontología de Procesos concretos (OntoOrg) y la ontología de la UOC (OntoUOC). Su objetivo es emular la ontología de procesos en entornos educativos (OntoProcED) pero a mayor nivel de concreción, fijada una organización, establecidos unos procesos concretos y en un contexto particular con sus propias normas y procedimientos.

Debido a la complementariedad entre ambas ontologías, la integración entre ellas es bastante. Dicha integración consiste en identificar los participantes y recursos de la ontología UOC y crear las correspondientes taxonomías. Además, por otro lado, las clases que representan instancias de participantes y recursos en la UOC se relacionan con las clases de OntoOrg de forma similar a como se relacionan las clases tipo correspondientes en OntoProc.

A continuación, se detallan los pasos seguidos para llevar a cabo la integración de OntoOrg y OntoUOC:

- 1) Creación de la taxonomía de participantes mediante la creación de la clase *Participante*.

Esta taxonomía tiene en un primer nivel de la jerarquía la clases *Persona* la clase *Persona*, lo cual permite distinguir entre participantes humanos o sistemas. En un segundo nivel, como subclase de *Persona*, se sitúa la clase *Miembro UOC* que representa cualquier persona de la comunidad UOC.

- 2) Creación de la taxonomía de recursos mediante la creación de la clase **Recurso**.

Esta taxonomía está formada por todas las clases de OntoUOC que no son Miembros UOC, ni enumeraciones. Esto significa que como subclases de la clase **Recurso** recién creada se sitúan las clases **AsignacionEncargoDocente**, **Aula**, **Curso_uoc**, **FuenteDeInformacion**, **HerramientasDeSoporte**, **ModuloDidactico** y **ParteCurso**.

- 3) Establecimiento de relaciones entre las clases **Participante** y **Recurso** recién creadas y las clases **Mensaje** y **Proceso** de OntoOrg.

En la tabla 5.12 se muestran estas relaciones.

Tipo de relación	Descripción	Dominio	Rango	Restricción
concreto_utiliza	Recurso que utiliza un proceso	Proceso	Recurso	Cardinalidad Mín.1
concreto_puedeEnviar	Mensaje que puede enviar un participante	Participante	Mensaje	
concreto_puedeRecibir	Mensaje que puede recibir un participante	Participante	Mensaje	
concreto_debeSerRecibidoPor	Participante que puede recibir este mensaje	Mensaje	Participante	
concreto_debeSerEnviadoPor	Participante que puede enviar este mensaje	Mensaje	Participante	
concreto_contieneRecurso	Recurso contenido en un mensaje	Mensaje	Recurso	Cardinalidad Mín.1
concreto_estaContenidoEnMensaje	Recurso contenido en un mensaje	Recurso	Mensaje	
concreto_esUtilizadoPor	Recurso utilizado por un proceso	Recurso	Proceso	

Cuadro 5.12: Tipos de relación entre las clases de OntoOrg y OntoUOC

- 4) Creación de la taxonomía de tipos enumerados mediante la creación de la clase **Enumerados**.

Esta taxonomía tiene en un primer nivel de la jerarquía las clases **TipoAula_uoc**, **TipoEncargo_uoc**, **TipoEvaluacion_uoc**, **TipoMaterial_uoc**, **TipoMiembro_uoc**, **TipoCalificacion_uoc**, **TipoCalificacionPV_uoc**, **TipoTablaEC-Final_eimt**, **TipoTablaExPract_eimt**.

5.4.2. Creación de la ontología de escenarios educativos en el contexto de la UOC

La ontología OntoUOC pretende refinar la ontología de procesos en entornos educativos (OntoProcED) en cuanto a escenarios educativos se refiere. Por tanto, su integración en OntoProcED básicamente consiste en la especialización de patrones de escenas mediante participantes y recursos específicos de la UOC, así como la incorporación de las restricciones relativas a ellos.

Durante el proceso de integración algunas clases de OntoUOC han especializado clases de OntoED tal como muestra la tabla 5.13.

Clase padre (OntoProcED)	Clase hija (OntoUOC)	Comentarios
ProfesorCoordinador	PRA	Añade nuevos tipos de relación para la coordinación del equipo docente y la supervisión de las aulas
Profesor	Consultor	Incorpora la propiedad TipoConsultor
GestorDocente	TecnicoApoyoDocente	Incorpora nuevas relaciones con las partes del curso que gestiona
Estudiante	Estudiante_uoc	Añade nuevos tipos de relación como miembro de la comunidad UOC y/o miembro de aula
Curso	Curso_uoc	Añade dos nuevas propiedades TablaCruceExaPrac y TablaCruceFinalEC y se compone de partes de curso
Clase	Aula	Incorpora la propiedad TipoAula y se compone de dos o más partes de aula
Material	FuenteDeInformacion, HerramientaDeSoporte, MóduloDidactico	Permite crear una taxonomía de materiales
AsignacionTareaDocente	AsignacionEncargoDocente	Delimita los tipos de encargos de consultores asignados a aulas

Cuadro 5.13: Clases de OntoProcED que se especializan en clases de OntoUOC

También, se han especializado algunas clases enumeradas de OntoProcED con clases enumeradas de OntoUOC con el fin de establecer el rango de valores para ciertas clases enumeradas genéricas de acuerdo con la normativa de la institución. Por ejemplo, en el caso de la UOC la tipología de materiales accesibles desde las aulas es el que determina el rango de valores de la clase enumerada *TipoMaterial_uoc*, los tipos de encargos docentes que la UOC reconoce como tareas docentes retribuíbles se identifican por los valores posibles de la clase enumerada *TipoEncargo_uoc*, los tipos de evaluación que puede adoptar cada curso a partir de los que reconoce la institución y que

vienen dados por la clase enumerada ***TipoEvaluacion_uoc*** o incluso, las calificaciones que pueden obtenerse como resultado de la evaluación de las distintas actividades entregadas mediante el rango de valores determinado por las clases ***TipoCalificacion_uoc*** o ***TipoCalificacionPV_uoc*** según el tipo de actividad.

En la tabla 5.14 se presentan las clases enumeradas de OntoUOC que especializan las clases enumeradas de OntoProcED y los valores que constituyen cada clase.

Clase enumerada padre (OntoProcED)	Clase enumerada hija (OntoUOC)	Valores
TipoMaterial	TipoMaterial_uoc	m_Actividad, m_ActividadEC, m_ActividadFinal, m_Examen, m_PV, m_PEC, m_Practica, Software, m_ModuloDidactico, m_HerramientaDeSoporte, m_FuenteDeInformacion
TipoTarea	TipoEncargo_uoc	EBC, ACTPD, CORREXA, TROBI, OtroEncargo
TipoEvaluación	TipoEvaluacion_uoc	EC EXA_EC PV_EC EXA_PV_EC
TipoCalificación	TipoCalificacion_uoc	A Bmas Bmenos Cmas Cmenos D NP
TipoCalificación	TipoCalificacionPV_uoc	Valida NoValida

Cuadro 5.14: Clases enumeradas de OntoUOC que especializan clases de OntoProcED

Además de la especialización de clases y clases enumeradas, se requiere la creación de dos nuevos *Object Types* con el fin de relacionar las instancias de participantes y recursos de OntoUOC con las instancias de participantes tipo y recursos tipo de OntoProcED. Estos nuevos tipos de relación son los mostrados en la tabla 5.15.

Tipo de relación	Descripción	Dominio	Rango	Restricción
tipoDeParticipante	ParticipanteTipo al que se asocia un participante concreto	Participante	ParticipanteTipo	Card.Mín=1
tipoDeRecurso	RecursoTipo al que se asocia un recurso específico	Recurso	RecursoTipo	Card.Máx=1

Cuadro 5.15: Tipos de relación entre participantes y recursos de OntoProcED y OntoUOC

Sin embargo, aparte de relacionar el participantes y recursos tipo con participantes y recursos concretos, interesa también la especialización de propiedades mediante subpropiedades, creando nuevos *Object Types* tal como se muestra en la tabla 5.16.

Propiedad que especializa	Subpropiedad
clases	aulasDelCurso
aulasDelCurso	aulaTeoDelCurso
aulasDelCurso	aulaLabDelCurso
aulasDelCurso	aulaFinalDelCurso
esDeUnCurso	perteneceACurso
perteneceACurso	perteneceAteoCurso
perteneceACurso	perteneceAlabCurso
perteneceACurso	perteneceAfinalCurso
profesorAsignadoAClase	consultorAsignado
consultorAsignado	consultorTeoAsignado
consultorAsignado	consultorLabAsignado
consultorAsignado	consultorFinalAsignado
claseAsignadaAProfesor	AulaAsignada
claseAsignada	AulaAsignada
AulaAsignada	AulaTeoAsignada
AulaAsignada	AulaLabAsignada
AulaAsignada	AulaFinalAsignada
tareasDocentesDelcurso	encargosDocentesDelCurso
cursoAsigandoAunaTarea	cursoDelEncargoDocente
tareasDocentesDelProfesor	enccargosDocentesDeConsultor
profesorAsignadoAunaTarea	consultorDelEncargoDocente

Cuadro 5.16: Propiedades de OntoUOC que especializan propiedades de OntoProcED

5.4.3. Integración del nivel de patrones de escenas con el nivel de procesos concretos

La integración de la ontología de la Organización (OntoOrg) en la ontología de procesos para entornos educativos (OntoProcED) permite garantizar que las escenas concretas son de acuerdo a los patrones de escenas establecidos.

Para conseguir esta finalidad es necesario que se establezca algún tipo de relación *tipoDe* que permita asociar a cada instancia concreta, que forma parte de la descripción de un proceso específico en una escena y organización concreta, su tipo correspondiente.

En la tabla 5.17 se presentan las relaciones de tipos de proceso, mensaje y secuencia para los casos en que existan.

Y en las tabla 5.18 se muestran los *Objet Types* creados para determinar los tipos de

Tipo de relación	Descripción	Dominio	Rango	Restricción
tipoDeProceso	ProcesoTipo al que se asocia un proceso concreto	Proceso	ProcesoTipo	Card.Máx.=1
tipoDeProcesoComplejo	ProcesoComplejoTipo al que se asocia un proceso complejo concreto	ProcesoComplejo	ProcesoComplejoTipo	Card.Máx.=1
tipoDeMensaje	MensajeTipo al que se asocia un mensaje específico	Mensaje	MensajeTipo	Card.Máx.=1
tipoDeSecuencia	SecuenciaTipo a la que se asocia a una secuencia específica	Secuencia	SecuenciaTipo	Card.Máx.=1

Cuadro 5.17: Tipos de relación entre clases de OntoOrg y OntoProcED

conectores en los casos que existan.

Tipo de relación	Dominio	Rango	Restricción
TipoDeConector	Conector	ConectorTipo	Card.Máx=1
TipoConectorAgregacion	ConectorAgregacion	ConectorAgregacionTipo	Card.Máx=1
TipoDeConectorDesagregacion	ConectorDesagregacion	ConectorDesagregacionTipo	Card.Máx=1
TipoDeConectorOr	ConectorOr	ConectorOrTipo	Card.Máx=1
TipoDeConectorXor	ConectorXor	ConectorXorTipo	Card.Máx=1
TipoDeConectorSecuencial	ConectorSecuencial	ConectorSecuencialTipo	Card.Máx=1
TipoDeConectorDiscriminacion	ConectorDiscriminacion	ConectorDiscriminacionTipo	Card.Máx=1

Cuadro 5.18: Tipos de relación entre clases de OntoOrg y OntoProcED

5.5. Resumen

En este capítulo se ha descrito la ontología de escenas educativas en el contexto de una organización concreta, la UOC (OntoEDUOC).

Dicha ontología representa una concreción de la ontología de procesos en entornos educativos (OntoProcED) adaptada, tanto a nivel de escenarios educativos (OntoUOC) como de procesos específicos (OntoOrg), para la organización UOC. Además, ha sido creada en lenguaje OWL y extendida mediante el lenguaje SWRL para la incorporación de restricciones.

Capítulo 6

Ontología de Implementación para el caso de la especificación OKI-OSID

Le mieux est l'ennemi du bien.

Francois Marie Arouet (Voltaire)

En este capítulo se describe el tercer nivel de la ontología propuesta en este trabajo de tesis: la ontología de implementación (*OntoOKI*). Se trata de una ontología de aplicación (Guarino 1998) que formaliza gran parte de la especificación OKI v2.0.0.

En primer lugar se formaliza el dominio de conocimiento de la ontología mediante la creación de un metamodelo UML para la especificación OKI-OSID. En segundo lugar se presenta la creación de dicha ontología en OWL extendida con el lenguaje de reglas SWRL y finalmente, su integración en la arquitectura ontológica multinivel propuesta en este trabajo de tesis.

6.1. Formalización de OntoOKI

Tal como se ha comentado en el capítulo 2, Estado de la Cuestión, la especificación OKI-OSID define un conjunto de interfaces de servicios web para entornos de enseñanza virtual dirigidos a la comunidad educativa de estudios superiores.

OKI, desde un punto de vista próximo a la implementación, propone un conjunto de interfaces y los organiza en packages. Sin embargo, **no existe un mapa conceptual que defina las clases propuestas por la especificación, ni sus interrelaciones.**

Aunque esta especificación está compuesta por dieciocho OSID (*Open Service Interface Definitions*), en este trabajo serán analizados sólo algunos de ellos. Concretamente aquellos que describen servicios propios de entornos virtuales de aprendizaje superior a 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, Authorization y Authentication.

Los OSID están relacionados entre sí y aunque algunos de ellos se definen independientemente (Agent/Repository), otros hacen referencia a clases externas en su definición. Sólo para definir el OSID Workflow se hace referencia a un OSID que no figura entre los elegidos (Authorization). El OSID Authorization ofrece interfaces para métodos que invocan servicios relacionados con permisos y derechos de acceso, sin embargo en este trabajo este aspecto es tratado a más alto nivel, concretamente en la ontología genérica o de dominio.

En la tabla 6.1 se presentan las relaciones entre el conjunto de OSID objeto de estudio.

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

Cuadro 6.1: Relaciones entre OSIDs

Para un conocimiento en profundidad de la especificación OKI-OSID v2.0.0 se remite al lector a la propia especificación¹.

Como resultado del estudio de la especificación OKI-OSID v2.0.0 se ha descrito un metamodelo para dicha especificación utilizando diagramas de clase UML.

¹<http://okicomunity.mit.edu/staticpages/index.php?page=OverOKI>

Aunque interesa la visión global de todos ellos, se ha considerado oportuno presentar distintos diagramas, uno para cada OSID, con el objetivo de facilitar la legibilidad y comprensión de dicho metamodelo, al igual que la tabla 6.1 mostrada anteriormente.

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

- 1) El correspondiente diagrama de clases UML, en el que las clases propias del OSID se detallan a nivel de propiedades, mientras que las externas aparecen simplemente vacías, sin ningún atributo.
- 2) Las propiedades de cada clase, su descripción, el tipo de dato asociado y sus restricciones se presentan en la tabla de propiedades correspondiente a la clase.
- 3) 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*.
- 4) Los tipos de relación existentes, entre las clases del OSID en cuestión y los de otras clases externas al mismo, se recogen en una tabla de relaciones que muestra una descripción de cada relación, el dominio y el rango de la misma y sus restricciones. Además, para aquellos tipos de relación que puedan ser derivados a partir de otros, se presenta textualmente la correspondiente regla de derivación.
- 5) 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.
- 6) Las limitaciones detectadas en la especificación OKI-OSID para el OSID analizado.

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

6.1.1. OSID Agent

El OSID *Agent* está diseñado para crear, gestionar y consultar agentes. Existen grupos de agentes que pueden contener otros grupos que también son gestionados y accedidos por interfaces de este package.

En la figura 6.1, se muestra el diagrama de clases correspondiente al package Agent de OSID-OKI v2.0.0 en el que se distinguen dos clases: **Agent** y **Group**.

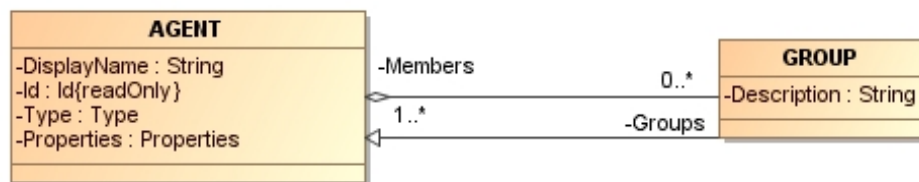


Figura 6.1: Diagrama de clases del OSID Agent

Clase Agent

Un *Agent* o agente es una entidad (individuo o proceso) que invoca servicios específicos. En la tabla 6.2 se muestran las propiedades de esta clase:

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un agente	String	Cardinalidad=1
Id	Identificador de agente, único e inamovible	Id	Cardinalidad=1
Properties	Propiedades de un agente a definir por la implementación	Properties	
Type	Caracterización de un agente	Type	Cardinalidad=1

Cuadro 6.2: Propiedades de la clase Agent

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, por lo que se presentan dichas relaciones al definir los tipos de relación de las demás clases relacionadas con ella.

En el caso de ejemplo, María sería una instancia de agente que actuaría como profesora responsable de asignatura y por tanto le correspondería el valor PRA en la propiedad *Type*. El equipo de consultores de la asignatura (Juan, Ana, Jaime, Pilar, Óscar, Manuel,

Antonio, Ángel y Marta) serían instancias de agente con valor PDC en la propiedad *Type*. Así mismo, Lidia como técnico de apoyo docente vendría representada por instancias de agente con el valor TAD en la propiedad *Type*.

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 tal como se muestra en la tabla 6.3:

Propiedad	Descripción	Tipo de dato	Restricción
Description	Descripción de un grupo	String	Cardinalidad=1

Cuadro 6.3: Propiedades de la clase Group

En el caso de ejemplo, podrían identificarse varios grupos: el grupo de consultores de la asignatura, el grupo de consultores activos y el grupo de técnicos de apoyo docente.

El grupo de consultores de la asignatura Bases de Datos I tendría como miembros a Juan, Ana, Jaime, Pilar, Óscar, Manuel, Antonio, Ángel, Marta, Ángel y Marta. El grupo de consultores activos de la asignatura Bases de Datos I estaría formado por todos los anteriores a excepción de Ángel y Marta, que formarían parte de un grupo complementario a éste, el grupo de consultores inactivos de la asignatura. Por otro lado, Lidia sería uno de los miembros del grupo de técnicos de apoyo docente.

Tipos de relación

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

En la tabla 6.4 se muestran los tipos de relación definidos entre agentes y grupos de este OSID:

Tipo de relación	Descripción	Dominio	Rango	Restricción
Members	Miembros que constituyen un grupo	Group	Agent	Cardinalidad Mín.1
Groups	Grupos a los que pertenece un agente	Agent	Group	

Cuadro 6.4: Tipos de relación de las clases del OSID Agent

Métodos

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

Limitaciones de OKI-OSID

Esta especificación 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, María es profesora responsable de asignatura, pero a su vez podría ejercer de consultora y formar parte del equipo docente de Bases de Datos I, en cuyo caso deberían de existir dos representaciones de María, una en calidad de PRA cuya propiedad tipo de agente sería PRA y otra en calidad de consultora cuya propiedad tipo de agente sería PDC.

6.1.2. OSID Assessment

El OSID *Assessment* tiene como finalidad proporcionar interfaces de servicios para crear, organizar, administrar, evaluar, almacenar y recuperar actividades de curso.

Las actividades llamadas *Assessment* siguen un ciclo de vida: primeramente se crean, luego se publican, posteriormente pueden ser tomadas por los estudiantes y una vez realizadas son entregadas con el objetivo de que sean evaluadas.

Además, puesto que las actividades se consideran formadas por unidades o componentes reutilizables, este OSID incluye otras clases como *Section* e *Item* que permiten

su composición y evaluación de forma independiente. Sin embargo, la entrega se realiza únicamente a nivel de actividad, pudiendo incluir ésta ejercicios y preguntas.

Una actividad puede encontrarse en distintos estados dependiendo del momento en que se encuentre en su ciclo de vida. Para cada posible estado de una actividad existe una clase distinta: *Assessment*, *AssessmentPublished* y *AssessmentTaken*. Análogamente sucede lo mismo para sus partes, *Section* y *SectionTaken* e *Item* e *ItemTaken*.

En la figura 6.2 se muestra el diagrama de clases correspondiente a este OSID.

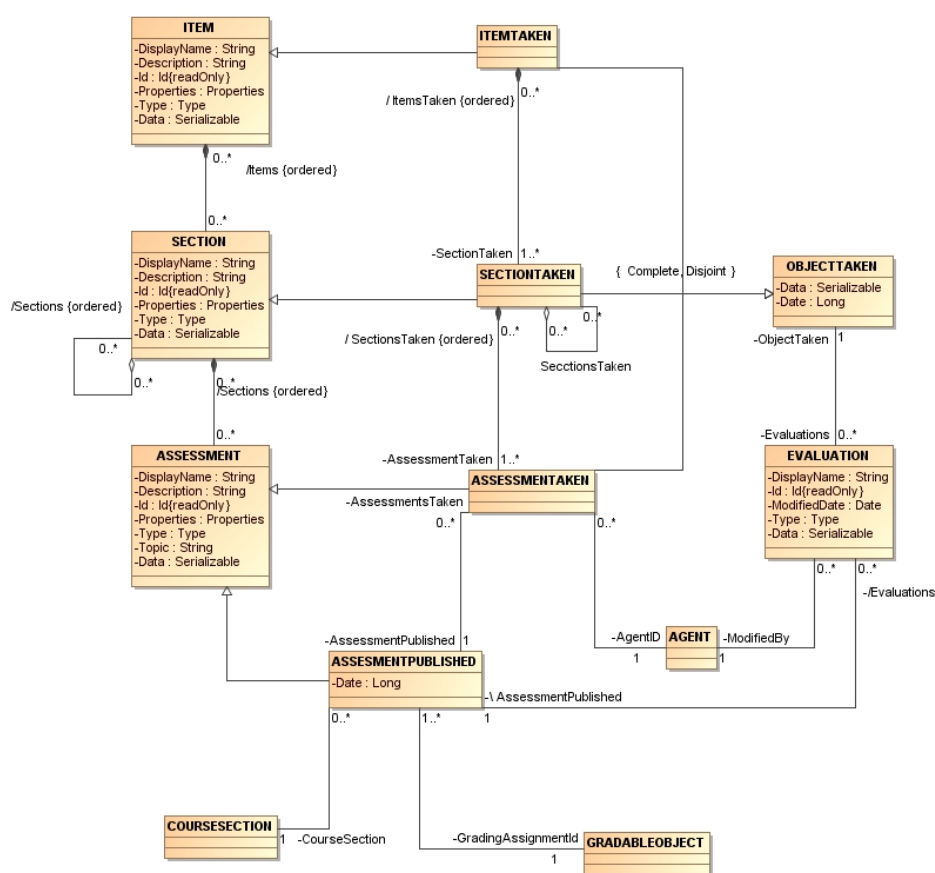


Figura 6.2: Diagrama de clases correspondiente al OSID Assesment

Las actividades entregadas son evaluadas. La información relativa a la evaluación se encuentra en la clase *Evaluation*. Debe notarse que aunque las actividades, ejercicios o preguntas puedan ser corregidos de forma independiente, la entrega siempre es de una actividad que contiene uno más ejercicios y este una o más preguntas.

Clase Assessment

Un *Assessment* o actividad constituye la estructura de organización de más alto nivel de este OSID. Se corresponde con el concepto de actividad entendida ésta como una unidad estructurada.

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.

Toda actividad tiene un nombre, una descripción y un identificador. Además es de un tipo determinado y tiene un tópico. El contenido asociado a dicha actividad se encuentra en la propiedad Data.

En la tabla 6.5 se presentan las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una actividad	String	Cardinalidad=1
Description	Descripción de una actividad	String	Cardinalidad=1
Id	Identificador de actividad	Id	Cardinalidad=1
Properties	Propiedades de una actividad a definir por la implementación	Properties	
Type	Tipo de actividad	Type	Cardinalidad=1
Data	Contenido de la actividad	Serializable	Cardinalidad=1
Topic	Tema asociado	String	Cardinalidad=1

Cuadro 6.5: Propiedades de la clase Assessment

En el caso de ejemplo, un Assessment sería cualquier actividad del curso de Bases de Datos I.

Puesto que el proceso Preparación del curso de Bases de Datos I supone crear una nueva edición del curso, deben de considerarse por un lado, los materiales y recursos docentes de cursos anteriores que se utilizarán de nuevo y por otro las actividades (PAC, práctica o examen) que se crean a propósito para la edición del nuevo curso.

Ejemplos de instancias de esta clase relativas al caso de ejemplo podrían ser: la primera PAC del curso Bases de Datos I (PAC1_2009_2010_1_BDI), la segunda práctica (PRA2_2009_2010_1_BDI) y el examen correspondiente al primer turno de examen (EX1_2009_2010_1_BDI) por poner algún ejemplo.

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 se muestran en la tabla 6.6 y, a excepción del tema, son las mismas que tienen la clase *assessment*.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una actividad	String	Cardinalidad=1
Description	Descripción de una actividad	String	Cardinalidad=1
Id	Identificador de actividad	Id	Cardinalidad=1
Properties	Propiedades de una actividad a definir por la implementación	Properties	
Type	Tipo de actividad	Type	Cardinalidad=1
Data	Contenido de la actividad	Serializable	Cardinalidad=1

Cuadro 6.6: Propiedades de la clase *Assessment*

En el caso de ejemplo, a diferencia de las PAC y prácticas que las prepara un único consultor, los exámenes son preparados por más de un miembro del equipo docente dado que su preparación se realiza mediante ejercicios independientes.

Cada examen está formado 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 (EJ1_EX1_2009_2010_1_BDI, EJ2_EX1_2009_2010_1_BDI y EJ3_EX1_2009_2010_1_BDI) respectivamente.

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 también puede ser creada independientemente sin formar parte de ningún ejercicio o actividad para ser reutilizada posteriormente.

Las propiedades de esta clase coinciden con las de la clase *Section* por lo que no se presentan de nuevo.

Suponiendo el caso de ejemplo, instancias de esta clase son las preguntas de los exámenes del curso de Bases de Datos I. Si el ejercicio 1 del primer turno de examen (EJ1_EX1_2009_2010_1_BDI) consta de dos preguntas tendríamos como instancias de esta clase PRG1_EJ1_EX1_2009_2010_1_BDI y PRG2_EJ1_EX1_2009_2010_1_BDI.

Clase `AssessmentPublished`

Un *AssessmentPublished* 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 se muestra en la tabla 6.7.

Propiedad	Descripción	Tipo de dato	Restricción
Date	Fecha de publicación	Date	Cardinalidad=1

Cuadro 6.7: Propiedades de la clase `AssessmentPublished`

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_2009_2010_1_BDI.

Clase `ObjectTaken`

La clase *ObjectTaken* u objeto evaluable es cualquier 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* que se detallan a continuación de ésta, 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.

Las propiedades de la clase *ObjectTaken* son las mostradas en la tabla 6.8.

Propiedad	Descripción	Tipo de dato	Restricción
Data	Contenido o Información sobre el objeto entregado	Serializable	Cardinalidad=1
Date	Fecha de entrega del objeto	Long	Cardinalidad=1

Cuadro 6.8: Propiedades de la clase `ObjectTaken`

Según la especificación OKI-OSID, el control de autorizaciones para las entregas de actividades según el curso en el que está matriculado el estudiante y el aula en que está inscrito debe realizarse en tiempo de ejecución.

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. No obstante, una vez iniciado el curso, podrían ser instancias de esta clase el primer ejercicio del primer turno de examen (EJ1_EX1_2009_2010_1_BDI) o la primera PAC (PAC1_2009_2010_1_BDI) entregada por un estudiante concreto.

Clase `AssesmentTaken`

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* por lo que no se presentan en este apartado.

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 (PAC1_2009_2010_1_BDI) o el primer examen (EX1_2009_2010_1_BDI).

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 por lo que tampoco se detallan en este apartado.

En el caso de ejemplo, como se ha comentado anteriormente no existen instancias de

esta clase, pero en un estadio más avanzado del curso podría serlo el primer o segundo ejercicio (EJ1_EX1_2009_2010_1_BDI, EJ2_EX1_2009_2010_1_BDI) del primer turno de examen entregado por un estudiante determinado.

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 corresponder alguna instancia de la clase *SectionTaken*. Análogamente, no se presentan sus atributos por los mismos motivos que se han comentado para las dos clases anteriores.

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 (PREG1_EJ1_EX1_2009_2010_1_BDI) entregado por un estudiante dado.

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 la tabla 6.9 se muestran las propiedades que caracterizan la clase *Evaluation*:

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de la evaluación	String	Cardinalidad=1
Id	Identificador de evaluación	Id	Cardinalidad=1
Type	Tipo de evaluación	Type	Cardinalidad=1
Data	Información específica sobre la evaluación	Serializable	Cardinalidad=1
ModifiedDate	Fecha de actualización de la evaluación	Long	Cardinalidad=1

Cuadro 6.9: Propiedades de la clase *Evaluation*

Respecto al caso de ejemplo, dado que no se consideran actividades entregadas tampoco existen resultados de la evaluación. Sin embargo, suponiendo que un estudiante concreto ha entregado la primera PAC (PAC1_2009_2010_1_BDI), existiría una instancia

de evaluación que en la propiedad *Data* almacenaría toda la información sobre la evaluación de la PAC entregada y en la propiedad *ModifiedBy* tendría la fecha de la última vez en que se ha evaluado.

Tipos de relación

Los tipos de relación entre clases de este OSID y otras clases externas son los mostrados en la tabla 6.10.

