

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Informática
Departamento de Lenguajes y Sistemas Informáticos



Tesis Doctoral

Una Arquitectura Distribuida basada en
Agentes Software para el Desarrollo y el
Soporte de Espacios de Aprendizaje Ubicuos

Carlos Celorrio Aguilera
Ingeniero Superior Informático

2010

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Informática
Departamento de Lenguajes y Sistemas Informáticos



Una Arquitectura Distribuida basada en Agentes Software para el Desarrollo y el Soporte de Espacios de Aprendizaje Ubicuos

Carlos Celorrio Aguilera

Ingeniero Superior Informático por la Universidad Complutense de Madrid

Directora

M^a Felisa Verdejo Maíllo

*Catedrática de Universidad del Departamento de Lenguajes y Sistemas
Informáticos de la Universidad Nacional de Educación a Distancia*

2010

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi directora de tesis, Felisa, su magnífica ayuda durante el desarrollo de esta tesis y en general darla las gracias por su apoyo a lo largo de todos estos años que he tenido el privilegio de trabajar con ella. Ha sido excepcional contar con alguien de su prestigio tanto a nivel nacional e internacional y poder aprender junto a ella los entresijos del mundo de la investigación.

Por otro lado, quiero agradecer también a Paco su ayuda prestada con la plataforma ICARO-T y por las interesantes y productivas largas charlas sobre agentes e ingeniería del software con las que me ha obsequiado, gracias a las cuales ha sido posible orientar una gran parte de este trabajo.

No puedo olvidarme en estos agradecimientos de todos mis compañeros de trabajo; Beti, Nacho, Javi, Emilio, Marta, Tim, Yoli, Miguel... y en definitiva a todos los miembros de grupo de investigación LTCS. También mencionar a todos los compañeros y amigos del departamento LSI, especialmente a los incansables deportistas y a los integrantes de la LSI Big Band. Gracias a todos por los buenos ratos que he pasado junto a vosotros durante el desarrollo de esta tesis.

Finalmente, y no menos importante, me gustaría dedicar esta tesis por su incondicional apoyo a mi familia, a mis padres y hermanas, a todos mis amigos (incluso a aquellos que nunca llegaron a creer que la terminaría), y por supuesto a mi novia, Claudia. Te quiero, pequeñita.

Este trabajo de investigación ha sido realizado dentro del proyecto ENLACE (TIN2004-04232), gracias a la financiación del Ministerio de Ciencia y Tecnología y a la beca FPI asociada (BES-2005-7178), de la que ha disfrutado su autor.

RESUMEN

En este trabajo de investigación se explora y desarrolla toda una arquitectura que da soporte a nuevos paradigmas tecnológicos educativos emergentes de carácter colaborativo, ubicuo y distribuido buscando diseñar una serie de modelos computacionales que los sustenten.

Para este desarrollo se ha utilizado como estrato tecnológico el paradigma de sistemas multi-agente para dar soporte al aprendizaje que use dispositivos móviles y sistemas ubicuos en general. El nombre que recibe esta arquitectura es DAEDALUS (*Distributed Agent-based EDucational Architecture for Learning using Ubiquitous Systems*).

DAEDALUS está diseñado como un sistema multi-agente. Los agentes son capaces de observar el entorno de aprendizaje, analizarlo y actuar en consecuencia. La arquitectura de DAEDALUS está basada en ICARO-T, una plataforma para el desarrollo de sistemas multi-agente.

La arquitectura DAEDALUS se estructura a lo largo de varios niveles (Nivel de Infraestructura, Nivel de Semántica y Contexto, Nivel de Organización Educativa y Nivel de Interacción) y de forma general aporta:

1. Una plataforma software basada en agentes capaz de funcionar en un rango heterogéneo de dispositivos, incluidos dispositivos móviles.
2. Un modelo de organización de agentes de carácter completamente distribuido y con capacidad de adaptarse dinámicamente a distintas topologías de red.
3. Una ontología de conceptos comunes a todos los agentes de la organización educativa que incluye una taxonomía de eventos y un modelo de contexto.

4. Una representación computacional de dichos modelos que permita el intercambio de información y conocimiento entre todos los dispositivos de la infraestructura.
5. Un conjunto de agentes especializados que aporten servicios educativos inteligentes, básicos para el desarrollo y el apoyo de actividades de aprendizaje.

Una idea principal empleada a lo largo de este trabajo consiste en integrar el proceso de desarrollo de la propuesta en torno a un diseño basado en escenarios. Los escenarios son específicos, incluyen a las personas dentro de ellos y están situados en el mundo real. Mediante el empleo de escenarios durante el desarrollo es posible enfocar el diseño hacia detalles que precisan mayor atención desde el punto de vista de los usuarios.

ABSTRACT

This research work explores and develops a whole new architecture that supports new emerging educational technology paradigms of collaborative, ubiquitous and distributed nature, in order to design a series of computer models that enables them.

This development has been carried out using a multi-agent system paradigm as the technological layer for supporting student learning using mobile and ubiquitous systems in general. The name given to this architecture is DAEDALUS (Distributed Agent-based Architecture for Educational using Ubiquitous Learning Systems).

DAEDALUS is designed as a multi-agent system — agents are able to observe the learning environment, analyze it and act accordingly. DAEDALUS architecture is based on ICARO-T, a platform for the development of multi-agent systems.

DAEDALUS architecture is structured over several levels (Level of Infrastructure, Level of Semantics and Context, Level of Educational Organization and Level of Interaction) and generally provides:

1. An agent-based software platform capable of operating in a heterogeneous range of devices, including mobile devices.
2. A model of organization fully distributed and capable of dynamically adapting to different network topologies.
3. An ontology of concepts common to all actors in the educational organization which includes a taxonomy of events and a context model.

4. Computational representation of these models to enable the exchange of information and knowledge between all the devices of infrastructure.
5. A set of specialized agents to provide basic intelligent educational services for the development and support of learning activities.

A main idea used throughout this work is to integrate the development process around a scenario-based design. Scenarios are specific, include people in them and are located in the real world. Through the use of scenarios during development is possible to approach the design details that require further attention from the user's point of view.

CONTENIDOS

PARTE I	INTRODUCCIÓN	1
1	PREFACIO Y MOTIVACIÓN	3
1.1	OBJETIVOS	6
1.2	MARCO DE DESARROLLO	8
1.3	METODOLOGÍA	10
1.4	ESTRUCTURA DEL DOCUMENTO	11
2	ESTADO DEL ARTE	13
2.1	TEORÍAS Y EVOLUCIÓN DE LAS TECNOLOGÍAS EDUCATIVAS	14
2.2	DISEÑO Y EVALUACIÓN DE TECNOLOGÍAS EDUCATIVAS	17
2.2.1	<i>Diseño basado en Escenarios</i>	17
2.2.2	<i>Evaluación Formativa</i>	21
2.3	APRENDIZAJE COLABORATIVO	22
2.3.1	<i>Objetos de Aprendizaje Emergentes</i>	23
2.3.2	<i>Repositorios de Objetos de Aprendizaje</i>	26
2.4	APRENDIZAJE MÓVIL	28
2.4.1	<i>Hacia el Aprendizaje Ubicuo</i>	30
2.5	EL PROYECTO ENLACE	32
2.5.1	<i>Actividades de Aprendizaje de ENLACE</i>	33
2.5.2	<i>La Infraestructura Tecnológica de ENLACE</i>	34
2.5.3	<i>El Repositorio de Objetos de Aprendizaje de ENLACE</i>	35
2.6	SISTEMAS ADAPTADOS AL CONTEXTO	36
2.6.1	<i>Modelado del Contexto</i>	37
2.6.2	<i>Sistemas Educativos Adaptados al Contexto</i>	39
2.7	TECNOLOGÍA DE AGENTES	41

2.7.1	<i>Agentes Inteligentes</i>	41
2.7.2	<i>Agentes Inteligentes en la Educación</i>	44
2.7.3	<i>Sistemas Multi-Agente y Plataformas</i>	45
2.8	LA PLATAFORMA ICARO-T	47
PARTE II SOLUCIÓN PROPUESTA		51
3	HACIA UNA ARQUITECTURA DISTRIBUIDA PARA EL APRENDIZAJE UBICUO	53
3.1	ANÁLISIS DE REQUISITOS	54
3.1.1	<i>Escenario de actividad de aprendizaje en ENLACE</i>	57
3.1.2	<i>Comentarios acerca del escenario</i>	59
3.1.3	<i>Especificación de requisitos para el diseño</i>	60
3.2	DECISIONES DE DISEÑO Y DESARROLLO	62
3.2.1	<i>Elecciones de diseño</i>	62
3.2.2	<i>Casos de Uso</i>	63
3.2.3	<i>Solución propuesta</i>	65
4	LA ARQUITECTURA DAEDALUS	67
4.1	ESPACIOS DE APRENDIZAJE UBICUOS	68
4.1.1	<i>Interconexión de espacios de aprendizaje ubicuos</i>	69
4.1.2	<i>Estructura de un espacio de aprendizaje ubicuo</i>	70
4.2	ARQUITECTURA GENÉRICA	73
4.2.1	<i>Estratificación de la Arquitectura Genérica</i>	75
4.3	CONCEPTUALIZACIÓN GLOBAL DE LA ARQUITECTURA DAEDALUS	78
4.3.1	<i>Elementos del Nivel de Infraestructura</i>	78
4.3.2	<i>Elementos del Nivel de Semántica y Contexto</i>	80
4.3.3	<i>Elementos del Nivel de Organización Educativa</i>	81
4.3.4	<i>Elementos del Nivel de Interacción</i>	82
5	NIVEL DE INFRAESTRUCTURA	85
5.1	ARQUITECTURA ORIENTADA A AGENTES	86
5.1.1	<i>Sistemas complejos multi-agente distribuidos</i>	87
5.2	LA ORGANIZACIÓN EN DAEDALUS	89
5.2.1	<i>Acercando ICARO-T a los requerimientos de DAEDALUS</i>	91
5.3	LOS AGENTES EN DAEDALUS	96
5.3.1	<i>Patrón Agente (Micro-)Reactivo</i>	98
5.3.2	<i>Patrón Agente Cognitivo</i>	100
5.4	COMUNICACIÓN EN DAEDALUS	103
5.4.1	<i>Envío y recepción de mensajes</i>	104

5.4.2	<i>Subscripción y distribución de eventos.</i>	105
5.4.3	<i>Descubrimiento dinámico de nodos</i>	106
5.4.4	<i>Registro de agentes y recursos</i>	106
6	NIVEL DE SEMÁNTICA Y CONTEXTO	109
6.1	REPRESENTACIÓN DEL CONOCIMIENTO DE DAEDALUS	110
6.1.1	<i>Ontologías</i>	111
6.1.2	<i>Integración con otras ontologías y extensibilidad</i>	112
6.1.3	<i>Agentes como manipuladores de contexto distribuido</i>	113
6.1.4	<i>Difusión del conocimiento: modelo estático y modelo dinámico</i>	116
6.2	DOMINIO DE CONTEXTO	118
6.2.1	<i>Declaraciones de contexto</i>	119
6.2.2	<i>Entidades y relaciones de contexto</i>	120
6.2.3	<i>Dimensiones del contexto</i>	121
6.3	TAXONOMÍA DE SUCESOS DE CONTEXTO	129
6.3.1	<i>Clasificación de sucesos de contexto</i>	130
7	NIVEL DE ORGANIZACIÓN EDUCATIVA	137
7.1	SERVICIOS PEDAGÓGICOS INTELIGENTES Y AGENTES DE LA ORGANIZACIÓN EDUCATIVA	138
7.1.1	<i>Sistematización de SPIs en DAEDALUS</i>	139
7.1.2	<i>Clasificación de agentes pedagógicos en DAEDALUS</i>	140
7.2	DISEÑO DE SERVICIOS PEDAGÓGICOS INTELIGENTES COMO AGENTES	149
7.2.1	<i>Modelando responsabilidades dentro de los agentes pedagógicos</i>	151
8	NIVEL DE INTERACCIÓN	159
8.1	ELEMENTOS DEL NIVEL DE INTERACCIÓN	160
8.2	RECURSOS DE INTERACCIÓN	161
8.2.1	<i>Recursos de Interfaces de Usuario</i>	161
8.2.2	<i>Recursos de Herramientas de Aprendizaje</i>	162
8.3	AGENTES DE INTERACCIÓN	162
8.3.1	<i>Agentes de Usuario</i>	163
8.3.2	<i>Agentes de Aplicación</i>	163
	PARTE III APLICACIÓN Y RESULTADOS	165
9	LA ARQUITECTURA DAEDALUS Y EL ESCENARIO ENLACE	167
9.1	INFRAESTRUCTURA PARA EL ESCENARIO ENLACE	168
9.2	SPIs Y RESPONSABILIDADES EN EL ESCENARIO ENLACE	168
9.2.1	<i>Agentes Pedagógicos para el escenario ENLACE</i>	170

9.2.2	<i>Recursos para el escenario de ENLACE</i>	177
9.3	PUESTA EN MARCHA: SALIDA AL MONTE DEL PARDO	179
9.3.1	<i>Arranque del Nodo Servidor</i>	179
9.3.2	<i>Arranque del Nodo Móvil</i>	180
9.3.3	<i>Caso de Prueba 1: Administración de la infraestructura distribuida</i>	183
9.3.4	<i>Caso de Prueba 2: Fomento contextualizado de la colaboración</i>	187
9.3.5	<i>Caso de Prueba 3: Regulación y Control sobre la Actividad</i>	189
9.3.6	<i>Caso de Prueba 4: Reconfiguración eventual durante Actividades</i>	190
10	LA ARQUITECTURA DAEDALUS FRENTE A UN NUEVO ESCENARIO	191
10.1	DEFINICIÓN DEL ESCENARIO 'AULA INTELIGENTE'	192
10.1.1	<i>Descripción del escenario</i>	194
10.1.2	<i>Estructuración del Aula</i>	195
10.1.3	<i>Diseño de los procesos pedagógicos e interacción</i>	197
10.2	DAEDALUS EN EL 'AULA INTELIGENTE'	204
10.2.1	<i>Ampliación del Nivel de Semántica y Contexto</i>	205
10.2.2	<i>Diseño de la Organización Educativa</i>	207
10.2.3	<i>Implementación de la Organización Educativa</i>	217
10.3	PUESTA EN MARCHA DEL ESCENARIO	219
11	DISCUSIÓN FINAL Y CONCLUSIONES	223
11.1	RECOMENDACIONES DE APLICACIÓN Y EXTENSIBILIDAD	224
11.2	CONTRIBUCIONES	226
11.3	CONCLUSIONES	228
11.4	TRABAJO FUTURO	230
	REFERENCIAS	233
	PARTE IV APÉNDICES	249
	APÉNDICE A. IMPLEMENTACIÓN	251
	APÉNDICE B. ONTOLOGÍA OWL/RDF	255
	APÉNDICE C. DIAGRAMAS AGENTES PEDAGÓGICOS	269
	APÉNDICE D. PUBLICACIONES	277

LISTA DE FIGURAS

Figura 2.1 El aprendizaje en función de la movilidad y la integración	30
Figura 2.2 Flujo de actividades de ENLACE.....	33
Figura 4.1 Elementos presentes en un Espacio de Aprendizaje Ubicuo	71
Figura 4.2 Niveles de la Arquitectura Genérica	76
Figura 4.3 Elementos principales de cada nivel de la arquitectura	78
Figura 5.1 Esquema de la organización	90
Figura 5.2 Diferencias entre la organización en ICARO-T y la organización en DAEDALUS	92
Figura 5.3 Ejemplo de organización distribuida por distintos escenarios.....	93
Figura 5.4 Topologías de funcionamiento de comunicación	94
Figura 5.5 Tipos de Patrones de Agente	95
Figura 5.6 Patrón Agente (Micro-)Reactivo.....	98
Figura 5.7 Elementos para la definición de agentes (micro-)reactivos	99
Figura 5.8 Patrón Agente Cognitivo	100
Figura 5.9 Procesamiento de la información del agente cognitivo.....	101
Figura 5.10 Elementos para la definición de agentes cognitivos.....	102
Figura 5.11 Envío y recepción de mensajes	104
Figura 5.12 Suscripción y recepción de eventos.....	105
Figura 6.1 Ontología del modelo de conocimiento modelada con Protégé	117
Figura 6.2 Reificación de la declaración de contexto.....	120
Figura 6.3 Nomenclatura del lenguaje gráfico de representación	121
Figura 6.4 Partición del dominio de contexto en DAEDALUS.....	121
Figura 6.5 Dimensión de Artefacto Educativo	123
Figura 6.6 Dimensión de Individuo	124
Figura 6.7 Dimensión de Comunidad	125
Figura 6.8 Dimensión de Proceso Pedagógico	126
Figura 6.9 Dimensión de Espacio-Tiempo	127

Figura 6.10 Dimensión de Infraestructura	129
Figura 6.11 Reificación del suceso de contexto.....	130
Figura 6.12 Clasificación de Sucesos de Contexto	131
Figura 7.1 Clasificación de agentes pedagógicos.....	141
Figura 7.2 Tipos de agentes pedagógicos en DAEDALUS	142
Figura 7.3 Mismo agente, distintas perspectivas	143
Figura 7.4 Del diseño educativo, al diseño con agentes.....	151
Figura 7.5 Esquema general de autómata de agente pedagógico (micro-)reactivo...	153
Figura 7.6 Esquema general de una regla de un agente pedagógico cognitivo	154
Figura 7.7 Esquema de regla de creación de objetivos	155
Figura 7.8 Esquema de regla de focalización de objetivos.....	156
Figura 7.9 Esquema general de resolución de objetivos.....	157
Figura 8.1 Elementos del Nivel de Interacción	160
Figura 9.1 Agentes Pedagógicos para el escenario ENLACE	171
Figura 9.2 Responsabilidades de LearningDataMiningAgent	172
Figura 9.3 Responsabilidades de GlobalContextAgent.....	173
Figura 9.4 Responsabilidades de DeviceManagerAgent.....	174
Figura 9.5 Responsabilidades de DeviceAgent	175
Figura 9.6 Responsabilidades de CollaborationAgent	176
Figura 9.7 Responsabilidades de Personal Agent	177
Figura 9.8 Arranque del Nodo Servidor.....	180
Figura 9.9 Arranque del Nodo Móvil.....	181
Figura 9.10 Interfaz de Usuario de Inicio	182
Figura 9.11 Interfaz de Información al Usuario.....	182
Figura 9.12 Interfaz de Información de recursos para la actividad	183
Figura 9.13 Instalación de herramientas en dispositivos	185
Figura 9.14 Configuración personalizada de dispositivos.....	186
Figura 9.15 Sincronización de Información de Dispositivos.....	187
Figura 9.16 Fomento contextualizado de la colaboración.....	188
Figura 9.17 Regulación y Control sobre la Actividad	189
Figura 9.18 Reconfiguración eventual durante Actividades	190
Figura 10.1 Evolución del Desarrollo del Escenario	193
Figura 10.2 Distribución del aula	196
Figura 10.3 Instalación de los Proyectors	197
Figura 10.4 Flujo de la Actividad.....	198
Figura 10.5 Boceto de Visualización de Grupo.....	199
Figura 10.6 Boceto de Visualización de Clase	200
Figura 10.7 Boceto de la Interfaz de Usuario del Dispositivo Personal durante la Fase de Resolución de Problemas	201

Figura 10.8 Esquema de Relaciones entre los Artefactos del Escenario	203
Figura 10.9 Ampliación del Dominio de Contexto	206
Figura 10.10 Responsabilidades de los Agentes.....	209
Figura 10.11 Responsabilidades del Agente SmartClassroomCoordinator	210
Figura 10.12 Objetivos y Tareas del Agente SmartClassroomCoordinator	211
Figura 10.13 Responsabilidades del Agente MathActivity	212
Figura 10.14 Objetivos y Tareas del Agente MathActivity	213
Figura 10.15 Responsabilidades del Agente MathGroup.....	214
Figura 10.16 Objetivos y Tareas del Agente MathGroup	215
Figura 10.17 Interacción entre los agentes	216
Figura 10.18 Descripción de la Organización para el Escenario	218
Figura 10.19 Recurso de Visualización del Agente GlobalContext	219
Figura 10.20 Desarrollo de la Actividad	220
Figura 10.21 Interfaz de Usuario	220
Figura A.1 Descripción de la arquitectura.....	252

LISTA DE TABLAS

Tabla 3.1 Perfil de los participantes en el escenario salida al campo de ENLACE ...	56
Tabla 6.1 Funciones de agentes de contexto	115
Tabla 6.2 Elementos del espacio de aprendizaje ubicuo asociados a las dimensiones del dominio de contexto.....	122
Tabla 6.3 Sucesos de Artefacto Educativo	132
Tabla 6.4 Sucesos de Individuo	132
Tabla 6.5 Sucesos de Comunidad	133
Tabla 6.6 Sucesos de Proceso Pedagógico	134
Tabla 6.7 Sucesos de Espacio-Tiempo	135
Tabla 6.8 Sucesos de Infraestructura.....	135
Tabla 7.1 Sistematización y ejemplos de SPIs en función de los niveles inferiores ..	140
Tabla 9.1 SPIs y Responsabilidades para el escenario ENLACE	169
Tabla 9.2 Regla de Desarrollo del objetivo DeriveContextInfo del GlobalContextAgent	184
Tabla 10.1 Contenido y Metadatos de los Objetos de Aprendizaje del Escenario ...	204
Tabla 10.2 Sucesos para los Procesos Pedagógicos en el Aula Inteligente	207

PARTE I

Introducción

1

PREFACIO Y MOTIVACIÓN

El constante avance de las tecnologías de la información y la comunicación está convirtiéndose paulatinamente en un catalizador dentro del campo de la educación para la búsqueda de escenarios pedagógicos innovadores. Poco a poco, investigadores y diseñadores instruccionales empiezan a abandonar la exploración de modelos en los cuales el nexo entre la tecnología y el estudiante es una metáfora de la tradicional relación entre el profesor y el alumno, con el fin de dar paso a nuevos paradigmas que explotan aspectos socio-constructivistas, colaborativos, situacionales y contextuales del aprendizaje (Chan et al., 2006).