Tipos de relación	Descripción	Dominio	Rango	Restricción
Sections	Sections o ejercicios que ordenados forman una actividad u otro ejercicio	Assessment	Section	Transitiva
Items	Items o preguntas que ordenadas forman un ejercicio	Section	Item	
GradingAssignmentId	Criterios de evaluación de una actividad publicada	AssessmentPublished	GradableObject	Cardinalidad=1
CourseSectionId	Curso donde se está publicada una actividad concreta	AssessmentPublished	CourseSection	Cardinalidad=1
AgentId	Estudiante que entrega una actividad	AssessmentTaken	Agent	Cardinalidad=1
AssessmentPublished	Actividad publicada de referencia de una actividad entregada o una evaluación	AssessmentTaken, Evaluation	AssessmentPublished	Cardinalidad=1
AssessmentTaken	Actividad entregada de la que forma parte un ejercicio entregado	SectionTaken	AssessmentTaken	Cardinalidad=1
AssessmentsTaken	Actividad entregada de una actividad publicada	AssessmentPublished	AssessmentTaken	
SectionTaken	Ejercicio entregado del que forma parte una pregunta entregada	ItemTaken	SectionTaken	Cardinalidad=1
SectionsTaken	Ejercicio de una actividad entregada u otros ejercicios	SectionsTaken	SectionsTaken	Transitiva
ItemsTaken	Preguntas de un ejercicio entregado	SectionsTaken	ItemsTaken	
ObjectTaken	Objeto al que se refiere una evaluación	Evaluation	ObjectTaken	Cardinalidad=1
Evaluations	Evaluaciones relativas al objeto entregado o una actividad publicada	ObjectTaken, AssessmentPublished	Evaluation	
ModifiedBy	Agente evaluador de una evaluación	Evaluation	Agent	Cardinalidad=1

Cuadro 6.10: Tipos de relación de las clases del OSID CourseManagement

La organización de actividades como conjuntos de ejercicios ordenados y los ejercicios como un conjunto de preguntas ordenadas para promover la reusabilidad, requiere saber qué ejercicios forman una actividad (Sections) u otro ejercicio y como están ordenados, así como qué preguntas constituyen un ejercicio (ítems) y su ordenamiento. 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 tiene establecidos unos criterios de evaluación (GradingAssignmentId). Los estudiantes toman estas actividades para su realización y una vez entregadas, se evalúan, obteniéndose así los resultados de su evaluación (Evaluations) para cada actividad.

Las actividades entregadas provienen de alguna actividad publicada (AssessmentPublished) que ha sido realizada y entregada por un estudiante (AgentId). Puesto que una actividad entregada puede estar formada por un conjunto de ejercicios, que pueden ser evaluados independientemente, interesa saber qué ejercicios entregados constituyen una actividad entregada (SectionsTaken) u otros ejercicios contenidos en cada uno de ellos con la finalidad de para poder ser evaluados y al revés, a partir de un ejercicio entregado debe de poderse obtener la actividad entregada a evaluar que lo contiene (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 interesa saber a 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 ordenamiento pueden obtenerse a partir de todas las partes que forman la secuencia de ejercicios que determina la actividad (22, 23). Análogamente para las preguntas u ejercicios que componen otro ejercicio (24, 25), así como para la composición de actividades y ejercicios entregados.

Regla 22 (Rule-FirstSection):

El conjunto ordenado de ejercicios que componen una actividad incluye el primer ejercicio de la misma.

Regla 23 (Rule-NextSection):

El conjunto ordenado de ejercicios que componen una actividad incluye cualquier ejercicio que sea precedido por otro perteneciente a dicho conjunto.

Regla 24 (Rule-FirstItem):

El conjunto ordenado de ejercicios o preguntas que componen un ejercicio incluye el primer ejercicio o pregunta del mismo.

Regla 25 (Rule-NextItem):

El conjunto ordenado de ejercicios o preguntas que componen un ejercicio incluye cualquier ejercicio que sea precedido por otro perteneciente a dicho conjunto.

También mediante reglas de derivación (26, 27 y 28) puede describirse cuál es la actividad publicada de referencia de alguna actividad entregada para su evaluación o de sus partes (preguntas u ejercicios).

Regla 26 (Rule-AssessmentPublishedOfEvaluatedAssessment):

La actividad publicada de referencia de una actividad evaluada es la actividad de referencia de la actividad entregada.

Regla 27 (Rule-AssessmentPublishedOfEvaluatedSection):

La actividad publicada de referencia de un ejercicio evaluado es la actividad de referencia a la cual pertenece el ejercicio entregado.

Regla 28 (Rule-AssessmentPublishedOfEvaluatedItem):

La actividad publicada de referencia de una pregunta evaluada es la actividad de referencia de la actividad a la cual pertenece el ejercicio que contiene la pregunta entregada.

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`).

Limitaciones de OKI-OSID

La especificación OKI-OSID no permite compartir una misma actividad publicada entre distintas aulas de un curso, exigiendo la publicación de una misma actividad en cada una de las aulas. Sería interesante pues que las publicaciones pudieran establecerse a nivel de curso.

Esta especificación carece de interfaces para conocer cuales son las actividades entregadas por cada estudiante y las evaluaciones llevadas a cabo por cada agente evaluador. No obstante, esta información puede obtenerse a partir de un simple recorrido por las actividades entregadas o las evaluaciones realizadas.

6.1.3. OSID CourseManagement

El objetivo del OSID *CourseManagement* es soportar la creación y gestión del catálogo de cursos. En la figura 6.3 se muestra el diagrama de clases UML correspondiente este OSID.

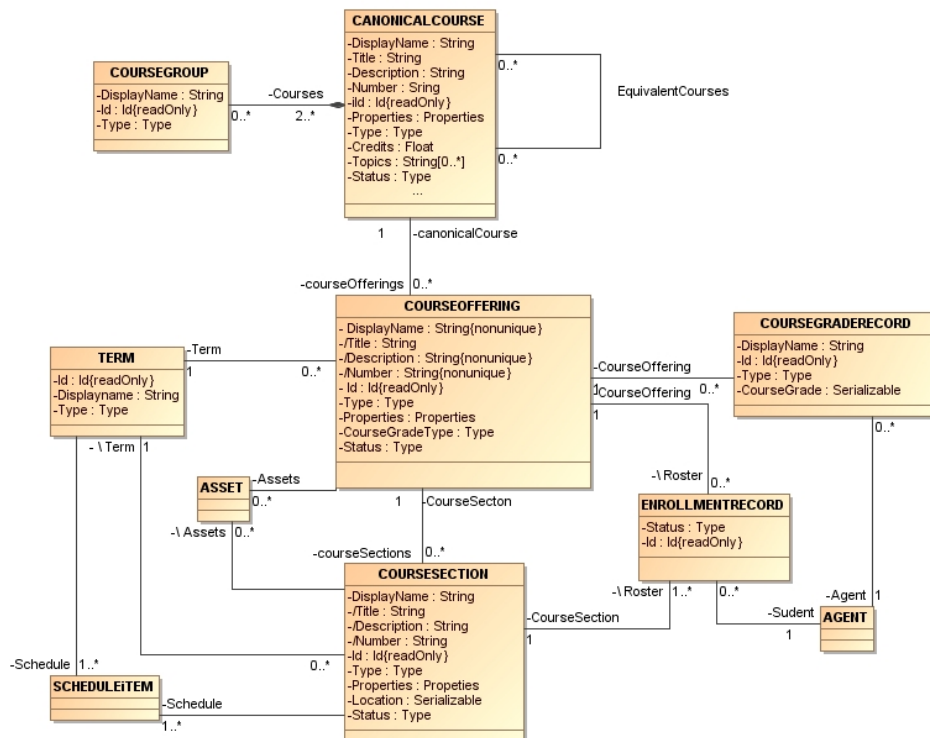


Figura 6.3: Diagrama de clases del OSID CourseManagement

El catálogo de cursos está organizado principalmente en tres niveles: **CanonicalCourse**, **CourseOffering** y **CoursesSection**. 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.

A continuación se describen cada una de las clases que constituyen este diagrama de clases UML.

Clase CanonicalCourse

Un *CanonicalCourse* o curso canónico define la estructura de organización de más alto nivel en el package CourseManagement.

También puede entenderse que un curso canónico representa el concepto de asignatura definida a nivel de plan de estudios, es decir con independencia de cursos académicos concretos y aulas donde se imparta. Y dentro de este contexto en relación 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 de otras propiedades propias del curso como pueden ser el tipo de curso o el estado del mismo.

En la tabla 6.11 se presentan las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un curso canónico	String	Cardinalidad=1
Title	Título del curso	String	Cardinalidad=1
Description	Descripción del curso canónico	String	Cardinalidad=1
Number	Número de curso	String	Cardinalidad=1
Id	Identificador de curso	Id	Cardinalidad=1
Properties	Propiedades de un curso a definir por la implementación	Properties	Ninguna
Type	Tipo del curso	Type	Cardinalidad=1
Credits	Número de créditos del curso	Float	Cardinalidad=1
Topics	Tópicos del curso	String	Ninguna
Status	Estado del curso	Type	Cardinalidad=1

Cuadro 6.11: Propiedades de la clase CanonicalCourse

En el caso de ejemplo, una instancia de curso canónico es la asignatura Bases de

Datos I que se imparte en dos titulaciones: la Ingeniería Informática de Gestión y la Ingeniería Informática de Sistemas.

Clase CourseOffering

Un *CourseOffering* o curso ofrecido es esencialmente la concreción de un curso canónico con un tipo de evaluación para un periodo académico determinado, es decir una asignatura que se imparte en un periodo académico concreto.

En la tabla 6.12 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un curso ofrecido	String	Cardinalidad=1
Title	Título del curso	String	Cardinalidad=1
Description	Descripción del curso ofrecido	String	Cardinalidad=1
Number	Número de curso	String	Cardinalidad=1
Id	Identificador de curso	Id	Cardinalidad=1
Properties	Propiedades de un curso a definir por la implementación	Properties	Ninguna
Type	Tipo de curso	Type	Cardinalidad=1
CourseGradeType	Tipo de evaluación del curso	Type	Cardinalidad=1
Status	Estado del curso	Type	Cardinalidad=1

Cuadro 6.12: Propiedades de la clase CourseOffering

Las propiedades *Title*, *Description*, *Number* pueden ser derivadas de las correspondientes en la clase *CanonicalCourse* o redefinidas. Además, todo curso que se ofrece tiene definido un tipo de evaluación, se le asocia un tipo de curso y tienen un indicador del estado del mismo.

En el caso de ejemplo, el curso que se prepara para el semestre 2009-2010-1, es el curso de Bases de Datos I 2009-2010-1 correspondiente a la concreción de curso Bases de Datos I en para el periodo académico de otoño 2009. Este curso sigue un modelo de evaluación tradicional basado en examen y evaluación continuada.

Clase Term

Un *Term* es un periodo académico que corresponde a un tipo determinado. Debe entenderse como un periodo durante el cual están planificadas un conjunto de tareas

necesarias para la impartición de un curso.

En la tabla 6.13 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un curso académico	String	Cardinalidad=1
Id	Identificador de curso académico	Id	Cardinalidad=1
Type	Tipo de curso	Type	Cardinalidad=1

Cuadro 6.13: Propiedades de la clase Term

En el caso de ejemplo, la preparación del curso de Bases de Datos I es para el periodo académico 2009-2010-1, cuyo tipo de curso es Otoño 2009.

Clase CourseSection

Un *CourseSection* o aula es la concreción de un curso ofrecido fijando la localización donde se imparte el curso de acuerdo con una planificación dada.

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* y además, tiene asociadas otras propiedades como son el tipo o el estado del curso.

En la tabla 6.14 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un aula	String	Cardinalidad=1
Title	Título de aula	String	Cardinalidad=1
Description	Descripción del aula	String	Cardinalidad=1
Number	Número de aula	String	Cardinalidad=1
Id	Identificador de aula	Id	Cardinalidad=1
Properties	Propiedades de un aula a definir por la implementación	Properties	Ninguna
Type	Tipo de aula	Type	Cardinalidad=1
Location	Ubicación del aula	Serializable	Cardinalidad=1
Status	Estado del curso	Type	Cardinalidad=1

Cuadro 6.14: Propiedades de la clase CourseSection

En el caso de ejemplo, existirían tantas instancias de `CourseSection` como aulas de Bases de datos, por lo que se crearían instancias de aulas de tipo Teórico (BDI.1, BD.I.2, BD.I.3, BD.I.4) e instancias de aulas de tipo Laboratorio (Lab_BD.I.1, Lab_BD.I.2), tantas como aulas deban de crearse de cada tipo.

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 la tabla 6.15 se muestran las propiedades de la clase *CourseGradeRecord*.

Propiedad	Descripción	Tipo de dato	Restricción
<code>DisplayName</code>	Denominación del registro de evaluación	String	Cardinalidad=1
<code>Id</code>	Identificador del registro de evaluación	Id	Cardinalidad=1
<code>Type</code>	Tipo del registro de evaluación	Type	Cardinalidad=1
<code>CourseGrade</code>	Evaluación del curso	Serializable	Cardinalidad=1

Cuadro 6.15: Propiedades de la clase `CourseGroupRecord`

En el caso de ejemplo, no existen instancias esta clase, ya que un registro de evaluación queda definido cuando un estudiante entrega una actividad para ser evaluada y esto no sucede antes de haberse iniciado el curso. Suponiendo el curso iniciado, podríamos mencionar como ejemplo la PAC1 entregada por un estudiante concreto, sea por ejemplo el registro de evaluación correspondiente a la PAC1_2009_2010_1_BDI entregada por un estudiante concreto.

Clase `EnrollmentRecord`

Un *EnrollmentRecord* o registro de matrícula almacena información sobre las matriculas, es decir los estudiantes que se han matriculado en cursos, especialmente el estado de la matrícula.

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

En la tabla 6.16 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
Status	Estado de la matriculación de un estudiante en un curso ofrecido y/o aula	Type	Cardinalidad=1
Id	Identificador del registro de matrícula	Id	Cardinalidad=1

Cuadro 6.16: Propiedades de la clase EnrollmentRecord

En el caso de ejemplo, serían instancias de esta clase cada una de las matrículas asociadas a cada aula del curso Bases de Datos I. Debe de observarse que el registro de matrículas sumariadas por curso ofrecido se puede derivar a partir de la unión de todas las matrículas de estudiantes inscritos en aulas concretas.

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, ya sea por una cadena de prerrequisitos, correquisitos o bien porque constituyan algún 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 se circunscribe a la preparación de un único curso sin considerar su pertenencia a grupos de asignaturas.

En la tabla 6.17 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación del grupo de cursos canónicos	String	Cardinalidad=1
Id	Identificador del grupo de cursos	Id	Cardinalidad=1
Type	Tipo del grupo de cursos	Type	Cardinalidad=1

Cuadro 6.17: Propiedades de la clase GroupCourse

Tipos de relación

Entre las clases del OSID *CourseManagement* y otras clases externas relacionadas con él se establecen ciertos tipos de relación. Éstos se presentan en la Tabla 6.18.

Tipo de relación	Descripción	Dominio	Rango	Restricción
EquivalentCourses	Cursos canónicos equivalentes	CanonicalCourse	CanonicalCourse	Simétrica y Transitiva
Courses	Cursos canónicos que forman un grupo de cursos	CourseGroup	CanonicalCourse	Mínimo dos
CourseOfferings	Cursos ofrecidos asociados a un curso canónico	CanonicalCourse	CourseOffering	
CourseSections	Aulas asociadas a un curso ofrecido	CourseOffering	CourSection	
Term	Curso académico asociado a un curso ofrecido o aula	CourseOffering, CourseSection	Term	Cardinalidad=1, Simétrica
CanonicalCourse	Curso canónico correspondiente a un curso ofrecido	CourseOffering	CanonicalCourse	Cardinalidad=1
CourseOffering	Curso al que se asocia un aula, un registro de evaluación o una matrícula	CourseSection, CourseGradeRecord, EnrollmentReccord	CourseOffering	Cardinalidad=1
CourseSection	Aula a la que se asociada a una matrícula	EnrollmentReccord	CourseSection	Cardinalidad=1
Assets	Materiales asociados a un curso ofrecido o un aula	CourseOffering, CourseSection	Asset	Simétrica
Roster	Matrículas asociadas a un curso o aula	CourseOffering, CourseSection	EnrollmentRecord	
Schedule	Tareas planificadas asociadas a un aula	CourseSection	ScheduleItem	Cardinalidad Mín.1
AgentId	Agente que define un registro de evaluación o que se matricula en un aula de un curso	CourseGradeRecord, EnrollmentRecord	Agent	Cardinalidad=1

Cuadro 6.18: Tipos de relación de las clases del OSID CourseManagement

Los cursos canónicos pueden tener cursos canónicos equivalentes (EquivalentCourses) y cada asignatura o curso canónico se puede ofrecer en uno o más cursos (CourseOfferings).

Para cada curso ofrecido existen relaciones que permiten determinar a qué curso canónico está asociado (CanonicalCourse), a qué curso académico pertenece (Term), qué materiales utiliza (Assets), qué estudiantes están matriculados en él (Roster) y cuáles son las aulas donde se imparte (CourseSections).

Las aulas o CourseSections están relacionadas con el curso ofrecido que imparten (CourseOffering), el período académico en cuestión (Term), la planificación del curso entendida como un conjunto de tareas (Schedule), el conjunto de materiales utilizados (assets) y todas sus matriculas (Roster).

La relación de un curso canónico o asignatura respecto a un grupo de cursos viene determinada por la relación (Courses) de manera que permite saber qué cursos forman cada grupo de cursos. La especificación OKI-OSID no proporciona interfaces para la

navegabilidad en sentido contrario para saber a qué grupo de cursos pertenece un curso canónico.

Un registro de evaluación pertenece a un curso ofrecido (*CourseOffering*) y es definido por el 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 que se han asignado a un aula o curso y cuáles son 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, como: *Description*(29,30), *Title*(31,32), *Number*(33,34), *Roster* (35,36), *Term* (37) y *Assets*(38).

Regla 29 (Rule-DescriptionOfCourseOffering):

La descripción de un curso ofrecido puede coincidir con la descripción del curso canónico al que se asocia.

Regla 30 (Rule-DescriptionOfCourseSection):

La descripción de un aula puede coincidir con la descripción del curso canónico al que se asocia el curso ofrecido que se imparte en dicha aula.

Regla 31 (Rule-TitleOfCourseOffering):

El título de un curso ofrecido puede coincidir con el título del curso canónico al que se asocia.

Regla 32 (Rule-TitleOfCourseSection):

El título de un aula puede coincidir con el título del curso canónico al que se asocia el curso ofrecido que se imparte en dicha aula.

Regla 33 (Rule-NumberOfCourseOffering):

El número de curso ofrecido puede coincidir con el número de curso canónico al que se asocia.

Regla 34 (Rule-NumberOfCourseSection):

El número de aula puede coincidir con el número de curso canónico al que se asocia el curso ofrecido que se imparte en dicha aula.

Regla 35 (Rule-RosterOfCourseSection):

Las matrículas asociadas a un aula son las matrículas de los estudiantes matriculados en el curso que se imparte en dicha aula.

Regla 36 (RosterOfCourseOffering):

Las matrículas asociadas a un curso son las matrículas de los estudiantes matriculados en todas las aulas donde se imparte dicho curso.

Regla 37 (Rule-TermOfCourseSection):

Las aulas donde se imparte un curso tiene asociado el mismo período académico que el curso que se imparte en ellas.

Regla 38 (Rule-AssetOfCourseSection):

Las aulas donde se imparte un curso tienen asociados los mismos materiales que el curso que se imparte en ellas.

Métodos

La especificación además 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`). Así como para obtener periodos académicos a partir de una fecha determinada (`getTermByDate`).

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

Limitaciones OKI-OSID

En lo que respecta a gestión de cursos se observa que la especificación OKI-OSID no contempla la posibilidad de más de un evaluador por curso. Podría darse el caso de que la evaluación de las actividades teóricas y la evaluación de las actividades prácticas las llevara a cabo distinto agente o que la evaluación de preguntas de examen se distribuyera entre un conjunto de evaluadores.

Tampoco permite la asignación de tareas a nivel de curso, sino que la asignación de tareas se realiza a nivel de aula, impidiendo así la compartición de tareas asociadas a un curso entre todas las aulas del curso.

Además, se observa que esta especificación no proporciona interfaces para saber qué cursos o aulas pertenecen a un periodo académico dado.

Por otro lado, aunque la clase *Agent* no ofrece interfaces para determinar en qué cursos está matriculado un estudiante, ni a qué aulas ha sido asignado, sí que es derivable esta información a partir de la información almacenada en la clase *EnrollmentRecord*. Análogamente ocurre con los resultados de las evaluaciones de un curso o las correcciones realizadas por un evaluador determinado.

6.1.4. OSID Grading

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

En la figura 6.4 se muestra el diagrama de clases UML correspondiente a este OSID con las clases *GradableObject* y *GradeRecord* y otras clases externas relacionadas.

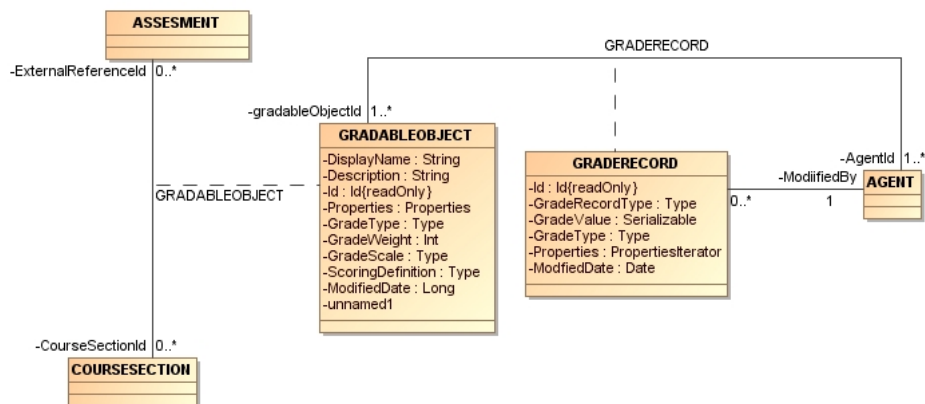


Figura 6.4: Diagrama de clases correspondiente al OSID Grading

La especificación de la evaluación de alguna actividad entregada requiere de cuatro elementos: un valor, un tipo, una escala y una definición de la calificación.

Clase *GradableObject*

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

En la tabla 6.19 se presentan las propiedades de un objeto de evaluación.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación del objeto evaluable	String	Cardinalidad=1
Descripción	Descripción del objeto de evaluación	String	Cardinalidad=1
Id	Identificador de objeto evaluable	Id	Cardinalidad=1
Properties	Propiedades del objeto evaluable a definir por la implementación	Properties	Ninguna
GradeType	Tipo de evaluación	Type	Cardinalidad=1
GradeScale	Escala o intervalos de valores posibles	Type	Cardinalidad=1
GradeWeight	Ponderación	Int	Cardinalidad=1
ScoringDefinition	Descripción de calificaciones	Type	Cardinalidad=1
ModifiedDate	Fecha de la última actualización	Long	Cardinalidad=1

Cuadro 6.19: GradabeObjectProperties

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. Para el caso de las PAC tendríamos dos objetos evaluables (GO_PAC1.2009_2010.1_BDI y GO_PAC2.2009_2010.1_BDI) para cada PAC del curso.

Clase GradeRecord

Un GradeRecord o registro de calificación almacena la calificación obtenida por cada estudiante y actividad entregada. Las propiedades de esta clase se muestran en la tabla 6.20.

Propiedad	Descripción	Tipo de dato	Restricción
Id	Identificador de objeto evaluable	Id	Cardinalidad=1
GradeRecordType	Tipo del registro de evaluación	Type	Cardinalidad=1
GradeValue	Calificación obtenida	Serializable	Cardinalidad=1
GradeType	Tipo de evaluación	Type	Cardinalidad=1
Properties	Propiedades del objeto evaluable a definir por la implementación	Properties	
ModifiedDate	Fecha de la última actualización	Long	Cardinalidad=1

Cuadro 6.20: Propiedades de la classe GradeRecord

Para cada registro de evaluación, además del valor correspondiente al resultado de

la evaluación de la actividad en cuestión, debe de especificarse el tipo de evaluación aplicado y la fecha en que se ha evaluado.

En el caso de ejemplo, no existen instancias de esta clase puesto que los registros de calificaciones se generan una vez se ha procedido a la evaluación de las actividades y ello sucede de acuerdo con el calendario académico un vez iniciado el curso.

Suponiendo entregada y evaluada la PAC1, serían instancias de esta clase los registros de calificación para cada estudiante para esta actividad.

Tipos de relación

Las relaciones entre las clases de este OSID y otras clases externas son las que se detallan en la tabla 6.21.

Tipos de relación	Descripción	Dominio	Rango	Restricción
CourseSectionId	Aula a la que se asocia un objeto evaluable	GradableObject	CourseSection	Cardinalidad=1
ExternalReferenceId	Actividad a la que se asocia un objeto evaluable	GradableObject	Assessment	Cardinalidad=1
GradableObjectId	Objeto evaluable asociado a un registro de evaluación	CourseGraderecord	GradableObject	Cardinalidad=1
AgentId	Estudiante al que corresponde un registro de evaluación	CourseGraderecord	Agent	Cardinalidad=1
ModifiedBy	Agente que ha actualizado el registro de evaluación	CourseGraderecord	Agent	Cardinalidad=1

Cuadro 6.21: Tipos de relación de las clases del OSID Grading

Los objetos de evaluación establecen los criterios de evaluación de cada actividad asociada a un aula, por lo que es posible determinar a qué aula se refieren (CourseSectionId) y a qué actividad (ExternalReferenceId).

Los registros de calificaciones en cambio existen para cada evaluación de una actividad entregada por algún estudiante, por lo que es posible determinar qué estudiante ha realizado la actividad (AgentId) y cuales son los criterios de evaluación utilizados (GradableObjectId).

También se almacena el agente que ha realizado la evaluación (ModifiedBy) de la actividad evaluada.

Métodos

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

Limitaciones de OKI-OSID

Esta especificación no permite definir objetos de evaluación a nivel de curso y compartirlos entre las distintas aulas donde se imparte el curso. Para que ello fuera posible debería de definirse el objeto de evaluación a nivel de curso ofrecido.

6.1.5. OSID Scheduling

El objetivo del OSID *Scheduling* es la asociación de agentes a tareas específicas que tienen asignado un periodo de realización. 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 6.5.

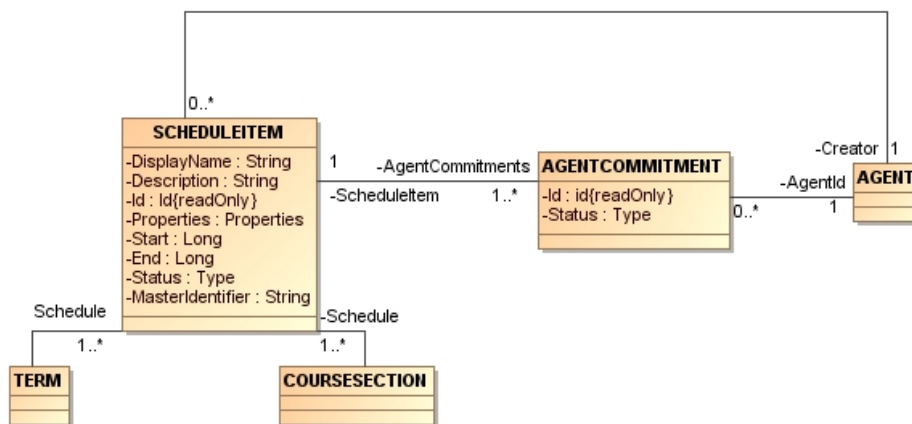


Figura 6.5: Diagrama de clases correspondiente al OSID Scheduling

Clase ScheduleItem

La clase *ScheduleItem* o ítem de planificación captura información relativa a cada una de las tareas que debe de llevarse 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. En la tabla 6.22 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de un ítem de planificación	String	Cardinalidad=1
Description	Descripción de ítem de planificación	String	Cardinalidad=1
Id	Identificador de ítem de planificación	Id	Cardinalidad=1
Properties	Propiedades del ítem de planificación definidas por la implementación	Property	
MasterIdentifier	Identificador opcional proporcionado por otra aplicación	String	Cardinalidad=1
Start	Fecha de inicio	Long	Cardinalidad=1
End	Fecha de finalización	Long	Cardinalidad=1
Status	Estado del ítem de planificación	Type	Cardinalidad=1

Cuadro 6.22: Propiedades de la clase *ScheduleItem*

Debe notarse que la especificación OKI-OSID utiliza el tipo de dato *Long* para las fechas.

En el caso de ejemplo de este trabajo, un encargo de consultoría sería una instancia de *ScheduleItem*. Así pues, el encargo de actualización del plan docente del Curso Bases de Datos I sería un ejemplo concreto, el cual se asignaría a todos los consultores activos de la asignatura del periodo académico 2009-2010-1, es decir todos ellos adquirirían un compromiso de realización y tendría una fecha de inicio y una fecha de finalización para la actualización.

Clase *AgentCommitment*

Un *AgentCommitment* o registro de compromiso almacena el grado de compromiso que adquiere cada agente en la tarea que participa. Por tanto se trata de mantener y gestionar las asignaciones de personas a tareas y el estado de la realización de las mismas, todo ello con independencia las características de la tarea y del periodo de tiempo durante el cual se va a llevar a cabo que son descritas en la clase *ScheduleItem*.

Las propiedades de esta clase son el identificador de la asignación y el estado de la asignación tal como se muestra en la tabla 6.23.

Propiedad	Descripción	Tipo de dato	Restricción
Id	Identificador de ítem de planificación	Id	Cardinalidad=1
Status	Estado que indica el nivel de compromiso adquirido	Type	Cardinalidad=1

Cuadro 6.23: Propiedades de la clase AgentCommitment

En el proceso *PrepararCursoBDI-0910-UOC* María como PRA asigna a cada uno de los consultores distintas tareas, para cada una de ellas y cada consultor involucrado existirá una instancia de esta clase. Por ejemplo, si Juan tuviera encomendada la preparación de la primera PAC existiría una instancia de esta clase asociada al agente Juan o si por el contrario Anna tuviera el encargo de preparar la primera pregunta de los tres turnos de examen, existirían tres instancias distintas de esta clase asociadas a Anna.

Tipos de relación

Los tipos de relación para las clases de este OSID son los que se detallan en la tabla 6.24 .

Tipo de relación	Descripción	Dominio	Rango	Restricción
AgentId	Agente a quien se encarga una tarea con cierto grado de compromiso	AgentCommitment	Agent	Cardinalidad=1
ScheduleItem	Tarea planificada	AgentCommitment	ScheduleItem	Cardinalidad=1
Creator	Agente que planifica la asignación de tarea	ScheduleItem	Agent	Cardinalidad=1
AgentCommitments	Encargos asignados en la planificación de una tarea	ScheduleItem	AgentCommitment	Card.Mín.1

Cuadro 6.24: Tipos de relación de las clases del OSID Scheduling

Para cada tarea planificada y agentes involucrados en ella existe un compromiso de realización de la misma por parte de éstos. Cada encargo asignado o compromiso de realización se asocia a un agente (AgentId) y tiene 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), tantos como agentes intervengan en ella.

Lo que no contempla la especificación OKI-OSID es la relación de tareas en las que

puede hallarse comprometido un agente y todas las tareas que tengan un mismo nivel de compromiso.

Métodos

La especificación ofrece interfaces para métodos consultores que permitan la consulta de ítems de planificación por identificador (`getScheduleItem`).

También ofrece interfaces adicionales para determinar disponibilidades coincidentes entre los agentes que intervienen en una misma tarea (`getAvailableTimes`), lo cual que puede resultar muy útil antes de crear un nuevo ítem de planificación, así como para conocer las tareas asignadas a un grupo de agentes durante un periodo de tiempo (`getScheduleItemsForAgents`).

Limitaciones de OKI-OSID

Este OSID no soporta la planificación de tareas constituidas por varias etapas si no es conjuntamente con el OSID Workflow. Por tanto únicamente con este OSID no sería posible la preparación de exámenes en varias etapas.

6.1.6. 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 como son: *Part*, *PartStructure*, *Record*, y *RecordStructure*, que permiten determinar la estructura y contenido de los contenidos almacenados en el repositorio.

Aunque para el caso de ejemplo se consideran las actividades de curso como un todo, es decir a nivel de *asset*, a continuación, en la Figura 6.6 se presenta la descripción de todo el OSID.

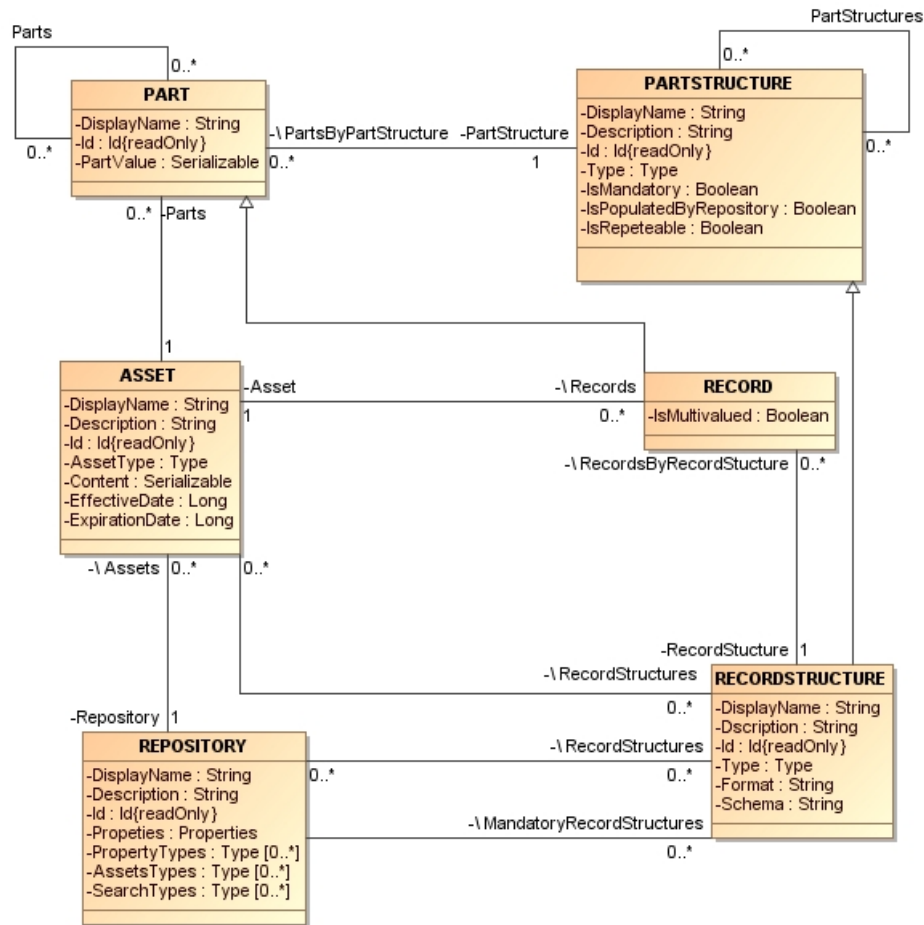


Figura 6.6: Diagrama de clases correspondiente al OSID Repository

Clase Asset

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

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).

Todo asset tiene un tipo y tiene un contenido, además de otras propiedades que se presentan en la tabla 6.25.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de recurso digital	String	Cardinalidad=1
Description	Descripción de recurso digital	String	Cardinalidad=1
Id	Identificador de recurso digital	Id	Cardinalidad=1
AssetType	Tipo de recurso digital	Type	Cardinalidad=1
EffectiveDate	Fecha de efecto	Long	Cardinalidad=1
ExpirationDate	Fecha de expiración	Long	Cardinalidad=1
Content	Contenido	Serializable	Cardinalidad=1

Cuadro 6.25: Propiedades de la clase Asset

De nuevo se observa que se utiliza el tipo de dato *Long* para las fechas, de acuerdo con OKI-OSID.

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*.

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 toda parte de asset forma parte de un asset. Además, una parte de asset puede ser estructurada y organizarse de acuerdo con una estructura determinada.

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 se presentan en la tabla 6.26.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una parte de recurso digital	String	Cardinalidad=1
Id	Identificador de la parte	Id	Cardinalidad=1
PartValue	Contenido	Serializable	Cardinalidad=1

Cuadro 6.26: Propiedades de la clase Part

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 (EJ1_EX1_2009_2010_1_BDI).

Clase PartStructure

Un *PartStructure* o estructura de una parte de recurso digital permite definir la estructuración de alguna parte de asset.

En la tabla 6.27 se presentan las propiedades de esta clase y se observa que la estructuración de la partes de assets se realiza determinando si las estructuras de registro son obligatorias u opcionales, repetibles o no y si son pobladas por algún repositorio.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una estructura de una parte de recurso digital	String	Cardinalidad=1
Description	Descripción de una estructura de una parte de recurso digital	String	Cardinalidad=1
Id	Identificador de estructura de recurso digital	Id	Cardinalidad=1
Type	Tipo de de estructura de parte de una parte de recurso digital	Type	Cardinalidad=1
IsMandatory	Indicador de obligatoria/opcional	Boolean	Cardinalidad=1
IsPopulatedByRepository	Indicador del tipo de población	Boolean	Cardinalidad=1
IsRepeteable	Indicador de repetición o no	Boolean	Cardinalidad=1

Cuadro 6.27: Propiedades de la clase PartStructure

En el caso de ejemplo, al considerarse los ejercicios de examen como 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, estos metadatos formarían parte de las estructuras de registros y serían instancias de esta clase.

Clase Record

Un *Record* o registro es una parte de asset que se define de acuerdo 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 a su vez estar formados por otras partes.

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

En la tabla 6.28 se muestran las propiedades de la clase *Record*.

Propiedad	Descripción	Tipo de dato	Restricción
IsMultivalued	Indicador de registro multivalor	Boolean	Cardinalidad=1

Cuadro 6.28: Propiedades de la clase *Record*

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.

Clase *RecordStructure*

Un *RecordStructure* o estructura de registro define la estructuración de un registro o parte de asset. Es decir, es una estructura de parte de asset que permite definir el formato y el esquema de representación del registro asociado, así como determinar si dicha estructura es repetible o no.

Sus propiedades se presentan en la tabla 6.29.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una estructura de registro	String	Cardinalidad=1
Description	Descripción de una estructura de registro	String	Cardinalidad=1
Id	Identificador de registro	Id	Cardinalidad=1
Type	Tipo de estructura de registro	Type	Cardinalidad=1
IsRepeteable	Indicador de estructura de registro repetitiva	Boolean	Cardinalidad=1
Format	Formato que usa	String	Cardinalidad=1
Schema	Esquema de representación que usa	String	Cardinalidad=1

Cuadro 6.29: Propiedades de la clase *RecordStructure*

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.

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.

Las propiedades que definen un repositorio básicamente están relacionadas con los tipos de contenidos digitales y los criterios para su localización. Se muestran en la tabla 6.30 .

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Denominación de una estructura de registro	String	Cardinalidad=1
Description	Descripción de una estructura de registro	String	Cardinalidad=1
Id	Identificador de registro	Id	Cardinalidad=1
Type	Tipo de repositorio	Type	Cardinalidad Mín.1
Properties	Propiedades del repositorio definidas por la implementación	Properties	Cardinalidad=1
PropertyTypes	Tipos de propiedades	Properties	
AssetTypes	Tipo de recursos digitales	Type	
SearchTypes	Tipos de búsqueda	Type	
Status	Estados para cada tipo de búsqueda	Type	

Cuadro 6.30: Propiedades de la clase Repository

En el caso de ejemplo, se mencionan varias instancias de la clase repositorio, por ejemplo el repositorio UOC (RepositoryUOC) que almacena todo tipo de recursos digitales de la organización y otros repositorios más concretos para guardar recursos digitales relativos a aulas (RepositoryAulas), matriculaciones (RepositoryMatrículas), evaluación (RepositoryEvaluación), recursos docentes (RepositoryRecursosDocentes), etc.

Tipos de relación

Los tipos de relación que se establecen entre las clases del OSID Repository y otras

clases externas son las que se presentan en la tabla 6.31 y se describen a continuación.

Tipo de relación	Descripción	Dominio	Rango	Restricción
Repository	Repositorio donde se almacena el recurso digital	Asset	Repository	Cardinalidad=1
Assets	Recursos digitales almacenados en un repositorio	Repository	Asset	
Parts	Recursos digitales componentes de un asset o parte de asset	Asset	Part	Transitividad
Records	Registros de un asset	Asset	Record	
Asset	Asset que corresponde a un registro	Record	Asset	
RecordStructure	Estructura de registro que determina la organización de un registro	Repository	RecordStructure	
RecordsByPartStructure	Registros que tienen una estructura de registro determinado	RecordStructure	Record	
PartStructure	Estructura de una parte de asset	Part	PartStructure	Cardinalidad=1
PartsByPartStructure	Partes que tienen una estructura de parte de asset determinada	PartStructure	Part	
PartStructures	Estructura de una parte de asset que contiene otras estructuras de partes de asset	PartStructures	PartStructures	Transitividad
RecordStructure	Estructura de registro que corresponde a un registro	Record	RecordStructure	Cardinalidad=1
RecordStructures	Estructuras de registros que reconoce un repositorio o definen un asset	Repository, Asset	RecordStructure	
MandatoryRecordStructures	Estructuras de registro obligatorias para los assets de un repositorio	Repository	RecordStructure	

Cuadro 6.31: 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), la cual a su vez puede estar formada por otras estructuras de partes (PartStructures), 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 ciertas estructuras de registro (RecordStructures), algunas de las cuales son obligatorias (MandatoryRecordStructures).

Existen tipos de relación derivados de otros que se definen mediante reglas de derivación como *Assets*(39), *Records*(40), *RecordByRecordStructure*(41), *PartsByPartStructure*(42), *RecordStructures* (43,44) y *MandatoryRecordStructures* (45).

Regla 39 (Rule-AssetsFromRepository):

Los assets asociados a un repositorio son la unión de todos los assets almacenados en él.

Regla 40 (Rule-RecordsFromAsset):

Los registros asociados a un asset son la unión de todos los registros que contienen dicho asset.

Regla 41 (Rule-RecordsByRecordStructureFromRecordStructure):

Los registros definidos de acuerdo con cierto tipo de estructura son la unión de todos los registros que tienen dicho tipo de estructura.

Regla 42 (Rule-PartsByPartStructureFromPartStructure):

Las partes de assets definidas de acuerdo con cierto tipo de estructura son la unión de todas las partes de assets que tienen dicho tipo de estructura.

Regla 43 (Rule-RecordStructuresOfAsset):

Las estructuras de registro de un asset son la unión de todas las estructuras de registro acorde a las cuales se definen los registros que componen dicho asset.

Regla 44 (Rule-RecordStructuresOfRepository):

Las estructuras de registro que reconoce un repositorio son todas aquellas estructuras acorde a las cuales se definen los assets almacenados en él o sus partes.

Regla 45 (Rule-MandatoryRecordStructures):

Las estructuras de registro obligatorias que reconoce un repositorio son todas aquellas estructuras obligatorias acorde a las cuales se definen los assets almacenados en él o sus partes.

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

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*).

Por otro lado, para comprobar la validez de una parte o un registro existen interfaces que permiten comprobar si el contenido está organizado de acuerdo con la correspondiente estructura (*validatePart*, *validateRecord*).

Limitaciones de OKI-OSID

Como limitación de esta especificación mencionar que no se considera ciertas formas de composición de *assets* utilizadas en otros estándares como por ejemplo la composición jerárquica o secuencial. Sería interesante poder extender el número de interfaces OKI teniendo en cuenta el estándar IEEE-LOM.

6.1.7. 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.

Puesto que el envío de mensajes es entre agentes, este OSID está directamente relacionado con el OSID *Agent*, cuya clase *Agent* debe de ser ampliada para posibilitar el envío/recepción por suscripción.

En la figura 6.7 se presenta el diagrama de clases de este OSID.

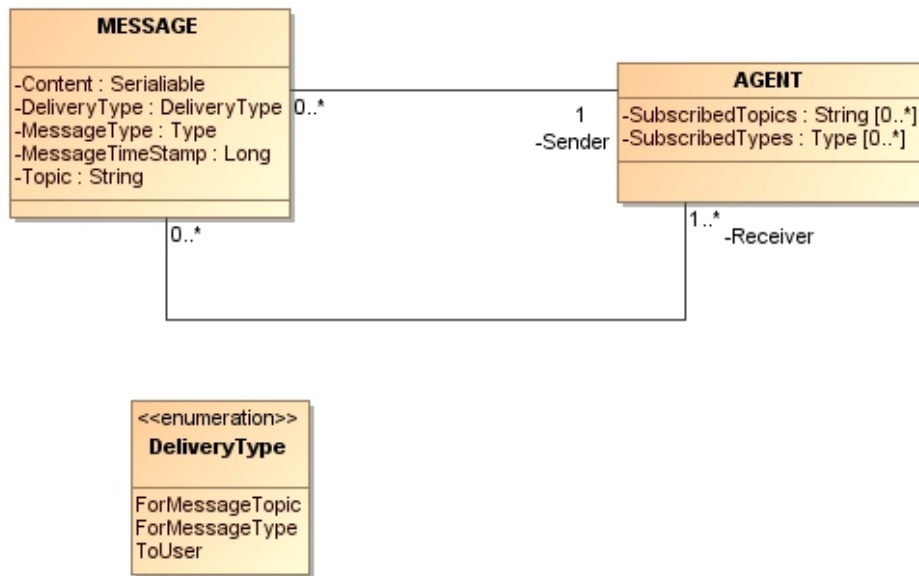


Figura 6.7: Diagrama de clases correspondiente al OSID UserMessaging

Se observa que un mensaje siempre es enviado por un único agente y puede ser recibido por varios agentes.

Clase Message

Un *Message* o mensaje es todo aquello que envía algún agente en un momento determinado. Las propiedades de esta clase se presentan en la tabla 6.32.

Propiedad	Descripción	Tipo de dato	Restricción
Content	Contenido del mensaje	Serializable	Cardinalidad=1
DeliveryType	Tipo de envío	Type	Cardinalidad=1
MessageTimeStamp	Instante del envío	Long	Cardinalidad=1
MessageType	Tipo de mensaje	Type	Cardinalidad=1

Cuadro 6.32: Propiedades de la clase Message

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

En el caso de ejemplo, los mensajes están representados por los flujos de información que se transmiten entre María y los consultores a través del sistema, Un ejemplo de mensaje sería aquel que María envía al grupo de consultores activos de la asignatura Bases de Datos I para asignar la tareas de consultoría, cuyo contenido serían las tareas asignadas. Como el grupo de consultores es reducido podría enviarse sin subscripción es decir directamente al grupo de consultores.

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) tal como se muestra en la tabla 6.33.

Tipo de relación	Descripción	Tipo de dato	Restricción
Sender	Agente que envía el mensaje	Agente	Cardinalidad=1
Receiver	Agentes a quién se dirige un mensaje	Agente	Ninguna

Cuadro 6.33: Tipos de relación del OSID User Messaging

Debe de tenerse en cuenta que para que sea posible la recepción de mensajes por tópico o por tipo de mensaje es necesario que exista una subscripción. Con esta finalidad, tal como se ha comentado anteriormente, la clase *Agent*, externa a este OSID, ha sido extendida con dos nuevas propiedades que permitan respectivamente la subscripción de un agente a diferentes tópicos (*SubscribedTopics*) y tipos de envío *SubscribedTypes*. Debe notarse que la subscripción es opcional por lo que estas propiedades no tienen ningún tipo de restricción de cardinalidad.

Métodos

Este OSID además de las propiedades y relaciones mencionadas ofrece interfaces de métodos para la gestión de subscripciones (*Subscribe/Unsubscribe*) de un agente o la eliminación de todas las subscripciones (*UnsubscribeAll*).

Como interfaces de métodos consultores ofrece la posibilidad de saber qué agentes son subscriptores (*getSubscribers*), qué agentes que se han suscrito a un determinado

tópico (getSubscribersByTopic) y qué mensajes han sido recibidos por tópico (getReceiveByTopic) o por tipo de mensaje (getReceiveForMessageType).

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

Limitaciones de OKI-OSID

No se ha detectado ninguna carencia en este OSID al margen de la necesidad de extender la clase Agent con nuevas propiedades para poder gestionar recepciones de mensajes por tópico y tipo de mensaje.

6.1.8. OSID Workflow

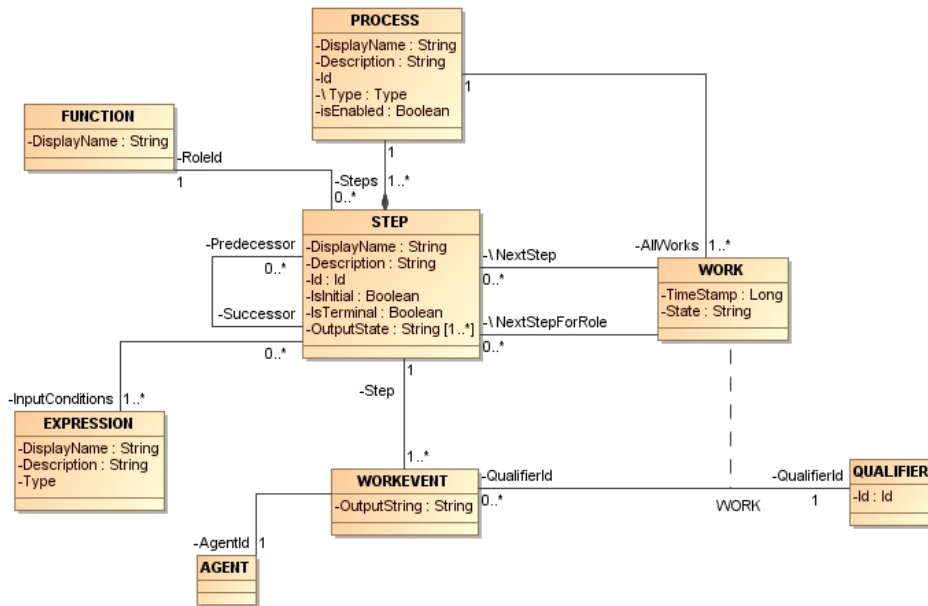


Figura 6.8: Diagrama de clases correspondiente al OSID Workflow

En la figura 6.8 se muestra el diagrama de clases UML correspondiente al 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.

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.

En la tabla 6.34 se muestran las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Nombre de proceso	String	Cardinalidad=1
Description	Descripción de un proceso	String	Cardinalidad=1
Id	Identificador de proceso	Id	Cardinalidad=1
Properties	Propiedades de un proceso a definir por la implementación	Properties	Ninguna
IsEnabled	Determina si el proceso está habilitado o no	Boolean	Cardinalidad=1
WorkState	Estado del proceso en unión de las tareas o trabajos	ProcessState	Cardinalidad=1

Cuadro 6.34: Propiedades de la clase Process

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

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 reconocida por el sistema y que se corresponde con una de las clases del OSID *Authorization*.

En la tabla 6.35 se describen las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Nombre de etapa	String	Cardinalidad=1
Description	Descripción de una etapa	String	Cardinalidad=1
Id	Identificador de etapa	Id	Cardinalidad=1
IsInitial	Determina si es la primera etapa de un proceso	Boolean	Cardinalidad=1
IsTerminal	Determina si es la última etapa de un proceso	Boolean	Cardinalidad=1
OutputStates	Descripción de los estados finales de una etapa	String	Cardinalidad Mín.1

Cuadro 6.35: Propiedades de la clase Step

Siguiendo con el caso de ejemplo y concretamente con el proceso *EstablecerAulas*, se identifican dos pasos o etapas: 1) *DeterminarAulasPorDefecto* que permite determinar el número de aulas por defecto que se asignan a una clase y es la etapa inicial cuya salida es una propuesta de número de aulas por tipo y 2) *AjustarAulasPorDefecto* que permite ajustar el número de aulas determinado previamente en la anterior etapa y produce como resultado la lista de aulas definitivas por tipo del curso y constituye la última etapa del proceso *EstablecerAulas*.

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.

Propiedad	Descripción	Tipo de dato	Restricción
DisplayName	Nombre de la condición inicial de una etapa	String	Cardinalidad=1
Description	Descripción de una condición de entrada de una etapa	String	Cardinalidad=1
Type	Tipo de expresión	Type	Cardinalidad=1

Cuadro 6.36: Propiedades de la clase Expression

Las propiedades de esta clase son las que se definen en la tabla 6.36.

Dentro del caso de ejemplo y concretamente considerando la etapa *DeterminarAulasPorDefecto* las condiciones iniciales para poderse llevar a cabo esta etapa son disponer del

identificador de asignatura (*IdAsignatura*) y del identificador de semestre (*IdSemestre*).

Clase Work

Un *Work* o tarea es la realización de un trabajo sobre un objeto determinado, que se corresponde con la clase *Qualifier* del OSID Authorization.

En la tabla 6.37 se muestran los dos únicas propiedades de esta clase: el instante de tiempo en que se llevará a cabo el trabajo y el estado en el que se encuentra.

Propiedad	Descripción	Tipo de dato	Restricción
TimeStamp	Instante en que se lleva a cabo el trabajo	Long	Cardinalidad=1
State	Estado en que se encuentra una tarea	String	Cardinalidad=1

Cuadro 6.37: Propiedades de la clase Work

Por ejemplo, suponiendo la etapa *DeterminarAulasPorDefecto* las tareas a llevar a cabo para finalizar esta etapa son: 1) la localización de los datos de la asignatura, 2) la localización de la lista de matrículas del curso, 3) la localización de los tipos de aulas correspondientes a la asignatura y 4) la localización de los ratios de estudiantes por aula.

Clase WorkEvent

Un *WorkEvent* o evento de trabajo es cada una de las tareas que lleva a cabo 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.

En la tabla 6.38 se definen las propiedades de esta clase.

Propiedad	Descripción	Tipo de dato	Restricción
OutputState	Estado de salida	String	Cardinalidad=1

Cuadro 6.38: Propiedades de la clase WorkEvent

Considerando los trabajos mencionados en el anterior apartado como ejemplos de trabajo, los eventos de trabajo correspondientes serían realizados por el sistema y se obtendría un estado de salida para cada uno de ellos. Por ejemplo, la localización de

los datos de la asignatura tendría como estado de salida: se dispone de los datos de la asignatura.

Tipos de relación

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

Tipo de relación	Descripción	Dominio	Rango	Restricción
Steps	Etapas que componen un proceso	Process	Step	Card. Mfn.1
Predecessor	Etapas anteriores	Step	Step	
Successor	Etapas posteriores	Step	Ninguna	
RoleId	Identificador de rol o función reconocida por el sistema	Step	Function	Cardinalidad=1
InputConditions	Condiciones de entrada de una etapa	Step	Expression	Card.Mfn.1
AllWorks	Tareas que se realizan en un proceso	Process	Work	Cardinalidad Mfn.1
NextSteps	Etapas que proceden a una tarea o trabajo	Work	Step	
NextStepForRole	Etapas que proceden a una tarea o trabajo que tienen un mismo rol	Work	Step	
AgentId	Agente que realiza un evento de trabajo	WorkEvent	Agent	Cardinalidad=1
QualifierId	Calificador sobre el que se realiza el trabajo	Work	Qualifier	Cardinalidad=1

Cuadro 6.39: Tipos de relación del OSID Workflow

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

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.

Limitaciones de OKI-OSID

Aunque este OSID tiene como finalidad representar flujos de trabajo, presenta algunas limitaciones que llevan a descartarlo como base de la ontología de procesos:

- 1) Las condiciones iniciales y los estados de salida expresados de forma textual no resultan útiles para el tratamiento automático, ni la alineación de precondiciones y postcondiciones.
- 2) No permite la reusabilidad de procesos, tareas ni etapas.
- 3) No propicia la modularidad, ya que no existe ningún tipo de relación entre procesos, lo cual impide de entrada la composición de los mismos.
- 4) Cada etapa tiene asociado un sólo rol, lo cual no permite la representación de procesos complejos en que un mismo participante asuma más de un rol.
- 5) Exige considerar el OSID Authorization, lo cual no interesa en este trabajo, ya pretende ser tratado a más alto nivel como se ha comentado anteriormente.

6.2. Construcción de OntoOKI en OWL

Al igual que para las ontologías anteriores (OntoProc, OntoED, OntoUOC) se han aplicado los mismos criterios para la transformación del diagrama de clases UML correspondiente a una ontología en OWL. Sin embargo ha sido necesarias algunas consideraciones a parte.

En la figura 6.9 se presenta la jerarquía de clases de OntoOKI con el fin de obtener una visión de conjunto de la misma.

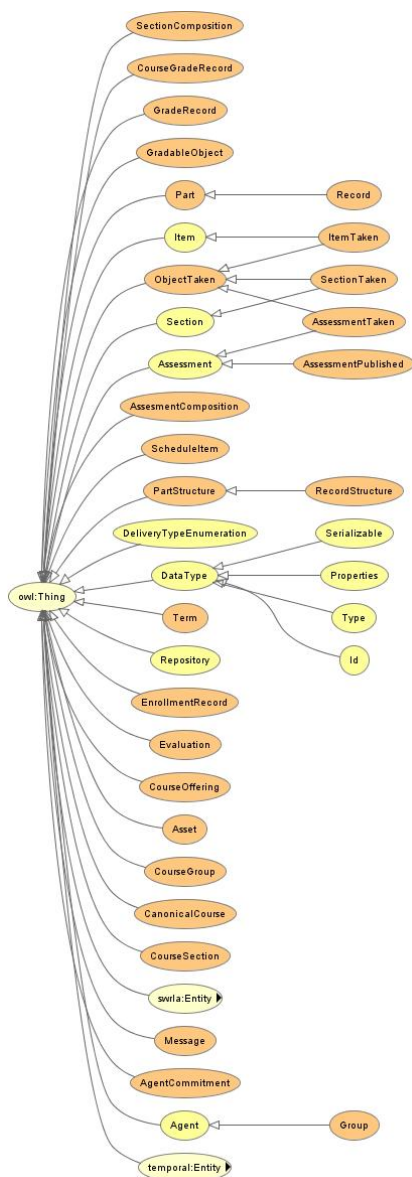


Figura 6.9: Jerarquía de clases y relaciones *is-a* de la ontología OntoOKI

En la figura 6.9 pueden observarse algunas clases que no han sido mencionadas en la formalización de OntoOKI.

Se ha introducido la clase *DataType* para generalizar los tipos de datos que se utilizan en la implementación de la especificación OKI-OSID y que se requieren para definir propiedades de tipo *ObjectType*. Esta clase se especializa en las siguientes clases: *Type*, *Id*, *Properties* y *Serializable*.

Además, ha sido necesario introducir otras dos nuevas clases: la clase *SectionComposition* y la clase *AssessmentComposition* para representar el ordenamiento de preguntas dentro de un ejercicio o de ejercicios dentro de actividades.

A continuación y mediante la figura 6.10 se explica como se ha resuelto el ordenamiento de ejercicios dentro de una actividad mediante la clase *AssessmentComposition*.

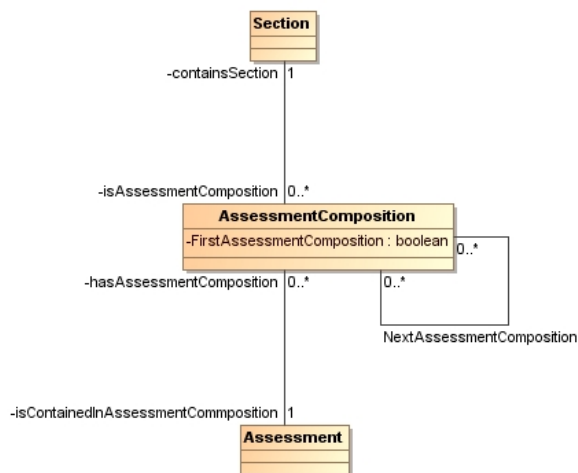


Figura 6.10: Composición de assessments

Cada componente de la actividad, es decir cada instancia de la clase *AssessmentComposition*, queda definida por la actividad que actúa de continente (*isContainedInAssessmentComposition*) y el ejercicio que está contenido en ella (*containsSection*) y además, tiene un valor que indica si es el primer componente de la actividad (*FirstAssessmentComposition*).

Los componentes que constituyen una actividad se relacionan con las clases *Assessment* y *Section*. Concretamente, la relación *isContainedInAssessmentComposition* permite hacer referencia a la actividad de la cual es parte y la relación *containsSection* a las partes contenidas en ella. De la misma manera los ejercicios que constituyen una actividad también se relacionan con los componentes de una actividad, la relación *containsSection* permite saber qué ejercicio es parte de una actividad y la relación *isAssessmentComposition* de qué actividades forma parte un ejercicio. Cada una de estas relaciones se traduce en una propiedad de tipo *ObjectType* en OWL.

Tal como puede apreciarse en la Figura 6.10, la ordenación de partes de una actividad queda establecida por dos propiedades:

- 1) el *dataType FirstAssessmentcomposition*, cuyo valor será cierto para las instancias de ejercicio que inician una actividad y falso para los demás ejercicios que componen la misma actividad y
- 2) el *ObjectType Next*, cuyo valor es otro componente de la actividad, concretamente, el que precede al actual

De forma análoga se ha procedido a representar el ordenamiento de preguntas dentro de alguna actividad utilizando la clase *SectionComposition*. Y en lo que respecta a actividades y ejercicios entregados, la composición y ordenación de sus componentes se obtendría por herencia. No obstante, no se ha desarrollado por estar fuera del contexto del caso de ejemplo.