Si nos paramos a pensar, vivimos actualmente un momento de revolución socio-tecnológica en el cual toda una multitud de dispositivos móviles y embebidos han pasado, casi sin que nos demos cuenta, a formar parte habitual de nuestras vidas. Se trata de toda una amalgama de aparatos con capa-

cidad de proceso equivalente a la de los ordenadores personales de hace tan sólo unos años, pero con el añadido de poder acceder a información de forma inalámbrica y manipularla en cualquier lugar. Teléfonos móviles inteligentes, agendas personales o PDAs, lectores de libros electrónicos, mini-ordenadores ultra portátiles, etc., forman parte de este elenco tecnológico que está presente en nuestro día a día de una forma totalmente nueva: nos acompaña, va con nosotros, se mueve con nosotros, pero lo más importante, está disponible para nuestro uso en cualquier lugar y en cualquier momento. Es por ello que, debido a su portabilidad, estos artilugios están volviendo a redefinir el término *personal* con mucho más significado inclusive que el que tenía en el concepto *ordenador personal* de finales del siglo XX. De hecho esta tecnología *móvil y personal* se está convirtiendo en un elemento tan integrado en nuestras vidas que dentro de poco gracias a la progresiva miniaturización e incorporación en objetos cotidianos se podrá considerar ubicua hasta llegar al punto que nos resultará *invisible* y no notaremos su presencia (Weiser, 1991).

Desde un punto de vista pedagógico, el potencial que tienen todos estos dispositivos móviles permite imaginar nuevas oportunidades dentro de la tecnología aplicada a la enseñanza que no se limitan a la actual interacción de los alumnos con “cajas” encima de la mesa del aula de informática del colegio o de su casa. Partiendo del hecho de que las oportunidades de aprendizaje no son las mismas cuando el alumno está situado en clase, en un museo, visitando una fábrica o en una excursión a una reserva natural, el carácter móvil y ubicuo de estos nuevos aparatos posibilita que los estudiantes puedan aprender en cualquier sitio y momento en el que sientan curiosidad, tanto dentro (Tatar, Roschelle, Vahey, & Penuel, 2003; Roschelle, 2003) como fuera del colegio (Y. Chen, T. C. Kao, & J. P. Sheu, 2003; Rogers et al., 2004; Faux, McFarlane, Roche, & Facer, 2006), y pudiendo promover de este modo *actividades auténticas* situadas en un contexto (John Seely Brown, Collins, & Duguid, 1989). Esto es lo que está dando lugar a lo que se conoce como la teoría del *aprendizaje móvil* (Sharples, Taylor, & Vavoula, 2005).

Por consiguiente, una consecuencia implícita de todas estas ideas es que el contexto en el cual se desarrolla el aprendizaje tiende a ganar inherente importancia. Dado que el aprendizaje puede no desarrollarse en un contexto fijo, es primordial que la tecnología educativa pueda mantener información constantemente actualizada acerca del alumno y los elementos que le rodean, como por ejemplo en qué lugar se encuentra, qué está haciendo, qué personas tiene alrededor, qué dispositivos hay accesibles, cuáles son las condiciones físicas del entorno, etc. Esto promueve la necesidad de construir mecanismos

mediante los cuales poder definir el entorno y hacer referencia a los elementos contextuales que rodean a los sistemas y sus usuarios, además de trasladar dichos elementos a un formato computable para poder ser utilizados por los propios sistemas y que de esta forma la tecnología tenga consciencia (“*awareness*”) de lo que hay a su alrededor.

Resulta todo un reto el diseñar e implementar una infraestructura adecuada a estas premisas y enfocada a la educación que no sólo permita gestionar y desarrollar actividades de aprendizaje en distintos entornos, ya sean estos formales como el aula de un colegio o bien informales como una visita al campo en medio de un bosque, sino que permita además la transferencia entre lugares de aprendizaje de una manera fluida e ininterrumpida y haciendo que la tecnología sirva de nexo entre todos estos ‘*espacios de aprendizaje ubicuos*’. Un tema abierto dentro de la investigación del aprendizaje móvil es el lograr una interconexión e integración tanto a nivel pedagógico como tecnológico de estos espacios de aprendizaje. La tecnología por tanto ha de adquirir un rol de elemento integrador pero no únicamente en términos espaciales, sino también, algo que pocas veces se considera en la literatura, integrando actividades a lo largo de una dimensión temporal, con el fin de dar una continuidad a las actividades de aprendizaje que sirva de refuerzo a los conceptos adquiridos y promoviendo de este modo una creación del conocimiento global, situada y continuada.

Se necesitan por tanto nuevas arquitecturas con las que poder hacer frente a la integración de escenarios pedagógicos situados, distribuidos y posiblemente repletos de dispositivos heterogéneos que deben ser conscientes del contexto que les rodea y capaces de adaptarse a él. La Inteligencia Artificial se muestra como una plausible opción para lograr esta tarea y en concreto la Tecnología de Agentes se presenta como una candidata prometedora. Los agentes inteligentes son entidades software que pueden trabajar por separado o en conjunto colaborando entre ellas para ofrecer a los usuarios, alumnos y profesores, servicios personalizados y contextualizados. Dada su naturaleza distribuida, reactiva y proactiva son una solución apta para construir aplicaciones en entornos dinámicos y cambiantes como los que se plantean en el *aprendizaje móvil*.

Mediante este trabajo de investigación pretendemos explorar y desarrollar toda una arquitectura que dé soporte a estos nuevos paradigmas tecnológicos educativos de carácter ubicuo y distribuido además de una serie de modelos computacionales que los sustenten. Para este desarrollo pretendemos

utilizar como estrato tecnológico el paradigma de sistemas multi-agente. Detallamos los objetivos de nuestro trabajo en el siguiente apartado.

1.1 Objetivos

Los objetivos de este trabajo tratan de explorar nuevos modelos pedagógicos como los introducidos en el apartado anterior y aportar soluciones tecnológicas aplicadas a la educación que los soporten de manera integradora.

DESDE UN PUNTO DE VISTA PEDAGÓGICO

Se busca brindar soporte al desarrollo y a la integración de actividades de aprendizaje colaborativas formales e informales que hagan uso de tecnologías móviles inalámbricas y de información contextual. Esta integración no ha de asociarse únicamente a una dimensión espacial, es decir a lo largo de distintos escenarios localizados en diferentes entornos y contextos, sino también dando importancia al eje temporal, con flujos de actividades enlazados y diseñados para ser realizados a corto, medio y largo término. Es por consiguiente necesario que dicho soporte haya de realizarse tanto durante como antes y después de las actividades.

Soporte a la configuración y preparación antes de las actividades:

- Facilitar el despliegue de actividades que requieran el uso de tecnologías y dispositivos heterogéneos y móviles.
- Permitir una configuración adaptada de los dispositivos en base al contexto social del usuario, de actividad/tarea, localización, infraestructura computacional y de comunicación, y condiciones físicas del entorno.
- Selección personalizada de artefactos con el fin de buscar la reutilización de resultados previos y la continuidad entre actividades.
- Configuración/adaptación automática de las herramientas y los recursos.

Soporte durante el desarrollo de actividades:

- Soportar actividades espontáneas mediante eventos recibidos y generados por el entorno en los lugares de interés.
- Potenciar la interacción entre alumnos con el fin de promover colaboraciones alrededor de lugares de interés.
- Poder descubrir, unirse y participar en actividades que se realizan alrededor.
- Evitar que la tecnología aisle al usuario del entorno y le prive de posibilidades de aprendizaje.

Soporte acabada la actividad:

- Sincronizar y metadocumentar los resultados de los alumnos en las actividades para su posterior reutilización.
- Guardar un cuaderno de bitácora / historial de eventos de las actividades.

DESDE UN PUNTO DE VISTA TECNOLÓGICO

Se pretende crear todo un conjunto de tecnologías que den apoyo a los objetivos pedagógicos descritos y que provean de un conjunto de modelos y de una arquitectura enfocada hacia la reutilización y la extensibilidad para futuros usos.

Con respecto a los modelos a construir se busca:

- Aportar una ontología en la que se representen los elementos de la arquitectura empleados por los agentes, tales como herramientas, actividades, tareas, dispositivos, usuarios, grupos, artefactos/objetos de aprendizaje, agentes, recursos, protocolos de comunicación, eventos...
- Aportar un modelo computacional del contexto de usuario, social, de actividad/tarea, localización, infraestructura computacional y de comunicación, y condiciones físicas del entorno.
- Definir una taxonomía de eventos que será empleada para transmitir información entre los componentes software de la arquitectura.
- Implementar dicho modelo contextual y dicha taxonomía de eventos dentro de la arquitectura.

Con respecto a la arquitectura a construir se persigue:

- Investigar cómo una aproximación tecnológica basada en tecnología de agentes software puede ser aplicada para dar soporte a escenarios de aprendizaje móvil y colaborativo.
- Implementar la arquitectura siguiendo una solución basada en Agentes Software usando la librería de componentes y patrones de agentes ICARO-T.
- Implementar en la arquitectura mecanismos de aprovechamiento de información contextual que permitan a los agentes reaccionar y generar eventos en función de las situaciones.
- Lograr una arquitectura junto con un modelo computacional que sea extensible y fácilmente adaptable a nuevas necesidades y escenarios pedagógicos.

DESDE UN PUNTO DE VISTA DEL DISEÑO Y LA EVALUACIÓN

Hacer un diseño metodológico orientado a los usuarios y una evaluación integrada en todo el proceso.

- Utilizar un enfoque metodológico basado en escenarios con diseño participativo.
- Trabajar con escenarios que incluyan casos de uso relevantes para la arquitectura diseñada y que guíen nuestro desarrollo.
- Evaluar la arquitectura contrastándola con los escenarios diseñados y casos de uso extraídos de éstos.
- Aportar recomendaciones sobre extensibilidad y pragmática del diseño para un futuro uso.

1.2 Marco de Desarrollo

El marco de desarrollo que aquí se presenta se encuentra vinculado a distintos proyectos, tanto nacionales como internacionales, en los que ha participado el autor de este trabajo de investigación. Los proyectos COLDEX y ENLACE, junto con una serie de participaciones en grupos de la red de investigación Kaleidoscope (KALEIDOSCOPE, 2008), son los que han ido perfi-

lando las nociones básicas y conceptos pedagógicos que prevalecerán a lo largo del desarrollo del trabajo de investigación y cuya propuesta que se expondrá en la Parte II de esta memoria.

En primer lugar hay que destacar el proyecto europeo COLDEX como uno de los artífices en comenzar a definir el marco conceptual que se emplea en este trabajo. El proyecto COLDEX (COLDEX, 2001) (*“Collaborative Learning and Distributed Experimentation”*) trata temas y retos en el campo del soporte tecnológico para el aprendizaje colaborativo en áreas de ciencia y tecnología, con un especial interés en el aprendizaje basado tanto en experimentación local como remota. Esta experimentación es llevada a cabo en los distintos escenarios diseñados. Un escenario en COLDEX (Milrad M. et al., 2002) puede definirse como una colección de actividades educativas dentro de dominios específicamente concretos que son apoyadas por un flujo de trabajo y una serie de herramientas. Los distintos ejemplos de escenarios colaborativos de COLDEX, tanto remotos (Sánchez, F. et al., 2004) como locales (Otero, N. et al., 2003), se sitúan en dominios como la sismología, biodiversidad, química y astronomía. Dentro de este proyecto, las experiencias del aprendizaje y los resultados basados en esta experimentación obtenidos en los diversos escenarios, son considerados como sujeto de intercambio dentro de una comunidad internacional, formada por comunidades locales de menor tamaño. El medio principal de intercambio es un repositorio de objetos de aprendizaje. Éste provee facilidades para la búsqueda y detección de intereses similares en términos de los objetos producidos o artefactos.

Esta línea de investigación ha sido posteriormente continuada y extendida dentro del marco del proyecto nacional ENLACE. El proyecto ENLACE (ENLACE, 2005) tiene como objetivo el diseño de entornos educativos innovadores que ofrezcan soporte tecnológico para realizar un amplio abanico de actividades de aprendizaje en dominios relacionados con las ciencias de la naturaleza. La tecnología en ENLACE busca dar soporte activo de actividades en organizaciones complejas, acentuando el modelado y diseño de un entorno abierto de soporte inteligente a comunidades de aprendizaje en donde las estrategias de aprendizaje, los mecanismos de colaboración y las herramientas disponibles, ofrezcan una funcionalidad adaptada a diferentes ámbitos de trabajo, escenarios de experimentación y niveles formales de organización social: individual, pequeños grupos, grandes grupos, comunidades, y en particular analizar y ajustar las funcionalidades necesarias para articular la noción de comunidad desde las diferentes perspectivas: el mundo escolar, el mundo científico, el mundo profesional y la sociedad. ENLACE trata de integrar sus

diversos escenarios pedagógicos empleando distintos tipos de tecnología, incluidos dispositivos móviles, pero siempre tratando de aportar un eje de continuidad temporal e intentando imbricar las actividades en un flujo continuo y fluido a corto, medio y largo plazo. Por resultar de gran importancia para el entendimiento del contexto de este trabajo de investigación, hablaremos más detenidamente del proyecto ENLACE y sus actividades de en el apartado 2.5.

A parte de su participación en varios proyectos de investigación, el autor de esta tesis ha tenido la oportunidad de realizar dos estancias breves en universidades extranjeras donde progresar en las líneas de trabajo definidas en estos proyectos de investigación y en su trabajo de tesis en sí. Estas estancias concretamente tuvieron lugar en la Universidad de Berkeley, California, durante el año 2008 con el grupo de la profesora Marcia Linn, y en la Universidad de Toronto, Canadá, durante el año 2009 con el grupo del profesor James Slotta. Específicamente el trabajo efectuado en esta última estancia ha servido como escenario de aplicación adicional de nuestra arquitectura y una prueba real de su uso como se relatará en el capítulo 10.

1.3 Metodología

Como parte fundamental de la investigación, nos proponemos un diseño participativo y un desarrollo mediante una metodología que focalice en la usabilidad y esté centrada en el usuario, además de emplear una evaluación formativa para comprobar la viabilidad de la propuesta a desarrollar.

La idea principal consiste en integrar el proceso de desarrollo de la propuesta en torno a un diseño basado en escenarios (Rosson & Carroll, 2002a). Los escenarios son específicos, incluyen a las personas dentro de ellos y están situados en el mundo real. Mediante el empleo de escenarios durante el desarrollo es posible enfocar el diseño hacia detalles que precisan mayor atención desde el punto de vista de los usuarios. El uso extensivo de ejemplos coherentes extraídos de la experiencia real nos servirá para separarnos de la abstracción y encaminarnos hacia un entendimiento concreto de las necesidades de las personas que utilizarán el sistema. De esta manera los escenarios nos proveerán de puntos de partida concretos a la vez que nos introducirán de forma anticipada y gradual en los distintos aspectos y consideraciones de importancia del diseño de un modo secuencial ordenado.

Por otro lado, incluiremos un proceso de evaluación continuo e incorporado a las diferentes fases de desarrollo. Mediante una evaluación formativa dispondremos de un método de control de calidad para ayudarnos a mejorar la efectividad pedagógica de nuestra propuesta.

Siempre dentro de estas premisas, el protocolo seguido para desarrollar este trabajo ha sido:

- Explorar el estado del arte y el trabajo relacionado con la propuesta de tesis que proponemos en las tres áreas concernientes, aprendizaje humano, arquitecturas distribuidas y sistemas adaptados al contexto, haciendo un repaso de las teorías del aprendizaje que tienen relación con nuestro enfoque, la tecnología de agentes y el modelado del contexto.
- Realizar sucesivas iteraciones en un ciclo que incluye:
 - Definir los requisitos de la propuesta teniendo en cuenta los requerimientos pedagógicos que son expuestos con la ayuda de escenarios.
 - Detallar las decisiones tecnológicas de diseño en base a los requisitos de nuestra propuesta.
 - Adaptar si es necesario la tecnología de la que partimos.
 - Diseñar una arquitectura basada en agentes:
 - Diseñar modelos computacionales y ontologías para los agentes.
 - Diseñar la organización definiendo tanto los agentes como los recursos que la componen.
 - Implementar la arquitectura.
 - Evaluar con los escenarios la arquitectura desarrollada.
- Y finalmente, extraer unas recomendaciones de uso y extensibilidad, junto con unas conclusiones finales.

1.4 Estructura del Documento

En este primer capítulo se ha ofrecido una introducción con el fin de mostrar la motivación, los objetivos y la metodología de nuestro trabajo de

tesis. A continuación, en el capítulo 2, vamos a proceder a contextualizar el trabajo aportando una panorámica de las teorías educativas y conceptos dignos de mención e introduciremos las tecnologías necesarias para situar la solución propuesta.

En la Parte II se describe la solución que proponemos. El capítulo 3 expondrá el escenario de diseño que utilizaremos a lo largo de todo este trabajo, junto con el análisis de requisitos y las decisiones de diseño y desarrollo. El capítulo 4 introduce los conceptos básicos de la arquitectura que hemos desarrollado y presenta el concepto de espacio de aprendizaje ubicuo para enmarcar esta arquitectura. Los capítulos 5 al 8 describirán detalladamente los distintos niveles en los que se divide la arquitectura.

Finalmente, en la Parte III se relatan las pruebas realizadas para evaluar nuestro trabajo junto con los resultados obtenidos. En el capítulo 9 se hace una revisión de los casos de uso de nuestra arquitectura frente al escenario de diseño. En el capítulo 10 se pone a prueba nuevamente nuestra arquitectura frente a un nuevo escenario de aplicación. Por último, el capítulo 11 se hace una discusión final, en la que se incluyen unas recomendaciones de aplicación y extensibilidad de nuestra arquitectura, junto con una relación de las contribuciones y una exposición del trabajo futuro.

2

ESTADO DEL ARTE

En esta sección se dará un repaso al estado del arte relacionado con la problemática planteada en la Parte II con la intención de establecer una base para poder presentar y discutir el marco tecnológico que se presentará a lo largo de esta memoria de tesis.

En primer lugar, presentaremos en esta sección una breve introducción a la historia evolutiva de las tecnologías educativas y las teorías de aprendizaje asociadas. También comentaremos sobre el desarrollo de estas tecnologías educativas, haciendo hincapié en el marco teórico de los procesos de diseño y evaluación que utilizaremos en este trabajo. A continuación, hablaremos más profundamente sobre las teorías de aprendizaje con las que trabajaremos y que fundamentan nuestra propuesta: el *aprendizaje colaborativo* y el *aprendizaje móvil*. Dentro del aprendizaje colaborativo, resultará de importancia para este trabajo introducir nuestra noción de *objeto de aprendizaje*, como elemento clave para la construcción colaborativa del conocimiento dentro de una comunidad de aprendizaje, y de los *repositorios de objetos de*

aprendizaje, como las herramientas para poder almacenar, manipular y recuperar dicho conocimiento.

Con el objetivo de poder situar la propuesta que se va a presentar en esta tesis, resulta antes necesario encuadrarla dentro de los conceptos y tecnologías que las que se basa. En primer lugar se ofrecerá en esta sección una amplia visión del proyecto de investigación ENLACE y su red tecnológica educativa, que definen las nociones y el marco de trabajo en el que se engloba nuestra propuesta. En segundo lugar se hará una introducción al estado del arte del modelado y la representación del contexto. Después se introducirá la tecnología de agentes, pieza fundamental de la arquitectura que pretendemos desarrollar, y se situará dentro del dominio educativo con las aportaciones que se han realizado dentro de este campo. Finalmente se mostrará la plataforma de desarrollo de sistemas multi-agente ICARO-T, pieza fundamental en el diseño y desarrollo de este trabajo de investigación.

2.1 Teorías y Evolución de las Tecnologías Educativas

Mucho antes de que las primeras computadoras vieran la luz ya se especulaba con la posibilidad de disponer de una enseñanza automática con material auto-guiado (Thordike, 1912). Incluso se exploró con rudimentarias máquinas que sirvieran para corregir y puntuar de forma automática (Pressey, 1926), buscando de esta forma conseguir un método de educación programado que mejorase la enseñanza al ofrecer una realimentación instantánea a las respuestas del estudiante. Esto no sólo sugiere que la idea central de las tecnologías aplicadas a la educación precedió de largo el advenimiento del ordenador, sino que de modo más relevante muestra también como el objetivo de la investigación en las tecnologías aplicadas a la educación está asociado, indistinguiblemente de hecho, con el objetivo convencional de la investigación en la educación en sí, es decir se busca la mejora de la enseñanza en la forma en la que ésta se define operacionalmente (G. Stahl, Koschmann, & Suthers, 2006).

Sin embargo no fue hasta mediados del siglo XX cuando estas ideas cuajasen, gracias ya a la presencia de los ordenadores, en el nacimiento del campo de las tecnologías aplicadas a la educación y desencadenasen una evo-

lución que continua actualmente y que obviamente va íntimamente ligada a la evolución de las tecnologías de la información y la comunicación.

Las primeras aplicaciones educativas en aparecer en los comienzos de la década de 1960 se han agrupado bajo el llamado paradigma de la *Instrucción Asistida por Ordenador* o CAI (del inglés, *Computer Assisted Instruction*). La característica más remarcable de todas estas aplicaciones es que se encuentran influenciadas por la teoría psicológica que prevalecía en aquel momento, el conductismo (Skinner, 1965). Los conductistas postulan que el aprendizaje se mide en función de resultados, es decir del comportamiento final, y que está condicionado por una secuencia de estímulo, respuesta y realimentación. Los sistemas CAI buscan promover la efectividad en términos de resultados promovidos por principios de diseño instructivo conductista, que aún siguen presentes en gran parte del software actual, y que se basan en la descomposición de la información en unidades y secuenciación de actividades de interacción con el usuario seguidas de un refuerzo. El refuerzo ha de ser establecido por el diseñador instructivo con el fin de alcanzar los objetivos de enseñanza.

Con el fin de mejorar los sistemas CAI, a principios de 1970 aparece un nuevo paradigma que trata de aplicar los últimos avances en Inteligencia Artificial a estos sistemas con la intención de mejorar de manera sustancial los procesos de aprendizaje y añadirles un componente tutorial para poder generar propuestas individualizadas de enseñanza de forma dinámica en función del alumno y su estado mental. Este nuevo paradigma es conocido como *Sistemas Tutores Inteligentes* o ITS (del inglés, *Intelligent Tutoring Systems*) (Sleeman & J. S. Brown, 1982) y está basado en una filosofía cognitivista que analiza el aprendizaje del alumno en términos de modelos mentales y que por tanto, rechaza en cierto modo el conductismo como punto de vista válido por no prestar importancia a cómo los alumnos procesan y representan el conocimiento.

Los sistemas ITS han logrado cotas de éxito relativamente altas en comparación con los sistemas CAI, convirtiéndose en un paradigma ampliamente estudiado y generalizado (Wenger, 1987) y que ha marcado el comienzo de la relación entre la Inteligencia Artificial y las tecnologías educativas. Los sistemas ITS son todavía un área activa de investigación dentro de las tecnologías de la educación. No obstante, como cualquier otro sistema basado en conocimiento, se encuentran limitados por el inherente problema de la repre-

sentación particular del conocimiento de dominio para definir los modelos mentales del alumno.

Pretendiendo contraponerse a la metáfora del “ordenador como tutor”, a partir de la década de 1980 surgen nuevas propuestas que se centran en un enfoque cognitivista basándose particularmente en la escuela psicológica del constructivismo (Piaget & T. Brown, 1985). Para los constructivistas el aprendizaje es un proceso activo de construcción individual del significado por parte del sujeto. El alumno adquiere una representación personal del conocimiento por sí mismo a partir de las interpretaciones de su propia experiencia personal. Este modo de pensar abre el camino a nuevos sistemas educativos en los cuales el alumno pasa de ser un mero receptor pasivo de información, a ser un elemento activo en el proceso de aprendizaje (Papert, 1988).

Otra teoría de importancia derivada del cognitivismo y que surge al añadir una perspectiva social al aprendizaje es el *socio-constructivismo* (Vygotsky, 1978). Contraponiéndose al constructivismo individual de Piaget y Papert, el socio-constructivismo postula que la cognición es un proceso que surge primeramente a través de las interacciones sociales. En esta dimensión social también se encuentra la llamada *cognición situada*, que postula la visión del conocimiento como dependiente de un contexto y que no debería ser desacoplado de la situación en la que es construido (John Seely Brown et al., 1989).

Todas estas últimas teorías e ideas socio-constructivistas y cognitivistas situadas van a ser la base que sustente nuevos paradigmas de aprendizaje que se definen y desarrollan a finales del siglo XX y principios del siglo XXI. Un ejemplo de éstos es el aprendizaje por descubrimiento científico o por indagación, conocido en inglés como *inquiry learning* (de Jong, 2006a), donde los alumnos han de proponer sus propias conjeturas e hipótesis acerca de un determinado fenómeno o cuestión a estudiar, realizar experimentación y sacar unas conclusiones al respecto. En todo este proceso, el profesor toma el rol de un guía que se dedica a dirigir a los alumnos y a encaminarles hacia el descubrimiento del conocimiento. Con el fin de dar soporte a los alumnos, la tecnología ha de aportar una serie de herramientas para potenciar el afán descubridor de los alumnos y permitirles hacer modelos, simulaciones y experimentos, tanto reales como virtuales, en lugares de interés o laboratorios (de Jong, 2006b).

Otros paradigmas actuales que van a guiar este trabajo de investigación y que trataremos más detenidamente respectivamente en los apartados

2.3 y 2.4 por merecer especial atención son el *aprendizaje colaborativo* y el *aprendizaje móvil*, pero antes hablaremos sobre el diseño y la evaluación de las tecnologías educativas.

2.2 Diseño y Evaluación de Tecnologías Educativas

Una primera consideración que se debe hacer cuando se desarrolla cualquier tipo de tecnología es plantearse qué métodos y procesos se pretenden emplear para el diseño y la evaluación de dicha tecnología. Las tecnologías educativas además, por representar sistemas que están dirigidos directamente hacia las personas, no deben de olvidar un factor clave: la usabilidad. Tres principios fundamentales de diseño son subrayados por (Gould & Lewis, 1985) para incidir en la usabilidad. En primer lugar se necesita un enfoque precoz en los usuarios y las tareas, de modo que los diseñadores puedan comprender quienes serán los usuarios y sus necesidades. En segundo lugar es recomendable desde fases tempranas del desarrollo que los usuarios puedan usar simulaciones y prototipos con el fin de poder tomar medidas empíricas acerca de cuál es su experiencia con el sistema. Y finalmente todo el proceso de diseño ha de ser iterativo, con el fin de poder tener un ciclo de diseño, prueba, evaluación y rediseño que se repita tan a menudo como sea necesario.

Con el fin de incidir de esta forma en el concepto de usabilidad, y tal y como se ha comentado en el apartado 1.3, el diseño y desarrollo de nuestro trabajo de investigación estará encaminado por una *Ingeniería de Usabilidad* (Rosson & Carroll, 2002b) basada en escenarios y por una *evaluación formativa* de la tecnología a desarrollar, que se comentan en los siguientes apartados.

2.2.1 Diseño basado en Escenarios

El diseño basado en escenarios es un conjunto de técnicas mediante las cuales se define el futuro uso de un sistema desde las primeras etapas del proceso de desarrollo. Se utilizan descripciones narrativas que sirven para ofrecer una imagen de posibles episodios de uso aportando una visión general que se

empleará para guiar el desarrollo del sistema con el fin último de habilitar dichas posibles experiencias de uso de los usuarios.

Los escenarios son específicos, incluyen a las personas dentro de ellos y están situados en el mundo real. Básicamente los escenarios son historias. Éstas transcurren en un lugar o dentro de un contexto bien definido y describen a los actores que intervienen con sus motivaciones, conocimientos y habilidades, junto con las herramientas y objetos que se encuentran y manipulan. Un escenario describe una secuencia de acciones y eventos que finalmente desembocan en un resultado. Mediante el empleo de un desarrollo basado en escenarios es posible enfocar el diseño hacia detalles que precisan mayor atención desde el punto de vista de los usuarios. El uso extensivo de ejemplos coherentes extraídos de la experiencia real sirve para separarse de la abstracción y focalizar hacia un entendimiento concreto de las necesidades de las personas. De esta manera los escenarios proveen de puntos de partida concretos a la vez que nos introducen de forma anticipada y gradual en los distintos aspectos y consideraciones de importancia del diseño de un modo secuencial ordenado.

Cabe destacar que el diseño basado en escenarios aporta una aproximación ligera, relajada pero sobre todo centrada en el usuario. En vez de situar el foco de atención del trabajo de diseño en la definición operacional del sistema, es decir en su especificación funcional, el diseño basado en escenarios lo sitúa en la descripción acerca de cómo las personas usarán el sistema para llevar a cabo tareas y actividades.

El diseño de sistemas interactivos es lo que se conoce como un problema mal definido. Esta clase de problemas se caracteriza por tratar elementos complejos y que por tanto no pueden ser fácilmente descritos de un modo conciso y completo. Este tipo de problemas habitualmente evocan una estrategia de resolución que es denominada *primera solución* (Cross, 2001) en la que los diseñadores generan y analizan una solución candidata al problema y que servirá como medio para clarificar el dominio del problema y, basándose en su exactitud, obtener nuevos requerimientos del análisis. Una aproximación al diseño basada en *primera solución* es eficiente y efectiva, como se puede apreciar de la actual popularidad de sistemas de desarrollo como el *prototipado rápido* (Wasserman & Shewmake, 1982) y la *programación extrema* (Beck, 1999). No obstante, también entraña peligros ya que los diseñadores tienden a generar soluciones de forma rápida comprometiéndose de forma prematura a una visión que en caso de tener problemas con ella, puede

ser difícil de abandonar. Otro problema aparece cuando los diseñadores tienden a reutilizar las mismas soluciones empleadas en problemas anteriores para los nuevos, aun cuando estas no sean las más apropiadas para los usuarios, sin explorar profusamente todo el posible espacio de soluciones. Finalmente, también se corre el peligro de perderse en el desarrollo sin prestar atención ni cuestionarse durante éste si se ajusta al diseño e incluso si el diseño es el correcto.

(Rosson & Carroll, 2002a) aportan tres características clave acerca de las ventajas que reporta usar un diseño basado en escenarios para minimizar estos peligros. Primeramente los escenarios son concretos aunque incompletos. Segundo, los escenarios mantienen la orientación hacia los usuarios y sus necesidades. Y por último los escenarios son evocativos y generan preguntas.

LOS ESCENARIOS SON CONCRETOS AUNQUE INCOMPLETOS

El análisis del diseño siempre resulta un proceso indeterminado y voluble, propiciado porque el mismo acto de diseño cambia el mundo dentro del cual las personas interactúan y experimentan, pero también por el innegable hecho de que los requerimientos de un sistema siempre cambian (Brooks, 1995). Esto se agrava más cuando los diseños incorporan tecnologías que tienden a evolucionar rápidamente con el tiempo. Un buen proceso de diseño tiene que poder adaptarse a estas ambigüedades.

Las representaciones de diseño que son concretas pero a la vez flexibles ayudan a poder tratar situaciones ambiguas y que además tienen un alto componente de dinamismo. Un escenario ofrece una visión de diseño concreta, pero que al mismo tiempo puede estar bosquejada a diferentes niveles de detalle. Al principio del desarrollo, aun capturando la esencia del diseño, un escenario puede estar poco elaborado. Según avanza este desarrollo, el escenario ofrece un marco observable en el que se pueden ir incluyendo nuevas ideas expresadas de una manera relajada e informal y que permite de este modo hacer visible el progreso del desarrollo.

LOS ESCENARIOS MANTIENEN LA ORIENTACIÓN HACIA LOS USUARIOS

Es a menudo posible que desarrolladores con años de experiencia en determinadas aproximaciones o tecnologías puedan tener una visión sesgada basada en sus previas soluciones de diseño. Por ejemplo un grupo de desarrollo con amplia experiencia en aplicaciones web basadas en formularios puede

rechazar de primeras una aproximación en tiempo real posiblemente más conveniente basada en Web 2.0. No obstante, debido a que se centran en las personas y sus experiencias, los escenarios desvían la atención hacia la apropiada validez de uso desde la perspectiva de usuario de las ideas de diseño, obligando a los desarrolladores a no atrincherarse en una determinada solución o tecnología y considerar otras opciones.

Por otro lado, aunque bien es cierto que la utilización del lenguaje técnico tiende a favorecer la precisión en la comunicación entre los diseñadores de un proyecto, este lenguaje técnico también tiende a excluir a posibles participantes en el diseño, como por ejemplo miembros del equipo menos entrenados o los propios usuarios. Los escenarios ayudan a tratar dicho problema al ofrecer un lenguaje universalmente accesible. Todos los miembros del proyecto pueden hablar el mismo idioma, el idioma de los escenarios.

Es por ello que el diseño basado en escenarios ofrece un proceso simple y natural de *diseño participativo*, es decir el trabajo de diseño se lleva a cabo como una colaboración entre los desarrolladores y la gente que va a usar el sistema (M. J. Muller & Kuhn, 1993). Los escenarios ayudan a integrar distintos tipos de conocimiento y experiencia procedente de distintos tipos de expertos y usuarios, promoviendo la comunicación y coordinación entre los diferentes estamentos del grupo de desarrollo.

LOS ESCENARIOS SON EVOCATIVOS Y GENERAN PREGUNTAS

Otro peligro que aparece en la Ingeniería de Usabilidad es el conflicto entre el pensar y el hacer. Los desarrolladores tienen la imperiosa necesidad de actuar para ofrecer un progreso rápido, pero siempre a costa de no reflexionar acerca del análisis y la implementación necesaria para conseguir una solución de diseño de calidad. Las reuniones de grupo de trabajo pueden servir en ocasiones para lograr este fin y reflexionar sobre la dirección del desarrollo, no obstante lo que suele ocurrir en tales reuniones es que los resultados son evaluados en términos de progreso, resolución de objetivos y completitud de especificaciones, sin llegar a cuestionarse problemas de usabilidad.

La naturaleza evocativa de los escenarios ayuda a tratar este problema. Los escenarios generan preguntas que surgen al comparar el desarrollo actual con el diseño. Estas preguntas además aparecen a todos los niveles. De esta forma es más fácil establecer problemas actuales, e incluso poder vislumbrar posibles dificultades futuras.

El proceso de desarrollo del diseño basado en escenarios es el resultado de una continuada iteración entre distintas fases de análisis, diseño, prototipado y evaluación, y aconsejablemente solapándose entre ellas. Se distingue entre un tipo de evaluación que ha de estar integrada durante todo el proceso de desarrollo con el fin de guiar el rediseño del sistema, *evaluación formativa*, y otra que sirve de función de verificación del sistema, *evaluación sumativa*. En el siguiente apartado vamos a tratar con más detalle esta primera por ser la que nos ayudará a depurar nuestra solución a medida que la vamos desarrollando.

2.2.2 Evaluación Formativa

Mientras que la evaluación sumativa aporta preguntas del estilo: ¿Hemos logrado el sistema que habíamos visionado y especificado? ¿Hemos alcanzado o superado los objetivos pedagógicos y de usabilidad cuantificados en las especificaciones? La evaluación formativa se plantea preguntas como: ¿Qué es lo que no está funcionando bien? ¿Por qué? ¿Qué cambios son necesarios para arreglarlo?

De este modo para el momento en el que el equipo de desarrollo efectúe una evaluación sumativa bastará con saber si los objetivos han sido alcanzados. Por el contrario, la evaluación formativa está enfocada a mejorar el diseño del prototipo, no meramente a medir la calidad global del sistema.

La evaluación formativa es uno de los tipos más antiguos de evaluación, siendo (Scriven, 1967) uno de los primeros en mencionar este término en el contexto pedagógico. (Thiagarajan, 1991) define la evaluación formativa como “un método de control de calidad orientado a mejorar, que no a probar, la efectividad instructiva” y “un proceso continuo incorporado a distintas etapas del desarrollo”.

La evaluación formativa es normalmente llevada a cabo por el diseñador o desarrollados, aunque en ocasiones grandes organizaciones requieren la ayuda de evaluadores externos. (Dessinger & Moseley, 2004) describen cuatro estrategias básicas para conducir una evaluación formativa:

1. Una revisión con un individuo o grupo experto familiar con el contenido y las necesidades.

2. Una evaluación uno-a-uno que incluya al diseñador o evaluador y al alumno o usuario.
3. Una evaluación virtual o en vivo con un grupo pequeño.
4. Un test de campo o piloto que incluya o bien partes o bien todo el sistema o producto.

2.3 Aprendizaje Colaborativo

El *aprendizaje colaborativo* puede entenderse como el proceso en el que dos o más personas aprenden o intentan aprender algo juntas (Dillenbourg, 1999). Como señala Dillenbourg, cada elemento de esta definición puede ser interpretado de distintas formas. De este modo, “dos o más” puede entenderse como una pareja, un grupo pequeño, una clase o hasta una comunidad entera de estudiantes. Lo mismo es aplicable a las nociones de “aprender” y “juntas”, ya que es difícil concretar definiciones precisas para aprendizaje y colaboración.

Dejando a un lado la terminología, bien es cierto que mediante esta aproximación a la enseñanza se busca el promover distintos mecanismos cognitivos y de comunicación que no están presentes en el aprendizaje individual. Al igual que en los sistemas cognitivos individuales se realizan actividades (leer, reflexionar, resumir,...) que fomentan ciertos mecanismos de aprendizaje (inducción, deducción, síntesis,...), en el aprendizaje colaborativo se distinguen otras actividades y mecanismos propios debidos a la interacción entre individuos. Esto no quiere decir que se suprima la cognición individual, sino que a ésta se le añaden además, gracias a la interacción entre individuos, otras actividades (explicación, negociación, desacuerdo, regulación mutua,...) que provocan mecanismos cognitivos extra (descubrimiento del conocimiento, interiorización, reducción de la carga cognitiva,...).

No obstante, a pesar de que estos mecanismos tienden a ocurrir más frecuentemente en el aprendizaje colaborativo que en condiciones individuales, no existe ninguna garantía en absoluto de que aparezcan en cualquier esquema colaborativo. Por este motivo, es necesario promover las interacciones entre sujetos que los provoquen mediante una mayor facilidad para la comunicación y la compartición. En este sentido, el campo del *Aprendizaje Colaborativo Apoyado por Ordenador* (Koschmann, 1996; Barros, 1999) (en adelante CSCL, *Computer Supported Collaborative Learning*), trata de esta-

blecer modelos de colaboración y plasmarlos en medios tecnológicos para apoyar y fomentar la co-construcción de conocimiento en el proceso de aprendizaje. La comunicación mediante ‘artefactos’, entendidos éstos en el sentido de objetos (co-)construidos por los estudiantes, es un principio básico usado en diversos entornos de trabajo compartido en sistemas CSCL. Los estudiantes tienen que crear y manipular representaciones comunes que les obliguen a compartir y reflexionar sobre sus ideas. De esta forma se persigue una aproximación en la que se desarrolle una construcción social del conocimiento (Scardamalia & Bereiter, 1994). Esto significa que los estudiantes tienen que manejar estos artefactos, otorgándoseles facilidades para modificarlos y desarrollarlos en conjunto, y así colaborativamente encontrar un estado final para éstos que satisfaga al grupo entero. Muy relacionado con todo esto se encuentra el concepto de *objetos de aprendizaje emergentes*, que se presenta a continuación en el siguiente apartado.

2.3.1 Objetos de Aprendizaje Emergentes

Durante los últimos años, mucha de la investigación en el campo de las tecnologías educativas se ha centrado en la noción de componentes reutilizables de contenido multimedia, los llamados *objetos de aprendizaje*. La fuerza que promueve el desarrollo a partir de este concepto es precisamente la reutilización a la que están orientados, ya que puede conducir a importantes ahorros de tiempo y de dinero en el desarrollo de nuevo entornos virtuales de aprendizaje, a la par que mejorar la calidad de las experiencias de aprendizaje digitales.