Finalmente, se presenta el conjunto de reglas SWRL que extienden la ontología OntoOKI:

22) Rule-FirstSection

$$\text{isContainedInAssessmentComposition}(?ac, ?a) \wedge \text{containsSection}(?ac, ?s) \wedge \\ \text{firstAssessmentComposition}(?ac, \text{true}) \rightarrow \text{sections}(?a, ?s)$$

23) Rule-NextSection

$$\text{isContainedInAssessmentComposition}(?ac1, ?a) \wedge \text{containsSection}(?ac1, ?s1) \wedge \\ \text{isContainedInAssessmentComposition}(?ac2, ?a) \wedge \text{containsSection}(?ac2, ?s2) \wedge \\ \text{nextAssessmentComposition}(?ac1, ?ac2) \rightarrow \text{sections}(?a, ?s2)$$

24) Rule-FirstItem

$$\text{isContainedInSectionComposition}(?sc, ?s) \wedge \text{containsItem}(?ac, ?i) \wedge \\ \text{firstSectionComposition}(?sc, \text{true}) \rightarrow \text{items}(?s, ?i)$$

25) Rule-NextItem

$$\text{isContainedInSectionComposition}(?sc1, ?s) \wedge \text{containsItem}(?sc1, ?i1) ? \wedge$$

$$\text{isContainedInSectionComposition}(?sc2, ?s) \wedge \text{containsItem}(?sc2, ?i2) \wedge$$

$$\text{nextSectionComposition}(?sc1, ?sc2) \rightarrow \text{items}(?s, ?i2)$$
26) Rule-AssessmentPublishedOfEvaluatedAssessment

$$\text{evaluations}(?e, ?ot) \wedge \text{AssessmentTaken}(?ot) \wedge \text{assessmentPublished}(?ot, ?a)$$

$$\rightarrow \text{assessmentPublished}(?e, ?a)$$
27) Rule-AssessmentPublishedOfEvaluatedSection

$$\text{evaluations}(?e, ?ot) \wedge \text{SectionTaken}(?ot) \wedge \text{assessmentTaken}(?ot, ?at) \wedge$$

$$\text{assessmentPublished}(?at, ?a) \rightarrow \text{assessmentPublished}(?e, ?a)$$
28) Rule-AssessmentPublishedOfEvaluatedItem

$$\text{evaluations}(?e, ?ot) \wedge \text{ItemTaken}(?ot) \wedge \text{sectionTaken}(?ot, ?st) \wedge$$

$$\text{assessmentTaken}(?st, ?at) \wedge \text{assessmentPublished}(?at, ?a)$$

$$\rightarrow \text{assessmentPublished}(?e, ?a)$$
29) Rule-DescriptionOfCourseOffering

$$\text{description}(?ca, ?d) \wedge \text{courseOfferings}(?ca, ?co) \rightarrow \text{description}(?co, ?d)$$
30) Rule-DescriptionOfCourseSection

$$\text{description}(?ca, ?d) \wedge \text{courseOfferings}(?ca, ?co) \wedge \text{courseSections}(?co, ?cs)$$

$$\rightarrow \text{description}(?cs, ?d)$$
31) Rule-TitleOfCourseOffering

$$\text{title}(?ca, ?t) \wedge \text{courseOfferings}(?ca, ?co) \rightarrow \text{title}(?co, ?t)$$
32) Rule-TitleOfCourseSection

$$\text{title}(?ca, ?t) \wedge \text{courseOfferings}(?ca, ?co) \wedge \text{courseSections}(?co, ?cs)$$

$\rightarrow \text{title}(\text{?cs}, \text{?t})$

33) *Rule-NumberOfCourseOffering*

$\text{number}(\text{?a}, \text{?n}) \wedge \text{courseOfferings}(\text{?a}, \text{?co}) \rightarrow \text{number}(\text{?co}, \text{?n})$

34) *Rule-NumberOfCourseSection* $\wedge \text{courseSections}(\text{?co}, \text{?cs})$

$\rightarrow \text{number}(\text{?cs}, \text{?n})$

35) *Rule-RosterOfCourseSection*

$\text{EnrollmentRecord}(\text{?m}) \wedge \text{agentId}(\text{?m}, \text{?a}) \wedge \text{courseSectionId}(\text{?m}, \text{?cs})$

$\rightarrow \text{roster}(\text{?cs}, \text{?m})$

36) *Rule-RosterOfCourseOffering*

$\text{roster}(\text{?cs}, \text{?m}) \wedge \text{courseOffering}(\text{?cs}, \text{?co}) \rightarrow \text{roster}(\text{?co}, \text{?m})$

$\rightarrow \text{term}(\text{?cs}, \text{?t})$

37) *Rule-TermOfCourseSection*

$\text{term}(\text{?co}, \text{?t}) \wedge \text{courseSections}(\text{?co}, \text{?cs}) \rightarrow \text{term}(\text{?cs}, \text{?t})$

38) *Rule-AssetsOfCourseSection*

$\text{assets}(\text{?co}, \text{?a}) \wedge \text{courseSections}(\text{?co}, \text{?cs}) \rightarrow \text{assets}(\text{?cs}, \text{?a})$

39) *Rule-AssetsFromRepository*

$\text{repository}(\text{?a}, \text{?r}) \rightarrow \text{assets}(\text{?r}, \text{?a})$

40) *Rule-RecordsFromAsset*

$\text{asset}(\text{?r}, \text{?a}) \rightarrow \text{records}(\text{?a}, \text{?r})$

41) *Rule-RecordsByRecordStructureFromRecordStructure*

$\text{recordStructure}(\text{?r}, \text{?rt}) \rightarrow \text{recordsByRecordStructure}(\text{?rt}, \text{?r})$

42) *Rule-PartsByPartStructureFromPartStructure*

$\text{partStructure}(?p, ?pt) \rightarrow \text{partsByPartStructure}(?pt, ?p)$

43) *Rule-RecordStructuresOfAsset*

$\text{records}(?a, ?r) \wedge \text{recordStructure}(?r, ?rs) \rightarrow \text{recordStructures}(?a, ?rs)$

44) *Rule-RecordStructuresOfRepository*

$\text{Repository}(?rp) \wedge \text{assets}(?rp, ?a) \wedge \text{records}(?a, ?r) \wedge$
 $\text{recordStructure}(?r, ?rs) \rightarrow \text{recordStructures}(?rp, ?rs)$

45) *Rule-MandatoryRecordStructures*

$\text{Repository}(?rp) \wedge \text{assets}(?rp, ?a) \wedge \text{records}(?a, ?r) \wedge$
 $\text{recordStructure}(?r, ?rs) \wedge \text{isMandatory}(?rs, \text{true}) \rightarrow \text{recordStructures}(?rp, ?rs)$

6.3. Integración de OntoOKI en OntoProcEDUOC

La integración de la ontología de implementación (OntoOKI) en la ontología de procesos en entornos educativos de la UOC (OntoProcEDUOC) se basará principalmente en equivalencias entre clases, propiedades de clases y relaciones entre clases con el fin de obtener especificaciones formales de escenas educativas en términos de la especificación OKI-OSID y por tanto más orientadas a la automatización.

Puesto que la ontología de la UOC es una extensión de la ontología de escenarios educativos se opta por establecer, en primer lugar, las correspondencias con la ontología de procesos en entornos educativos (OntoProcED) de forma que éstas se propaguen a las ontologías de organización que la especializan. Luego, se procede a buscar las correspondencias de la ontología de la organización con la ontología OKI para aquellos conceptos y relaciones propios o específicos de cada organización y que no especializan conceptos ni relaciones en la ontología que extienden.

6.3.1. Correspondencia entre OntoProcED y OntoOKI

A continuación se analizan las equivalencias entre clases, propiedades y relaciones de OntoProcED con OntoOKI y se formalizan mediante reglas de transformación escritas en SWRL.

- **Clase Titulacion:** Se corresponde con la clase *CourseGroup*. La propiedad *nombre* se corresponde con la propiedad *displayName* y la propiedad *acrónimo* no tiene representación en la especificación OKI-OSID.
- **Clase Asignatura:** Tiene correspondencia con la clase *CanonicalCourse* de OntoOKI. En concreto las propiedades de clases se corresponden de la siguiente manera: *nombre* con *title*, *codigo* con *number*, *descripcion* con *description*, *topics* con *contenidos* y *tipo* con *type*. Las propiedades *metodologia* y *objetivos* no tienen correspondencia pero podrían gestionarse mediante la propiedad *properties*.
- **Clase AsignaturaTitulacion:** No tiene correspondencia en OntoOKI, por tanto se pierde información acerca de la obligatoriedad de las asignaturas en distintas titulaciones o lo que es lo mismo, de un curso canónico dentro de un grupo de cursos.
- **Clase CursoAcademico:** Tiene correspondencia con la clase *Term* y la propiedad *descripcion* con *description*.
- **Clase Curso:** Se corresponde con la clase *CourseOffering*. Las propiedades de clases *nombre* y *codigo* se corresponden con las propiedades *title* y *number* respectivamente. Los valores por defecto de estas propiedades se inicializan con el correspondiente valor de la propiedad que tiene el mismo nombre en la clase *CanonicalCourse*, aunque pueda ser redefinido.
- **Clase Evaluacion:** Se corresponde con la propiedad *CourseGradeType* asociada a la clase *CourseOffering*. La propiedad descripción de la clase *evaluación* no tiene equivalente en OntoOKI.

- **Clase *Planificacion***: No tiene correspondencia con ninguna clase, ni propiedad de OntoOKI. Sin embargo desde distinto nivel de abstracción podría establecerse una correspondencia con la clase ***Qualifier*** del OSID Authorization.
- **Clase *Evento***: Tiene correspondencia con la clase ***ScheduleItem***. La propiedad de clase *descripcion* se corresponde con la propiedad *description* y la propiedad *fechaClave* puede corresponderse con la propiedad *start*, considerando que los eventos suceden en una fecha, no un intervalo de fechas. La propiedad *tipo* sin embargo, no tiene correspondencia en la ontología OKI, pero podrían utilizarse para este fin las propiedades *displayName* o *properties*.
- **Clase *Material***: Se corresponde con la clase ***Asset***. Sus propiedades *descripcion*, *tipo*, *codigo* se corresponden con las propiedades *description*, *assetTye*, *id* de la clase *Asset* de OntoOKI.
- **Clase *Clase***: Tiene correspondencia con la clase ***CourseSection***. En concreto, la propiedad *Codigo* se corresponde con *Number* y *LugarImparticion* con *Location*, mientras que la propiedad *CapacidadMaxima* no tiene correspondencia por lo que podría usarse la propiedad *Properties*.
- **Clase *Actividad***: Se corresponde con parte de las clases ***AssessmentPublished*** y ***GradableObject***. Las propiedades *Codigo*, *Descripcion* y *Contenido* de la clase *Actividad* se corresponden con las propiedades *Id*, *Description* y *Data* de la clase ***AssessmentPublished*** y la propiedad *CriteriosEvaluacion* se corresponde con la propiedad *Description* de la clase ***GradableObject***, a la cual se accede mediante la relación *GradeObjectId*. La propiedad *Objetivo* no tiene correspondencia por lo que podría usarse la propiedad *Properties*
- **Clase *Persona***: Se corresponde con la clase ***Agent***. Las propiedades de clase *Dni*, *Nombre* y *TipoPersona* de esta clase se corresponden con las propiedades *Id*, *DisplayName* y *Type* de la clase *Agent*. El resto de propiedades, es decir, *Direccion*, *Telefono* podrían mapearse mediante la propiedad *Properties*. Concretamente, los

valores de la propiedad *Type* quedan determinados por las instancias de la clase enumerada *TipoPersona*.

- **Clase *Profesor***: Se corresponde con una instancia de la clase *Group*, aquella cuya descripción es 'Profesor'.
- **Clase *ProfesorCoordinador***: Se corresponde con una instancia de la clase *Group* cuya descripción es 'ProfesorCoordinador'.
- **Clase *GestorDocente***: Se corresponde con una instancia de la clase *Group* que tiene por descripción el valor 'GestorDocente'.
- **Clase *Estudiante***: Se corresponde con una instancia de la clase *Group* cuya descripción es 'Estudiante'.

A partir de aquí ya no existe una correspondencia directa con una clase, propiedad o instancia, sino que se trata de correspondencias entre distintos tipos de relación.

- **Clase *EntregaActividad*** : No tiene correspondencia con una clase concreta de OKI, sino que se corresponde con propiedades de varias clases: *AssessmentTaken* y por consiguiente con la clase *ObjectTaken*, *Evaluation* y *GradeRecord*. La propiedad fecha de entrega de la clase *EntregaActividad* se corresponde con la propiedad *date* de la clase *AssessmentTaken* y la propiedad *calificacionParcial* con la propiedad *gradeValue* de la clase *GradeRecord* relacionada, no obstante no existe en el metamodelo una forma de relacionar la clase *GradeRecord* con *AssessmentTaken* por lo que no existe una correspondencia para la calificación final de una actividad entregada.
- **Clase *AsignacionEstudianteCurso***: Tampoco tiene una correspondencia directa con una clase de OKI, si no que se corresponde con propiedades de varias clases como son: *EnrollmentRecord* y *CourseGradeRecord*. Las propiedades *NumMatricula* y *Estado* de la matrícula se corresponden directamente con las propiedades *Id* y *Status* de la clase *EnrollmentRecord*. La calificación final del

curso descrita por la propiedad *CalificacionFinal* se corresponde con la propiedad *CourseGrade* de la clase *CourseGradeRecord*.

- **Clase *AsignacionTareaDocente***: Se corresponde a la vez con parte de las clases *ScheduleItem* y *AgentCommitment*. Las propiedades *Tarea*, *FechaInicio* y *FechaFin* se corresponden con las propiedades *Description*, *Start*, *End* de la clase *ScheduleItem*. La propiedad *Estado* en cambio se corresponde con la propiedad *Status* de la clase *AgentCommitment*.
- **Clase *PlanDocente***: No tiene correspondencia con ninguna clase en particular y sus propiedades difícilmente pueden instanciarse, ello es debido a que el plan docente se describe a nivel de curso y en OKI los apartados que éste incluye se definen a nivel de clase y pueden por tanto no ser únicos. Sin embargo, podría obtenerse haciendo la unión, a pesar que podría haber contradicciones por ser apartados textuales mayoritariamente.

Por otro lado las clases enumeradas de OntoED *EstadoTarea*, *EstadoMatricu-*
la, *TipoAsignatura*, *TipoEvaluacion*, *TipoEvento*, *TipoMaterial*, *TipoTarea*,
TipoCalificación y *TipoPersona* tienen correspondencia con especializaciones de la
clase *Type* de OntoOKI definidas como subconjuntos de datos de tipo *String*.

En lo que respecta los tipos de relación entre OntoProcED y OntoOKI son necesarias las siguientes correspondencias:

- ***asignaturaEquivalente***: Se corresponde con *equivalentCourses* que relaciona cursos canónicos.
- ***incluyeAsigantura***: Tiene correspondencia con la propiedad *courses*.
- ***incluidaEnTitulacion***: Es la inversa de la anterior y no tiene correspondencia en la ontología OKI.
- ***cursoAcademico***: Tiene correspondencia con la relación *Term*.
- ***curso***: Es la inversa de la anterior y no tiene correspondencia en la ontología OKI.

- *esDeUnaAsignatura*: Se mapea mediante la relación *canonicalCourse*.
- *cursos*: Es la relación inversa de la anterior y tiene correspondencia con *CourseOfferings*.
- *esDeUnCurso*: Se corresponde con la propiedad *courseOffering*.
- *clases*: Es la inversa de la anterior y se corresponde con la relación *courseSections*.
- *evaluacionCurso*: Se traduce, tal como se ha comentado en el anterior apartado, mediante la propiedad de clase *CourseGradeType*.
- *cursoEvaluado*: Es la inversa de la anterior, pero no tiene correspondencia en OKI.
- *estudianteQueEntrega*: Se corresponde con la propiedad *AgentId* cuyo dominio es la clase *AssessmentTaken*.
- *actividadEntregada*: No tiene correspondencia en la ontología OKI.
- *cursoAsignado*: Se corresponde con la propiedad *CourseOffering* con dominio la clase *EnrollmentRecord*.
- *claseAsignadaAEstudiante*: No tiene correspondencia con la ontología OKI.
- *estudianteAsignado*: Es la inversa de las dos propiedades anteriores y se corresponde con la propiedad *Student*.
- *materialDelCurso*: Se corresponde con la propiedad *Assets*.
- *materialDeAsignatura*: Se corresponde con la propiedad *MaterialDelCurso* aplicada a todos los cursos de dicha asignatura.
- *asignaturaAsociadaAMaterial*: Es la inversa de la anterior y no tiene correspondencia.
- *tareasDocentesDelCurso*: No tiene correspondencia con la ontología OKI.

- *tareasDocentesDelProfesor*: Tampoco tiene correspondencia con la ontología OKI.
- *actividadReferida*: No tiene correspondencia en OntoOKI.
- *planificacionCurso*: No tiene correspondencia en la ontología OKI.
- *eventos*: Tampoco tiene correspondencia.
- *eventosActividad*: No tiene propiedad correspondiente en OntoOKI.
- *referidoA*: Es la inversa de la anterior y no tiene correspondencia en la ontología OKI.
- *actividadCurso*: Se corresponde con la propiedad *externalReferenceId* aplicada sobre todas las clases del curso en cuestión.
- *actividadDeLaEntrega*: Se corresponde con la propiedad *courseSectionId* cuyo dominio es la clase *AssessmentTaken*.
- *asignaturaCoordinada*: No tiene correspondencia en OntoOKI.
- *profesorCoordinador*: Es la inversa de la anterior y tampoco tiene correspondencia en la ontología OKI.
- *profesor*: No tiene correspondencia con ninguna propiedad de la Ontología OKI.
- *profesorDe*: Es la inversa de la anterior y tampoco tiene propiedad en la ontología OKI.
- *cursoCoordinado*: No tiene correspondencia en la ontología OKI.
- *profesorCoordinadorDeCurso*: Es la inversa de la anterior y no tiene propiedad equivalente en la ontología OKI.
- *claseAsignadaAProfesor*: No tiene propiedad equivalente en OntoOKI.
- *profesorAsignadoAclase*: Es la inversa de la anterior y tampoco tiene propiedad equivalente en la ontología OKI.

En el apéndice H se encuentra todas las reglas de transformación definidas y formalizadas en lenguaje SRWL.

6.3.2. Correspondencia entre OntoUOC y OntoOKI

La correspondencia entre la ontología de escenarios de una organización y la ontología OKI en este caso se concreta mediante en la correspondencia entre la ontología UOC y la ontología OKI.

A nivel de participantes existe una correspondencia entre las figuras UOC o participantes del contexto de la UOC, en tanto en cuanto son especializaciones de los tipos de participantes de entornos educativos. Sin embargo, no existen sus equivalentes dado que OKI es una especificación de tipo general y no contempla particularidades.

En lo que respecta a recursos la correspondencia se hace más difícil, ya que en esta ontología se recogen únicamente aquellos recursos específicos de la UOC que no se consideran genéricos de instituciones educativas en general y por tanto, tampoco representables de acuerdo con el metamodelo OKI. Por ejemplo, los permisos de acceso y actualización relativos partes de aulas y clases que posiblemente podrían requerir el OSID *Authorization* y que no se ha considerado en este marco por los motivos mencionados al inicio de este capítulo.

Así pues las correspondencias entre los tipos participantes UOC y los agentes de OntoOKI se establecen de la siguiente manera:

- ***MiembroEquipoDocente***: Instancia de la clase ***Group*** cuya descripción es 'PRA o Consultor'.
- ***MiembroAula***: Instancia de la clase ***Group*** cuya descripción es 'MiembroEquipoDocente o Estudiante'.
- ***MiembroCurso***: Instancia de la clase ***Group*** cuya descripción es 'MiembroAula'.
- ***MiembroUOC***: Instancia de la clase ***Group*** cuya descripción es 'MiembroCurso o PRA o Consultor o Estudiante o TecnicoSoporteDocente'.

En consecuencia, se observa que para obtener una correspondencia entre las especificaciones formales proporcionadas por el marco propuesto, a nivel de escenarios de organizaciones concretas es necesario se requiere una nueva ontología de implementación, bastante más extensa que la proporcionada por OKI o bien adaptar la ontología OKI para incluir nuevos conceptos. Esto sin embargo, debe considerarse trabajo futuro.

6.4. Resumen

En este capítulo se ha descrito el tercer nivel de la ontología multinivel propuesta en este trabajo de tesis, la ontología de implementación para el caso concreto de la especificación OKI-OSID (ontoOKI).

La creación de dicha ontología en OWL y su extensión mediante el lenguaje de reglas SWRL ha requerido de un estudio en profundidad de dicha especificación, lo que ha permitido detectar algunas de las limitaciones de la especificación y la creación de un metamodelo de gran parte de la misma.

Finalmente, se presenta la integración de esta ontología como extensión del marco propuesto con el fin de comprobar su utilidad en el siguiente capítulo.

Capítulo 7

Validación del marco formal propuesto

Computers do not solve problems, they execute solutions.

Lauren Gasser

En este capítulo se presenta la evaluación del marco formal propuesto en el presente trabajo para la representación de escenas educativas. La evaluación que se presenta abarca tanto la verificación formal del marco y la comprobación de la compleción del mismo como su adecuación para resolver al problema planteado.

Posteriormente a la evaluación se procederá, en el siguiente capítulo, a la evaluación de los objetivos planteados en el primer capítulo.

7.1. Ámbitos a considerar en la validación

La validación de cualquier modelo de referencia siempre debe de realizarse considerando varios criterios, por lo menos debe de tenerse en cuenta el dominio de aplicación para el cual ha sido desarrollado y su corrección sintáctica. En este trabajo se propone una evaluación del marco formal propuesto desde los siguientes ámbitos:

- 1) Correctitud del marco. Se realiza un análisis del marco formal creado para comprobar que está bien construido, es decir que no contiene restricciones contradictorias y por tanto es satisfactible.

- 2) Completitud del marco. Se comprueba que el marco permite la representación de conjuntos de escenas educativas complejas que tienen lugar en entornos educativos reales. Para llevar a cabo tal comprobación se utilizará el caso de ejemplo, seleccionado a propósito por ser suficientemente representativo.
- 3) Utilidad del marco. Se comprueba que utilizando este marco para la generación de descripciones formales de escenas educativas se pueden obtener implementaciones parciales de dichas escenas, útiles para la automatización de las mismas.

7.2. Correctitud del marco propuesto

La verificación de correctitud del marco propuesto para la especificación de escenas educativas implica realizar un análisis del mismo con el fin de verificar que a nivel formal está bien escrito y que no contiene contradicciones.

Puesto que el marco en cuestión es una ontología para la especificación de patrones de escenas y de su adaptación a organizaciones concretas está descrito en OWL y extendido con SWRL (Horrocks et al. 2004) es necesario realizar dos comprobaciones: 1) comprobar que la parte de la ontología definida en OWL es correcta y 2) comprobar que también lo es la parte de la ontología definida en SWRL.

En el plano estrictamente formal, no se puede considerar que un modelo de referencia está correctamente definido hasta que se comprueba que el conocimiento reflejado en él es consistente y no presenta contradicciones y por lo tanto se puede satisfacer. Para garantizar este punto se emplea el razonador de lógica de descripciones **Pellet v1.5.2** que viene integrado con la versión de Protégé utilizada, así como el editor de reglas **SWRLJessTab** que lo integra.

Pellet, es un razonador Java de código abierto para OWL-DL el cual permite: razonamiento sobre tipos, respuesta conjuntiva, soporte para reglas, análisis y reparación de ontologías así como depuración de ontologías y razonamiento incremental. Por su parte **SWRLJessTab** es el editor de reglas, es de código abierto y utiliza **Jess** como motor de inferencia.

Se ha procedido al análisis de validez de la especificación formal en OWL utilizando el razonador Pellet. Para ello se ha editado la ontología de escenas educativas (Onto-EDUOC), que incluye la de patrones de escenas y sus concreciones, y se ha invocado al razonador Pellet. El resultado obtenido ha permitido verificar la correctitud de la ontología en lo que respecta a su descripción en OWL.

Por otra parte, la inclusión del conjunto de reglas SWRL como parte de ella, ha sido verificada anteriormente puesto que SWRLJessTab no incorpora reglas SWRL a la ontología que no sean formalmente correctas determinado cierto dominio de conocimiento.

Por tanto, se concluye que el marco formal propuesto para la representación de escenas educativas es formalmente correcto.

7.3. Completitud del marco propuesto

Determinar si el marco propuesto es completo implica comprobar que permite especificar toda la información relevante para la representación de escenas educativas.

Comprobar la adecuación del marco creado para definir todos los posibles patrones de escenas educativas, así como todas las posibles escenas concretas que tienen lugar en un contexto organizativo específico, y que se definen a partir de patrones de escenas establecidos, es evidentemente imposible. Por tanto, se estudia la compleción del marco instanciando un caso real y complejo. Obviamente no podremos decir que el marco es completo en general, pero sí para el caso que nos ocupa.

En consecuencia, se procede a utilizar el caso de ejemplo referenciado a largo de este trabajo, el cual se describe en detalle en el apéndice D. Tal como se ha comentado en la introducción, se trata de la escena Preparación de un curso en la UOC (*PrepararCursoBDI-UOC*), la cual se presenta como una adaptación de la escena genérica *PrepararCurso* definida para entornos educativos en general. Se parte de la correspondiente descripción gráfica mediante la notación propuesta descrita en el apéndice C y a partir de ella se instancia el marco formal que se propone.

En concreto, se ha representado una parte significativa de la escena Preparar curso en la UOC utilizando el marco formal propuesto. La representación de esta parte de la

escena muestra tres niveles de descomposición para patrones de escenas y cinco niveles de descomposición para la representación de concreciones en el seno de una organización, la UOC en este caso. Exactamente, se ha llegado hasta quinto nivel para definir el primer patrón de escena de la secuencia de patrones que describe *PrepararCurso*, es decir se ha profundizado en la escena *EstablecerClases*.

La escena *EstablecerClases* se ha utilizado como patrón para definir la escena *EstablecerAulasBDI* en el contexto de la UOC y ha sido refinada mediante dos niveles más de descomposición para incorporar los procedimientos y la normativa de la UOC teniendo en cuenta los ratios por tipo de aula establecidos por la organización.

Una vez establecida la parte del caso de ejemplo utilizada para la instanciación del marco formal e instanciada la ontología convenientemente, se obtienen instancias de diecinueve patrones de escenas (el lector las puede encontrar en el apéndice E después de la descripción del caso de ejemplo presentado en el apéndice C) dos de las cuales son su vez patrones complejos y por tanto, se describen en términos de secuencias de escenas componentes.

Los datos relativos a la instanciación de la ontología de patrones de escenas se presentan en la tabla 7.1.

Clase	Número de instancias
ProcesoTipo	17
ProcesoTipoComplejo	2
ParticipanteTipo	3
MensajeTipo	61
ConectorTipo	10
RecursoTipo	43

Cuadro 7.1: Instancias para la representación de patrones de escenas

La instanciación de las propiedades de cada clase y de las relaciones establecidas entre ellas pueden consultarse directamente en la ontología, la cual puede descargarse de <http://personal.uoc.edu/personalonto/ontologies>.

A nivel de escenas concretas definidas para organizaciones particulares, el caso de ejemplo ha permitido la instanciación de veintiocho procesos concretos definidos para

la UOC, cuatro de ellos complejos. Además, ha permitido la instanciación de cada una de las clases de la jerarquía de primer nivel de la ontología de escenas adaptada a la organización UOC tal como se resume en la tabla 7.2.

Clase	Número de instancias
Proceso	24
ProcesoComplejo	4
Participante	3
Mensaje	79
Conector	23
Recurso	50

Cuadro 7.2: Instancias para la representación de escenas UOC

Mencionar que dos de las escenas concretas complejas se han definido de acuerdo a los correspondientes patrones de escenas definidos en el primer nivel del marco y las otras dos, se han creado a partir del refinamiento de escenas concretas adaptándolas al contexto de la UOC. Esto explica que el número de instancias de este nivel sea superior al número de instancias del nivel de patrones de escenas.

En concreto, se han instanciado tres tipos de participantes: un *PRA*, un *Consultor* (podría haberse instanciado más de uno pero ello complicaría los diagramas y la instanciación de la ontología y no aportaría nada nuevo) y un *TécnicoApoyoDocente*. Y a nivel de recursos se han empleado, como recursos específicos de la UOC que refinan la taxonomía de recursos de la ontología de escenas educativas, los siguientes tipos de recursos: *AsignacionEncargoDocente*, *Asignatura*, *Curso_uoc* y *PlanDocente*, a parte de los demás cuarenta y tres recursos definidos en el nivel más alto de la jerarquía de recursos.

Así pues, se ha observado que la instanciación de participantes y recursos tiene lugar de forma diferente. Los participantes de las escenas UOC siempre son figuras de la organización definidas en los niveles más bajos de la jerarquía de participantes, mientras que los recursos, la mayoría de veces se han representado como recursos en el nivel superior de la jerarquía de recursos. Ello se debe a que no se ha tratado la composición de recursos, lo cual podría ser parte de otro trabajo de investigación que continuara el presente.

En lo que respecta a propiedades derivadas mediante reglas SWRL se muestra un ejemplo en la tabla 7.3. Supóngase la asignatura Bases de Datos de la titulación ITIG (Ingeniería Técnica en Informática de Gestión) de la UOC, coordinada por el PRA_BDI, y el curso CursoBDI_ITIG_Primavera2010 asociado a dicha asignatura. Partiendo de esta información y utilizando Jess como motor de inferencia, la regla 9 (Rule-CursoCoordinado) permite inferir que el profesor coordinador del curso CursoBDI_ITIG_Primavera2010 es PRA_BDI, una instancia de la clase *PRA* que especializa la clase *ProfesorCoordinador*. Por tanto, se comprueba que mediante la ejecución de la regla correspondiente se ha poblado la propiedad derivada *cursoCoordinado* de la clase *Profesor* y en consecuencia de la clase **PRA**.

Regla	$\text{asignaturaCoordinada}(?p, ?a) \wedge \text{cursos}(?a, ?c) \rightarrow \text{cursoCoordinado}(?p, ?c)$
Axiomas	$\text{cursos}(\text{AsignaturaBDI}, \text{CursoBDI_ITIG_Primavera2010})$ $\text{asignaturaCoordinada}(\text{PRA_BDI}, \text{AsignaturaBDI})$
Axiomas inferidos	$\text{cursoCoordinado}(\text{http://www.owl-ontologies.com/Ontology1254295609.owl\#PRA_BDI}, \text{CursoBDI_ITIG_Primavera2010})$

Cuadro 7.3: Ejemplo de aplicación de una regla

La instanciación de cada una de las ontologías y su integración ha originado la ontología multinivel OntoEDUOC. La ontología resultante ha sido probada al representar una parte representativa de las escenas educativas que conforman la escena *PrepararCursoBDI-UOC*, tal como se ha comentado anteriormente, primeramente a nivel de patrones de escenas y después de escenas concretas. La relación entre ambos niveles se ha llevado a cabo mediante las propiedades que definen para cada concreción el tipo correspondiente en el nivel de patrones (*tipoDe*).