Los objetos de aprendizaje han llegado a convertirse en los abanderados de una nueva filosofía dentro del campo de las tecnologías educativas y se presentan como la piedra angular de la nueva era del software educativo. Hoy en día existen distintos estándares que tratan sobre ellos, aunque, no obstante, todavía existe poco consenso general acerca de una definición precisa para ellos. Quizás la definición más apropiada en este sentido haya que debérsela a (Wiley, 2002), por ser suficientemente concreta en su enunciado pero amplia y flexible a la vez para abarcar el concepto:

Un objeto de aprendizaje es un recurso digital que puede ser reutilizado para dar apoyo al aprendizaje. Esta definición incluye todo aquello que puede ser distribuido a través de la red bajo demanda, ya sea grande o pequeño.

Probablemente, la característica más remarcable de esta definición de objeto de aprendizaje está justamente en relación con su capacidad de reutilización, pero también se encuentra en que aporta una visión suficientemente general de lo que puede englobarse como objeto de aprendizaje. Comúnmente, la visión que habitualmente se asocia a los objetos de aprendizaje es de contenidos pre-generados por los diseñadores instruccionales y destinados a ser consumidos por el alumno. Sin embargo, otra perspectiva más global permite verlos como elementos creados por cualquier tipo de usuario, incluidos los propios alumnos, permitiendo de este modo tratar como objetos de aprendizaje los resultados o productos intermedios que ‘*emergen*’ del desarrollo de las actividades que se les encomiendan. De aquí aparece el término *objeto de aprendizaje emergente*. El hecho de poder ser reutilizados por los mismos usuarios que los generaron, o incluso en otros contextos similares por usuarios diferentes, es lo que les hace tan interesantes y fundamenta una de las claves para el reaprovechamiento del conocimiento. Esta última visión más amplia de objeto de aprendizaje es la que se tendrá en cuenta a lo largo de este trabajo de investigación y es una de las piedras angulares del proyecto ENLACE (ENLACE, 2005), del que se habla más adelante en el apartado 2.5, y que comenzó a fraguarse en el desarrollo del proyecto COLDEX (COLDEX, 2001).

Partiendo del trabajo en sistemas CSCL dentro de áreas compartidas donde se permite la interacción, ya sea de forma síncrona o asíncrona, entre los estudiantes a través de artefactos, los miembros de una comunidad de aprendizaje pueden generar conocimiento (Gerry Stahl, 2000). A la hora de poder desarrollar comunidades virtuales de aprendizaje, es importante que éstas posean medios para intercambiar el conocimiento generado dentro de ellas, incluso permitiendo este intercambio a través de distintas escalas sociales (Mayorga et al., 2007). Ahora bien, resulta importante que los recursos y productos así generados sean almacenados para, posteriormente en un momento futuro, poder ser reutilizados bien por los mismos estudiantes que los generaron, o incluso por otros miembros pertenecientes a la comunidad de aprendizaje que potencialmente posean los mismos temas de interés. Esto es de hecho un mecanismo de cohesión para toda la comunidad, ya que es posible establecer vínculos entre los miembros de dicha comunidad gracias a la *navegación temática social* (Hoppe et al., 2005), que pone en contacto, posiblemente de forma indirecta, a usuarios con las mismas inquietudes o necesidades de conocimiento a través de los artefactos o recursos que producen.

Los objetos de aprendizaje emergentes muestran un carácter evolutivo. Un objeto de aprendizaje puede ir sufriendo distintos cambios a lo largo del ciclo de vida, mostrando las evoluciones del usuario que lo creó, o bien cuando se trata de un grupo de personas, el proceso colaborativo que se ha desarrollado sobre él. Todos estos datos referentes al ciclo de vida, junto con otros muchos más relativos al contexto y al contenido del objeto, se almacenan junto con éste. Son lo que se conoce como metadatos y va a ser el mecanismo por el que se logre la búsqueda reutilización.

METADATOS

A la hora de poder buscar y recuperar objetos de aprendizaje es necesario que éstos contengan cierta información acerca de los contenidos que poseen. Esta información, es conocida por el nombre de *metadatos*, datos sobre los datos, y permite especificar el contenido, objetivos, contexto, etc., del objeto de aprendizaje.

Se han presentado diversos formatos definiendo conjuntos de metadatos para objetos de aprendizaje (Weibel, Kunze, Lagoze, & Wolf, 1998; Friesen, Roberts, & Fisher, 2002; Anido et al., 2002). A pesar de ello, el que merece la pena destacar, sobre todo por ser del que parten muchos de ellos y que está siendo adoptado rápidamente por la comunidad de las tecnologías educativas, es el auspiciado por el IEEE: el estándar LOM (*Learning Object Metadata*) (IEEE Standards Department, 2002). Consta de 65 campos definidos y estructurados jerárquicamente en 9 categorías. Cada una de estas categorías agrupa una serie de elementos de metadatos que cubren aspectos específicos, tales como cuestiones tecnológicas o características educativas. Es importante reseñar que la estructura LOM puede ser extendida o bien puede escogerse un subconjunto de sus campos de metadatos, con el fin de adaptarlo a la comunidad de aplicación (Duval & Hodgins, 2003). Los esquemas de metadatos resultantes reciben el nombre de *perfiles de aplicación* (*application profiles*) (Heery & Patel, 2000), y son una manera apropiada de optimizar un conjunto de metadatos a los objetivos del contexto donde se desean aplicar.

Es de vital importancia que los objetos de aprendizaje emergentes se encuentren metadocumentados de la forma más completa y correcta posible, ya que esto posibilitará su búsqueda y localización de una manera más eficaz para poder ser compartidos por la comunidad. Sin embargo es justo aquí donde reside el gran problema de los metadatos. Normalmente los usuarios que crean contenidos no se detienen a cumplimentar los campos de los metadatos

asociados al objeto generado, sobre todo cuando el conjunto de esos campos es excesivamente grande. Es por ello que se hacen necesarios mecanismos de relleno automático de metadatos, basados típicamente en el contenido del objeto como en su contexto (Verdejo, Celorrio, & Lorenzo, 2006).

Resulta importante destacar que tomando como base la información que se puede obtener y utilizar a partir de las anotaciones de los objetos es posible aportar un mayor nivel de abstracción para el tratamiento de dichos objetos. Mediante una reificación de las relaciones de metadatos es posible vertebrar el conocimiento de una comunidad virtual de aprendizaje colaborativo en forma de una ontología, con el fin de permitir una recuperación y explotación semántica de dicho conocimiento (Mayorga et al., 2007). De esta forma los metadatos pueden considerarse como un primer paso hacia una representación semántica del contenido, la cual puede ser utilizada por los repositorios de objetos de aprendizaje.

2.3.2 Repositorios de Objetos de Aprendizaje

Los repositorios digitales pueden verse a grandes rasgos como lugares donde poder almacenar material digital. Ofrecen grandes facilidades para buscar y recuperar este material, promoviendo de forma eficaz la reusabilidad de sus contenidos. Por ello, cuando se utilizan para almacenar recursos educativos, es decir, objetos de aprendizaje, permiten aumentar las capacidades de reutilización de estos contenidos dentro de un contexto educativo. Gracias a estos repositorios, los usuarios pueden tener acceso y recuperar objetos de aprendizaje que hayan sido creados anteriormente, además de poder volver a almacenarlos como nuevas versiones disponibles para otros usuarios, una vez usados y modificados.

Los repositorios de objetos de aprendizaje pueden entenderse como almacenes donde los profesores y diseñadores instruccionales pueden compartir y encontrar elementos para la confección de sus clases y cursos. No obstante, existe también otro enfoque para el uso de estos. Dentro de una comunidad de estudiantes involucrados en la realización de actividades en las que se produzcan resultados o recursos asociados de cualquier tipo, tiene sentido el poder almacenar estos materiales producidos, con el fin de que puedan ser compartidos y reutilizados más tarde por los propios creadores, o por otros compañeros de la comunidad. Por tanto, se pueden utilizar estos repositorios con el fin de almacenar estos resultados y metadocumentarlos para su posterior búsqueda. De esta forma, se puede entender que el repositorio de objetos

de aprendizaje se configura así como una memoria colectiva perteneciente a esta comunidad que permite “recordar” a todos sus miembros experiencias pasadas, ya sea para enriquecer el proceso de aprendizaje o para acelerarlo con ejemplos prácticos anteriores.

TIPOS DE OBJETOS DE APRENDIZAJE

Debido a que se amplía la extensión del concepto de objeto de aprendizaje, como hemos comentado en el apartado 2.3.1 para incluir dentro de éste los productos emergentes creados por una comunidad de aprendizaje, y a que se establecen los repositorios de objetos de aprendizaje como lugares donde almacenar estos resultados para así crear una memoria colectiva, aparece en consecuencia el concepto de *tipo de objeto de aprendizaje* para poder definir, estructurar y clasificar los elementos almacenados en los repositorios (Celorrio & Verdejo, 2008).

Un *tipo de objeto de aprendizaje* define toda una clase de artefactos educativos que comparten características comunes. Así pues de la misma forma que las bases de datos aportan estructuras para almacenar datos e interrelacionarlos, los *tipos de objetos de aprendizaje* son un mecanismo a un nivel de abstracción mayor para estructurar y caracterizar las aportaciones de los usuarios. Dentro de un *tipo de objeto de aprendizaje* se especifica qué conjunto de esquemas de metadatos son aplicables a los objetos pertenecientes a dicho tipo y qué mecanismos específicos son empleados para extraer información para rellenarlos (Verdejo et al., 2006). Esto permite un instrumento potente para habilitar búsquedas específicas con metadatos sólo contemplados para ciertos objetos de aprendizaje. También dentro de la definición de un *tipo de objeto de aprendizaje* se incluye información acerca de qué clase de contenido tendrán todas las instancias de objeto de aprendizaje de un determinado tipo, permitiendo a las herramientas hacer un tratamiento específico de los contenidos y productos.

INTEROPERABILIDAD EN REPOSITORIOS

La interoperabilidad puede ser definida como la característica que permite que la información producida dentro de un contexto pueda ser utilizada en otro distinto de la manera más automática posible. En nuestro contexto, más específicamente, la interoperabilidad es un elemento práctico con el que se puede llegar a conseguir que objetos de múltiples y desconocidas

fuentes, posiblemente de sistemas o repositorios tecnológicamente heterogéneos, puedan ser (re)utilizados sin apenas problema (Hatala, Richards, Eap, & Willms, 2004).

La manera de lograr esto pasa por la estandarización de protocolos y por el establecimiento de formatos comunes. A parte de necesitar estándares que atañen directamente a los objetos de aprendizaje, para describir sus metadatos, contenido, empaquetamiento, secuenciación, etc., también es necesario la existencia de mecanismos comunes dentro de los repositorios que los contienen, para que puedan interoperar entre ellos o con otros sistemas o herramientas.

Es importante que los repositorios posean facilidades para almacenar, realizar búsquedas o recuperar objetos y que estas funcionalidades estén disponibles para herramientas externas, que así puedan operar con los objetos de aprendizaje que se encuentran en estos (Celorrio & Verdejo, 2008). No obstante, también es importante la interoperabilidad de repositorios entre sí, ya que posibilita la distribución y acceso de contenidos entre sistemas con contenidos educativos, que en principio pueden estar alojados en sistemas tecnológicamente muy distintos, pero que son capaces de interactuar entre sí gracias a que poseen protocolos de comunicación basados en estándares. Puede consultarse (Celorrio, Verdejo, & Barros, 2005, 2006) para más detalles al respecto junto con una solución de distribución basada en servicios web.

2.4 Aprendizaje Móvil

En la última década del siglo XX, Weiser (1991, 1993) acuñó el término *computación ubicua* para describir una visión del futuro en la cual las computadoras se encontrarían totalmente integradas en el mundo que nos rodea, ayudándonos de un modo cotidiano en nuestros quehaceres diarios. En la visión de Weiser la gente no interactuaba con ‘cajas’ encima de un escritorio. Por el contrario, las personas llevaban consigo pequeños dispositivos digitales, de la misma forma que llevamos una tarjeta de crédito o el documento nacional de identidad. Cuando estas personas entrasen en un nuevo recinto o lugar, estos dispositivos podrían interactuar con otros situados y, posiblemente, embebidos en el entorno con el fin de registrar nuestros intereses, acceder y mostrar información útil para nosotros e iniciar procesos para ayudarnos con nuestras tareas.

La tecnología actual nos está dirigiendo a un ritmo vertiginoso hacia esta visión de Weiser que para aquella época relativamente cercana podría parecer incluso ciencia ficción. Un amplio rango de pequeños dispositivos como agendas electrónicas (*PDA*s) o teléfonos móviles inteligentes (*Smartphones*) es ya común entre nosotros permitiendo lo que se denomina *computación móvil*, actividades realizadas con un computador que ocurren en un lugar alejado del puesto habitual de trabajo de una persona. Del mismo modo, se puede considerar de forma análoga que el *aprendizaje móvil* (del inglés *Mobile Learning*) es aquel que ocurre cuando el estudiante está alejado de su sitio habitual de aprendizaje. Una consecuencia del uso de dispositivos móviles es que el aprendizaje puede llevarse a cabo en lugares de interés (Soloway et al., 1999; Rogers et al., 2004; Y. Chen et al., 2003), pero también que es posible redefinir y ofrecer nuevas experiencias en ambientes formales (Tatar et al., 2003).

De esta forma, el aprendizaje móvil está ligado a los paradigmas ‘en cualquier momento’ y ‘en cualquier sitio’. Sin embargo, una matización importante que aportan algunos autores es que consideran que el aprendizaje es móvil es sí mismo, en tanto y cuanto el aprendizaje ocurre de forma continuada a lo largo de nuestra vida y en distintas dimensiones:

“El aprendizaje es móvil en términos de espacio, es decir puede ocurrir en el sitio de trabajo, en casa o en sitios de ocio; es móvil entre diferentes aspectos de la vida, por ejemplo puede estar relacionado con necesidades laborales, motivado por el afán de auto-superación, o por diversión; y es móvil con respecto al tiempo, puede ocurrir en diferentes momentos del día, en días laborales o los fines de semana.” (Vavoula & Sharples, 2002, pág. 152)

Es importante destacar que aunque autores previos habían incidido en “el aprendizaje [como] un proceso continuo y de por vida resultante de la actuación en situaciones” (John Seely Brown et al., 1989), el paradigma del *aprendizaje móvil* es el primero que integra la tecnología en las actividades diarias para dar soporte a un aprendizaje continuado (al que habitualmente se hace referencia en inglés como *life-long learning*) (Sharples, 2000).

En los últimos años muchas iniciativas para poner en práctica estas consignas del aprendizaje móvil. (Naismith & Futurelab, 2004; Faux et al., 2006) ofrecen un buen compendio de ejemplos al respecto. Es común que cuando se habla del *aprendizaje móvil* se utilicen habitualmente términos para describirlo como *personal*, *ubicuo* y *no intencional*. Personal como algo que

está adaptado a las necesidades del individuo y que le acompaña. Ubicuo en tanto en cuanto es accesible en todo momento y que se encuentra embebido en el contexto diario de uso. Y finalmente no intencional en el modo de que es usado cuando la ocasión aparece.

A tenor de esto, no hay que caer en la tentación de usar los dispositivos móviles como meros distribuidores de contenido pedagógico, sino buscar formas de potenciar todas estas características. Hay que mirar más allá y buscar nuevos paradigmas en los cuales los dispositivos móviles ofrezcan nuevas oportunidades y métodos de aprendizaje que pueden derivar en un mejor y más auténtico aprendizaje (Chan et al., 2006).

2.4.1 Hacia el Aprendizaje Ubicuo

A menudo se hace referencia en contextos muy similares al aprendizaje móvil a otros términos como *aprendizaje ubicuo* (*ubiquitous learning*) y *aprendizaje pervasivo* (*pervasive learning*). Habitualmente muchos autores los emplean indistintamente por presentar características casi idénticas al aprendizaje móvil.

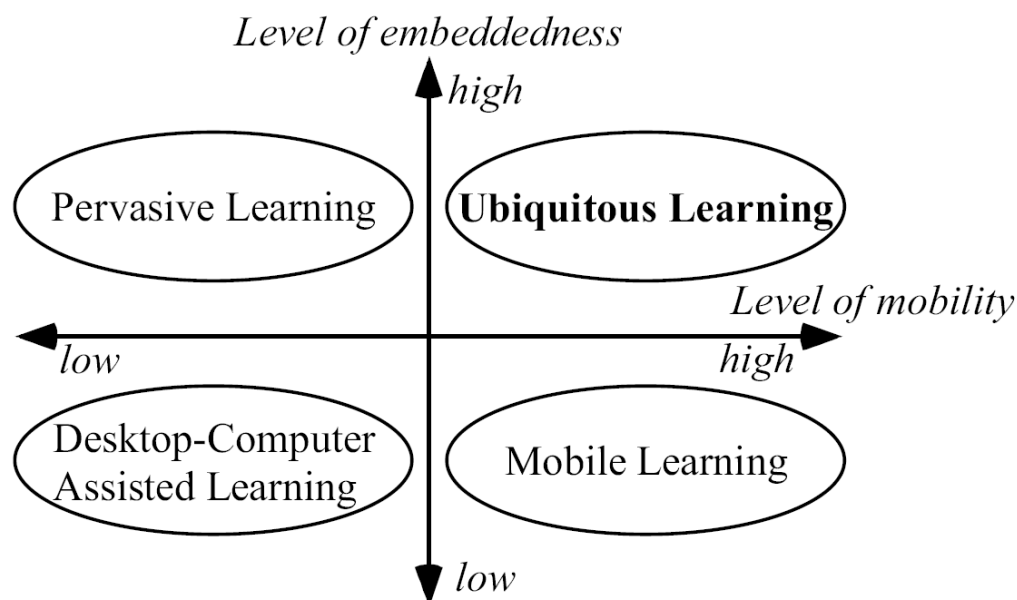


Figura 2.1 El aprendizaje en función de la movilidad y la integración

No obstante, (Ogata & Yano, 2004) ofrecen una clasificación, visible en la Figura 2.1, en la cual se puede observar una sutil diferencia. El *aprendizaje*

pervasivo se podría caracterizar, por tanto, por utilizar dispositivos y tecnología con un nivel alto de integración en el entorno y no necesariamente movilidad. Un ejemplo serían las denominadas ‘aulas inteligentes’, las cuales pueden contener dispositivos interconectados e integrados en el entorno como pantallas en las paredes para visualizar información de las actividades de los alumnos, cámaras para registrar sus acciones, sensores para captar sus posiciones, etc. En el momento en que la movilidad y la integración de la tecnología con el entorno se unen podemos considerar que tratamos con *aprendizaje ubicuo*.

En el *aprendizaje ubicuo* cada lugar puede resultar en un espacio de aprendizaje para el usuario que interconecta de forma inteligente tanto objetos, personas, dispositivos e incluso otros espacios de aprendizaje, y en el cual pueden aparecer oportunidades de aprendizaje adaptadas al contexto y personalizadas al usuario. Una forma de reflexionar sobre la sutil diferencia entre el aprendizaje móvil y el aprendizaje ubicuo es que mientras que el *aprendizaje móvil* saca las computadoras fuera de la clase al mundo, en el *aprendizaje ubicuo* el mundo se convierte en la clase y en la computadora (Ley, 2007).

Es por tanto que el mayor problema actual que existe para poner en práctica y dar apoyo a un tipo de aprendizaje como el ubicuo es, paradójicamente, su alto componente integrador. Un problema que reta a las tecnologías educativas a dar un soporte a la integración con el fin de mejorar los procesos de aprendizaje que se presenta tanto con dimensión tecnológica, para lograr integrar de la forma más transparente posible distintos dispositivos heterogéneos; dimensión espacial, para permitir el aprendizaje adaptado a contextos diferentes, en múltiples lugares y creando interconexiones entre estos; dimensión personal, para ofrecer contenidos y estilos de aprendizaje acondicionados a la persona; dimensión social, para potenciar la interacción entre grupos de individuos y la consiguiente confrontación cognitiva entre estos; dimensión curricular, para relacionar temas de contextos diferentes; como con dimensión temporal, que tenga en consideración no sólo el momento actual para ofrecer un aprendizaje oportunista, sino el pasado y el futuro con actividades para conectar y reforzar el aprendizaje a corto, medio y largo plazo. Nuestra aportación tratará de contribuir en este sentido buscando acortar la distancia entre la teoría y la práctica mediante una solución tecnológica distribuida con capacidad inteligente de interpretación y adaptación al contexto, que aporte soluciones a la integración en todas estas direcciones.

2.5 El Proyecto ENLACE

Dentro de los objetivos del proyecto ENLACE (ENLACE, 2005) se encuentra el diseño de entornos educativos innovadores, que ofrezcan soporte inteligente para realizar toda una sucesión de actividades de aprendizaje en dominios relacionados con las ciencias de la naturaleza. El proyecto ENLACE sigue las líneas marcadas en el proyecto anterior COLDEX (COLDEX, 2001), tratando de definir estructuras sociales a modo de comunidades de aprendizaje, en las que exista la posibilidad de crear y compartir conocimiento, construyéndolo de una forma social. Una aproximación pedagógica que se sigue a lo largo del proyecto es la de basar el proceso de aprendizaje en el descubrimiento científico o “aprendizaje por indagación” (en inglés *inquiry learning*), de forma que los alumnos tengan que formular sus hipótesis sobre los temas a estudiar, realizar experimentos y simulaciones, y finalmente aportar unas conclusiones, guiados en todo momento por el profesor (de Jong, 2006b).

De esta forma en ENLACE se busca desarrollar entornos informáticos abiertos, que integran herramientas de modelado para el análisis y construcción colaborativa de soluciones, a problemas científicos. Estas herramientas han de ofrecer mecanismos semánticos para guardar, compartir e intercambiar datos y artefactos multimedia, que faciliten la mediación y negociación para plantear, organizar y realizar tareas colaborativas. Todas estas tareas han de responder de forma dinámica a las necesidades de los diferentes modos y situaciones de trabajo y experimentación: individual o colectivo, en el aula, en el laboratorio, en la naturaleza, que ofrezcan flexibilidad para diseñar y llevar a cabo actividades que se adapten a las necesidades de cada grupo favoreciendo el mantenimiento y desarrollo de la “memoria colectiva” de la comunidad.

A lo largo del desarrollo del proyecto ENLACE se ha incorporado al equipo investigador expertos del dominio medioambiental y del campo educativo para cubrir los aspectos vinculados a nuevas formas de enseñar y aprender, de forma que la innovación tecnológica suponga también innovación pedagógica en convergencia con las necesidades pedagógicas y sociales expresadas en los objetivos planteados, y que esta consistencia pueda evaluarse en cada ciclo de desarrollo del proyecto.

2.5.1 Actividades de Aprendizaje de ENLACE

Las actividades en ENLACE están pensadas para ser desarrolladas a lo largo de un amplio período de tiempo que puede abarcar todo un semestre o incluso varios cursos académicos. Los alumnos realizan estas actividades en distintos espacios de aprendizaje como pueden ser el aula de clase, el laboratorio de ordenadores del colegio, su propia casa, una visita a un entorno natural o una visita a un museo (Verdejo, Celorrio, Lorenzo, Ruiz, & Sastre, 2007). Estos espacios de aprendizaje cuentan con distintos elementos y artefactos que dependen del entorno en el que se encuentran los alumnos y de la propia actividad que tienen que desempeñar.

El flujo de actividades viene secuenciado por una pregunta que sirve de hilo conductor a todo el conjunto de tareas que los alumnos tendrán que realizar. A modo de ejemplo, la profesora puede hacer una pregunta del estilo de “¿Por qué este animal sólo vive en este determinado ecosistema en verano?” que desencadenará todo un conjunto de actividades que servirá de pretexto para estudiar las características naturales y el hábitat de la fauna y flora de la zona.

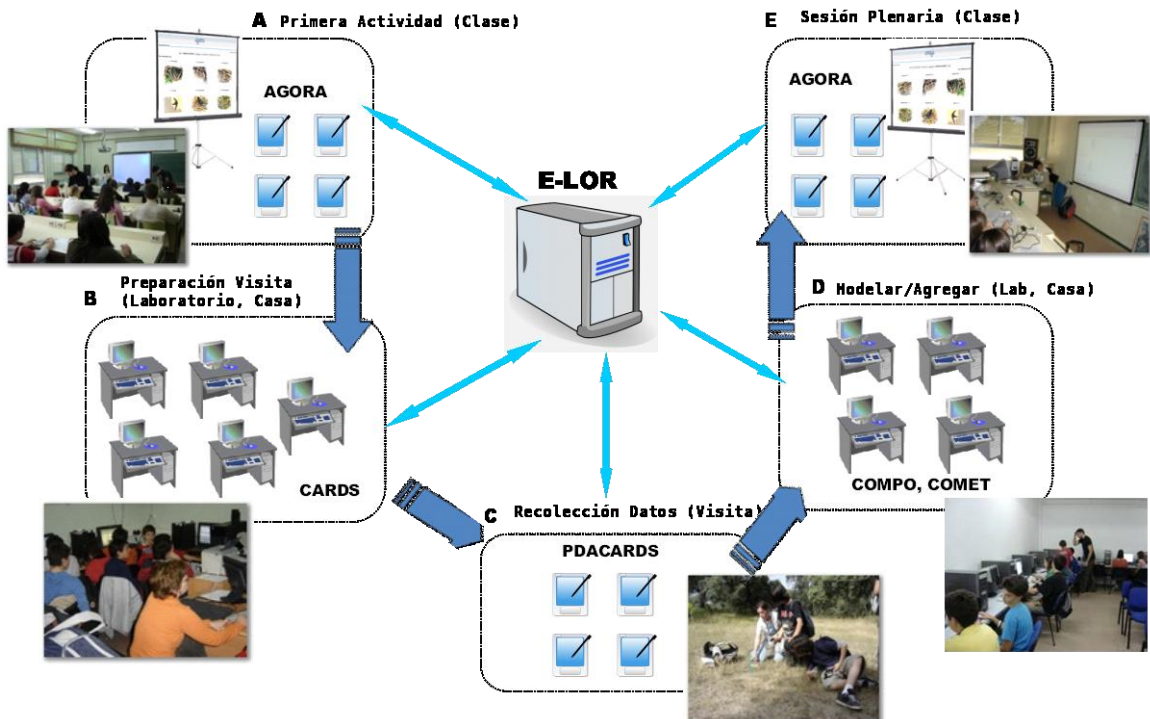


Figura 2.2 Flujo de actividades de ENLACE

En la *Figura 2.2* se puede apreciar a modo de ejemplo un flujo completo de actividades de ENLACE a lo largo de 5 escenarios etiquetados de la A a la E:

- A. Los alumnos realizan una actividad inicial para tomar contacto con los dispositivos móviles que van a utilizar a lo largo del resto de futuras actividades.
- B. Antes de efectuar una salida al campo los alumnos tienen que diseñar unas hipótesis para responder a la pregunta y un plan de trabajo para poder demostrarlas, que incluye posiblemente una división de tareas dentro del grupo.
- C. Los alumnos realizan una visita al campo con el fin de observar de primera mano y tomar datos de los elementos a estudiar.
- D. Una vez de vuelta en clase, los alumnos realizan simulaciones y agregan datos recogidos para formar modelos que den soporte a sus hipótesis.
- E. Finalmente, los alumnos presentan sus modelos generados en una sesión plenaria y reflexionan sobre los de sus compañeros de otros grupos.

2.5.2 La Infraestructura Tecnológica de ENLACE

Una importante aportación de ENLACE es la completa red tecnológica educativa compuesta por un conjunto de herramientas y servicios. Las herramientas desarrolladas son (Verdejo et al., 2009):

- *Agora*. Un sistema de votación diseñado para trabajar con dispositivos móviles y una pantalla de visualización general para la clase.
- *Cards*. Una herramienta web para diseñar y rellenar formatos de datos que pueden ser exportados a otras herramientas.
- *PDACards*. Una herramienta para PDAs que permite rellenar en cualquier lugar formatos de recolección de datos previamente creados con *Cards*.
- *Pelican*. Una plataforma para el soporte del aprendizaje colaborativo y la gestión de usuarios y actividades.
- *Compo*. Una herramienta web para agregar datos y realizar colaborativamente composiciones sobre imágenes usando dispositivos móviles.

- *Comet*. Una herramienta colaborativa para crear mapas conceptuales.
- *E-LOR*. Un servicio de repositorio de objetos de aprendizaje.

Debido a la importancia del repositorio E-LOR dentro del funcionamiento de la red tecnológica educativa de ENLACE, vamos a comentar más detalladamente sus características y su papel dentro del proyecto.

2.5.3 El Repositorio de Objetos de Aprendizaje de ENLACE

El flujo de actividades de ENLACE está soportado en concreto por un servicio de repositorio de objetos de aprendizaje o E-LOR (ENLACE Learning Object Repository) (Celorrio & Verdejo, 2008). Este repositorio actúa como una base de datos para almacenar no sólo todos los elementos requeridos por las actividades, sino también todos los artefactos que los alumnos producen durante el aprendizaje. Así pues, todos los resultados que los alumnos generan en cada uno de los escenarios son almacenados como objetos de aprendizaje junto con metadatos que los describen en el E-LOR (véase la Figura 2.2). De esta forma, en sucesivas actividades, los alumnos pueden buscar y recuperar resultados previos propios o ajenos para utilizarlos en estas, construyendo de este modo nuevo conocimiento sobre conocimiento previo de una forma social.

El repositorio E-LOR ofrece comunicación e interoperabilidad con cualquier herramienta externa (incluidas las de ENLACE, obviamente) mediante Servicios Web, mecanismos de auto-llenado de metadatos (Verdejo et al., 2006), junto con una estructura de espacios sociales configurable a las necesidades de los escenarios de aprendizaje y todo un sistema de metadatos extensible.

Una importante aportación relacionada con esto último, es la posibilidad de definir y añadir al repositorio E-LOR de forma declarativa lo que se han denominado *tipos de objetos de aprendizaje*. Un *tipo de objeto de aprendizaje*, como se ha mencionado previamente, es un mecanismo que sirve para describir por un lado qué clase de contenido albergará el objeto, para poder establecer una asociación entre los objetos y herramientas indicadas para poder manipularlos, y, por otro lado, el catálogo de metadatos que caracterizará a todos los objetos de dicho tipo. Se trata pues de un mecanismo muy potente incluido dentro de E-LOR, ya que gracias a los *tipos de objetos de aprendizaje* es posible definir un perfil de aplicación específico y adaptado a las necesidades concretas de cada herramienta, pudiendo incorporar de esta

manera información de dominio útil a los objetos para su búsqueda y manipulación.

2.6 Sistemas Adaptados al Contexto

Los sistemas adaptados al contexto (del inglés *Context Aware Systems*) son capaces de adecuar sus operaciones al contexto actual sin necesidad de intervención explícita del usuario, aumentando por tanto su usabilidad y efectividad al tener en cuenta características del entorno. Particularmente, cuando se trata de dispositivos móviles, es bastante conveniente que los programas y los servicios puedan reaccionar específicamente no sólo a la localización actual del usuario, sino a los elementos externos que le rodean y a su estado, tanto en términos sociales como de actividad.

Una de las primeras aplicaciones adaptadas al contexto que se conocen es la introducida por (Want, Hopper, Falcao, & Gibbons, 1992), en el que usaba tecnología infrarroja adherida a una chapa que llevaban los usuarios para determinar sus posiciones dentro del edificio y poder así redirigir las llamadas de teléfono a la habitación donde se hallasen. Aunque en los comienzos de estas aplicaciones una de las principales características contextuales era la localización, pronto surgieron nuevos intentos de capturar distintos tipos de información contextual útil para los sistemas. Esto enlaza con una de las tareas más controvertidas para los investigadores de este campo, que no es sino la propia definición de ‘contexto’.

En la literatura el término *context-aware* aparece por primera vez en (Schilit & Theimer, 1994), donde los autores describen el contexto como la localización de uso, el conjunto de personas cercanas y los dispositivos accesibles, además de los cambios a dichos objetos con respecto al tiempo. Muchos otros autores han seguido este método de definición de contexto mediante la enumeración de elementos. (Schmidt, Beigl, & Gellersen, 1999) por ejemplo separa el contexto en factores humanos que incluye al propio usuario, su entorno social y sus tareas, y en factores físicos que incluyen las condiciones del entorno, de infraestructura y la localización. Por otro lado, (Zimmermann, Lorenz, & Oppermann, 2007) ofrece una estructuración en cinco categorías: individualidad, actividad, localización, tiempo y relaciones.

No obstante, dos buenas definiciones de contexto desde un punto de vista del sistema que lo va a utilizar son las que se refieren al contexto en

términos de “los elementos del entorno del usuario que el ordenador tiene conocimiento” (P. J. Brown, 1996) o de “cualquier información que puede ser usada para caracterizar la situación de entidades (cualquier persona, lugar o objeto) que es considerada relevante para la interacción entre el usuario y la aplicación, incluyendo al usuario y la aplicación en sí mismos” (Dey, 2001).

Entre los aspectos a tener en cuenta a parte de la definición están temas como de dónde extraerá el sistema el contexto que utilizará y cómo lo procesará. Las fuentes de extracción del contexto son muy variadas e incluyen desde múltiples sensores situados en dispositivos y habitaciones, hasta información obtenida por los propios usuarios. En cuanto a cómo se procesará esta información de contexto va muy ligado íntimamente a la forma en la que se modelará dicho contexto.

2.6.1 Modelado del Contexto

Un modelo de contexto es necesario para poder definir, manipular y almacenar la información contextual de forma procesable por una máquina. El modelado del contexto es un factor muy importante cuando se quiere dotar a un sistema de mecanismos de consciencia del contexto (*context-awareness*) y es necesario poder identificar todos los aspectos que se desea incluir en el sistema para que éste sepa interpretarlos y poder reaccionar en consecuencia. Desarrollar ontologías de contexto suficientemente flexibles y útiles para proporcionar un amplio rango de contextos es una tarea compleja. (Strang & Linnhoff-Popien, 2004) hacen una sistematización y evaluación de los métodos de modelado del contexto más relevantes en función de las estructuras de datos usadas para representar e intercambiar información contextual en el sistema:

MODELOS CLAVE-VALOR. Representan la estructura de datos más simple para modelar información contextual. Debido a su simplicidad son modelos fáciles de utilizar, pero la falta de mecanismos formales y de validación los hace inadecuados para mantener e inferir información de ellos.

MODELOS DE ESQUEMA DE MARCADO. Aprovechan la capacidad expresiva y declarativa de los lenguajes de marcado para ofrecer estructuras jerárquicas de contenido y atributos. Los *perfiles* representan típicos modelos representados mediante esquema de marcado, normalmente codificados en RDF/S (W3C, 2004a), pudiéndose encontrar distintos ejemplos en (Strang & Linnhoff-Popien, 2004).

MODELOS GRÁFICOS. Emplean representaciones gráficas basadas en UML para modelar los elementos de contexto, aunque a menudo se necesita realizar ciertas extensiones con el fin de aumentar la expresividad para definir los elementos contextuales. Un ejemplo es la extensión llamada ORM (Object-Relation Modeling) de (K. Henricksen, J. Indulska, & Rakotonirainy, 2003). Aunque altamente expresivos, estos modelos son normalmente usados con propósito de ser tratados por humanos y como paso previo a un mapeo a otro tipo de modelo.

MODELOS LÓGICOS. Tienen un alto grado de formalidad. Expresiones lógicas, hechos y reglas son utilizados para definir el modelo de contexto. Un sistema con un motor de inferencia de lógica formal se suele emplear para agregar, actualizar y borrar hechos, además de derivar nuevos hechos. Uno de los primeros en emplear modelos lógicos para definir el contexto fue (McCarthy & Buvac, 1998).

MODELOS ORIENTADOS A OBJETOS. Utilizan técnicas comunes orientadas a objetos (herencia, reusabilidad, encapsulación, polimorfismo...) para definir los elementos de contexto como clases e instancias para representar datos contextuales concretos. Además este tipo de modelos favorecen la distribución y compartición gracias a la encapsulación de los datos contextuales como objetos y una validación parcial a nivel de compilador. Es común que muchos de los tipos de modelos aquí presentados acaben siendo mapeados a un lenguaje de programación orientado a objetos para poder tratarlos de forma programática.

MODELOS ONTOLÓGICOS. Por último, los modelos basados en descripciones ontológicas representan el contexto en forma de conceptos y relaciones, y se tratan de una aproximación con alto nivel de formalidad y expresividad, que permiten fácilmente la aplicación de distintas técnicas ontológicas de razonamiento. Un buen ejemplo de este paradigma es COBRA-ONT (H. Chen, Finin, & Joshi, 2004a), que define una modelo contextual para ‘espacios inteligentes’ y que está basado en el lenguaje ontológico OWL (W3C, 2004b).

Como conclusión a la evaluación de todos estos tipos de modelos, (Strang & Linnhoff-Popien, 2004) determinan que el modelo más expresivo y que puede cumplir la mayoría de expectativas es el ontológico, seguido de cerca del orientado a objetos. Sin embargo habitualmente en la práctica es fácil observar sistemas que hacen un uso combinado de varias de estas aproximaciones (Baldauf, Dustdar, & Rosenberg, 2007).

(G. Chen & Kotz, 2000) ofrecen una lista muy completa de ejemplos de aplicaciones de tipo general que usan información contextual, sin embargo, dentro de la literatura también podemos encontrar distintos casos de uso del contexto con el fin de mejorar la experiencia educativa.

2.6.2 Sistemas Educativos Adaptados al Contexto

Se considera de forma general que un entorno de aprendizaje es adaptativo si es capaz de: monitorizar las actividades de sus usuarios, interpretarlas en base a modelos específicos de dominio, inferir requerimientos y preferencias del usuario a partir de estas actividades interpretadas, representar éstos en los modelos asociados y, finalmente, actuar en consecuencia utilizando el conocimiento disponible sobre los usuarios, con el fin de facilitar los procesos de aprendizaje (Paramythis & Loidl-Reisinger, 2004). Observándolos con esta perspectiva general, los entornos de aprendizaje adaptativos realizan una adaptación del sistema básicamente en correspondencia a un modelo del usuario. Dentro de los elementos que pueden ser adaptados encontramos:

- *Interacción.* Refiriéndose a la adaptación que tiene lugar en la interfaz del sistema para facilitar o dar soporte a la interacción entre el usuario y el sistema.
- *Contenido.* Tomando en cuenta las técnicas adaptativas para descubrir y ensamblar contenido pedagógico para el alumno.
- *Navegación.* Buscando la adaptación en la secuenciación del contenido y la capacidad de recomendar cursos o actividades adaptadas.
- *Soporte a la Colaboración.* Dando apoyo a la comunicación entre múltiples personas, y por tanto a la interacción social, con el fin último de potenciar la cooperación y colaboración.

Ahora bien, concretando estos elementos en sistemas que den soporte al aprendizaje móvil todas estas formas de adaptación pueden ser potenciadas, debido a una mayor presencia de características contextuales para la adaptación, haciendo necesario no únicamente un modelo de usuario, sino una generalización de éste hacia un modelo de contexto en el que se amplíe la información acerca del usuario situándolo en un lugar y un momento, y que se añadan nuevas dimensiones que consideren dispositivos y conectividades empleadas (Drira, Tirellil, Laroussi, Derycke, & Benghezala, 2006). La interacción y la presentación de contenidos será necesario que estén adaptados al tipo de dispositivo (tamaño de pantalla, interfaz de entrada...) y en función de

las capacidades de éste ciertos contenidos no podrán ser utilizados (Trifonova & Ronchetti, 2004). También, los dispositivos móviles a menudo presentan periodos de desconexión tanto intencionalmente, cuando por ejemplo la conexión puede resultar costosa en términos económicos, o no, cuando no hay infraestructura de red disponible o su presencia es intermitente (Goh & Kins-huk, 2002), por lo que hay que tener en cuenta mecanismos de filtrado de elementos multimedia pesados o realizar transferencias de los contenidos de forma previa a las actividades cuando exista conexión. La secuenciación de contenidos y recomendación de actividades también puede ser contextualizada tomando en consideración características como por ejemplo la localización del estudiante (si está en un laboratorio o en su casa) o al tiempo (si tiene 20 minutos para repasar antes de un examen).

Una aproximación empleada en distintos prototipos de sistemas para el aprendizaje móvil es utilizar la información de contexto para seleccionar contenido al alumno y facilitar recomendaciones dependiendo de la posición donde éste se encuentre. En el proyecto *MobiLearn* (Lonsdale, Baber, Sharples, & Arvanitis, 2004) se emplea la información de situación del estudiante dentro de la sala de un museo junto con la tarea que está intentando resolver para poder proporcionarle una lista de recomendaciones y recursos multimedia en referencia al artefacto del museo frente al que esté. Otro ejemplo similar se aprecia en JAPELAS (Ogata & Yano, 2004), un sistema de aprendizaje del lenguaje que proporciona expresiones educadas del japonés dependiendo de la información social y situación del alumno.