En consecuencia, se concluye que ha sido posible representar escenas educativas y patrones de escenas, pudiendo definir las segundas a partir de las primeras y adaptándolas al contexto de una organización concreta. Por tanto, toda información relativa al caso de ejemplo ha podido ser representada utilizando el marco formal propuesto.

7.4. Utilidad del marco propuesto

La utilidad del marco formal propuesto se ha evaluado comprobando hasta que punto las descripciones formales obtenidas a partir de la ontología multinivel pueden ser útiles para la representación de escenas educativas y la generación de implementaciones parciales de las mismas.

La descripción de este marco ha servido para definir un DSL que permite la representación gráfica de escenas educativas y sus patrones, así como la instanciación automática de una parte de la ontología que conforma dicho marco. En el anexo E se puede encontrar información adicional sobre esta herramienta CASE, la cual es actualmente un prototipo que facilita la representación gráfica de escenas educativas y la instanciación automática de parte del marco propuesto, concretamente la que se refiere a la ontología de procesos. Sin embargo, constituye una prueba fehaciente de que es posible crear una herramienta para la instanciación automática del marco a partir de las representaciones gráficas de escenas educativas.

En lo que se refiere a generación de implementaciones parciales de escenas educativas, el marco propuesto también puede servir para generar implementaciones parciales de las especificaciones de escenas educativas. Para ello se ha utilizado la ontología OKI y el conjunto de reglas que la extienden con el fin de facilitar la población automática de dicha ontología de implementación a partir de las descripciones formales obtenidas mediante el marco propuesto.

Concretamente, en el caso de ejemplo se han utilizado las siguientes reglas de transformación: las reglas 48 a 51 y 53 para traducir la asignatura Bases de Datos I a *CanonicalCourse*, las reglas 55 a 57 para traducir el curso de Bases de Datos I del semestre Primavera 2010 a *CourseOffering* y la regla 72 para traducir cada persona a *Agent*. Estas reglas puede encontrarlas el lector en el anexo H y en la ontología.

Debe notarse, que el grado de población de la ontología OKI no es muy elevada en comparación con la ontología multinivel. Ello se debe en parte, a que los recursos utilizados en la representación de escenas educativas han sido definidos mayoritariamente a alto nivel, sin tener en cuenta su composición y por tanto, no se dispone de detalles que

puedan hacerse corresponder con propiedades de las clases que configuran los OSID. Por otro lado, el problema es que la ontología OKI describe el metamodelo de la especificación OKI-OSID que es genérico. Esta generalidad implica que no contemple particularidades de organizaciones concretas, por lo que para obtener implementaciones más completas de las especificaciones del marco debería de adaptarse OKI o disponer de una ontología de implementación más adecuada.

Así pues, se concluye que el marco formal propuesto permite la instanciación de parte de la ontología OKI y su integración como tercer nivel del mismo, lo que posibilita la generación de implementaciones parciales de forma automática a partir de las especificaciones de escenas educativas representadas por él, hecho que prueba también la utilidad de este *framework*. Con una ontología de implementación más amplia y una representación de los recursos en términos de sus componentes se conseguiría una implementación más completa.

7.5. Resumen

En este capítulo se presenta la validación del marco formal propuesto para la representación de escenas educativas. La validación se ha llevado a cabo desde tres perspectivas esenciales: corrección formal, prueba de completión y evaluación de su utilidad. La valoración positiva de ese análisis, es el punto de partida para examinar la consecución de los objetivos planteados en el capítulo 1, que se abordará en el próximo capítulo.

Capítulo 8

Conclusiones y trabajo futuro

La vida es linda, lo malo es que muchos confunden lindo con fácil.

Mafalda

Una vez presentado el marco formal para la representación de escenas educativas reutilizables y su evaluación, sólo resta extraer y analizar las conclusiones que sintetizan el trabajo realizado.

Se presenta, en primer lugar, un resumen de los objetivos y se analiza el grado de satisfacción de los mismos. Después, se describen las aportaciones originales que han surgido del proceso de investigación y las conclusiones obtenidas.

Finalmente, se presentan las líneas de investigación que quedan abiertas en el campo de investigación, así como otras posibilidades de continuación del esfuerzo realizado.

8.1. Satisfacción de los objetivos propuestos

En esta sección se describe cómo se han alcanzado tanto el objetivo general enunciado en el capítulo 1, como los objetivos específicos, indicando en ambos casos el proceso de evaluación realizado. Este proceso de evaluación permitirá indicar el grado de cumplimiento de los mismos.

8.1.1. Resumen del objetivo general

La finalidad fundamental de esta tesis ha sido realizar una aportación significativa al área de la especificación de procesos en tecnología educativa como un paso más hacia la automatización de procesos en entornos educativos.

Más concretamente, es una aportación al área de la especificación de escenas educativas reutilizables, definiendo *un marco formal para la representación de escenas educativas reutilizables, orientado a facilitar la automatización o semi-automatización de los procesos que tienen lugar en ellas, adaptando patrones de escenas educativas a escenas concretas en el contexto de una organización y generando implementaciones parciales de las mismas.*

Para cumplir los objetivos planteados, en primer lugar se ha propuesto la notación gráfica para la representación de escenas educativas descrita en el anexo C. Esta notación permite representar gráficamente escenas educativas en términos de secuencias de procesos reutilizables y cada proceso mediante interacciones de los participantes y los recursos utilizados. Tal secuenciación describe cómo debe de ser una futura implementación de la escena con el fin de alcanzar un objetivo.

En segundo lugar, se ha propuesto un marco formal constituido conceptualmente por dos niveles: el primero con la finalidad de representar patrones de escenas educativas y el segundo para representar escenas concretas adaptadas a un contexto organizativo específico. Posteriormente, se han integrado ambos niveles para dar lugar a un mecanismo de representación de escenas educativas, tanto a nivel genérico como particular para organizaciones concretas.

Finalmente, se ha construido un metamodelo OKI-OSID de parte de la especificación OKI-OSID para poder probar que las descripciones formales obtenidas a partir del marco propuesto (OntoEDUOC) pueden ser utilizadas para generar automáticamente implementaciones parciales de escenas educativas.

Todo lo que se ha expuesto resalta las características novedosas que el presente trabajo supone con respecto al estado de la cuestión, y postula el marco formal propuesto como un modelo de referencia extensible para la representación de escenas educativas en el contexto de las instituciones de enseñanza superior tal como se indica más adelante.

8.1.2. Satisfacción de los objetivos específicos

Satisfacer el objetivo principal de este trabajo de tesis supone conseguir un marco formal que permita representar patrones de escenas educativas, adaptar dichos patrones a escenas educativas concretas en el contexto de una organización y generar una implementación parcial de las mismas. Para alcanzar este objetivo ha sido necesario satisfacer otros objetivos más específicos. A continuación se detalla, para cada objetivo específico el grado de consecución del mismo:

- **Objetivo 1:** Proponer un lenguaje que facilite la representación gráfica de escenas educativas.

La consecución de este objetivo se ha alcanzado al definir una notación gráfica para la especificación de escenas educativas que se ha descrito en el anexo C, la cual ha sido utilizada para la representación gráfica de patrones de escenas y escenas concretas del caso de ejemplo utilizado en este trabajo de investigación. La validez de esta notación ha sido demostrada en tanto en cuanto los diagramas obtenidos son fácilmente comprensibles por los humanos y adecuados para la instanciación de la ontología del marco propuesto.

- **Objetivo 2:** Obtener especificaciones formales de patrones de escenas en términos de participantes involucrados, recursos utilizados y la secuencias de procesos reutilizables para componer procesos más complejos.

Este objetivo ha sido alcanzado mediante la creación e instanciación de la ontología de procesos (OntoProc). La creación de la ontología de procesos ha permitido la descripción formal de escenas en términos de los procesos que describen las actividades que tienen lugar en ellas, los participantes que interactúan con dichos procesos y los recursos que se utilizan en ellos. La instanciación de la misma por su parte ha permitido la descripción formal de patrones de escenas educativas reutilizables, tal como queda demostrado más adelante al definir escenas educativas para organizaciones concretas a partir de ellos.

- **Objetivo 3:** Obtener especificaciones formales de escenarios educativos que permitan contextualizar los procesos en entornos educativos.

Análogamente, este objetivo se ha alcanzado con la creación e instanciación de la ontología de escenarios educativos (OntoED). Esta ontología permite describir los recursos y participantes que caracterizan un entorno educativo como especializaciones de recursos y participantes tipo, así como las restricciones relativas a dicho entorno definidas mediante reglas y enumeraciones que restringen el rango de valores de las relaciones entre ellos. La instanciación de la misma para el caso de ejemplo ha permitido probar que cumple su cometido.

- **Objetivo 4:** Obtener una integración de las especificaciones descritas en los objetivos 2 y 3 para representar patrones de escenas educativas.

La creación de la ontología de primer nivel del marco propuesto, la ontología de patrones de escenas educativas o de procesos en entornos educativos (OntoProcED) y su instanciación a partir de la integración de las dos ontologías anteriores ha satisfecho la consecución del cuarto objetivo. Se ha conseguido mediante el establecimiento de un conjunto de relaciones entre ambas ontologías de manera que se permita la incorporación de taxonomías de participantes y recursos del entorno educativo en los patrones de escenas, así como las relaciones entre participantes y recursos tipo con participantes y recursos concretos. La representación de patrones de escenas del caso de ejemplo mediante ontologías así lo demuestra.

- **Objetivo 5:** Obtener especificaciones formales de procesos para organizaciones concretas.

Este objetivo se ha alcanzado mediante la creación e instanciación de la ontología de procesos concretos (OntoOrg). Esta ontología describe secuencias de procesos válidos para cualquier organización educativa. Así pues, su instanciación ha facilitado la obtención de descripciones formales procesos concretos para organizaciones específicas, definidas como secuencias de procesos reutilizables dentro de organizaciones concretas. El caso de ejemplo ha permitido probar que es posible la

definición de escenas concretas en la UOC, como ejemplo de organización, a partir de la composición de sus escenas componentes en dicho contexto organizativo y por tanto, la especificación de los procesos que se llevan a cabo en ella incorporando procedimientos específicos adaptados a su normativa, funcionamiento e idiosincrasia.

- **Objetivo 6:** Obtener especificaciones formales de participantes y recursos de una organización concreta.

Para alcanzar este objetivo se ha creado la ontología de escenarios educativos para la UOC (OntoUOC) de manera que pueda representarse formalmente el contexto de esta organización en lo que se refiere a participantes específicos y a recursos particulares de la misma, así como los aspectos normativos a tener en cuenta en los procedimientos que se llevan a cabo en dicha institución. Mediante la instanciación de esta ontología para el caso de ejemplo también se ha probado que este objetivo ha sido alcanzado.

- **Objetivo 7:** Obtener especificaciones formales de escenas educativas adaptadas a alguna organización.

La integración de las ontologías que forman parte del segundo nivel de la arquitectura ontológica propuesta, es decir la ontología de procesos concretos (OntoProc) y la de escenarios educativos UOC (OntoUOC) permiten la definición de escenas concretas para la organización UOC como ejemplo de organización específica.

Este objetivo se ha alcanzado creando la ontología de escenas educativas en el contexto de una organización (OntoEDUOC). La consecución de este objetivo la demuestra el hecho de posibilitar la representación de escenas educativas adaptadas al contexto y normativa de la organización, en este caso la UOC, como una concreción de patrones de escenas educativas.

- **Objetivo 8:** Obtener un metamodelo de una parte de la especificación OKI-OSID para automatizar o semi-automatizar las especificaciones formales de escenas educativas.

La formalización de una parte de la especificación OKI-OSID mediante diagramas de clases UML ha permitido la definición de dicho metamodelo. Este metamodelo se ha implementado en OWL para crear la ontología OKI.

Este objetivo se ha conseguido mediante la especificación del metamodelo OKI-OSID, la definición de un conjunto de reglas que establecen la correspondencia entre las descripciones formales de escenas educativas y su implementación mediante elementos de OKI-OSID. Prueba de que se ha alcanzado este objetivo es la población automática de parte de la ontología OKI a partir de la ejecución de este conjunto de reglas definidas en la ontología OKI para el caso de ejemplo.

8.2. Aportaciones originales

La consecución de estos objetivos ha permitido realizar una serie de aportaciones, siendo la aportación fundamental de este trabajo de investigación un **marco formal para la representación de patrones de escenas educativas, la adaptación de dichos patrones a escenas educativas concretas en el contexto de una organización y la generación de una implementación parcial de las mismas.**

Aplicando la técnica divide-y-vencerás para la consecución del principal objetivo se han alcanzado otros objetivos más específicos que han dado lugar a otras aportaciones más concretas:

- **Aportación 1:** Una notación gráfica que permita la representación de escenas educativas tanto a nivel genérico como concreto. Ha sido descrita en el anexo C y utilizada para la representación de diagramas de escenas a lo largo de todo este trabajo.
- **Aportación 2:** Una ontología de procesos que permite formalizar especificaciones de procesos en términos de participantes involucrados, recursos utilizados y secuencias de procesos reutilizables para componer procesos más complejos: OntoProc. Los detalles sobre la misma se han presentado en la primera parte del capítulo 4.

- **Aportación 3:** Una ontología de escenarios educativos que facilite la especificación formal de escenarios educativos mediante la descripción de los tipos de participantes y recursos propios de un entorno educativo: OntoED. Los detalles de esta ontología han sido descritos en la segunda parte del capítulo 4.
- **Aportación 4:** Un marco formal para especificar patrones de especificaciones de escenas educativas a partir de la integración de las ontologías anteriores: Onto-ProcED. Se ha presentado como ontología resultado de la integración de las dos anteriores y ha sido descrita en la tercera parte del capítulo 4 del presente trabajo.
- **Aportación 5:** Una ontología de definición de procesos de acuerdo a las necesidades funcionales de las distintas instituciones educativas: OntoOrg. Los detalles sobre la misma se han presentado en la primera parte del capítulo 5.
- **Aportación 6:** Una ontología de una organización concreta, por ejemplo la UOC: OntoUOC. Los detalles de esta ontología han sido descritos en la segunda parte del capítulo 5.
- **Aportación 7:** Una marco formal para especificar escenas educativas adaptadas a las necesidades de una organización concreta. Se ha presentado como ontología resultado de la integración de todas las anteriores y ha sido descrita en la tercera parte del capítulo 5 del presente trabajo.
- **Aportación 8:** Una ontología de implementación basada en la especificación OKI-OSID extendida con un conjunto de reglas que permiten la población automática de la ontología de implementación: OntoOKI. La ontología OKI y las reglas comentadas se encuentran en el capítulo 6.
- **Aportación 9:** Una herramienta CASE que permita la representación gráfica de patrones de escenas educativas utilizando la notación referida en la aportación 1. En el anexo E se presenta brevemente el prototipo creado de esta herramienta.

8.3. Conclusiones

El marco formal que se propone en este trabajo para la representación de escenas educativas es una arquitectura ontológica formada, a nivel conceptual, por dos niveles: 1) un nivel genérico o de especificación de patrones de escenas y 2) un nivel concreto o de especificación de escenas, que permite concretar patrones de escenas para una organización en particular.

El marco propuesto facilita: 1) la descripción formal de escenas de acuerdo con algún patrón de escena educativa establecido, lo que permite la reutilización de patrones de escenas y escenas entre organizaciones y 2) la implementación parcial de escenas educativas para ofrecer soporte a la automatización de las mismas.

Con el fin de facilitar la reutilización y extensión del marco propuesto uno de los criterios utilizados para el diseño del mismo ha sido la modularidad. Por este motivo el marco en cuestión está formado por la integración de varias ontologías, cada una de ellas con una misión concreta y la mayoría de ellas reutilizables.

Por un lado, la ontología de procesos puede ser utilizada para describir secuencias de procesos reutilizables, con independencia del entorno donde se lleven a cabo las escenas. Por otro, la ontología de procesos concretos es válida para la descripción de secuencias de procesos de cualquier escena y organización. Y además, el primer nivel conceptual del marco puede ser extendido por un segundo nivel, que varía en función de la organización que se considere, permitiendo así la concreción de patrones de escenas a escenas particulares que tengan en cuenta las normas y procedimientos de cada institución educativa. El proceso seguido para la integración de la ontología UOC en el *framework* propuesto sería análogo para cualquier otra organización educativa, bastaría con establecer las correspondencias entre la ontología que describa el escenario educativo de la nueva organización y la ontología de primer nivel.

También se ha creado un DSL para ofrecer soporte a la representación gráfica de escenas educativas y a la instanciación de la ontología. Esta herramienta utiliza como lenguaje una notación gráfica creada con el fin de permitir definir los patrones de escenas semánticamente de forma concisa y clara, además de poder ser utilizada para compartir

información sobre el funcionamiento de una organización educativa. Aunque el prototipo creado instancia una parte del marco, concretamente la ontología de procesos, puede ser ampliado para ofrecer soporte en la instanciación de todas las ontologías que constituyen el marco.

A partir de la estructura ontológica instanciada con representaciones de escenas educativas se pueden extraer especificaciones formales que pueden ser traducidas en términos de código ejecutable. Así pues, el conocimiento extraído a partir del marco formal propuesto puede servir para la automatización de los procesos que se llevan a cabo en entornos educativos y por tanto, contribuir a la automatización de escenas educativas.

Otro de los posibles usos de este marco es la validación del comportamiento del sistema de aprendizaje previa implementación de los procesos que se llevan a cabo en las escenas educativas que se desea automatizar. El análisis realizado como consecuencia de la especificación de dichas escenas permite la optimización e innovación de los procesos que se desarrollan en entornos educativos, así como la agilización de procedimientos que habitualmente ocurren en las instituciones educativas, facilitando el desarrollo y mantenimiento de las funcionalidades del sistema de aprendizaje y reduciendo costes de desarrollo por eliminación de errores.

Finalmente, constatar que la validación del marco formal propuesto ha sido realizada en términos de corrección formal, completitud y utilidad, lo que garantiza su satisfactibilidad y efectividad y, en consecuencia, satisface los objetivos establecidos en el capítulo primero.

8.4. Líneas de trabajo futuras

Tanto a lo largo del desarrollo de este trabajo de tesis, como a la finalización del mismo, se han observado posibles líneas de trabajo futuras que incluyen, por un lado, extensiones del marco presentado y, por otro lado posibles usos del mismo.

A continuación, se describen brevemente las líneas de trabajo futuras más relevantes que podrían continuar este trabajo de tesis.

8.4.1. Extensiones del marco propuesto

El marco propuesto presentado puede extenderse de la siguiente manera:

- Integrando como ontologías de segundo nivel ontologías que describan la semántica de las especificaciones y estándares existentes en el área de la tecnología educativa. Por ejemplo para describir recursos mediante el estándar IEEE-LOM o actividades de evaluación de acuerdo con IMS-QTI, por citar algunos. La utilización de estas especificaciones y estándares en la descripción formal de escenas educativas ayudaría a promover la interoperabilidad de sistemas de aprendizaje y la portabilidad de las especificaciones.
- Creando un tercer nivel en la arquitectura ontológica que permita definir las descripciones formales de escenas educativas en términos de su implementación con independencia del marco de referencia o *framework*. En este trabajo de tesis se ha propuesto, como ontología de implementación particular, la ontología OKI pero podría haber sido otra. Por lo cual se propone la creación de un nuevo nivel dirigido a la especificación de implementaciones en general, de la misma manera que se ha creado el primer nivel para la descripción de patrones de escenas educativas.
- Definiendo una jerarquía de recursos que describa la composición de recursos utilizados en las especificaciones de distintos patrones y escenas descritos con el marco propuesto según el nivel de detalle que les corresponda. Esta extensión permitiría incorporar nuevas restricciones de integridad al modelo ontológico para comprobar la consistencia en el uso de los recursos empleados en los distintos niveles de definición de las escenas y patrones de escenas educativas. Esta jerarquía permitiría definir mejor los recursos y por tanto generar implementaciones más completas.

8.4.2. Posibles usos del marco presentado

Otras formas potencialmente interesantes de utilizar el marco propuesto en el futuro son las siguientes:

- Ampliar el prototipo de herramienta CASE desarrollado para ofrecer soporte en la instanciación de todas las ontologías de la arquitectura ontológica propuesta, no sólo a nivel de especificación de patrones de procesos, sino para procesos concretos y su contextualización.
- Ampliar las funcionalidades de la herramienta CASE permitiendo la extracción del conocimiento almacenado en la ontología multinivel para obtener una implementación de las escenas descritas.
- Utilizar el marco desarrollado para una aproximación *model-driven* que permita obtener implementaciones de procesos que forman las escenas a partir de la definición formal de éstas que proporciona el marco propuesto.
- Construir un catálogo de procesos para entornos educativos que facilite la reutilización de los mismos para la composición de nuevas escenas educativas y que a la vez facilite la comparación de los procesos que se llevan a cabo en distintas instituciones educativas. La creación de dicho catálogo y las escenas que puedan describirse a partir de éstos, permitiría verificar el cumplimiento de normas y criterios empleados en los distintos procesos, de manera que pueda garantizarse la calidad de los procesos y facilitarse su comparación. Disponer de este catálogo también puede, a la larga, ser útil para la estandarización de procesos en entornos de aprendizaje.

Apéndice A

Ontologías del marco formal propuesto

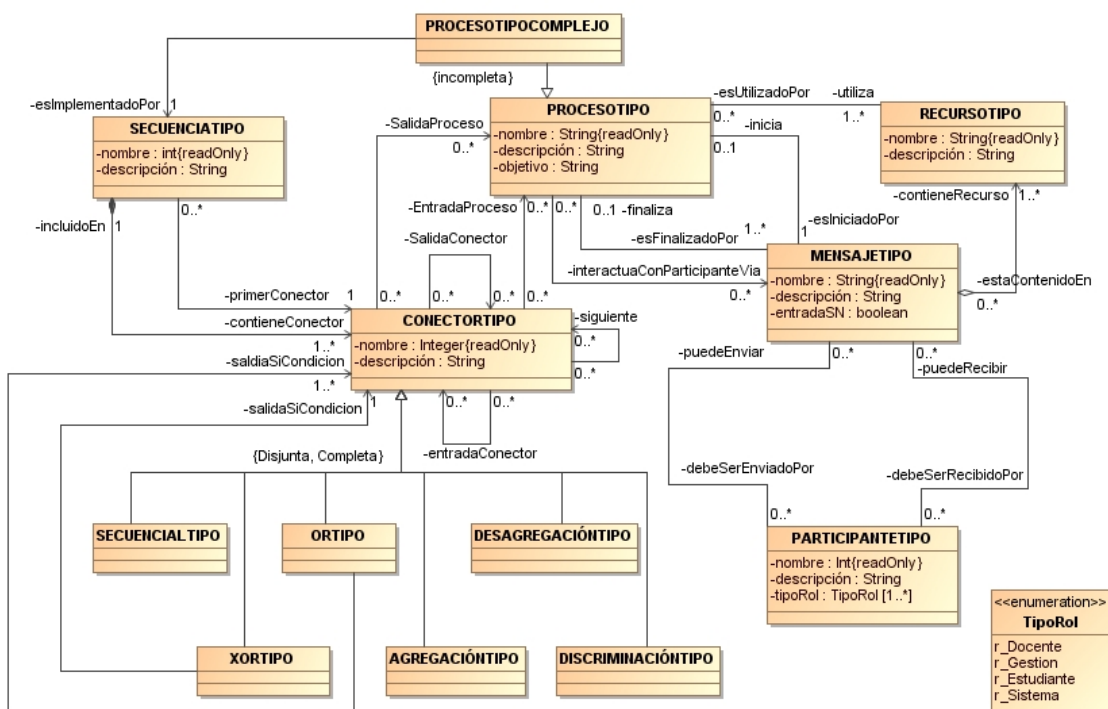


Figura A.1: Ontología de procesos

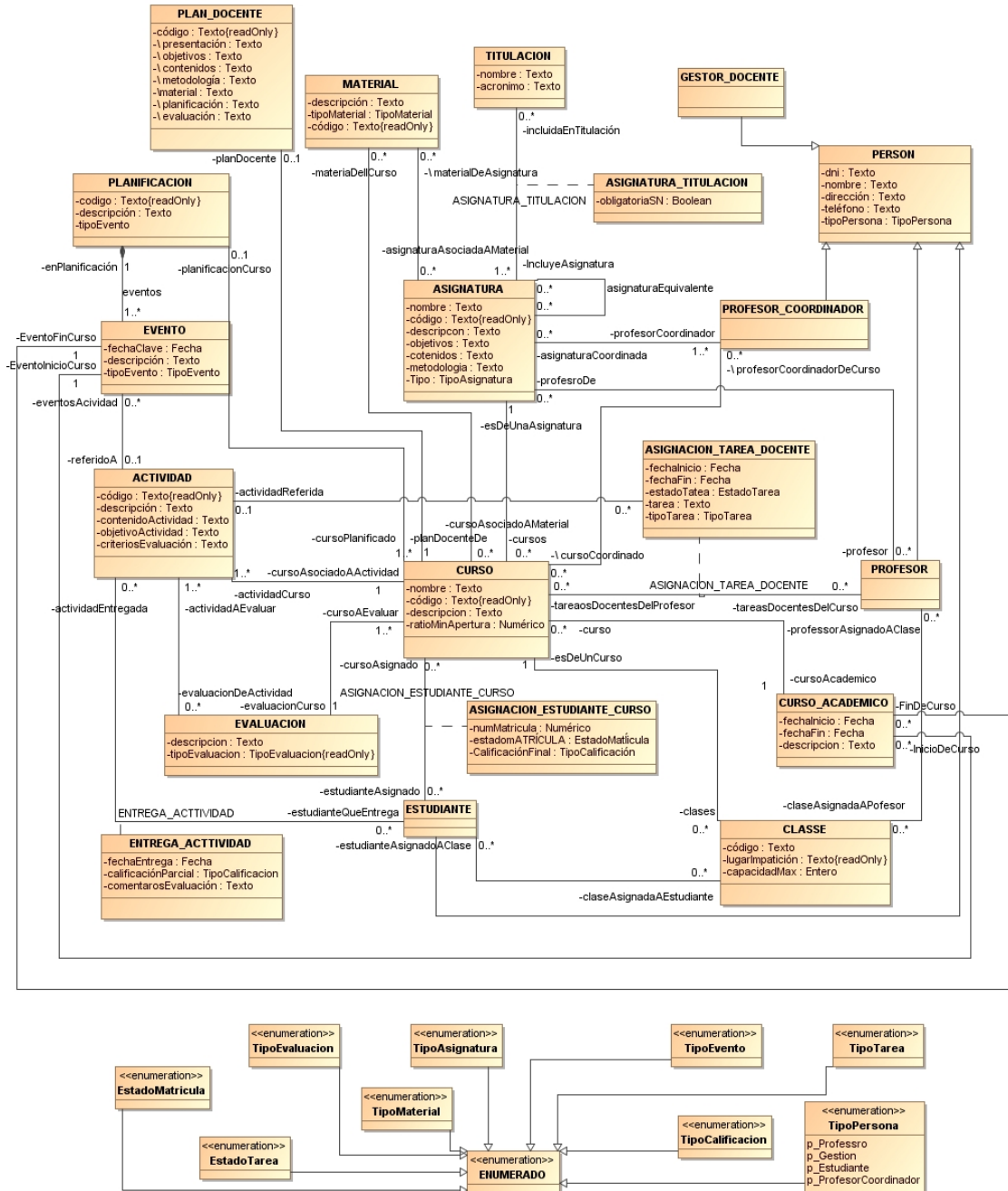


Figura A.2: Ontología de escenarios educativos

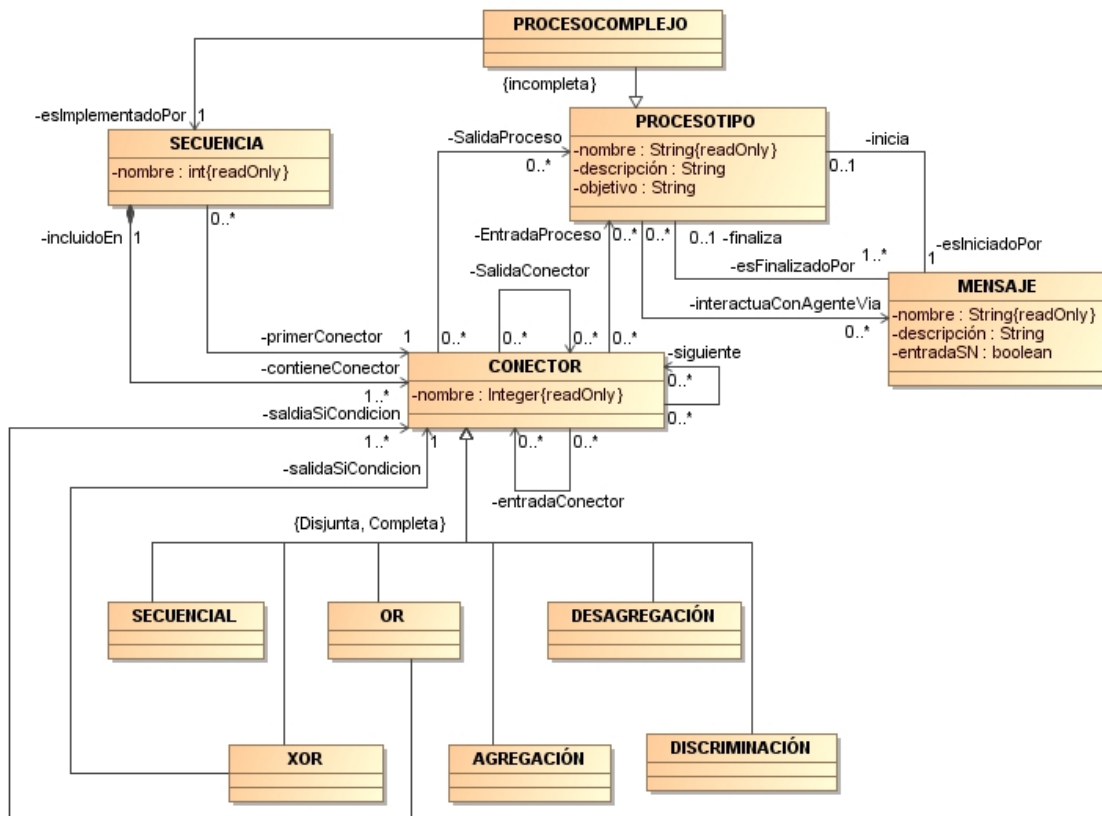


Figura A.3: Ontología de procesos concretos para la UOC

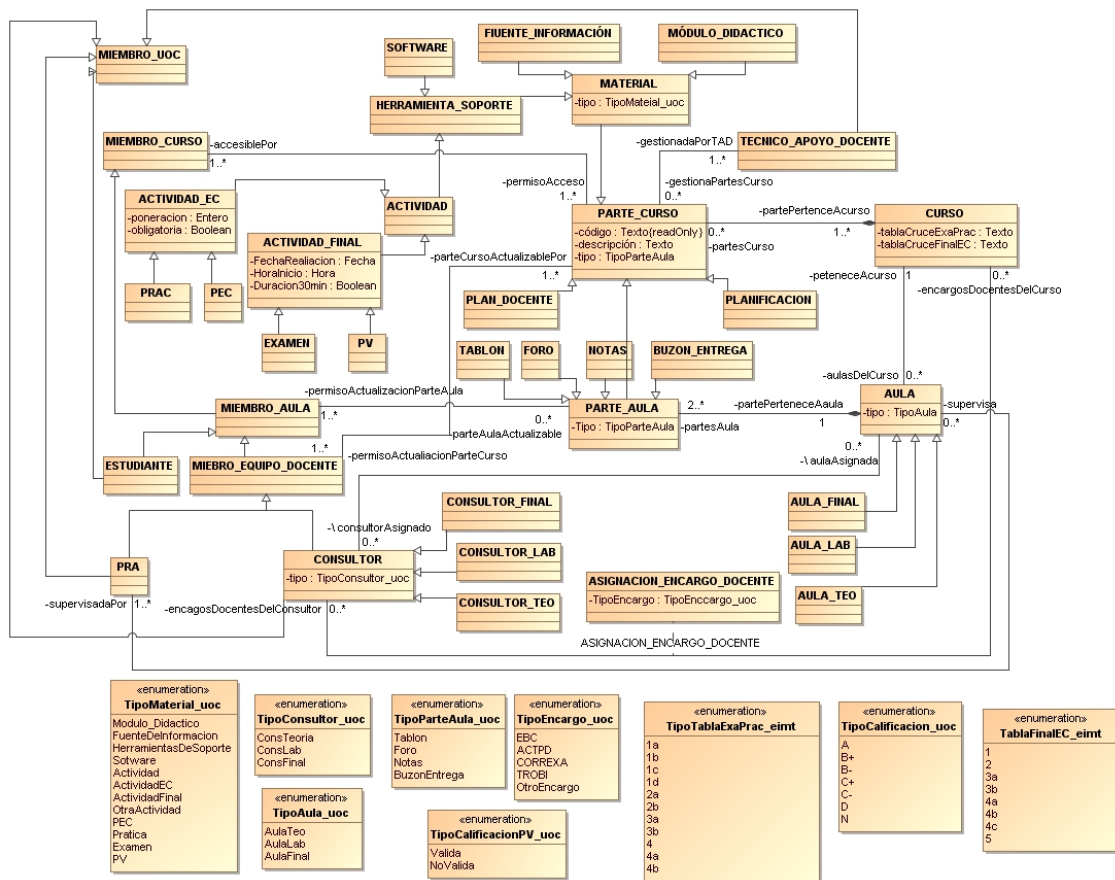


Figura A.4: Ontología de escenas educativas en el contexto de la UOC

Apéndice B

Guía práctica para la transformación de un diagrama UML a ontología OWL

De acuerdo con (Dean et al. 2004) una ontología en OWL se compone de los elementos siguientes:

- 1) **Clases**: grupos abstractos o colecciones de objetos jerárquicamente organizados. Existen por tanto, subclases o superclases según la clase que se tome como referencia.
- 2) **Propiedades** o *Slots*: pueden ser de tres tipos:
 - a) **Object Properties**: es una propiedad cuyo valor es un enlace a otra clase.
 - b) **Datatype Properties**: es una propiedad cuyo valor es un tipo de dato.
 - c) **Annotations**: es una propiedad que almacena comentarios sobre la clase para facilitar el diseño y comprensión de la ontología.
- 3) **Restricciones** o *Facets*: determinan como aplicar propiedades a las clases, pudiendo indicar que propiedades son obligatorias u opcionales para una clase, el número de instancias de otra clase con la que pueden relacionarse, etc.
- 3) **Instancias**: objetos concretos de la clase con valores concretos para cada propiedad.