Como se ha comentado anteriormente, el contexto puede servir para recomendar actividades en función del tipo de usuario, su localización, el momento actual, el dispositivo que esté utilizando, etcétera. El sistema de (Yau & Joy, 2007) toma en consideración las preferencias y la agenda de eventos del estudiante para seleccionar contenidos y planificarle actividades en función del tiempo que dispone. Otro buen ejemplo es el sistema desarrollado por (Martín & Carro, 2009). Esta aplicación para el aprendizaje móvil guía a los estudiantes a través de un conjunto de actividades, adaptándose a distintos tipos de aprendizaje y pudiendo ser utilizada de forma individual o colaborativa. Dependiendo de las preferencias de aprendizaje, el nivel de los usuarios, sus conocimientos previos, el lugar donde se encuentren y el tiempo del que disponen, el sistema es capaz de filtrar y estructurar las actividades a realizar en base a un conjunto de reglas estructurales y de contexto.

Por otro lado, el contexto se puede emplear para ofrecer recomendaciones de personas con las que interactuar. (Schmidt, 2007) define información acerca de relaciones sociales entre personas para poder recomendar expertos que puedan ayudar de forma contextualizada. Otro método es el uso de información contextual social con el fin de guiar y orientar en nuevos entornos. El sistema *ActiveCampus* (William et al., 2003) en el cual los miembros de la comunidad universitaria tienen acceso a información contextual de otros miembros para ponerles en contacto y ayudarles en su día a día por el campus.

Se han presentado diversas arquitecturas conceptuales para dar soporte al aprendizaje móvil mediante el contexto (Schmidt, 2007; Drira et al., 2006; Wang, 2004; Trifonova & Ronchetti, 2004), siendo una característica común entre la mayoría de ellas el presentar una estructuración centralizada, habitualmente basada en modelos web cliente-servidor. Cabe destacar que una importante propiedad que busca nuestra propuesta con respecto de estas arquitecturas y el conjunto de sistemas que hemos tratado en este apartado, es que nuestra propuesta busca una *solución distribuida*, en contraposición a soluciones de carácter centralizado, con el fin de aportar flexibilidad a la hora de definir y desarrollar espacios de aprendizaje.

2.7 Tecnología de Agentes

Los agentes son una tecnología surgida de la Inteligencia Artificial que utiliza el concepto de agente como componente principal para el diseño y desarrollo de sistemas inteligentes y distribuidos.

En este apartado vamos a profundizar en la noción de agente inteligente que utilizaremos en este trabajo, situaremos la tecnología de agentes dentro del contexto educativo y haremos un repaso de las plataformas de Sistemas Multi-Agente, con énfasis en las que permiten utilizar dispositivos móviles.

2.7.1 Agentes Inteligentes

En términos generales, un agente es una entidad que actúa, o tiene la autoridad para actuar, buscando satisfacer los intereses de otra. De la misma forma, en términos de las ciencias de la computación un agente es una entidad computacional que actúa en favor de un usuario humano, entidad softwa-

re u otro agente. Los agentes nacen de la investigación en el terreno de la Inteligencia Artificial, y más concretamente dentro del campo de la Inteligencia Artificial Distribuida. Es difícil encontrar una única definición consensuada de agente inteligente ya que de hecho el término agente siempre ha sido bastante controvertido en la literatura. Distintos autores (Wooldridge & Jennings, 1995; Franklin & Graesser, 1997) han optado por dar una lista de características que debe exhibir un agente para poder considerarse inteligente:

- *Autonomía.* Los agentes operan sin la intervención directa de cualquier otra entidad ya que poseen cierto control sobre sus acciones.
- *Habilidad Social.* Los agentes se comunican y colaboran con otras entidades, incluidos otros agentes.
- *Reactividad.* Los agentes perciben el entorno y reaccionan a los cambios de éste de forma apropiada.
- *Proactividad.* Los agentes muestran un comportamiento orientado a objetivos por el cual son capaces de tomar la iniciativa.
- *Capacidad de inferencia.* Los agentes pueden tomar decisiones basándose en su conocimiento actual sobre ellos mismos, su entorno y sus objetivos generales.
- *Persistencia temporal.* Los agentes tienen atributos a modo de identidad y de estado interno que se mantienen y evolucionan durante el tiempo.

Se puede afirmar que tampoco existe un consenso general sobre en qué medida estas habilidades han de estar presentes dentro del agente, y que, dependiendo del tipo de aproximación utilizado para diseñar e implementar el agente, algunas características estarán más presentes que otras. Si bien es cierto que habitualmente estos son los elementos que comúnmente se consideran aceptados para poder considerar a un agente como ‘inteligente’, otras habilidades complementarias son observables:

- *Personalidad.* Un agente puede mostrar cualidades que muestran un carácter propio.
- *Adaptabilidad.* Un agente puede aprender basándose en la experiencia y modificar su comportamiento futuro en base a ello.

- *Movilidad.* Un agente puede migrar, transportando su comportamiento y estado interno, del dispositivo o nodo en el que reside a otro.

Un agente que posee esta última habilidad recibe el nombre de *agente móvil* y aunque éstos muestran atractivas ventajas (Lange & Oshima, 1999) son también cuestionados debido a los problemas de seguridad que presentan y a que dichas ventajas pueden conseguirse mediante otros medios (Chess, Harrison, & Kershenbaum, 1997). Cabe destacar que un agente móvil no tiene nada que ver con el tipo de dispositivo en el que reside. Es decir, que un agente no es móvil por residir en una PDA o en un teléfono móvil, sino por su capacidad de migrar entre dispositivos, sean éstos móviles o no.

No obstante, aunque como hemos dicho no existe una definición consensuada de qué es un agente, en este trabajo vamos a optar por una visión del agente como sistema que tomaremos prestada de la Inteligencia Artificial clásica. Desde un punto de vista clásico de la Inteligencia Artificial, un agente puede verse como un sistema que en vez de trabajar dentro del *nivel simbólico* (es decir, mediante expresiones, fórmulas, variables...), es capaz de trabajar en el *nivel de conocimiento* (Newell, 1982). Los componentes dentro del nivel de conocimiento se caracterizan por tener un grado de abstracción mayor y se dividen en *objetivos y acciones*. De esta forma, un agente está compuesto por un conjunto de acciones que es capaz de realizar y un conjunto de objetivos que le es necesario satisfacer. El medio dentro del *nivel de conocimiento* es, como apunta Newell, obviamente el conocimiento. Finalmente, la principal ley de comportamiento que gobierna al agente es el denominado *Principio de Racionalidad*:

Las acciones de un agente son seleccionadas para lograr satisfacer los objetivos de dicho agente.

Una característica importante es que el agente normalmente no dispone información total de lo que pasa a su alrededor, sino que dispone de conocimiento parcial del mundo, de forma similar a lo que le ocurre a cualquier persona. Con el fin de cumplir el *Principio de Racionalidad*, el agente es capaz de utilizar el conocimiento que tiene sobre sí mismo y el entorno que le rodea para poder seleccionar las acciones más apropiadas.

2.7.2 Agentes Inteligentes en la Educación

Son diversas las aportaciones en la literatura que usan los agentes dentro del campo de la educación, algunas enfocando más la parte de la teoría pedagógica y otras resaltando más la parte técnica.

Desde el punto de vista más pedagógico, (Jafari, 2002) da un enfoque generalista en el que podría considerarse a los agentes como elementos integrados dentro del aprendizaje y la enseñanza, aportando ejemplos de agentes que podrían ayudar a los profesores y estudiantes como un ‘Agente Profesor Asistente’, que asista al profesor en las tareas pedagógicas y agregue información sobre el progreso de los alumnos, un ‘Agente Tutor’, capaz de encontrar y recomendar contenidos al alumno, y un ‘Agente Secretario’, que ayude con las tareas burocráticas. De un modo similar, (Gregg, 2007) habla de un hipotético entorno de enseñanza virtual en el cual un conjunto de agentes son utilizados para conseguir un aprendizaje adaptado y personalizado. Entre los agentes que introduce el autor, todo siempre desde una perspectiva teórica, se encuentran distintos agentes para seleccionar y personalizar la distribución de contenidos en función del alumno, junto con un ‘Agente de Colaboración’ capaz de ponerle en contacto mediante chat con otros alumnos.

Desde un punto de vista más tecnológico, varias aportaciones utilizan la tecnología de agentes para aportar nuevas formas de implementar sistemas educativos ya existentes. Un ejemplo es la utilización de los sistemas multiagentes para implementar Sistemas de Tutor Inteligentes (Capuano, Marsella, & Salerno, 2000; Gascueña & Fernández-Caballero, 2005).

Por otro lado, varios sistemas se alinean más en consonancia con elementos presentes en nuestro trabajo, como son los agentes y utilización del contexto en dispositivos móviles. (L. T. Kinshuk, 2004) aboga por el uso de agentes móviles, que puedan moverse entre los dispositivos clientes y el servidor, para dar soporte a entornos de aprendizaje móvil. (Esmahi & Badidi, 2004) presentan una arquitectura que cuenta con repositorios de perfiles de usuarios, perfiles de dispositivos y de contenidos educativos y que son empleados por diversos agentes móviles para ofrecer a los alumnos contenidos personalizados. (Ko, Hur, & Kim, 2005) presentan un sistema basado en agentes que pretende ser empleado para el aprendizaje del idioma inglés utilizando dispositivos móviles. En (Graf et al., 2008) se discute acerca de una infraestructura de agentes para desarrollar actividades basadas en problemas

dentro de entornos con dispositivos móviles. No obstante, se ofrecen pocos detalles acerca del escenario pedagógico de aplicación y los detalles técnicos del sistema, al comentarse que está en fase de implementación. De la misma forma, *GlobalEdu* (D.N.F. Barbosa et al., 2006; Nino et al., 2007) se presenta como una arquitectura adaptada al contexto para el aprendizaje ubicuo, pero tanto su modelo de contexto como la arquitectura de agentes se comentan muy superficialmente en la literatura que hay sobre ella. Esto desgraciadamente es común en la mayoría de las propuestas de desarrollo de sistemas basados en agentes, las cuales bastantes veces no llegan a implementarse o se quedan sólo en prototipos muy básicos.

Uno de los principales retos a los que se enfrenta la investigación en este campo es el modelado de los agentes pedagógicos, junto con su funcionalidad y estructuras internas, apareciendo todos estos elementos muy vagamente descritos en la literatura en general de los sistemas basados agentes para la educación y en concreto de los sistemas basados en agentes para el aprendizaje móvil. Uno de los objetivos de nuestra propuesta es incorporar modelos de agentes que describan su funcionalidad y mecanismos internos de forma detallada.

2.7.3 Sistemas Multi-Agente y Plataformas

Los agentes raramente existen de forma aislada en las aplicaciones, sino que usualmente forman una comunidad de agentes que recibe el nombre de Sistema Multi-Agente o SMA (en inglés *Multi-Agent System* o *MAS*). La tecnología de agentes ofrece todo su potencial cuando se diseñan y desarrollan sistemas en los cuales distintos agentes colaboran entre sí para perseguir los objetivos del sistema.

Desde un punto de vista de desarrollo, diseñar aplicaciones con una perspectiva SMA ofrece diversas ventajas. Una primera de estas ventajas fomentada por el uso explícito del agente como componente principal de diseño es que los sistemas orientados a agentes cuentan con una separación bien definida de componentes. Una aproximación orientada a agentes facilita intrínsecamente la separación de módulos funcionales (es decir, agentes) que pueden ser implementados por separado, resultando por lo tanto en un diseño más claro y una organización de los equipos de desarrollo más sencilla (Fabio Bellifemine, Giovanni Caire, Agostino Poggi, & Giovanni Rimassa, 2008). En segundo lugar, el agente funciona como una unidad de encapsulación a un nivel de abstracción mucho mayor que, por ejemplo, los objetos en la Progra-

mación Orientada a Objetos. Un agente encapsula una parte bien definida del control de la aplicación y se describe mediante objetivos, acciones y creencias del agente. Este grado de abstracción facilita la comunicación entre clientes y desarrolladores al proveer conceptos de alto nivel para realizar el diálogo, facilitando el análisis de requisitos y el rediseño. Por otro lado los SMA son diseñados con un inherente carácter distribuido en mente. Los componentes de un SMA pueden ser repartidos entre distintos servidores, garantizando de este modo la flexibilidad y la escalabilidad del sistema.

Los agentes software ofrecen un acercamiento alternativo al modelado e implementación de soluciones software. Cualquier aplicación que puede ser realizada usando acercamientos tradicionales también puede ser realizada utilizando agentes inteligentes. No obstante, una duda que puede asaltar es por qué, gozando de tan buena consideración dentro de la comunidad científica, la Programación Orientada a Agentes todavía no se ha consolidado dentro del mundo industrial y empresarial. A pesar de la multitud de aplicaciones que se describen en la literatura, los agentes todavía no se han usado en software convencional con la extensión con la que han sido originalmente contemplados. Las razones que expliquen esto pueden ser distintas, pero el problema clave es que la comunidad investigadora de agentes todavía no ha demostrado las circunstancias en las cuales una solución basada en agentes es mejor que los acercamientos convencionales. Esto junto con la falta de fuerza de desarrollo de herramientas y metodologías ha hecho que las aplicaciones basadas en agentes sean sólo un nicho reducido en la industria software.

No obstante, esto tiende a cambiar recientemente con la puesta en marcha de propuestas para ofrecer una visión de ingeniería del desarrollo basado en agentes, como por ejemplo con metodologías para el diseño de sistemas multi-agente como INGENIAS (Jorge Gómez-Sanz, 2002; Jorge J. Gómez-Sanz, Juan Pavón, & Francisco Garijo, 2002), que junto con herramientas y plataformas para el desarrollo hagan que la tecnología de agentes esté encaminándose hacia lo que se podría considerar una Ingeniería Software Orientada a Agentes.

PLATAFORMAS DE AGENTES

Hay toda una miríada de diferentes plataformas para el desarrollo mediante SMA, y cada día aparece una nueva. No obstante, dos de las más conocidas son JADE (F. Bellifemine, A. Poggi, & G. Rimassa, 1999) y FIPA-OS (Poslad, Buckle, & Hadingham, 2000). Ambas plataformas están profunda-

mente basadas en los estándares FIPA para la comunicación entre agentes. FIPA (O'Brien & Nicol, 1998) es una organización que promueve la estandarización de las tecnologías basadas en agentes y la interoperabilidad entre estas tecnologías. Aunque FIPA tiene distintas iniciativas encaminadas a estandarizar las plataformas de agentes con temas como la definición de una arquitectura estándar (FIPA, 2002a) y la gestión de agentes en ésta (FIPA, 2002b), la categoría clave en la que principalmente basa todos sus esfuerzos es la definición de la estructura de mensajes y de protocolos de comunicación para la interoperabilidad entre agentes de distintas plataformas (FIPA, 2002c, 2002d).

Tanto JADE como FIPA-OS poseen sendas versiones ligeras diseñadas para trabajar con dispositivos móviles: JADE-LEAP y MicroFIPA-OS. Por un lado, JADE-LEAP (*Lightweight Extensible Agent Platform*) (Fabio Bellifemine et al., 2008) fue la primera plataforma FIPA que funcionaba en PDAs y teléfonos móviles. Para la comunicación en entornos inalámbricos, la plataforma LEAP usa un protocolo llamado JICP para la comunicación intra-plataforma. MicroFIPA-OS (Laukkanen, Tarkoma, & Leinonen, 2002) por el otro lado, es una plataforma de desarrollo de agentes basada en la FIPA-OS. El sistema está diseñado para trabajar en dispositivos móviles de recursos medios y altos como PDAs.

Una consecuencia directa de las plataformas de agentes basadas en FIPA como JADE y FIPA-OS, y sus extensiones LEAP y MicroFIPA-OS, es que tienden a establecer el foco en los estándares y la infraestructura de comunicación. Sin embargo, no ofrecen soporte para el desarrollo de las entidades de comunicación en sí, es decir, los agentes. De este modo, el analista carece de modelos de diseño y patrones a seguir a la hora de desarrollar un sistema multi-agente cuando el desarrollo sale fuera del ámbito de la comunicación entre agentes. ICARO-T es una plataforma que ofrece una solución a este problema ofreciendo patrones y componentes reusables. Al ser el pilar base para el desarrollo de la arquitectura multi-agente de nuestra propuesta, ICARO-T es presentado con detenimiento en el siguiente apartado.

2.8 La Plataforma ICARO-T

ICARO-T (ICARO-T, 2008) es una plataforma para el desarrollo de sistemas multi-agente de Telefónica I+D (TID) que aporta una arquitectura

para el desarrollo de aplicaciones basada en agentes (Francisco J Garijo, Spina, & Polo, 2009). Esta plataforma MAS ha sido usada para desarrollar varias aplicaciones industriales distribuidas (Francisco J. Garijo, Bravo, Gonzalez, & Bobadilla, 2004) totalmente operativas en máquinas de bajo costo. Como resultado de la experiencia acumulada, la plataforma ICARO-T ha elaborado, refinado y validado no únicamente conceptos y modelos, sino también patrones arquitectónicos y componentes software flexibles y reusables que están íntegramente documentados en UML (Pavon, F. Garijo, & Gomez-Sanz, 2008). La arquitectura del sistema fuerza el uso de estos patrones con la intención de guiar al desarrollador en la identificación de las características relevantes del sistema.

ICARO-T ofrece cuatro categorías de modelos de componentes reutilizables:

1. Modelos de organización para describir y administrar la estructura global de la aplicación.
2. Modelos de agentes para describir los principios de comportamiento interno de los agentes.
3. Modelos de recursos para encapsular componentes que dan servicio a los agentes
4. Entidades básicas de computación para construir nuevos modelos de agentes y modelos de recursos. Esta categoría incluye tipos abstractos de datos, librerías especializadas, ontologías de domino, procesadores de reglas, etc.

La organización de ICARO-T está dividida en dos capas distintas: la capa de control, que está compuesta por componentes agente, y la capa de recursos, que incluye recursos de visualización (como interfaces de usuario), de persistencia (como bases de datos), de registro y de comunicación de agentes. Los agentes son los controladores de la aplicación, y los componentes recurso son usados por éstos para ayudarles a cumplir sus objetivos. Dentro de la capa de control de ICARO-T se distingue entre gestores y especialistas. Los gestores son componentes agente responsables de los aspectos de administración del sistema, mientras que los especialistas están a cargo de alcanzar la funcionalidad del sistema en sí. La organización de ICARO-T está estructurada de manera jerárquica mediante tres agentes gestores: el Gestor de Agentes, que supervisa a los agentes especialistas, el Gestor de Recursos, que supervisa el correcto funcionamiento de los componentes recurso, y el Gestor de la Orga-

nización, que es responsable de la creación y supervisión de los dos anteriores. Los elementos de la organización pueden ser distribuidos a lo largo de diferentes nodos, sin embargo, todos los nodos son dependientes de los nodos que contienen a los gestores.

ICARO-T proporciona al desarrollador dos tipos diferentes de modelo de agente, un patrón de agente reactivo y un patrón de agente cognitivo. El patrón de agente reactivo puede ser usado para el desarrollo de agentes con un comportamiento simple. Los elementos de percepción han de ser identificados en este patrón además de las acciones asociadas a realizar dependiendo del estado mental del agente, este último descrito mediante una máquina de estados finita. En contraposición al patrón reactivo, el patrón de agente cognitivo ofrece un comportamiento más complejo y potente que permite inferencia de conocimiento y razonamiento. El desarrollador ha de concentrarse en la definición de objetivos, tareas y el proceso cognitivo que encadena dichas tareas para resolver los objetivos. El motor de inferencia del patrón cognitivo está construido sobre un motor de reglas.

En resumidas cuentas, ICARO-T ofrece una plataforma de desarrollo muy adecuada para conseguir un desarrollo software altamente componentizado y reutilizable. Busca promover buenos hábitos en el desarrollo de sistemas multi-agente y además incluye armazones software preparados para ser empleados. Es por tanto que el elemento diferenciador de ICARO-T respecto a otros frameworks o herramientas de desarrollo basadas en agentes es considerar un agente como un componente software que se instancia a partir de un patrón. ICARO-T permite definir nuevos patrones de agentes y proporciona normas de ingeniería sencillas para su diseño, su integración en la infraestructura y su posterior reutilización.

PARTE II

Solución Propuesta

3

HACIA UNA ARQUITECTURA DISTRIBUIDA PARA EL APRENDIZAJE UBICUO

Con el fin de alcanzar los objetivos descritos en el apartado 1.1 de esta memoria, se presenta la siguiente propuesta de arquitectura distribuida basada en agentes para el soporte y el desarrollo de actividades de aprendizaje en escenarios ubicuos y con dispositivos móviles. Esta propuesta ha sido refinada a lo largo de distintos trabajos previos (Celorrio & Verdejo, 2007; Verdejo & Celorrio, 2007), dando como resultado la solución arquitectónica que expondremos durante la Parte II de esta memoria de tesis.