Si disponemos de un diagrama de clase UML con los elementos de representación básicos podemos construir una ontología OWL aplicando las siguientes correspondencias:

- Una clase UML se corresponde con una clase OWL.
- Cada propiedad de una clase, definida en la sección anterior dentro de la tabla de propiedades de la cada clase, tienen una correspondencia directa con un *Data Property*.
- Cada propiedad de una clase definida como tipo de relación se les hace corresponder con un *Object Property*, siendo el rango de éste el tipo de dato (o clase) de la tabla de tipos de relación de la clase. En el caso de tipos de relación simétricos se establece una única propiedad bidireccional de manera que su inversa es ella misma.
- La definición de cada clase se guarda en la propiedad *Annotation* de la clase a modo de comentario.
- Toda restricción de propiedad de una clase que sólo pueda tener un valor se marca como funcional en el correspondiente *Datatype Property*.
- Toda restricción de cardinalidad relativa a un tipo de relación se representa como restricción de cardinalidad al definir cada clase.

Posteriormente, una vez creada la estructura ontológica deben de incorporarse aquellas reglas que determinan el conjunto de propiedades derivadas y/o restricciones e integridad no representables en OWL pero representables mediante su extensión en SWRL.

Apéndice C

Lenguaje gráfico para la representación de escenas educativas

En este anexo se describe la notación gráfica propuesta para la representación de escenas educativas.

Las escenas educativas reutilizables se describen como procesos simples que interactúan con participantes y utilizan recursos. Los componentes básicos para describir tales escenas se representan en la figura C.1








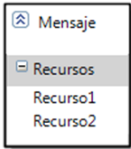

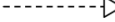
Elemento	Representación visual	Elemento	Representación visual
Escenario		Escenario complejo	
Recurso		Participante	
Inicio		Fin	
Excepción		Mensaje	
Flujo de control + datos		Flujo de datos exclusivamente	

Figura C.1: Componentes del lenguaje para describir escenas educativas reusables

Los flujos de flujos de datos son utilizados para indicar que un participante o proceso consume un recurso dado y los flujos de control en este tipo de escenas inician o finalizan el proceso. Los mensajes asociados a flujos de datos siempre contienen datos.

Para representar escenas complejas deben de utilizarse alguno de los conectores presentados en la figura C.2:






Elemento	Representación visual	Elemento	Representación visual
Desagregación		Agregación	
Or		XOr	
Discriminación			

Figura C.2: Componentes del lenguaje para describir escenas educativas complejas

En el apéndice D pueden encontrarse numerosos ejemplos de escenas representados con este lenguaje gráfico.

Apéndice D

Patrones de escenas utilizados en el caso de ejemplo

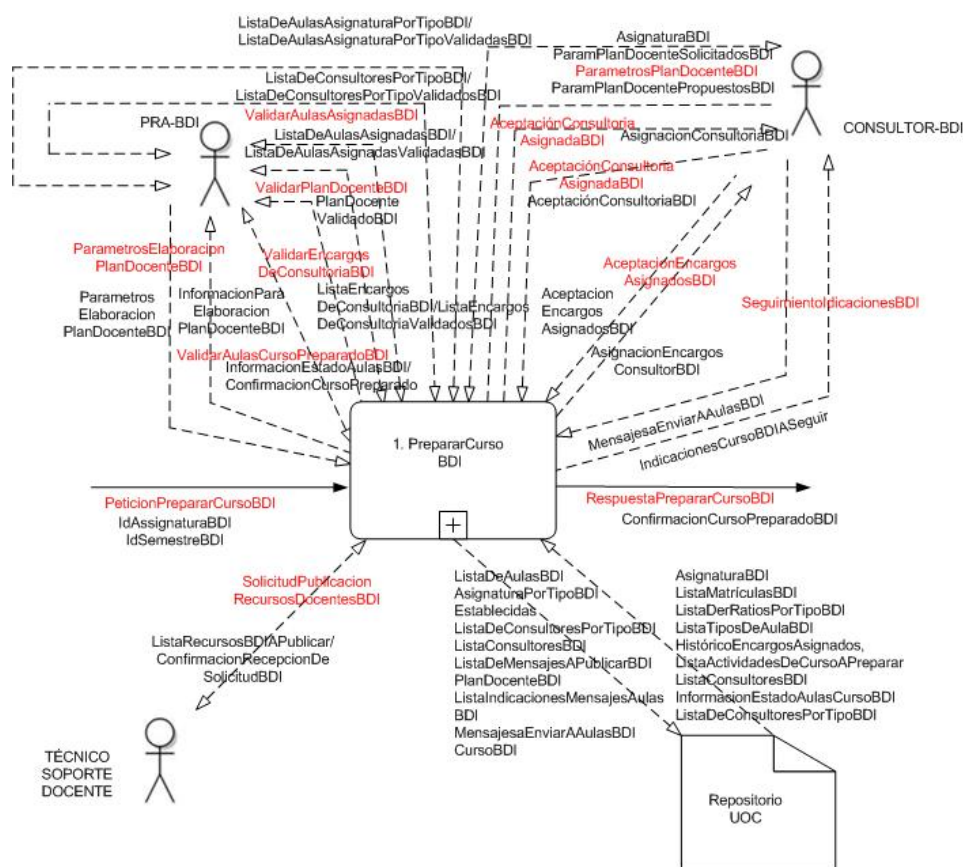


Figura D.1: Escena PrepararCurso

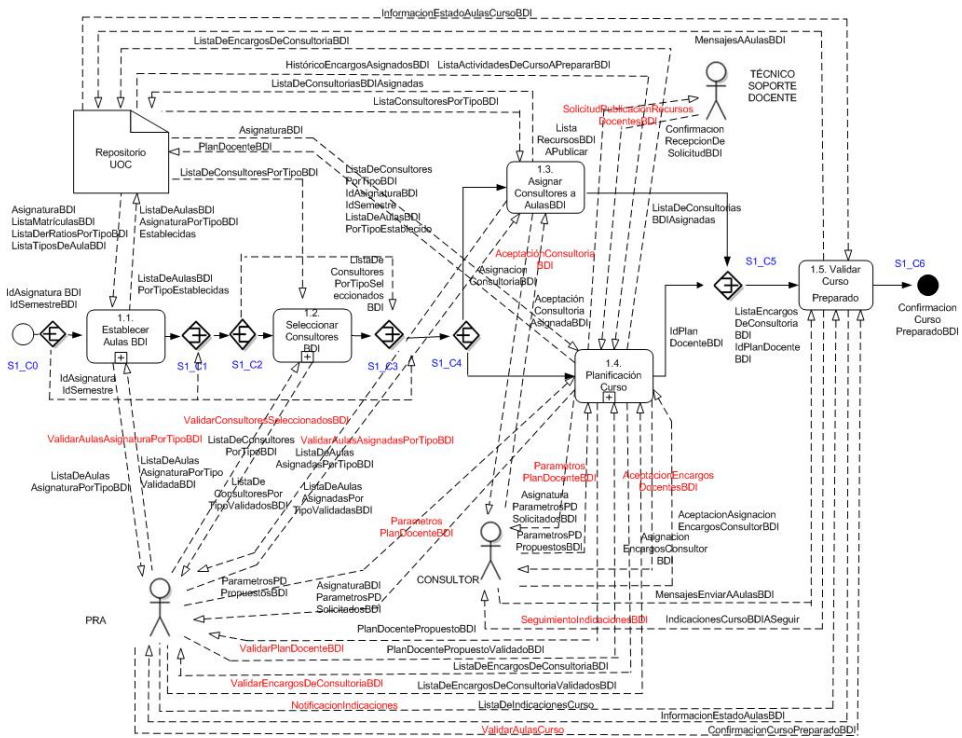


Figura D.2: Secuencia que representa la escena compleja PrepararCurso

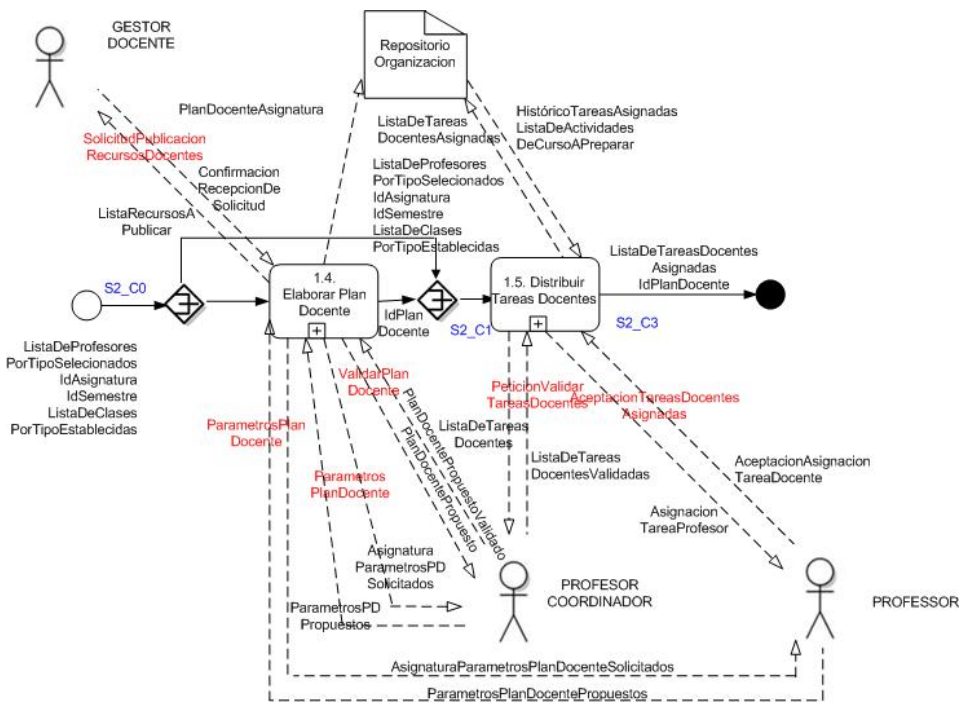


Figura D.3: Secuencia que representa la escena compleja PlanificarCurso

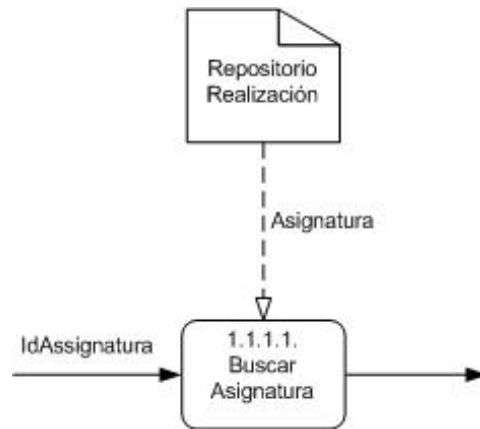


Figura D.4: Escena BuscarAsignatura

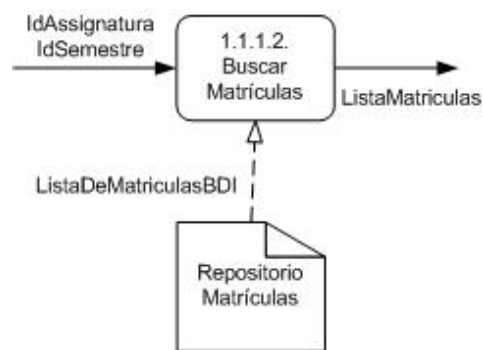


Figura D.5: Escena BuscarMatriculas

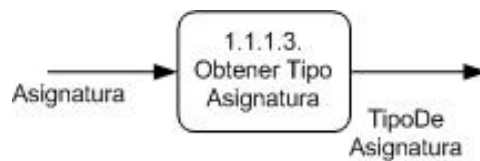


Figura D.6: Escena ObtenerTipoAsignatura



Figura D.7: Escena Contarelementoslista

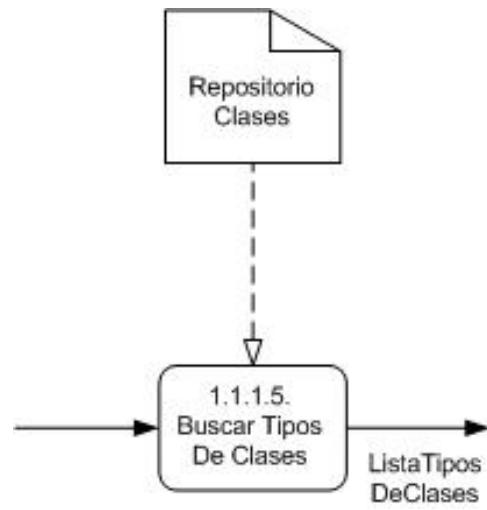


Figura D.8: Escena Buscar tipos de clases

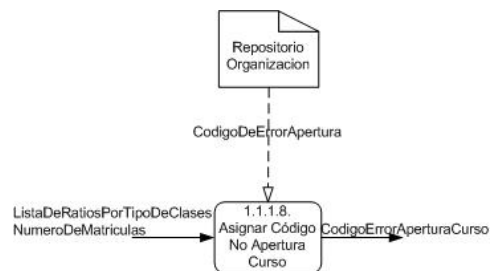


Figura D.9: Escena AsignarCodigoNoAperturaCurso

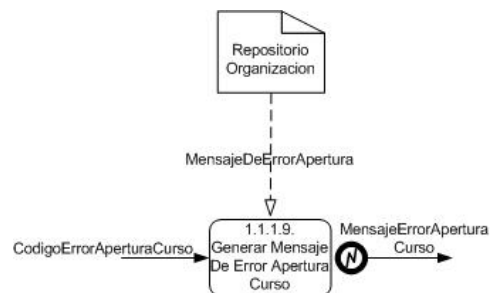


Figura D.10: Escena GenerarMensajeDeErrorAperturaCurso

Apéndice E

Diagramas del caso de ejemplo

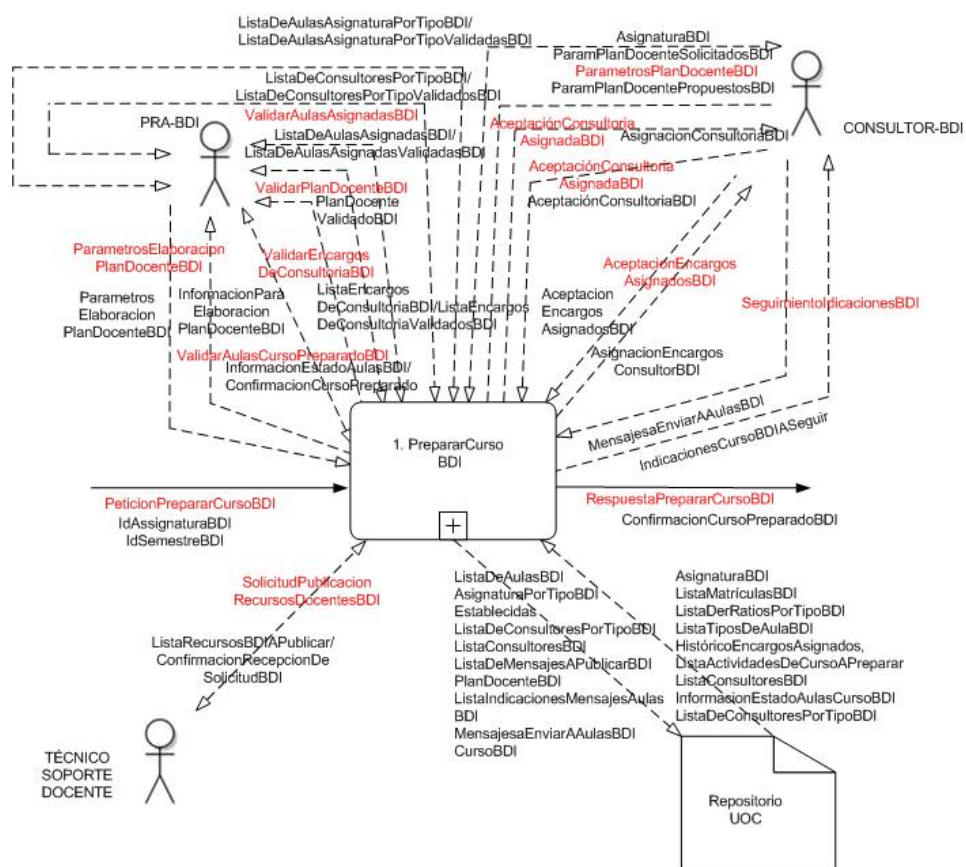


Figura E.1: Escena PrepararCursoBDI en el contexto de la UOC

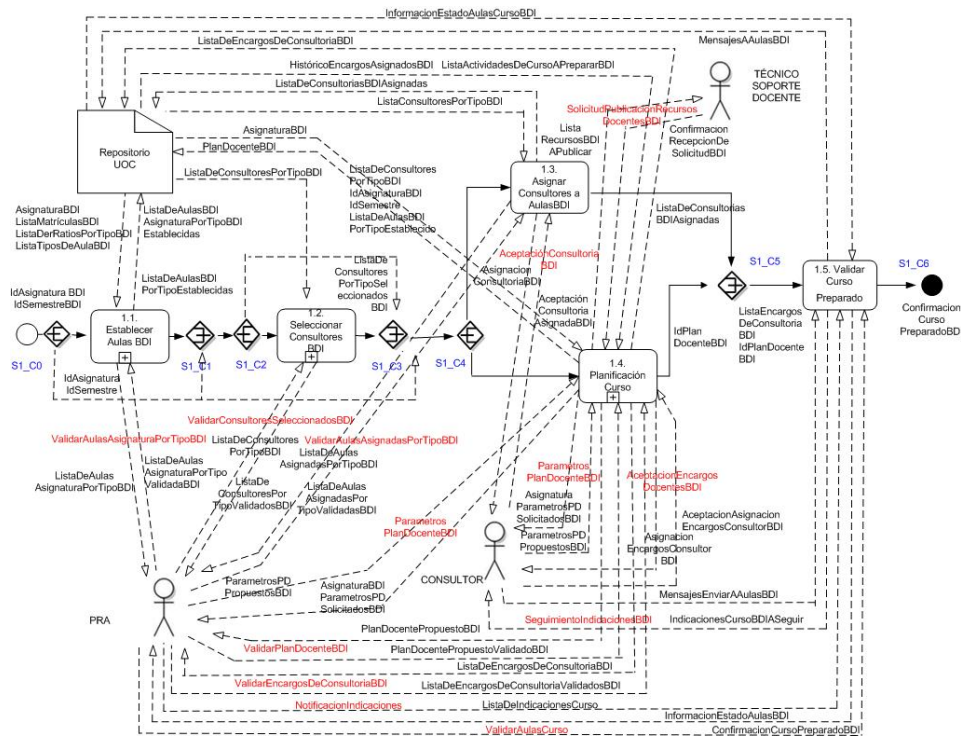


Figura E.2: Secuencia que representa la escena compleja PrepararCursoBDI en el contexto de la UOC

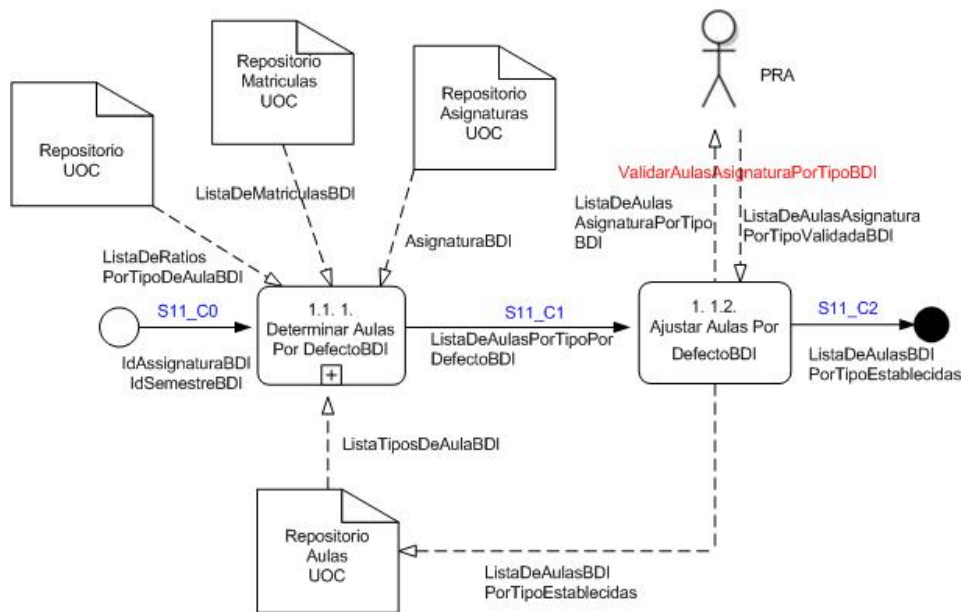


Figura E.3: Secuencia que representa la escena compleja EstablecerAulasBDI en el contexto de la UOC

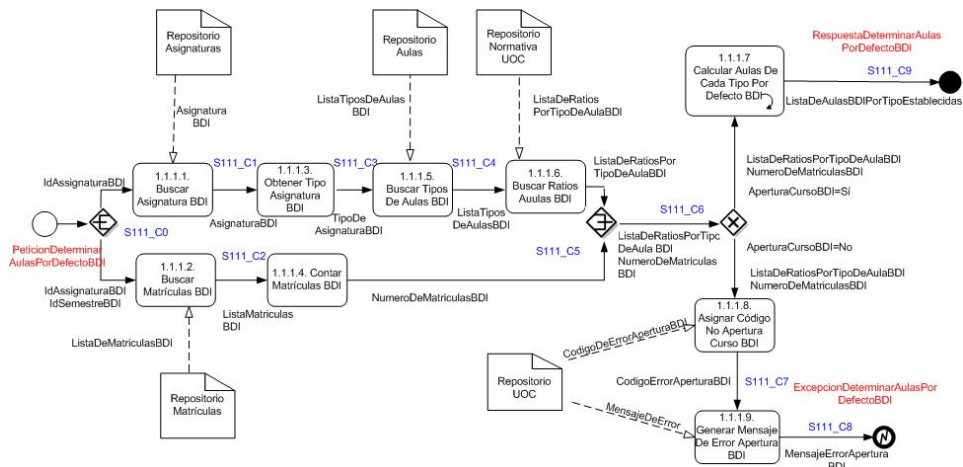


Figura E.4: Secuencia que representa la escena compleja DeterminarAulasPorDefectoBDI.

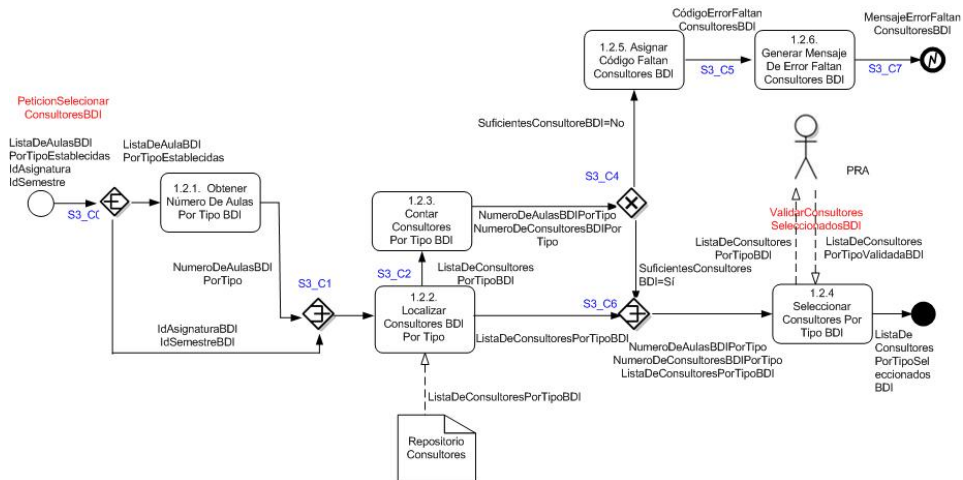


Figura E.5: Secuencia que representa la escena compleja SeleccionarConsultoresBDI.

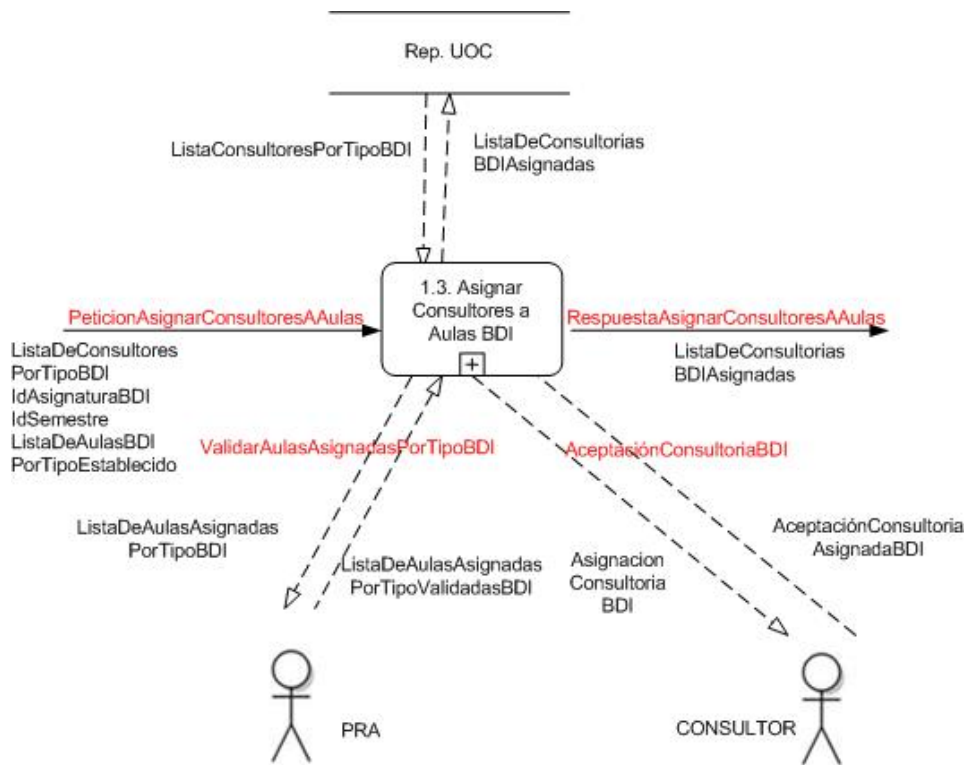


Figura E.6: Escena AsignarConsultoresAAulasBDI.