A tal propósito comenzaremos este capítulo detallando las decisiones de diseño y desarrollo, con el fin de describir y analizar el escenario que nos ayudará a guiar nuestro trabajo y que motivará la elección de la tecnología de agentes y la plataforma ICARO-T como herramientas para lograr nuestra solución.

Además, en este capítulo vamos a analizar y formalizar las distintas decisiones de diseño y desarrollo que van a dirigir nuestra propuesta. Con la intención de seguir como comentamos en el apartado 1.3 una metodología de *Diseño Basado en Escenarios* y que hemos descrito en el apartado 2.2.1, vamos a proceder a enmarcar nuestro problema mediante un escenario que nos guíe en primer lugar en la realización de un análisis de requisitos. Para ello, se expondrá una descripción narrativa que nos permitirá abstraer los requisitos funcionales necesarios para nuestra propuesta y que servirán para poder formular las elecciones tecnológicas oportunas que estarán presentes a lo largo de nuestro trabajo de investigación.

3.1 Análisis de Requisitos

Dentro del análisis de requisitos del *Diseño Basado en Escenarios*, la definición y las circunstancias del problema son estudiadas a través del contacto con los clientes (es decir, los usuarios y participantes en los escenarios), estudios de campo sobre la situación actual e intercambios de ideas entre los usuarios y los desarrolladores. Toda esta posible información recopilada sirve como entrada para formular los escenarios del problema, que sirven para expresar importantes características acerca de los usuarios, las tareas tanto típicas como críticas en las que están involucrados, las herramientas que usan, junto con el contexto organizativo que envuelve todo lo anterior. El análisis y el refinamiento de estos escenarios se realiza apoyándose en el estudio de los perfiles de usuarios, con el fin de conocer cuáles son sus expectativas y preferencias, además de establecer una serie de declaraciones del problema (*'claims'* en terminología inglesa) que muestran una lista de características importantes relativas a las circunstancias del problema y su impacto en las experiencias de los usuarios. En el proceso de transición del análisis hacia el diseño, todas estas declaraciones van siendo transformadas y derivadas en una lista de rasgos funcionales y características deseables que dan como resultado una especificación de requisitos.

En nuestro caso concreto, el análisis y el diseño de la arquitectura que pretendemos desarrollar van a estar gobernados por un escenario que se ha repetido sucesivas veces a lo largo del proyecto ENLACE y en el cual nos hemos encontrado en contacto directo con los distintos participantes. Tomando por tanto como referencia y punto de partida este escenario de problema, nuestro objetivo es obtener toda una especificación de requisitos fruto del análisis y generalización de éste, y que guíe el desarrollo de nuestra propuesta.

El escenario característico de las actividades de ENLACE que pretendemos usar de guía para nuestro desarrollo es la recolección de datos en el lugar de interés durante una excursión. El espacio de aprendizaje donde se realiza este escenario es habitualmente un entorno informal y fuera del colegio. Sin embargo existe como elemento importante una imbricación con las actividades previas y posteriores que habitualmente son realizadas en entornos más formales o incluso en nuevas salidas.

Los participantes del escenario se encuentran estratificados en tres tipos de perfil, a saber, estudiantes, profesores y monitores. Los estudiantes son los elementos centrales del proceso de aprendizaje y las tareas que han de realizar están orientadas por unos determinados objetivos pedagógicos que pretenden ser alcanzados. Los profesores son los encargados de la autoría y monitorización de dichas tareas y actividades, buscando integrarlas con las clases y el currículo escolar. Por otro lado, durante las salidas a entornos naturales y museos en ENLACE, suele aparecer la figura de otro grupo de usuarios, los monitores, que se encargan de llevar a cabo las salidas, guiando a los estudiantes a través del parque natural o museo.

Cada perfil tiene unas determinadas expectativas puestas en torno al desarrollo del escenario y una serie de preferencias a la hora de llevarlo a cabo, tanto desde aspectos pedagógicos como tecnológicos, influyendo en la usabilidad del sistema. En la Tabla 3.1 pueden apreciarse las características generales de cada uno de estos perfiles para el escenario de salida al campo de ENLACE.

Participante	Características Generales del Grupo
Estudiante	<p><i>Descripción:</i> Los estudiantes son alumnos de Educación Secundaria que tienen como objetivo estudiar la flora y fauna de un parque natural. Para ello recogen datos y muestras que posteriormente usarán para poder modelar fenómenos de interés.</p> <p><i>Expectativas:</i> Ven la actividad como una oportunidad de escapar de la rutina de la clase. El uso de tecnología les aporta motivación extra de cara a las actividades.</p> <p><i>Preferencias:</i> Son capaces de adaptarse a nuevos paradigmas tecnológicos con facilidad, y están acostumbrados a la utilización de dispositivos móviles.</p>
Profesor	<p><i>Descripción:</i> El profesor o maestro es el encargado de diseñar las actividades que los alumnos realizarán tanto antes, durante, como después de la salida.</p> <p><i>Expectativas:</i> Buscan poder aprovechar la salida para encuadrarla en actividades del currículo, de forma que sirva para reforzar el aprendizaje. Quieren poder reutilizar recursos de actividades anteriores, incluyendo cuadernillos de visita preparados por los monitores.</p> <p><i>Preferencias:</i> No poseen necesariamente un alto grado de dominio informático.</p>
Monitor	<p><i>Descripción:</i> Los monitores son miembros de asociaciones que buscan concienciar de problemas medioambientales y fomentar el conocimiento de la Ciencia y la Naturaleza. Son guías que acompañan a los estudiantes durante su visita.</p> <p><i>Expectativas:</i> Facilidad en la autoría de los cuadernillos para la visita y poder utilizarlos. Mecanismos para controlar a los estudiantes durante la salida.</p> <p><i>Preferencias:</i> Se sirve de apoyo de cuadernillos de colección de datos para sus explicaciones. No quieren que la tecnología sea una distracción durante la salida.</p>

Tabla 3.1 Perfil de los participantes en el escenario salida al campo de ENLACE

3.1.1 Escenario de actividad de aprendizaje en ENLACE

Los alumnos de 13 años de primero de la ESO del Instituto de Educación Secundaria Diego Velázquez están particularmente motivados este año en su clase de Ciencias Naturales. Marta, su profesora, está trabajando de forma activa con ellos en un proyecto que durará todo el curso. En este proyecto los alumnos tendrán que trabajar para estudiar y describir distintos ecosistemas naturales con el fin de gradualmente crear un modelo global con todo el conocimiento que vayan adquiriendo sobre la fauna, flora y características de ciertos hábitats. Esto incluirá explorar y observar de primera mano los entornos estudiados para recopilar datos, analizar y representar la información obtenida, además de tener que agregarla de forma que los estudiantes puedan establecer conexiones entre la información (por ejemplo, mediante mapas conceptuales).

Esta semana los alumnos van a hacer una visita a El Monte Del Pardo en Madrid. La visita va a estar dirigida por los guías de SEO/Birdlife, una asociación sin ánimo de lucro para la defensa y conservación del medio ambiente. Tanto los estudiantes como los monitores de SEO van a estar equipados con dispositivos móviles, PDAs en concreto, para el desarrollo de algunas actividades *in situ*.

Marta ha preparado una serie de actividades para realizar durante la visita. Estas actividades tienen como objetivo recopilar información acerca de la fauna y flora de los distintos ecosistemas del lugar, sus condiciones meteorológicas y diferentes medidas físicas del entorno. Para ello, Marta ha creado una serie de formularios digitales de recolección de datos, mediante la aplicación CARDS, y los comparte con sus alumnos subiéndolos al repositorio de objetos de aprendizaje. Aunque durante la visita todos los alumnos cubrirán los mismos lugares, Marta quiere que cada uno se centre en las particularidades de diferentes hábitats. Por lo tanto, Marta divide a los estudiantes en grupos que tendrán que recorrer de forma más detallada o bien el ecosistema de ribera o bien el ecosistema de encinar. A partir de esto, ella animará a sus estudiantes a que planeen su propia visita y a que adapten los formularios de recolección de datos a sus necesidades y al ecosistema que les haya tocado.

Juan, María y Paco forman uno de los grupos responsables del ecosistema de ribera. En la fase de preparación, este grupo ha explorado activamente el tema para determinar la información que necesitarán para este escenario en concreto y las observaciones necesarias para identificarlo y describirlo.

Como resultado de este paso, el grupo modifica y amplía, usando la aplicación CARDS desde la sala de ordenadores del instituto, los formularios de recolección que había puesto a disposición la profesora en el repositorio.

El día anterior a la actividad, la PDA de Juan le muestra una serie de mensajes: “Hola Juan. Parece ser que mañana tienes una actividad de Visita a El Monte Del Pardo. Vas a necesitar la aplicación PDACARDS. También he encontrado una serie de recursos relacionados. ¿Quieres que te lo configure todo para la actividad en esta PDA?” Juan está de acuerdo, por lo que el sistema procede a seleccionar, instalar y configurar las herramientas necesarias para la actividad de mañana, y a personalizar el dispositivo con las colecciones de datos que había diseñado su grupo anteriormente para que esté todo disponible en la PDA.

El día de la visita los estudiantes llevan su PDA pre-configurada y se mueven alrededor de distintos puntos de interés, guiados en todo momento por los monitores. Usan la aplicación PDACARDS para recolectar información que se guarda de forma local en sus dispositivos, ya que no hay conexión a Internet en el campo. No obstante, las PDAs pueden comunicarse entre ellas mediante redes ad-hoc con WIFI y Bluetooth.

Mientras deambula cerca de la orilla del río, María se sitúa cerca de Aurora, miembro de otro grupo que está ensimismada rellenando datos en su PDA. Al estar próximas una de otra, las PDAs de ambas intercambian información de contexto y de repente la PDA de María muestra un mensaje en la pantalla: “Aurora está trabajando en la tarea Describir la Flora de Ribera. Tú tienes que realizar también esta tarea. ¿Quieres colaborar con Aurora?” María responde que sí, y su PDA automáticamente arranca la aplicación PDACARDS con las fichas de recolección de la tarea. Otro mensaje similar aparece en la PDA de Aurora y ambas se ponen a trabajar juntas para describir la flora de ribera.

Eventualmente algunos alumnos, incluido Paco, se juntan alrededor de Celia, una monitora de SEO/Birdlife que va a explicarles las características de la vegetación del bosque y en concreto las propiedades de un arbusto que han encontrado. No obstante, Paco está tan abstraído en la pantalla de su PDA tratando de rellenar datos que realmente no está prestando atención a las explicaciones de Celia. Por lo tanto, la monitora se percata de ello y tras indicar en su propia PDA que está dando una explicación, el sistema manda un mensaje a las PDAs de todos los alumnos de alrededor de Celia que deshabilita PDACARDS durante unos segundos. Paco y varios alumnos más no

tienen otra opción que levantar su mirada de la pantalla y prestar atención a los comentarios de la monitora, descubriendo de esta forma interesantes propiedades del arbusto Torvisco.

Finalmente en otro momento durante la visita, Juan encuentra una extraña y gigantesca pluma en el suelo. Juan se la muestra a Celia y la monitora le explica que pertenece a un tipo de buitre raro de ver en esta zona y que podría ser interesante encontrar más rastros de él. Juan entonces crea una nueva tarea en su PDA llamada 'Encontrar Rastros de Buitre'. Automáticamente la actividad es reconfigurada con una nueva tarea y es distribuida a todos los usuarios que están al alcance. Celia desde su PDA añade a la tarea una ficha con información acerca de los carroñeros, que también será distribuida a todas las PDAs de los alumnos.

Al día siguiente en clase, Marta pide a los alumnos que muestren en la pizarra electrónica la información recolectada durante la visita. En cuanto los alumnos han entrado en clase con su PDA, el sistema ha subido de forma autónoma todos los datos de la visita de ayer al repositorio de objetos de aprendizaje. Cuando a Juan, María y Paco les toca el turno de exponer, simplemente abren su vista de grupo en el repositorio de aprendizaje y muestran las fichas rellenas. Marta también observa que se creó una actividad nueva, con lo cual la clase se dedica a averiguar qué tipo de animal pudo ser a partir los datos recolectados de éste, lo que genera una animada discusión que deriva en una conversación sobre los animales carroñeros.

Finalizada la clase, toda la información recolectada continuará disponible cuando los alumnos necesiten realizar otras actividades en los días próximos como crear mapas conceptuales, agregaciones o simulaciones.

3.1.2 Comentarios acerca del escenario

Echemos una mirada más detenida a este escenario que se ha expuesto con el fin de poder entrever la lista de requisitos que un sistema ha de cumplir para poder aportar una experiencia de aprendizaje integrada de este estilo.

Primeramente, la tecnología debe facilitar la gestión y despliegue de todos los recursos necesarios del sistema a través de los dispositivos empleados. En el escenario, las herramientas de aprendizaje y sus recursos han sido automáticamente instalados y configurados en los dispositivos móviles de los usuarios, previendo actividades futuras de cada alumno en concreto. El siste-

ma ha sido capaz de buscar en el repositorio los objetos de aprendizaje que se requieren para configurar el desarrollo de la actividad y personalizar el dispositivo con otros artefactos útiles en función del usuario y su contexto: grupo al que pertenece, localización de la actividad, infraestructura tecnológica, etc.

Segundo, la tecnología debe hacer hincapié en la interacción entre usuarios para promover la colaboración. Durante la visita, el sistema ha sido capaz de mantener y seguir los contextos de los usuarios con el fin de plantear interacciones entre usuarios en momentos adecuados. Los dispositivos móviles han sido capaces de intercambiar información contextual actualizada acerca de los usuarios con el fin de poder proponer de forma inteligente colaboraciones situadas que inciten al diálogo y a la negociación en las actividades.

En tercer lugar, la tecnología no debe ser un elemento aislador del usuario con respecto al entorno y distraerlo de posibles oportunidades de aprendizaje. En el escenario se pueden observar mecanismos para re-enfocar la atención de los alumnos con el fin de atender cuando la situación lo requiere o dirigir la atención hacia elementos concretos del entorno.

Finalmente, la tecnología ha de ser capaz de hacer uso de oportunidades de aprendizaje situadas que pueden aparecer de forma espontánea durante actividades de aprendizaje móvil y que no fueron previamente planificadas. En el escenario se observa como el sistema ayuda a la creación y distribución de actividades espontáneas y a la distribución de recursos útiles para éstas entre el resto de miembros del grupo y usuarios, incluyendo además mecanismos para su integración a posteriori.

3.1.3 Especificación de requisitos para el diseño

Podemos sintetizar todos los comentarios sobre el escenario obtenidos en el apartado anterior y abstraer, mediante una lista de requisitos de diseño funcionales, las características demandadas en una infraestructura tecnológica enfocada a ofrecer soporte a actividades de aprendizaje móvil y colaborativo que cumplan dichas premisas:

- *Extracción de recursos.* El sistema debe encontrar y recolectar, posiblemente de diversas fuentes distribuidas, el conjunto de recursos necesarios para las actividades de aprendizaje incluyendo posiblemente aquellos de actividades anteriores e interesantes para potenciales actividades generadas.

- *Contextualización.* El sistema ha de mantener actualizados modelos de información acerca del entorno del alumno incluyendo su situación actual dentro de un contexto social, un contexto de actividad, un contexto tecnológico y un contexto general espacio-temporal.
- *Distribución.* El sistema ha de ser capaz de desplegar artefactos y compartir información contextual entre los elementos distribuidos de la infraestructura, en concreto los dispositivos de los alumnos.
- *Adaptación.* El sistema ha de tener en cuenta esta información de contexto para adaptar contenidos, procesos e interfaces a los dispositivos de los usuarios.
- *Instalación de herramientas.* El sistema ha de gestionar las herramientas software necesarias y asegurar su presencia en los dispositivos adecuados de una forma automática.
- *Configuración.* Es competencia del sistema la inicialización y parametrización de las herramientas requeridas para el escenario, gestionando sus datos y los objetos de aprendizaje de los usuarios para las actividades en los dispositivos.
- *Personalización.* El sistema debe decidir qué conjunto de actividades y recursos de aprendizaje son aplicables a un entorno de dispositivos/herramientas y ajustarlos si es necesario para cada determinado alumno.
- *Comunicación.* El sistema ha de ser capaz de facilitar la comunicación a lo largo de la infraestructura y funcionar adaptándose a distintas configuraciones de conexión tratando de forma eficaz las eventuales desconexiones de dispositivos.
- *Presencia.* El sistema puede poner en contacto a usuarios dentro de la infraestructura e informarles de acciones y actividades particulares de otros miembros.
- *Reconfiguración.* El sistema puede reaccionar a eventos excepcionales que pueden ocurrir en el desarrollo de las actividades y reconfigurar la infraestructura redistribuyendo recursos entre los usuarios y sus dispositivos si es necesario.
- *Ampliación y Extensión.* Una última consideración necesaria para dar valor añadido al sistema y permitir su futura reutilización es que éste ha de poseer mecanismos para permitir una extensibilidad de-

clarativa por parte de administradores y diseñadores de sus herramientas, recursos y modelos de información en general.

3.2 Decisiones de Diseño y Desarrollo

En base al listado de requisitos del apartado 3.1.3, podemos apreciar que la arquitectura tecnológica que buscamos se fundamenta fuertemente en dos clases de propiedades características que la influyen en todos sus aspectos funcionales: sus cualidades distribuidas y su dependencia de la información contextual. Estos dos rasgos principales se encuentran en concordancia con las necesidades desde un punto de vista tecnológico de las aproximaciones pedagógicas del *aprendizaje colaborativo* y el *aprendizaje móvil* presentadas respectivamente en los apartados 2.3 y 2.4. Es por tanto que, por un lado, las características distribuidas de una infraestructura compuesta potencialmente por múltiples dispositivos heterogéneos independientes, que han de ofrecer la comunicación entre ellos necesaria para promover el *aprendizaje colaborativo* entre estudiantes, abogarán por una topología de red flexible que soporte la conexión/desconexión de nodos de manera dinámica y descentralizada. Por otro lado, las características contextuales son requisito indispensable para un sistema enfocado al *aprendizaje móvil* y que permita su desarrollo en distintos ambientes ubicuos. Consecuentemente, esto hace imprescindible contar con modelos que permitan tratar y manipular la información contextual además de mecanismos para compartir estos modelos a lo largo de toda la infraestructura distribuida.

3.2.1 Elecciones de diseño

La tecnología de agentes parece ser una plausible candidata para satisfacer estas características distribuidas y contextuales. Una primera razón de peso es que los sistemas multi-agente son distribuidos por naturaleza. Los agentes pueden distribuirse por todo el conjunto de nodos (por ejemplo PDAs o teléfonos móviles) que conforma la red educativa, además de presentar capacidades para poder comunicarse entre ellos. De esta forma, los agentes pueden estar embebidos dentro de los dispositivos móviles de los alumnos, y al ser entidades autónomas proporcionar una experiencia inteligente, flexible y adaptada al alumno en cualquier momento y lugar, de manera autónoma aún incluso cuando no hay comunicación con el resto de la red educativa.

Otra de las razones para pensar en la tecnología de agentes como óptima aspirante a nuestras necesidades, es que desde sus inicios los agentes han sido una aproximación muy ligada a técnicas para mantener y compartir conocimiento de dominio (es decir manipular *ontologías*). Los agentes se presentan por tanto como una buena opción para manipular modelos de contexto compartidos e inferir nuevo conocimiento a partir de ellos. Esto se une a que los agentes presentan comportamientos diseñados para percibir el entorno y reaccionar ante él, pudiendo de esta manera proporcionar experiencias no intencionales a los alumnos cuando la ocasión surja.

La plataforma ICARO-T ha sido elegida como herramienta de desarrollo de agentes para nuestra propuesta de arquitectura. A pesar de no estar enfocada al desarrollo en dispositivos móviles ni contar con los mecanismos de comunicación distribuidos completamente independientes requeridos, su naturaleza altamente componentizada y las detalladas descripciones de diseño en UML que provee, nos permitirán la adaptación y extensión de sus funcionalidades para poder cubrir nuestras necesidades. No obstante, en este proceso nos encontraremos con retos, como la utilización de agentes en dispositivos de pequeñas prestaciones y la provisión de un mecanismo de comunicación adecuado entre éstos.

Por último, una decisión implícita de diseño resultante de utilizar ENLACE como marco de desarrollo y fuente de inspiración para nuestro escenario de diseño, es que nuestra arquitectura estará basada en los elementos tecnológicos de la red educativa de ENLACE, como por ejemplo su repositorio de objetos de aprendizaje y demás herramientas, haciendo necesaria la interoperabilidad con estos componentes software. No obstante, un objetivo importante es intentar abstraer las características de todos estos componentes para obtener como resultado una arquitectura genérica y extensible, que no se vea atada a los elementos de ENLACE y se pueda utilizar en otros escenarios. Para ello la noción de “componente recurso” que se tomará de la plataforma ICARO-T será de importancia para tal fin.