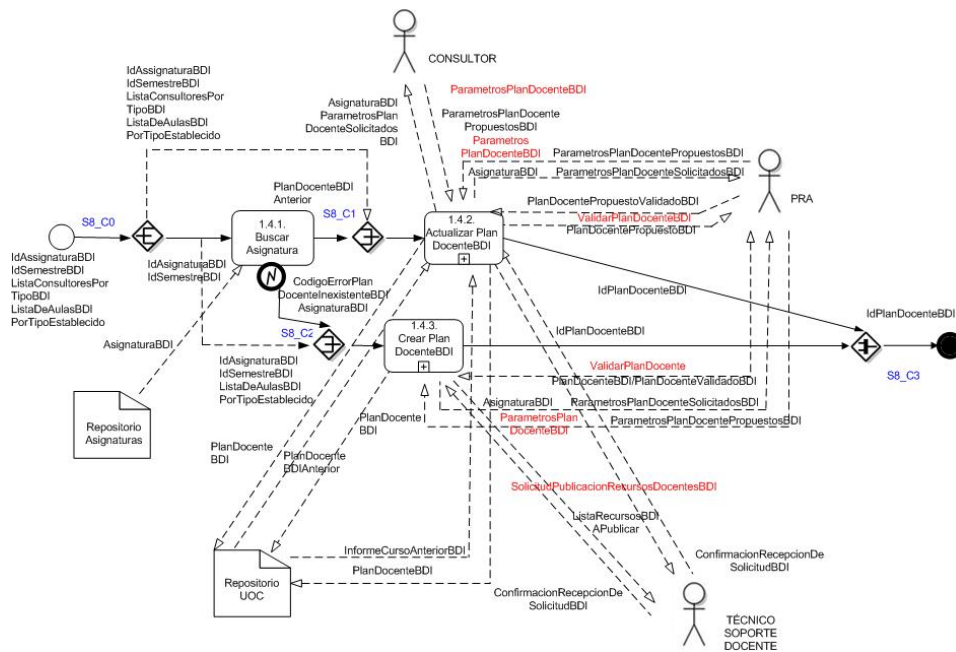


Figura E.7: Secuencia que representa la escena compleja ElaborarPlanDocenteBDI.

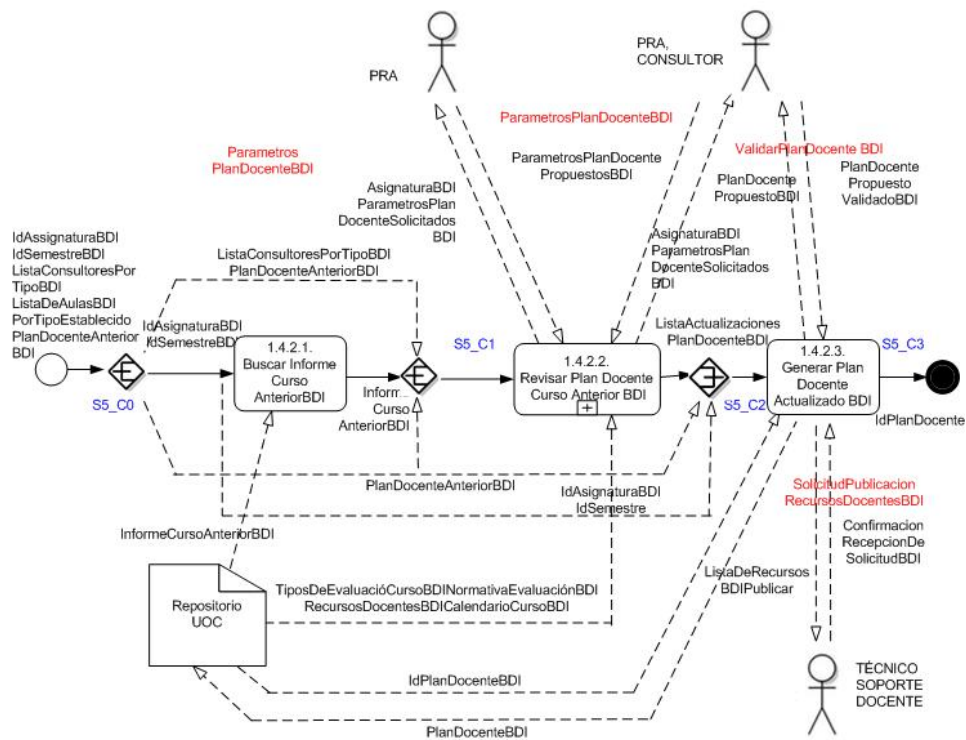


Figura E.8: Secuencia que representa la escena compleja ActualizarPlanDocenteBDI.

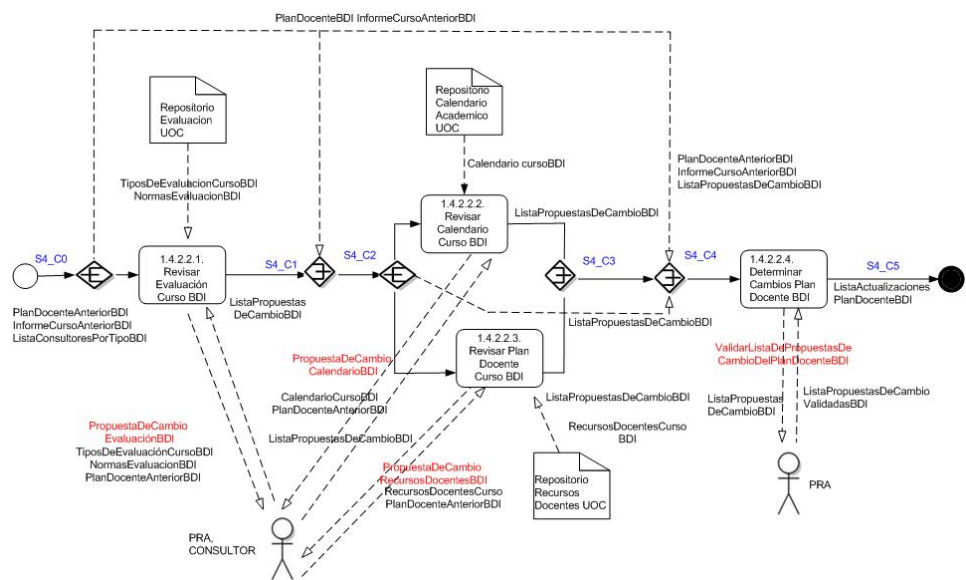


Figura E.9: Secuencia que representa la escena compleja RevisarPlanDocenteCursoAnteriorBDI.

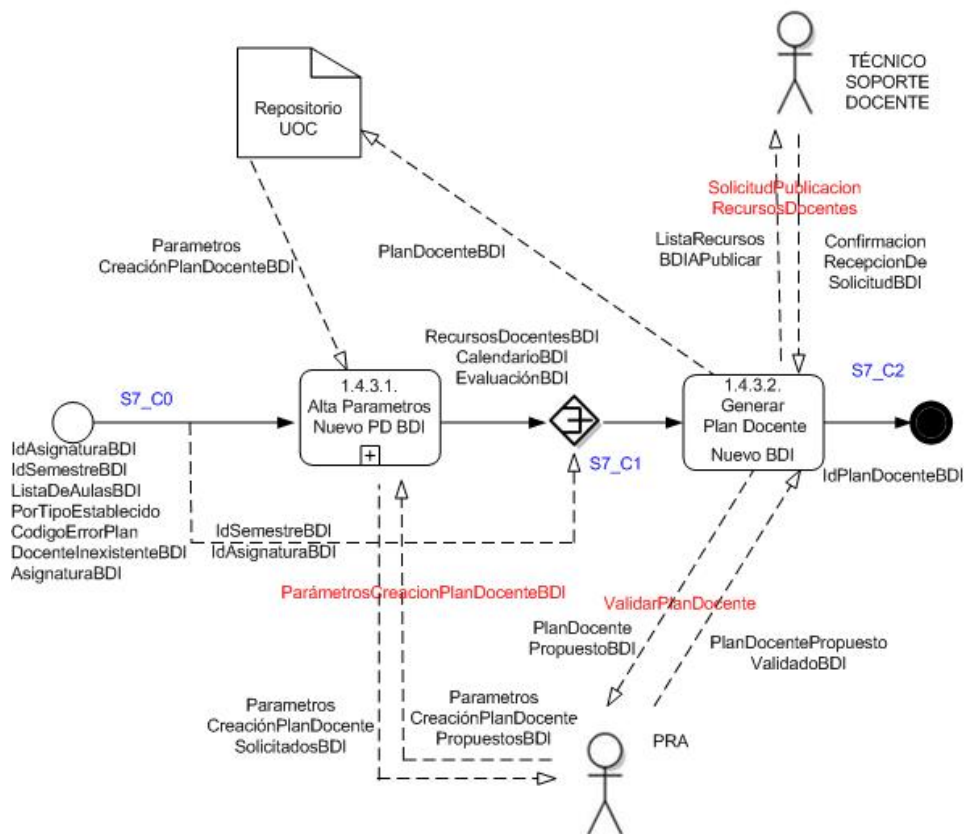


Figura E.10: Secuencia que representa la escena compleja CrearPlanDocenteBDI.

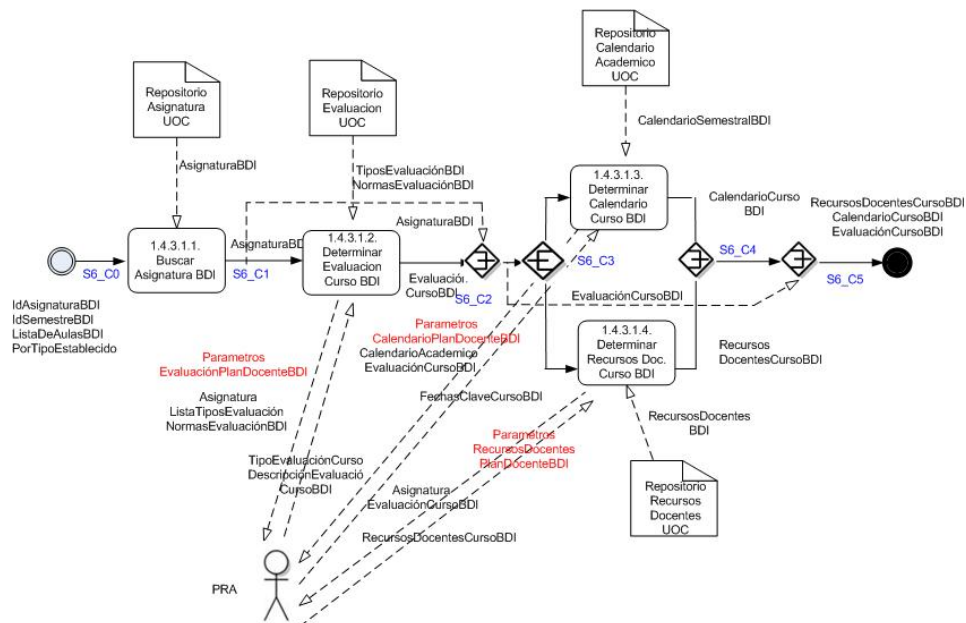


Figura E.11: Secuencia que representa la escena compleja AltaParametrosNuevoPlanDocenteBDI.

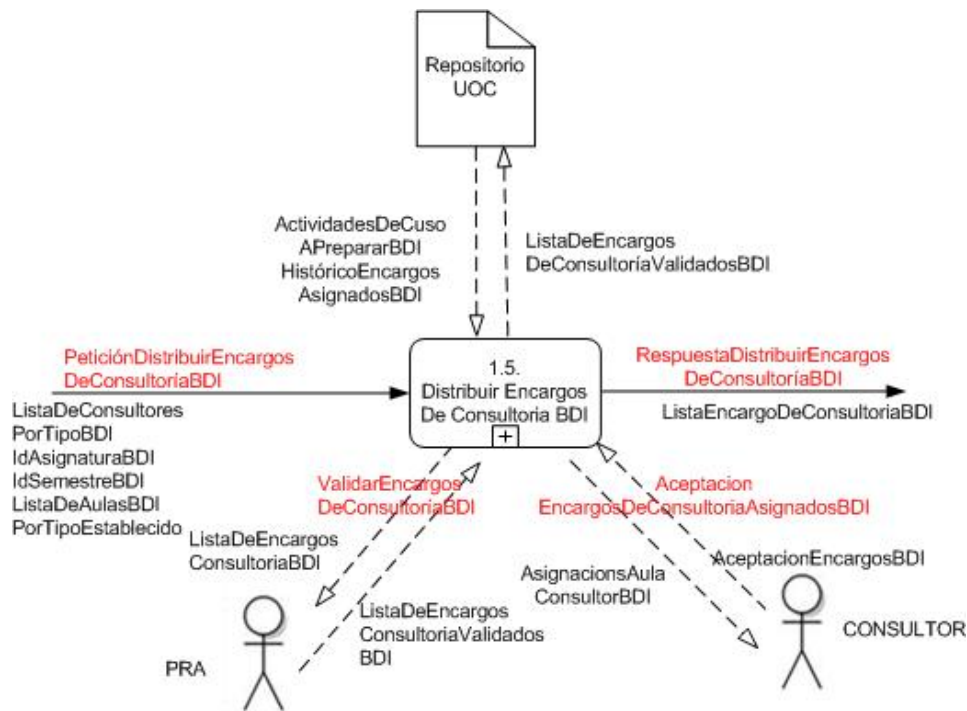


Figura E.12: Secuencia que representa la escena compleja DistribuirEncargosDeConsultoriaBDI.

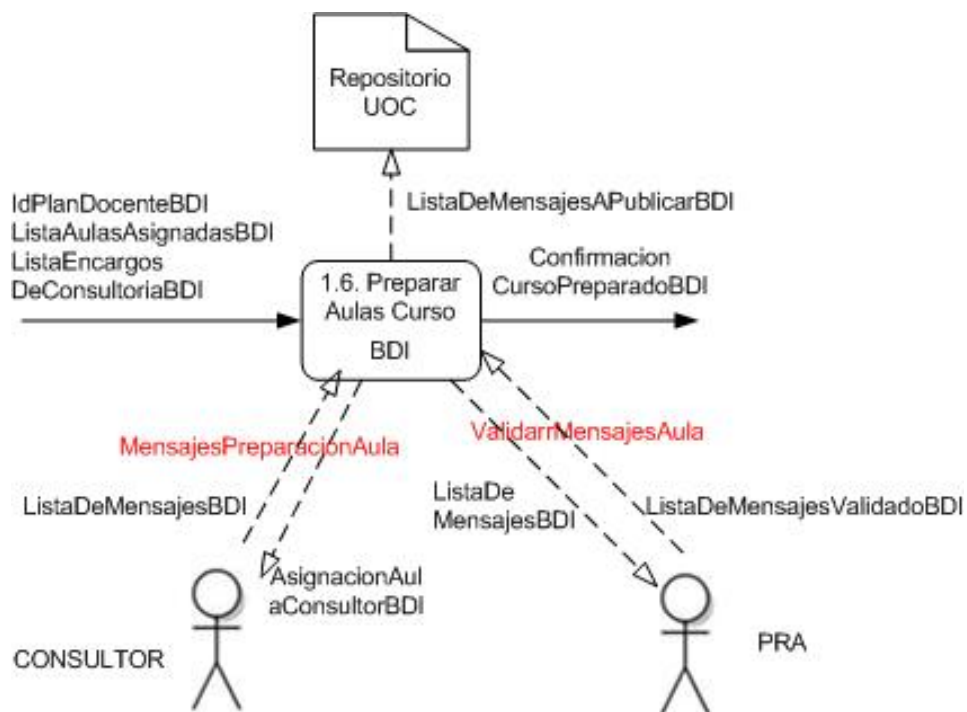


Figura E.13: Escena PrepararAulasCursoBDI.

Apéndice F

Una herramienta CASE para especificar escenas educativas

La herramienta CASE que se propone utiliza un DSL (Domain Specification Language) para ofrecer soporte a la descripción formal de escenas educativas.

Habitualmente, un DSL es considerado como la unión de un lenguaje, un dominio y una herramienta. En la figura F.1 se visualizan los tres componentes de esta herramienta.

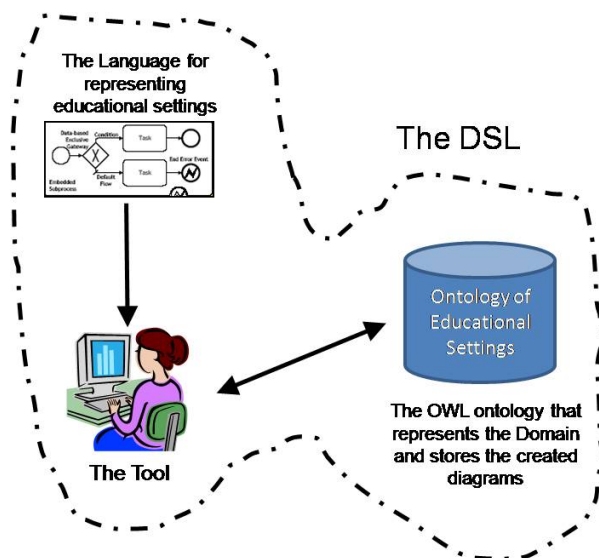


Figura F.1: Componentes del DSL propuesto

En este caso concreto, el DSL utiliza el lenguaje gráfico descrito en el anexo C, el dominio de OntoProc descrito en el capítulo 4 para la definición semántica de dicho lenguaje gráfico y la herramienta CASE que se describe en este anexo.

F.1. Utilidad de la herramienta

Esta herramienta CASE ha sido diseñada para ofrecer soporte en la instanciación de la ontología, así como en la diagramación de las especificaciones de escenas.

En concreto, esta herramienta debe de facilitar al usuario: la descripción de escenas educativas mediante el lenguaje gráfico propuesto y la instanciación automática de la ontología. Estas funcionalidades se consiguen mediante un DSL que permite reconocer automáticamente los elementos del lenguaje gráfico y a partir de diagramas que utilizan esta notación, lleva a cabo una instanciación de la ontología que describe la semántica del lenguaje.

En la figura F.2 puede visualizarse gráficamente el cometido de esta herramienta.

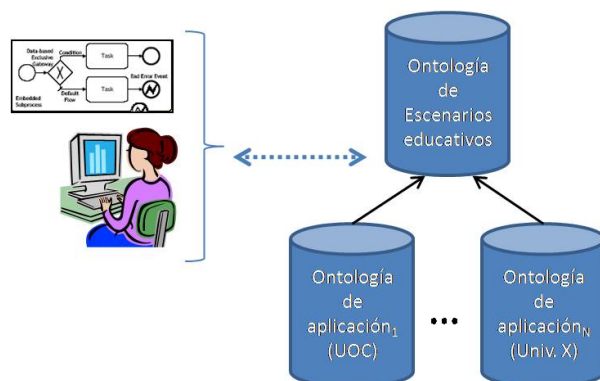


Figura F.2: Utilidad de la herramienta CASE

F.2. Arquitectura del prototipo

La construcción de esta herramienta mediante un prototipo se ha llevado a cabo utilizando la tecnología .NET, exactamente el *framework* Visual Studio DSL tools (VS-DSLTL).

VSDSLTL es un entorno que permite al desarrollador definir tanto el metamodelo del lenguaje a modelar como los elementos gráficos que establecen las clases y relaciones del dominio. También permite incorporar reglas de validación y restricciones de integridad.

Además, VSDLT permite generar a partir del modelo de instancias (un diagrama modelado con DSL): ficheros de código, un *report* o como en este caso un fichero con los elementos definidos en el diagrama.

En concreto, la herramienta creada con VSDSLT es un *plug-in* que puede ser instalado e integrado en Visual Studio IDE.

F.3. La interfície del prototipo

En este caso la herramienta diseñada con este DSL es una herramienta CASE que permite definir procesos mediante elementos gráficos tales como formas y líneas. Las formas representan instancias de clases del dominio, mientras que las líneas representan relaciones entre estas instancias. Dichos elementos son accesibles a través de una caja de herramientas y pueden ser seleccionados y arrastrados por el usuario hacia el diagrama que está construyendo.

Esta herramienta presenta una interfície compuesta por dos vistas. La primera permite definir la interfície del proceso, es decir, sus mensajes de entrada y de salida, las interacciones con participantes externos y los recursos que utiliza. La otra vista permite definir secuencias de procesos mediante la combinación de otros procesos y conectores y sus relaciones. El usuario puede cambiar de una vista a otra utilizando alguna opción disponible en el menú contextual del diagrama.

Al crear un diagrama nuevo el usuario puede elegir qué tipo de diagrama desea, simple o complejo (como combinación de otros). Esta selección determina la forma como el usuario interactúa con el diagrama, por ejemplo, un proceso simple no dispone de una vista de secuencias y en un diagrama complejo el usuario no puede añadir o eliminar participantes, ni recursos en la definición de la vista, ya que éstos son inferidos a partir de los procesos utilizados en su vista de secuencia. Además, debe tenerse en cuenta que el número de formas y elementos de relación disponibles en la caja de herramientas se ve condicionado por el tipo de proceso, simple o complejo. Por ejemplo, un diagrama que describa un proceso complejo no tendrá en su caja de herramientas participantes ni recursos, puesto que no es posible añadirlos en este tipo de diagramas.

Por otro lado, existe un conjunto de reglas que son comprobadas por la herramienta

para asegurar la integridad del modelo, la mayoría de ellas restricciones de cardinalidad, por ejemplo un estado inicial debe de tener una y sólo una actividad de salida (proceso o conector) pero no tiene entradas.

Cuando el diagrama ha sido validado, es decir no tiene errores, se grava y la herramienta automáticamente traduce los elementos definidos en él a instancias de la ontología OntoProc, generando el correspondiente fichero OWL.

En (Rius, Conesa & Gañán 2010) puede encontrarse información ampliada sobre esta herramienta.

Apéndice G

Reglas SWRL para la traducción del marco propuesto a OKI

La ontología que constituye el marco propuesto ha sido extendido con la ontología OKI. El espacio de nombres de la ontología OKI siguiente:

p1 <http://www.owl-ontologies.com/Ontology1254295609.owl#>

Regla 46 (Rule-Transf-TitulacionToCourseGroup):

La clase CourseGroup de OKI es semánticamente equivalente a la clase Titulación de OntoProcED.

$\text{Titulacion}(?x) \rightarrow p1:\text{CourseGroup}(?x)$

Regla 47 (Rule-Transf-NombreTitulacionToDiplayNameCourseGroup):

El propiedad displayName de clase CourseGroup de OKI es semánticamente equivalente a la propiedad nombre de la clase Titulación.

$\text{Titulacion}(?x) \wedge \text{nombre}(?x, ?n) \wedge p1:\text{CourseGroup}(?x) \rightarrow p1:\text{displayName}(?x, ?n)$

Regla 48 (Rule-Transf-AsignaturaToCanonicalCourse):

La clase CanonicalCourse de OKI es semánticamente equivalente a la clase Asignatura de OntoProcED.

$\text{Asignatura}(?x) \rightarrow p1:\text{CanonicalCourse}(?x)$

Regla 49 (Rule-Transf-NombreAsignaturaToTitleCanonicalCourse):

El propiedad title de clase CanonicalCourse de OKI es semánticamente equivalente a la propiedad nombre de la clase Asignatura de OntoProcED.

$\text{Asignatura}(?x) \wedge \text{nombre}(?x, ?n) \wedge p1:\text{CanonicalCourse}(?x) \rightarrow p1:\text{title}(?x, ?n)$

Regla 50 (Rule-Transf-CodigoAsignaturaToNumberCanonicalCourse):

El propiedad number de clase CanonicalCourse de OKI es semánticamente equivalente a la propiedad código de la clase Asignatura de OntoProcED.

$\text{Asignatura}(?x) \wedge \text{codigo}(?x, ?n) \wedge p1:\text{CanonicalCourse}(?x) \rightarrow p1:\text{number}(?x, ?n)$

Regla 51 (Rule-Transf-DescripcionAsignaturaToDescriptionCanonicalCourse):

El propiedad description de clase CanonicalCourse de OKI es semánticamente equivalente a la propiedad descripcion de la clase Asignatura de OntoProcED.

$Asignatura(?x) \wedge descripcion(?x, ?n) \wedge p1:CanonicalCourse(?x) \rightarrow p1:description(?x, ?n)$

Regla 52 (Rule-Transf-ContenidosAsignaturaToTopicsCanonicalCourse):

El propiedad topics de clase CanonicalCourse de OKI es semánticamente equivalente a la propiedad contenidos de la clase Asignatura de OntoProcED.

$Asignatura(?x) \wedge contenidos(?x, ?n) \wedge p1:CanonicalCourse(?x) \rightarrow p1:topics(?x, ?n)$

Regla 53 (Rule-Transf-TipoAsignaturaTypeToCanonicalCourse):

El propiedad type de clase CanonicalCourse de OKI es semánticamente equivalente a la propiedad tipo de la clase Asignatura de OntoProcED.

$Asignatura(?x) \wedge contenidos(?x, ?n) \wedge p1:CanonicalCourse(?x) \rightarrow p1:topics(?x, ?n)$

Regla 54 (Rule-Transf-CursoAcademicoToTerm):

La clase Term de OKI es semánticamente equivalente a la clase CursoAcademico de OntoProcED.

$CursoAcademico(?x) \rightarrow p1:Term(?x)$

Regla 55 (Rule-Transf-CursoToCourseOffering):

La clase CourseOffering de OKI es semánticamente equivalente a la clase Curso de OntoProcED.

$Curso(?x) \wedge p1:CourseOffering(?x)$

Regla 56 (Rule-Transf-NombreCursoToTitleCourseOffering):

El propiedad title de clase CourseOffering de OKI es semánticamente equivalente a la propiedad nombre de la clase Curso de OntoProcED.

$Curso(?x) \wedge nombre(?x, ?nc) \wedge p1:CourseOffering(?x) \rightarrow p1:title(?x, ?nc)$

Regla 57 (Rule-Transf-CodigoCursoToTitleCourseOffering):

El propiedad number de clase CourseOffering de OKI es semánticamente equivalente a la propiedad código de la clase Curso de OntoProcED.

$Curso(?x) \wedge codigo(?x, ?nc) \wedge p1:CourseOffering(?x) \rightarrow p1:number(?x, ?nc)$

Regla 58 (Rule-Transf-EvaluacionToCourseGradeType):

La propiedad CourseGradeType de la clase CourseOffering de OKI es semánticamente equivalente a la clase Evaluación de OntoProcED.

$Curso(?x) \wedge evaluacionCurso(?x, ?e) \wedge p1:CourseOffering(?x) \rightarrow p1:courseGradeType(?x, ?e)$

Regla 59 (Rule-Transf-EventoToScheduleItem):

La clase ScheduleItem de OKI es semánticamente equivalente a la clase Evento de OntoProcED.

$Evento(?x) \rightarrow p1:ScheduleItem(?x)$

Regla 60 (Rule-Transf-DescripcionEventoToDescriptionScheduleItem):

El propiedad description de clase ScheduleItem de OKI es semánticamente equivalente a la propiedad descripcion de la clase Evento de OntoProcED.

$Evento(?x) \wedge descripcion(?x, ?d) \wedge p1:ScheduleItem(?x) \rightarrow p1:description(?x, ?d)$

Regla 61 (Rule-Transf-FechaClaveEventoToStartScheduleItem):

El propiedad start de clase ScheduleItem de OKI es semánticamente equivalente a la propiedad fechaClave de la clase Evento de OntoProcED.

$Evento(?x) \wedge fechaClave(?x, ?n) \wedge p1:ScheduleItem(?x) \rightarrow p1:start(?x, ?n)$

Regla 62 (Rule-Transf-MaterialToAsset):

La clase Asset de OKI es semánticamente equivalente a la clase Material de OntoProcED.

$Material(?x) \rightarrow p1:Asset(?x)$

Regla 63 (Rule-Transf-DescripcionMaterialToDescriptionAsset):

El propiedad description de clase Asset de OKI es semánticamente equivalente a la propiedad descripcion de la clase Material de OntoProcED.

Regla 64 (Rule-Transf-TipoMaterialAssetToTypeAsset):

El propiedad tipo de clase Asset de OKI es semánticamente equivalente a la propiedad tipoMaterial de la clase Material de OntoProcED.

$Material(?x) \wedge tipoMaterial(?x, ?n) \wedge p1:Asset(?x) ? p1:assetType(?x, ?n)$

Regla 65 (Rule-Transf-CodigoMaterialToIdAsset):

El propiedad Id de clase Asset de OKI es semánticamente equivalente a la propiedad código de la clase Material de OntoProcED.

$Material(?x) \wedge codigo(?x, ?n) \wedge p1:Asset(?x) \rightarrow p1:id(?x, ?n)$

Regla 66 (Rule-Transf-ClaseToCourseSection):

La clase CourseSection de OKI es semánticamente equivalente a la clase Clase de OntoProcED.

$Clase(?x) \rightarrow p1:CourseSection(?x)$

Regla 67 (Rule-Transf-CodigoClaseIdToCourseSection):

El propiedad id de clase CourseSection de OKI es semánticamente equivalente a la propiedad código de la clase Clase de OntoProcED.

$Clase(?x) \wedge codigo(?x, ?n) \wedge p1:CourseSection(?x) \rightarrow p1:number(?x, ?n)$

Regla 68 (Rule-Transf-LugarDeImparticionToClaseLocationCourseSection):

El propiedad location de clase CourseSection de OKI es semánticamente equivalente a la propiedad lugarDeImpartición de la clase Clase de OntoProcED.

$Clase(?x) \wedge lugarDeImparticion(?x, ?n) \wedge p1:CourseSection(?x) \rightarrow p1:location(?x, ?n)$

Regla 69 (Rule-Transf-CodigoActividadToIdAssessmentPublished):

El propiedad id de clase AssessmentPublished de OKI es semánticamente equivalente a la propiedad código de la Clase Actividad de OntoProcED.

$Actividad(?x) \wedge codigo(?x, ?c) \wedge p1:AssessmentPublished(?x) \rightarrow p1:id(?x, ?c)$

Regla 70 (Rule-Transf-DescripcionActividadToDescriptionAssessmentPublished):

El propiedad description de clase AssessmentPublished de OKI es semánticamente equivalente a la propiedad descripción de la Clase Actividad de OntoProcED.

Actividad(?x) ∧ descripcion(?x, ?c) ∧ p1:AssessmentPublished(?x) → p1:description(?x, ?c)

Regla 71 (Rule-Transf-ContenidosActividadToDataAssessmentPublished):

El propiedad data de clase AssessmentPublished de OKI es semánticamente equivalente a la propiedad contenido de la clase Actividad de OntoProcED.

Actividad(?x) ∧ contenidos(?x, ?c) ∧ p1:AssessmentPublished(?x) → p1:data(?x, ?c)

Regla 72 (Rule-Transf-PersonaToAgent):

El clase Agent de OKI es semánticamente equivalente a la clase Persona de OntoProcED.

Persona(?x) → p1:Agent(?)

Regla 73 (Rule-Transf-DniPersonaToIdAgent):

El propiedad id de la clase Agent de OKI es semánticamente equivalente a la propiedad dni de la clase Persona de OntoProcED.

Persona(?x) ∧ dni(?x, ?d) ∧ p1:Agent(?) → p1:id(?x, ?d)

Regla 74 (Rule-Transf-NombrePersonaToDisplayNameAgent):

El propiedad displayName de la clase Agent de OKI es semánticamente equivalente a la propiedad nombre de la clase Persona de OntoProcED.

Persona(?x) ∧ nombre(?x, ?d) ∧ p1:Agent(?x) → p1:displayName(?x, ?d)

Regla 75 (Rule-Transf-TipoPersonaToTypeAgent):

El propiedad type de la clase Agent de OKI es semánticamente equivalente a la propiedad tipoPersona de la clase Persona de OntoProcED.

Persona(?x) ∧ tipoPersona(?x, ?d) ∧ p1:Agent(?x) → p1:type(?x, ?d)

Regla 76 (Rule-Transf-ProfesorToGroup):

La clase Group tiene una instancia profesor si existe alguna instancia de persona de OntoProcED que tiene como valor de la propiedad tipoPersona 'p_Profesor'.

tipoPersona(?x, p_Profesor) → p1:Group(Profesor)

Regla 77 (Rule-Transf-ProfesorCoordinadorToGroup):

La clase Group tiene una instancia profesorCoordinador si existe alguna instancia de persona de OntoProcED que tiene como valor de la propiedad tipoPersona 'p_ProfesorCoordinador'.

Persona(?x) ∧ tipoPersona(?x, p_ProfesorCoordinador) → p1:Group(ProfesorCoordinador)

Regla 78 (Rule-Transf-GestorDocenteToGroup):

La clase Group tiene una instancia gestorDocente si existe alguna instancia de persona de OntoProcED que tiene como valor de la propiedad tipoPersona 'p_GestorDocente'.

Persona(?x) ∧ tipoPersona(?x, p_GestorDocente) → p1:Group(GestorDocente)

Regla 79 (Rule-Transf-ProfesorCoordinadorToGroup):

La clase Group tiene una instancia estudiante si existe alguna instancia de persona de OntoProcED que tiene como valor de la propiedad tipoPersona 'p_Estudiante'.

$Persona(?x) \wedge tipoPersona(?x, p_Estudiante) \rightarrow p1:Group(Estudiante)$

Regla 80 (Rule-Transf-EntregaActividadToAssessmentTaken):

La clase AssessmentTaken de OKI tiene correspondencia con la clase EntregaActividad y la propiedad actividadDeLaEntrega de OntoProcED.

$EntregaActividad(?x) \wedge actividadDeLaEntrega(?x, ?y) \rightarrow p1:AssessmentTaken(?y)$

Regla 81 (Rule-Transf-FechaEntregaActividadToDateAssessmentTaken):

La propiedad date de la clase AssessmentTaken de OKI tiene correspondencia con la propiedad fechaEntrega de la clase EntregaActividad de OntoProcED.

$EntregaActividad(?x) \wedge actividadDeLaEntrega(?x, ?y) \wedge fechaEntrega(?y, ?c) \wedge p1:AssessmentTaken(?x) \rightarrow p1:date(?y, ?c)$

Regla 82 (Rule-Transf-AsignacionTareaDocenteToScheduleItem):

La clase ScheduleItem de OKI tiene correspondencia con la clase AsignacionTareaDocente de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge profesorAsignadoAClase(?ad, ?p) \wedge AsignacionTareaDocente(?ad) \wedge p1:courseSections(?c, ?cl) \wedge p1:schedule(?cl, ?s) \rightarrow p1:ScheduleItem(?s)$

Regla 83 (Rule-Transf-AsignacionTareaDocenteToAgentCommitment):

La clase AgentCommitment de OKI tiene correspondencia con la clase AsignacionTareaDocente de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge profesorAsignadoAClase(?ad, ?p) \wedge AsignacionTareaDocente(?ad) \wedge p1:courseSections(?c, ?cl) \wedge p1:schedule(?cl, ?s) \wedge p1:agentCommitments(?s, ?t) \rightarrow p1:AgentCommitment(?t)$

Regla 84 (Rule-Transf-EstadoTareaAsignacionTareaDocenteToStatusAgentCommitment):

La propiedad status de la clase AgentCommitment de OKI tiene correspondencia con la propiedad estadoTarea de clase AsignacionTareaDocente de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge AsignacionTareaDocente(?ad) \wedge p1:courseSections(?c, ?cl) \wedge p1:schedule(?cl, ?s) \wedge p1:agentCommitments(?s, ?t) \wedge p1:AgentCommitment(?t) \wedge estadoTarea(?ad, ?e) \rightarrow p1:status(?t, ?e)$

Regla 85 (Rule-Transf-TareaAsignacionTareaDocenteToDescriptionScheduleItem):

La propiedad description de la clase AssessmentTaken de OKI tiene correspondencia con la propiedad tarea de la clase EntregaActividad de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge profesorAsignadoAClase(?ad, ?p) \wedge AsignacionTareaDocente(?ad) \wedge p1:agentID(?s, p?) \wedge p1:Schedule(?c, ?s) \wedge tarea(?ad, t?) \rightarrow p1:description(?s, ?d)$

Regla 86 (Rule-Transf-FechaInicioAsignacionTareaDocenteToStartcheduleItem):

La propiedad fechaInicio de la clase AssessmentTaken de OKI tiene correspondencia con la propiedad start de la clase EntregaActividad de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge profesorAsignadoAClase(?ad, ?p) \wedge$
 $AsignacionTareaDocente(?ad) \wedge p1:agentId(?s, ?p) \wedge p1:schedule(?c, ?s) \wedge tarea(?ad,$
 $?t) \rightarrow p1:description(?s, ?t)$

Regla 87 (Rule-Transf-FechaFinAsignacionTareaDocenteToEndScheduleItem):

La propiedad fechaFin de la clase AssessmentTaken de OKI tiene correspondencia con la propiedad end de la clase EntregaActividad de OntoProcED.

$cursoAsignadoAunaTarea(?ad, ?c) \wedge profesorAsignadoAClase(?ad, ?p) \wedge$
 $AsignacionTareaDocente(?ad) \wedge p1:agentId(?s, ?p) \wedge p1:schedule(?c, ?s) \wedge$
 $fechaFin(?ad, ?t) \rightarrow p1:end(?s, ?t)$

Regla 88 (Rule-Transf-AsignacionEstudianteCursoToCourseGradeRecord):

La clase CourseGrade tiene correspondencia con la clase AsignacionEstudianteCurso.

$cursoEnAsignacion(?ae, c?) \wedge esudianteEnAsignacion(?ae, e?) \wedge$
 $AsignacionEstudianteCurso(?ae) \wedge p1:courseOffering(?gr, ?c?) \wedge p1:agentId(?gr?, e?)$
 $\rightarrow CourseGradeRecord(?gr)$

Regla 89 (Rule-Transf-NumMatriculaAsignacionEstudianteCursoToIdCourseGradeRecord):

La propiedad Id de la clase CourseGradeRecord de OKI tiene correspondencia con la propiedad numMatriucla de la clase AsignacionEstudianteCurso de OntoProcED.

$cursoEnAsignacion(?ae, c?) \wedge estudianteEnAsignacion(?ae, e?) \wedge$
 $AsignacionEstudianteCurso(?ae) \wedge p1:courseOffering(?gr, ?c?) \wedge p1:agentId(?gr?, e?)$
 $\wedge numMatricula(?gr, m?) \rightarrow p1:Id(?gr, ?m)$

Regla 90

(Rule-Transf-EstadoatriculaAsignacionEstudianteCursoToStatusCourseGradeRecord):

La propiedad status de la clase CourseGradeRecord de OKI tiene correspondencia con la propiedad estadoMatriucla de la clase AsignacionEstudianteCurso de OntoProcED.

$cursoEnAsignacion(?ae, c?) \wedge estudianteEnAsignacion(?ae, e?) \wedge$
 $AsignacionEstudianteCurso(?ae) \wedge p1:courseOffering(?gr, ?c?) \wedge p1:agentId(?gr?, e?)$
 $\wedge estadoMatricula(?gr, m?) \rightarrow p1:status(?gr, ?m)$

Regla 91 (Rule-Transf-AsignacionEstudianteCursoToEnrollmentRecord):

La clase EnrollmentRecord tiene correspondencia con la clase

AsignacionEstudianteCurso de OntoProcED.

$cursoEnAsignacion(?ae, c?) \wedge esudianteEnAsignacion(?ae, e?) \wedge$
 $AsignacionEstudianteCurso(?ae) \wedge p1:courseOffering(?er, ?c?) \wedge p1:student(?gr?, e?)$
 $\rightarrow CourseGradeRecord(?er)$

Regla 92

(Rule-Transf-CalificacionFinalAsignacionEstudianteCursoToCourseGradeEnrollmentRecord):

La propiedad courseGrade de la clase EnrollmentRecord de OKI tiene correspondencia con la propiedad calificacionFinal de la clase AsignacionEstudianteCurso de

OntoProcED. $cursoEnAsignacion(?ae, c?) \wedge estudianteEnAsignacion(?ae, e?) \wedge$
 $AsignacionEstudianteCurso(?ae) \wedge p1:courseOffering(?er, ?c?) \wedge p1:student(?gr?, e?)$
 $\wedge calificacionFinal(?gr, f?) \rightarrow CourseGradeRecord(?gr, f?)$

Regla 93 (Rule-Transf-asignaturaEquivalenteToequivalentCourses):

La propiedad de objeto *equivalentCourses* de la ontología OKI tiene correspondencia con la propiedad *asignaturaEquivalente* de *OntoProcED*. $asignaturaEquivalente(?x, ?y) \rightarrow equivalentCourses(?x, ?y)$.

Regla 94 (Rule-Transf[Rule-Transf-incluyeAsignaturaToCanonicalCourse):

La propiedad de *courses* de la clase *CourseGroup* de la ontología OKI tiene correspondencia con la propiedad *incluyeAsignatura* de *OntoProcED*. $incluyeAsignatura(?t, ?a) \wedge p1:CanonicalCourse(?c) \wedge p1:CourseGroup(?t) \rightarrow p1:courses(?t, ?c)$

Regla 95 (Rule-cursoAcademicoToTerm):

La clase *Term* de OKI tiene correspondencia con la clase *CursoAcademico* de *OntoProcED*. $CursoAcademico(?x) \rightarrow p1:Term(?x)$

Regla 96 (Rule-Transf[Rule-Transf-esDeUnaAsignaturaTocanonicalCourse):

La propiedad *canonicalCourse* de la ontología OKI tiene correspondencia con la propiedad *esDeUnaAsignatura* de *OntoProcED*. $esDeUnaAsignatura(?x, ?y) \rightarrow p1:canonicalCourse(?x, ?y)$

Regla 97 (Rule-Transf[Rule-Transf-coursesToCourseOfferings):

La propiedad *courseOfferings* de la ontología OKI tiene correspondencia con la propiedad *courses* de *OntoProcED*. $courses(?x, ?y) \rightarrow p1:courseOfferings(?x, ?y)$

Regla 98 (Rule-Transf[Rule-Transf-coursesToCourseOfferings):

La propiedad *courseOffering* asociada a la clase *CourseSeccion* de la ontología OKI tiene correspondencia con la propiedad *esDeunCurso* de *OntoProcED*. $courses(?x, ?y) \rightarrow p1:courseOfferings(?x, ?y)$

Regla 99 (Rule-Transf[Rule-Transf-clasesTocourseSections):

La propiedad *courseSections* asociada a la clase *CourseSeccion* de la ontología OKI tiene correspondencia con la propiedad *clases* de *OntoProcED*. $courses(?x, ?y) \rightarrow p1:courseOfferings(?x, ?y)$

Regla 100 (Rule-Transf[Rule-Transf-evaluacionCursoTocourseGradeType):

La propiedad *courseGradeType* de la ontología OKI tiene correspondencia con la propiedad *evaluación curso* de *OntoProcED*. $evaluacionCurso(?x, ?y) \wedge p1:courseGradeType(?x, ?y)$

Regla 101 (Rule-Transf[Rule-estudianteQueEntregaToAgentId):

La propiedad *agentId* de la clase *AssessmentTaken* de la Ontología OKI tiene correspondencia con la propiedad *estudianteQueEntrega* de *OntoProcED*. $estudianteQueEntrega(?x, ?y) \wedge p1:AssessmentTaken(?x) \rightarrow p1:agentId(?x, ?y)$

Regla 102 (Rule-Transf-cursoAsignadotocourseOffering):

La propiedad *courseOffering* de la clase *EnrollmentRecord* de la Ontología OKI tiene correspondencia con la propiedad *cursoAsignado* de *OntoProcED*. $cursoAsignado(?e, ?ae) \wedge AsignacionEstudianteCurso(?ae) \wedge p1:EnrollmentRecord(?x) \wedge curso(?ae, ?c) \wedge student(?x, ?e) \rightarrow p1:courseOffering(?x, ?e)$

Regla 103 (Rule-Transf-estudianteAsignadotostudent):

La propiedad student de la clase EnrollmentRecord de la Ontología OKI tiene correspondencia con la propiedad estudianteAsignado de OntoProcED.

estudianteAsignado(?e, ?ae) ∧ AsignacionEstudianteCurso(?ae) ∧ p1:EnrollmentRecord(?x) ∧ p1:courseOffering(?x, ?e) → student(?ae, ?c)

Regla 104 (Rule-Transf-materialDelCursoToassets):

La propiedad assets de la clase CourseOffering de la Ontología OKI tiene correspondencia con la propiedad materialDelCurso de OntoProcED.

materialDelCurso(?x, ?y) → p1:assets(?x, ?y)

Regla 105 (Rule-actividadCursoToexternalReferenceId):

La propiedad externalReferenceId de la clase CourseSection asociada a la clase CourseOffering de la Ontología OKI tiene correspondencia con la propiedad actividadDelCurso de OntoProcED.

actividadCurso(?c, ?a) ∧ p1:CourseOffering(?c) ∧ p1:courseSections(?c, ?cl) → p1:externalReferenceId(?cl, ?a)

Regla 106 (Rule-actividadDeLaEntregaToAssessmentTaken):

La clase AssessmentTaken de la Ontología OKI tiene correspondencia con la propiedad actividadDeLaEntrega de OntoProcED.

actividadDeLaEntrega(?e, ?a) ∧ p1:courseSectionId(?e, ?cl) ∧ p1:courseOffering(?cl, ?c) → p1:AssessmentTaken(?a)

Bibliografía

- ADL (2004), Sharable Content Object Reference Model Version 1.3.1, Technical report, Advanced Distributed Learning. note: Conformance Requirements available at: <http://www.adlnet.org> [última consulta: 28/01/2007].
- Amorim, R., Lama, M., Sánchez, E., Riera, A. & Vila, X. A. (2006), 'A Learning Design Ontology based on the IMS Specification', *Educational Technology & Society* **9**(1), 38–57.
- Anido, L., Fernández, M., Caeiro, M. & Santos, J. (2002), 'Educational metadata and brokerage for learning resources', *Computers & Education* **38**(4), 351–374.
- Avgeriou, P., Papasalouros, A., Retalis, S. & Skordalakis, M. (2003), 'Towards a Pattern Language for Learning Management Systems', *Educational Technology & Society* **6**(2), 11–24.
- Badakchani, H. (2004), 'Introduction to RosettaNet'. <http://dev2dev.bea.com/pub/a/2004/12/RosettaNet.html> [última consulta: 1/5/2008].
- Barros, B., Verdejo, M., Read, T. & Mizoguchi, R. (2002), Applications of a Collaborative Learning Ontology, in 'Micai 2002: Advances in Artificial Intelligence: Second Mexican International Conference on Artificial Intelligence', Springer, pp. 301–310.
- Bates, A. & Poole, G. (2003), *Effective Teaching with Technology in Higher Education. Foundations for Success*, Jossey-Bass, cop.
- Beeners-Lee, T., Hendler, J. & Lassila, O. (2003), 'The semantic web', *Scientific American* **284**(5), 34–43.

- Bucchiarone, A. & Gnesi, S. (2006), A Survey on Services Composition Languages and Models, *in* A. Bertolino & A. Polini, eds, 'Proceedings of International Workshop on Web Services Modelling and Testing (WS-MaTe2006)', pp. 51–63.
- Carlson, P. (1998), 'Advanced Educational Technologies – Promise and Puzzlement', *Journal of Universal Computer Science* **4**(3), 210–215.
- Carroll, J. (1995), *Scenario-based design: envisioning work and technology in system development*, John Wiley & Sons, Inc. New York, NY, USA.
- Coalition, W. M. (1997), *Workflow Handbook*, John Wiley and Sons, New York.
- Dean, M., Schreiber, G., Bechhofer, S., Harmelen, F. V., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. & Stein, L. A. (2004), 'OWL Web Ontology Language Reference', *W3C recommendation* **10**. <http://www.w3.org/TR/owl-ref/> [última consulta: 11/11/2009].
- Drucker, P. (2000), Need to Know: Integrating e-Learning with High Velocity Value Chains, Technical report, Delphy Group.
- Duart, J. M., Salomón, L. & Lara, P. (2006), 'La Universitat Oberta de Catalunya (UOC): innovación educativa y tecnológica en educación superior', *RIED.Revista iberoamericana de educación a distancia* **9**(1-2), 315–344.
- Eap, T., Htala, M. & Richards, G. (2004), Digital Repository Interoperability: Design, Implementation and Deployment of ECL Protocol and Connecting Middleware, *in* 'Proceedings of the 13th International World Wide Web Digital Repository Interoperability', ACM Press, pp. 19–27.
- Fahey, L. & Randall, R. M. (1997), *What is scenario learning?*, Learning from the Future: Competitive Foresight Scenarios, Wiley, pp. 3–21.
- Farance, F. & Tonkel, J. (2001), *Draft Standard for Learning Technology – Learning Technology Systems Architecture (LTSA)*, IEEE Learning Technology Standards Committee.

- Friesen, N. (2004), Final Report on the International LOM Survey, Technical report, Technical Report Document 36C087.
- Friesen, N. (2005), 'Interoperability and Learning Objects: An overview of E-Learning Standarization', *Interdisciplinary Journal of Knowledge and Learning Objects* **1**.
- Gruber, T. R. (1993), 'A translation approach to portable ontology specifications', *Knowledge Acquisition* **5**(2), 199–220.
- Guarino, N. (1998), Formal ontology in information systems, *in* 'Formal Ontology in Information Systems: Proceedings of the First International Conference (FOIS'98)', IOS press, Amsterdam, pp. 3–15.
- Helic, D. (2007), 'Formal Representations of Learning Scenarios: A Methodology to Configure E-Learning Systems', *Journal of Universal Computer Science* **13**(4), 504–530.
- Helic, D., Hrastnik, J. & Maurer, H. (2005), An Analysis of Application of Business Process Management Technology in E-Learning Systems, *in* 'Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (e-Learn 2005)', pp. 2937–2942.
- Horrocks, I., F., P., Boley, H., Tabet, S., Grosz, B. & Dean, M. (2004), 'SWRL: A Semantic Web Rule Language Combining OWL and RuleML', *W3C recommendation* **10**. <http://www.w3.org/Submission/SWRL/> [última consulta: 26/04/2010].
- Ikeda, M., Hoppe, U. & Mizoguchi, R. (1995), Ontological Issues of CSCL Systems Design, *in* 'Proceedings of the 7th World Conference on Artificial Intelligence in Education', pp. 16–19.
- IMS (2003a), IMS Digital Repositories Interoperability-Core Functions Information Model Version 1. Final Specification, Technical report, IMS Global Consortium.
- IMS (2003b), IMS Learning Design Information Model. Version 1.0. Final Specification, Technical report, IMS Global Consortium, Inc.

- Inaba, A., Tamura, T., Ohkubo, R., Ikeda, M., Mizoguchi, R. & Toyoda, J. (2001), Design and Analysis of Learners' Interaction based on Collaborative Learning Ontology, *in* 'Proceedings of the 1st European conference on computer-supported collaborative learning (Euro-CSCL'2001)', Maastricht, The Netherlands, pp. 308–315.
- Kabel, S., Wielinga, B. & de Hoog, R. (1999), Ontologies for Indexing Technical Manuals for Instruction., *in* 'AI-ED (Artificial Intelligence in Education) conference. Proceedings workshop on ontologies for intelligent educational systems', pp. 44–53.
- L'Allier, J. (1997), Frame of Reference: NETg's Map to the Products, Their Structure and Core Beliefs, Technical report, NetG. Whitepaper.
- Learning Technology newsletter* (2004), Vol. 6, IEEE Computer Society Technical Committee on Learning Technology (LTTC). note: lctf.ieee.org/learn_tech/issues/january2004/learn_tech_january2004.pdf [última consulta: 26/01/2007].
- LTSC WG12 (2002), Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, Technical report, IEEE Learning Technology Committee.
- Lytras, M., Tsilira, A. & Themistocleous, M. (2003), Towards the semantic e-learning: an ontological oriented discussion of the new research agenda in e-learning, *in* 'Proceedings of the Ninth American on Information Systems', pp. 2985–2997.
- Macarro, G., Arroyo, S., Sicilia, M. A. & Armas, R. (2006), Semantic orchestrated interactions in learning technology, *in* 'Proceedings of the 9th Intl Conf. on Interactive Computer aided Learning (ICL 2006)', Villach, Austria, pp. 26–29.
- McIlraith, S. & Zeng, T. C. (2001), 'Semantic web services', *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]* **16**(2), 46–53.
- Mendling, J. & Neumann, G. (2005), *A Comparison of XML Interchange Formats for Business Process Modelling*, Future Strategies Inc., Florida, USA.

- Mizoguchi, R., Sinita, K. & Ikeda, M. (1996), Task Ontology Design for Intelligent Educational/Training Systems, in 'Position Paper for ITS 96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs', Vol. 96, pp. 1–21.
- Moore, M. & Kearsley, G. (1996), *Distance Education: A system View.*, Wadsworth Publishing Company.
- Najjar, J., Ternier, S. & Duval, E. (2002), The Actual Use of Metadata in ARIADNE: An Empirical Analysis, p. 16.
- Noy, N. & McGuinness, D. L. (2001), 'Ontology Development 101: A Guide to Creating Your First Ontology. 2001', *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001 880*.
- OMG & BPML.org (2006), Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification. dtc/06-02-01, Technical report, Object Management Group (OMG).
- Pagés, C., Sicília, M. A., García-Barriocanal, E., Martínez, J. J. & Gutiérrez, J. M. (2003), On the Evaluation of Completeness Of Learning Object Metadata in Open Repositories, in 'Proceedings of the Second International Conference on Multimedia and Information & Communication Technologies in Education (Mi.ICTE 2003)', Vol. 3, Seville, Spain, pp. 1760–1764.
- Paulsen, M. (2002), 'Online Education Systems: Discussion and Definition of Terms', *NIK Distance Education*. Retrieved July 17, 2004.
- Pearsall, J. & Hanks, P. (1998), *The new Oxford dictionary of English*, Clarendon Press, Oxford.
- Peltz, C. (2003), 'Web Services Orchestration and Choreography', *Computer* **36**(10), 46–52.

- Polsani, P. R. (2003), 'Use and Abuse of Reusable Learning Objects', *Journal of Digital Information* **3**(4).
- Quinn, C. & Hobbs, S. (2002), 'Learning Objects and Instructional Components', *Educational Technology and Society* **3**(2).
- Rius, A., Conesa, J. & Gañán, D. (2010), A DSL Tool to assist specifications of educational settings, in 'The 2010th International Conference on Semantic Web and Web Services'. To be published.
- Rius, A., Santanach, F., Almirall, M., Conesa, J. & García-Barriocanal, E. (2010), 'An Open and Service-Oriented Architecture to Support the Automation of Learning Scenarios', *International Journal of Instructional Technology and Distance Learning* . To be published.
- Rius, A., Sicília, M. A. & García-Barriocanal, E. (2007a), Justificación y Descripción del Dominio de Conocimiento de una Ontología para la Formalización y Automatización de Escenarios Educativos, in 'IV Simposio Pluridisciplinar sobre Diseño, Evaluación y Desarrollo de Contenidos Educativos Reutilizables (SPDECE 2007)'. .
- Rius, A., Sicilia, M. A. & García-Barriocanal, E. (2008a), 'An Ontology to Automate Learning Scenarios? An Approach to its Knowledge Domain', *Interdisciplinary Journal of Knowledge and Learning Objects* **4**, 151–165.
- Rius, A., Sicília, M. A. & García-Barriocanal, E. (2008b), 'Towards Automated Specifications of Scenarios in Enhanced Learning Technology', *Int. J. of Web-Based Learning and Teaching Technologies* **3**(1), 68–78.
- Rius, A., Sicília, M. & García-Barriocanal, E. (2007b), A catalogue of primitive scenario-types. The first step to the automation of learning scenarios, in 'Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (e-Learn 2007)', pp. 6954–6959.
- Rius, A., Sicília, M., García-Barriocanal, E. & Macarro, G. (2006), Beyond contents and activities: specifying processes in learning technology, in 'Proceedings of the

Fourth International Conference on Multimedia and Information and Communication Technologies in education (M-ICTE 2006) Vol 1.', pp. 301–305.

Rodríguez, M. E., Serra, M., Cabot, J. & Guitart, I. (2006), Evolution of the Teacher Roles and Figures in E-learning Environments, *in* 'The 6th IEEE International Conference on Advanced Learning Technologies ICALT (06)', pp. 512–514.

RosettaNet (2001), Rosettanet implementation framework: Core specification, Technical report.

Serra, M., Huertas, M. A., Rius, A., Rodríguez, M. E. & Santamaria, E. (2005), Nuevos paradigmas de la educación: roles de acción docente en el entorno virtual de la UOC, *in* 'Actas de I Simposio Nacional de Tecnologías de la Información y de las Comunicaciones en la Educación (SINTICE 2005)', pp. 45–52.

Sheth, A. & Larson, J. (1990), 'Federated database systems for managing distributed, heterogeneous, and autonomous databases', *ACM Computing Surveys (CSUR)* **22**(3), 183–236.

Sicília, M. (2007), On the general structure of ontologies of instructional models, *in* 'IV Simposio Pluridisciplinar sobre Diseño, Evaluación y Desarrollo de Contenidos Educativos Reutilizables (SPDECE 2007)'.

Sicília, M. A. & García-Barriocanal, E. (2003), 'On the concepts of usability and reusability of learning objects', *International Review of Research in Open and Distance Learning* **4**(2).

Sicília, M. A., García-Barriocanal, E., Sánchez, S., Rius, A. & Pagés, C. (2004), Specifying semantic conformance profiles in reusable learning object metadata, *in* 'Proceedings of the Fifth International Conference on Information Technology Based Higher Education and Training, 2004 (ITHET 2004)', pp. 93–97.

Sicília, M. & García-Barriocanal, E. (2005), 'On the convergence of formal ontologies and standardized e-learning. journal of distance education technologies', *Journal of Distance Education Technologies* **3**(2), 12–28.

- Sicília, M., García-Barriocanal, E., Pagés, C., Martínez, J. & Gutiérrez, J. (2004), 'Complete Metadata Records in Learning Object Repositories: Some Evidence', *International Journal of Learning Technology* **1**(4), 411–424.
- Sicília, M. & Lytras, M. (2005), 'Scenario-oriented reusable learning object characterisations', *International Journal of Knowledge and Learning* **1**(4), 332–341.
- Simon, B., Massart, D., Assche, F. V., Ternier, S. & Duval, E. (2005), 'Simple Query Interface Specification', *Draft CEN Workshop Agreement (CWA). Version 1*, 11–18.
- Stojanovic, L., Staab, S. & Studer, R. (2001), eLearning based on the Semantic Web, *in* 'In Proceedings of the World Conference on the WWW and Internet (Webnet, 2001)', Orlando, Florida, USA, pp. 23–27.
- Toffolon, C. (2006), Learning Management System Scenario-Based Engineering, *in* D. Remenyi, ed., 'Proceedings of the 5th European Conference on e-Learning (ECEL 2006)', Academic Conferences Limited, pp. 397–406.
- Tood, C., Klagholz, L., Sheschter, E., Doolan, J., Jensen, J. & Nagler, I. (1999), *Learning scenarios*, New Jersey World Languages Curriculum Framework. A Document in support of the Core Curriculum Content Standards for World Languages, New Jersey State Department of Education, New Jersey State, pp. 75–81.
- Uchitel, S. (2003), 'Incremental Elaboration of Scenario-Based Specifications and Behaviour Models Using Implied Scenarios'.
- UN/CEFACT & OASIS (2001), ebXML Business Process Specification Schema Version 1.01, Technical report.
- Van der Aalst, W., Hofstede, A. H., Kiepuszewski, B. & Barros, A. (2003), 'Workflow Patterns', *Distributed and Parallel Databases* **14**(1), 5–51.
- Van der Aalst, W., Hofstede, A. H. & Weske, M. (2003), 'Business Process Management: A Survey', *LNCS* **2678**, 1–12.

- Vogten, H., Martens, H., Nadolski, R., P., C. R. T. & Koper, R. (2006), CopperCore Service Integration - Integrating IMS Learning Design and IMS Question and Test Interoperability, *in* 'Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies(ICALT '06)', IEEE Computer Society, Washington, DC, USA, pp. 378–382.
- Wagner, E. D. (2000), E-learning: where cognitive strategies, knowledge management, and information technology converge, *in* D. Brightman, ed., 'Emerging strategies for effective e-learning solutions', Vol. 3 of *Learning without limits*, Informania Inc., 444 De Haro Street, Suite 128, San Francisco, California 94107, pp. 9–19.
- White, S. (2004), 'Introduction to BPMN', *IBM Cooperation* .
- Wiley, D. A. (2002), *Connecting learning objects to instructional design theory: A definition, a metaphor and a taxonomy*, The instructional use of learning objects, Agency for Instructional Technology and Association for Educational Communications and Technology, Bloomington, Indiana, pp. 3–24.