3.2.2 Casos de Uso

A partir del análisis del escenario del problema que hemos definido y tenido en cuenta en los requisitos funcionales extraídos, vamos a diseñar una serie de casos de uso claves que nuestro sistema persigue y que nos ayudarán a evaluar más adelante el grado de idoneidad alcanzado por nuestra arquitectura. Diseccionando el escenario, extraemos un conjunto de cuatro casos de

uso característicos en situaciones dinámicas: administración de la infraestructura distribuida, que incluye la preparación para las actividades y la sincronización de dispositivos, fomento contextualizado de la colaboración entre estudiantes, mecanismos de control/regulación y reconfiguración eventual durante la actividad.

CASO DE USO 1: ADMINISTRACIÓN DE LA INFRAESTRUCTURA DISTRIBUIDA.

Este caso de uso comienza cuando los usuarios se conectan a sus dispositivos tanto antes como después de las actividades. En este proceso el sistema automáticamente tratará de preparar los dispositivos que tomarán parte en el desarrollo de las próximas actividades. Para ello, el sistema ha de tomar en cuenta elementos del contexto referentes a los usuarios, las propias actividades y los dispositivos que utilizarán, para poder instalar, configurar, distribuir y adaptar todos los recursos de aprendizaje necesarios. Por ejemplo, distintos dispositivos móviles de distintos usuarios tendrán que ser configurados y personalizados con recursos diferentes, en función de los objetos de aprendizaje que generaron en la actividad anterior, las tareas que tengan que realizar en las próximas actividades, dónde vayan a tener lugar estas actividades, etc.

Por otro lado, cada vez que un dispositivo se conecte también será necesario inspeccionar todos los recursos generados en el dispositivo durante las actividades con el fin de sincronizar toda la información con el resto de la infraestructura.

CASO DE USO 2: FOMENTO CONTEXTUALIZADO DE LA COLABORACIÓN.

Durante las actividades, el sistema ha de monitorizar el contexto de los usuarios, incluyendo sus tareas actuales y localización, es decir qué están haciendo y dónde lo están haciendo. Cuando sea posible establecer lazos de colaboración convenientes entre los participantes de las actividades, el sistema ha de poner en marcha mecanismos para incitar y promover dicha colaboración.

Este proceso comienza cuando un dispositivo detecta la presencia de otro dispositivo cercano y proceden a intercambiar información acerca del contexto de los usuarios que están utilizando sendos dispositivos (sus estados, sus localizaciones, sus tareas actuales...). Acto seguido, el sistema puede aplicar patrones predefinidos para identificar potenciales situaciones de colaboración sobre la información recién adquirida, que determinarán si se efectuarán

acciones para hacer efectiva dicha colaboración. Estas acciones pueden incluir el contacto efectivo entre los usuarios para proponerles en qué y cómo colaborar, ya sea por ejemplo informándoles de ello buscando entablar una comunicación presencial directa entre los usuarios implicados o bien modificando la interfaz del dispositivo que están utilizando buscando una colaboración a través de herramientas software instaladas en los dispositivos.

CASO DE USO 3: REGULACIÓN Y CONTROL SOBRE LA ACTIVIDAD.

Dependiendo de sus roles dentro del proceso de aprendizaje, ciertos miembros de la comunidad de aprendizaje, como profesores o monitores, han de tener mecanismos para controlar el transcurso de las actividades de aprendizaje. Este control incluye el acceso a la información de contexto de los estudiantes, que puede ayudar a pronosticar problemas en el desarrollo de las actividades, pero también incluye la regulación de los flujos predeterminados de la actividad, invalidando en cualquier momento el comportamiento por defecto, como por ejemplo deteniendo una actividad de recogida de datos en el campo para poder dar explicaciones acerca de un fenómeno recién avistado.

CASO DE USO 4: RECONFIGURACIÓN EVENTUAL DURANTE ACTIVIDADES.

En determinados momentos, es posible que algo inesperado ocurra que no estuviera planeado para la actividad pero que puede ser utilizado como una oportunidad de aprendizaje. El sistema ha de ser lo suficientemente flexible para poder cambiar en tiempo real durante el transcurso de las actividades y adaptarse a los nuevos sucesos del escenario. Estos sucesos pueden ser disparados por el contexto en el que están trabajando los usuarios. El sistema responderá a estos sucesos transmitiendo información y recursos a todos los miembros implicados, con el fin de reconfigurar las actividades o añadir nuevas tareas y artefactos educativos adecuados a los nuevos requisitos.

3.2.3 Solución propuesta

Guiándonos con el escenario propuesto en este capítulo, y teniendo en todo momento presente los requisitos funcionales que hemos obtenido, hemos desarrollado una arquitectura educacional distribuida basada en agentes para dar soporte al aprendizaje que use dispositivos móviles y sistemas ubicuos en general. El nombre que recibe esta arquitectura es DAEDALUS (*Distributed*

Agent-based EDucational Architecture for Learning using Ubiquitous Systems). La arquitectura DAEDALUS aporta (1) una plataforma software basada en agentes capaz de funcionar en un rango heterogéneo de dispositivos, incluidos dispositivos móviles, (2) un modelo de organización de agentes de carácter totalmente distribuido y con capacidad de adaptarse dinámicamente a distintas topologías de red, (3) una ontología de conceptos comunes a todos los agentes de la organización que incluye una taxonomía de eventos y un modelo de contexto, (4) una representación computacional de dichos modelos que permita el intercambio de información y conocimiento entre todos los dispositivos de la arquitectura, (5) un conjunto de agentes especializados que aporten servicios educativos básicos para el desarrollo y el apoyo de actividades de aprendizaje, basándonos en el escenario de ENLACE presentado en el apartado 3.1.1.

DAEDALUS está diseñado como un sistema multi-agente. Los agentes son capaces de observar el entorno, analizarlo y actuar en consecuencia. La arquitectura de DAEDALUS está basada en ICARO-T y comparte muchas nociones de su modelo de organización. Sin embargo, como se verá más adelante en el Capítulo 5, será necesario adaptar ICARO-T a nuestras necesidades mediante nuevos recursos y modelos en la organización que lo complementen y que servirán para definir lo que es DAEDALUS y que presentamos a continuación en el siguiente capítulo.

4

LA ARQUITECTURA DAEDALUS

La arquitectura DAEDALUS plantea ofrecer una solución al desarrollo de escenarios pedagógicos en los que priman componentes distribuidos para dar soporte al aprendizaje, con necesidades de comunicación entre ellos y un fuerte uso del contexto en el que se encuentran los usuarios. Esta solución basa sus cimientos en la Inteligencia Artificial, concretamente haciendo uso de la tecnología de agentes y de las tecnologías semánticas para el tratamiento del conocimiento que el sistema tiene acerca de los usuarios, el modo en que se organizan y colaboran, las actividades que realizan, los artefactos, dispositivos para ayudar al aprendizaje que utilizan, etc., y buscando una continuidad en dicho aprendizaje tanto en la dimensión espacial como en la temporal.

Teniendo presente los requisitos extraídos del capítulo anterior, en este capítulo daremos una introducción a DAEDALUS y al concepto de *espacio de aprendizaje ubicuo* que proponemos. A continuación procederemos a trazar la

arquitectura genérica en la que se sustenta DAEDALUS y definiremos los elementos fundamentales que son necesarios comprender para la implementación de *espacios de aprendizaje ubicuos* mediante la arquitectura DAEDALUS.

4.1 Espacios de Aprendizaje Ubicuos

Se puede argumentar que el espacio de aprendizaje clásico y por antonomasia ha sido y es el aula de la escuela. Es en este lugar donde tradicionalmente se produce la transferencia de conocimientos durante una gran parte de nuestra vida. Sin embargo, este espacio de aprendizaje está aislado y se presenta a menudo independiente de los procesos que ocurren en el mundo real. Tomando en cuenta las teorías del *aprendizaje móvil*, el aprendizaje es móvil de por sí, y puede ocurrir en cualquier parte y en cualquier momento. De hecho, el aprendizaje que se realiza imbuido en un contexto práctico y localizado tiende a tener un mayor valor añadido. Estas van a ser por tanto las principales premisas con la que vamos a contar en la definición de nuestra arquitectura. De este modo definimos para nuestra propuesta un *Espacio de Aprendizaje Ubicuo* como el lugar físico o virtual orientado y destinado a la adquisición de conocimientos mediante ‘servicios pedagógicos inteligentes’ que de forma general pueden ocurrir en cualquier parte y/o lugar. Dentro de esta definición la tecnología jugará un papel fundamental para poder ofrecer este conjunto de ‘servicios pedagógicos inteligentes’ dentro del espacio, debido a que los estudiantes accederán a ellos mediante diversos dispositivos móviles y/o embebidos en el entorno y que les permitirán la realización de sus actividades de aprendizaje.

Dado que el aprendizaje puede llevarse a cabo en cualquier sitio y en cualquier lugar, la tecnología que ha de dar soporte a este aprendizaje y que debe ayudar a los discentes no ha de suponer un impedimento a priori para estos prerrequisitos espaciales y temporales, sino que es más, la tecnología tiene la obligación de catalizarlos y potenciarlos. Es por ello que esto no sólo implica que la tecnología ha de estar presente, ya bien sea haciéndola portable para acompañar al alumno o integrándola en el ambiente, sino que también:

- Ha de poder sacar provecho de manera ‘inteligente’ a las características y circunstancias contextuales que rodean a los elementos incluidos en el espacio de aprendizaje.

- Ha de buscar hacer los procesos pedagógicos más productivos.
- Ha de fomentar aspectos sociales como la colaboración.
- Ha de poder establecer hilos conductores espaciales y temporales a lo largo de todo el transcurso del aprendizaje.

Una consecuencia implícita de esto último anterior es que los *escenarios de aprendizaje ubicuos* no se encuentran aislados entre sí y que por lo tanto pueden estar imbricados unos con otros mediante estos hilos conductores que los unen a pesar de tener localizaciones diferentes o de estar ordenados de forma temporal secuenciada.

4.1.1 Interconexión de espacios de aprendizaje ubicuos

Podemos plantear en base a las propias características ubicuas espacio-temporales que los definen que un *espacio de aprendizaje ubicuo* probablemente se encuentre relacionado con otros espacios e interconecten entre ellos a lo largo del transcurso del proceso pedagógico. Por ejemplo, una visita a un museo puede verse complementada previamente con una clase teórica en un aula y posteriormente con ciertos ejercicios prácticos de reflexión a realizar en casa. En cada uno de estos espacios se ejecutan una serie de actividades distintas, con dispositivos diferentes, usuarios distintos, organizados de distinta forma y en definitiva contextos distintos. La tecnología que debe dar soporte en cada una de estos espacios debe además de dar continuidad a lo largo de todos ellos.

Esta interconexión por consiguiente puede preponderar tanto en el sentido físico espacial, como en el sentido temporal. En el primer sentido se puede considerar la existencia de varios lugares donde se realice el aprendizaje, ya sea al mismo tiempo o en momentos diferentes. Y por el otro sentido puede plantearse existir una sucesión de actividades en el tiempo, en el mismo lugar o en distintos. Por tanto dependiendo de ello podemos hacer una división en dos tipos de interconexión:

- *Interconexión síncrona.* Aparece cuando existe una comunicación entre distintos espacios en tiempo real. Esto posibilita la colaboración entre usuarios situados en espacios diferentes como por ejemplo unos alumnos haciendo una visita a un parque natural y otros en clase siguiendo la visita y utilizando los datos recolectados por los primeros a la vez que les proponen a los primeros focalizar en determinados elementos del parque.

- *Interconexión asíncrona.* Aparece cuando se permite el almacenamiento de recursos, para poder reutilizarlos en otros escenarios o el mismo, pero siempre con variaciones en el hilo temporal. Este es el caso por ejemplo del escenario de ENLACE comentado en el apartado 2.5.1.

En el caso de la interconexión síncrona, la tecnología tiene que lograr aplanar la dimensión espacial, uniendo espacios situados en localizaciones físicas diferentes. Mientras que en el caso de la interconexión asíncrona la tecnología también tiene que ofrecer una continuidad temporal, haciendo disponibles los artefactos necesarios a través de los espacios, sin importar donde estén localizados ni cuando hayan sido producidos.

4.1.2 Estructura de un espacio de aprendizaje ubicuo

Con el fin de poder establecer la composición y estructuración de los recién definidos *espacios de aprendizaje ubicuos*, resulta de vital importancia caracterizar y enumerar los elementos pedagógicos que los componen. Además esto nos servirá como punto de partida para la identificación de los distintos elementos tecnológicos que deberán estar presentes en una arquitectura que de manera genérica dé soporte a estos espacios. Se tomará consecuentemente una visión del *espacio de aprendizaje ubicuo* tanto pedagógica como tecnológica (Figura 4.1), destacando cómo se interrelacionan las dos perspectivas entre sí y ofreciendo un símil entre los elementos de ambas.

Desde un punto de vista pedagógico (A, Figura 4.1), un *espacio de aprendizaje ubicuo* es un *entorno* en el cual tienen lugar una serie de *procesos pedagógicos*, siendo estos dos los primeros elementos característicos que identificamos (A.1 y A.2, respectivamente). El *entorno* es considerado *ubicuo* por no tener restricciones espaciales ni temporales. En teoría puede estar localizado en cualquier lugar (o lugares) y puede suceder en cualquier momento (o momentos), ya sea en ambientes más o menos formales como el aula de una clase, o ambientes informales como una salida al campo o un viaje en autobús a casa. Los *procesos pedagógicos* son actividades y tareas que suceden en el *entorno ubicuo* y dependen en gran medida de las condiciones que se dan en cada instante dentro de éste, incluyendo la situación particular de los propios elementos que lo integran. Debido a este dinamismo del *entorno ubicuo* se hace esencial que los *procesos pedagógicos* se ajusten activamente a las características de *contexto* actuales del *entorno*. Su importancia da pie a tratar al *contexto* como un elemento característico más (A.3).

El *espacio de aprendizaje ubicuo* está caracterizado por definición por contener artefactos interactivos que identificaremos como *dispositivos* (A.4). Los *dispositivos* se encuentran distribuidos por todo el *entorno ubicuo* y pueden estar embebidos en éste o ser móviles. Las *personas* (A.5) que entran en el espacio de aprendizaje emplean los *dispositivos* para poder llevar a cabo los *procesos pedagógicos*, ya que les permiten interactuar con el espacio de aprendizaje e incluso les ofrecen una realidad aumentada de este espacio. Las *personas* desempeñan diferentes papeles y tienen distintas responsabilidades, obligaciones y derechos dentro de los *procesos pedagógicos*. Es por tanto que el *rol* de las *personas*, es decir estudiantes, profesores, monitores, etc., es un elemento identificativo más dentro del espacio de aprendizaje (A.6).

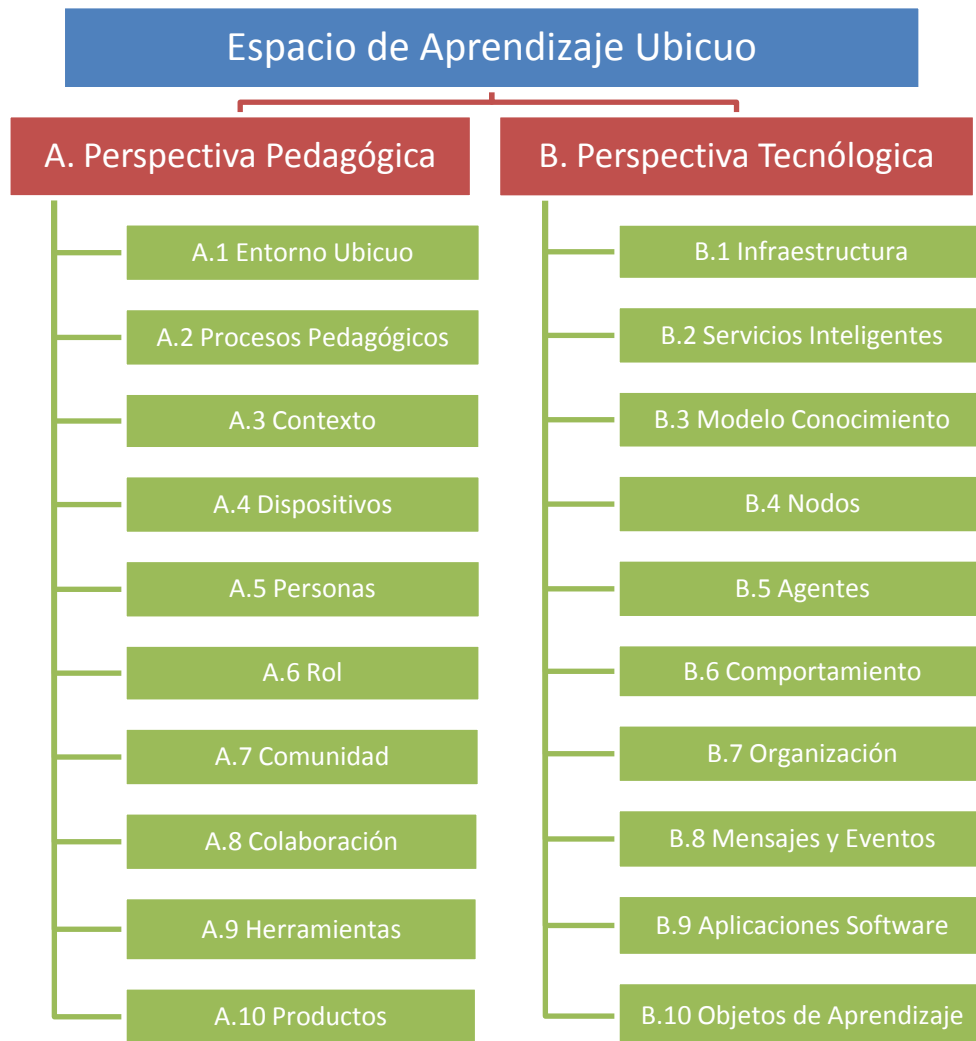


Figura 4.1 Elementos presentes en un Espacio de Aprendizaje Ubicuo

Todo el conjunto de *personas* y los *roles* que desempeñan éstas en el *espacio de aprendizaje ubicuo* imbrican una estructura social que se conoce como *comunidad de aprendizaje* (A.7). Dentro de esta *comunidad* priman distintas relaciones entre sus miembros, pero una a la que le hemos dado especial importancia es la *colaboración* (A.8). La *colaboración*, como se ha expuesto en el apartado 2.3, es un elemento deseable en los *procesos pedagógicos* y es materializada en estos a través de las *herramientas de aprendizaje* presentes dentro de los *dispositivos*. Las *herramientas* (A.9) ofrecen funcionalidades específicas para realizar tareas concretas para el aprendizaje empleando los *dispositivos*, como puede ser recolectar datos, aportar un comentario a una discusión, participar en una votación, etc. Las *herramientas* pueden dar soporte a una colaboración síncrona entre las *personas* si intercambian información en tiempo real, o de forma asíncrona si intercambian y reutilizan objetos temáticos como *productos* emergentes del aprendizaje (Hoppe et al., 2005). Estos *productos* (A.10) son generados mediante las *herramientas* como resultado de la interacción de las *personas*, principalmente estudiantes, y pueden ser reutilizados y aumentados por otras *personas* en diferentes *contextos*, *procesos pedagógicos* y en definitiva *entornos ubicuos*.

Desde un punto de vista tecnológico (B, Figura 4.1), y buscando el símil con la perspectiva pedagógica, un *espacio de aprendizaje ubicuo* incluye todo un conjunto de componentes tecnológicos necesarios para el *entorno ubicuo* y que recibe el nombre de *infraestructura* (B.1). La *infraestructura* tiene como misión aportar una colección de *servicios inteligentes* para dar soporte a los *procesos pedagógicos* (B.2). Se consideran ‘inteligentes’ porque estos *servicios* se ajustan a los *procesos* en todo momento ya que pueden interpretar el *contexto* y adaptarse a las características actuales del *entorno ubicuo*. A tal fin se establece un *modelo de conocimiento* que define los elementos del *contexto* que vayan a ser representados computacionalmente dentro de la *infraestructura* (B.3).

La *infraestructura* está distribuida físicamente a través de los *dispositivos* del espacio de aprendizaje, configurando lo que se conoce como *nodos* de la *infraestructura* (B.4). Cada *nodo* está ‘habitado’ por componentes software destinados a dar soporte proactivo a las *personas* y que se definen como *agentes* (B.5). Dependiendo de las responsabilidades que tengan los *agentes* en la provisión de los llamados *servicios inteligentes* se les identifica mediante distintos *comportamientos* (B.6). Para poder proveer de *servicios* de carácter complejo a toda la *comunidad de aprendizaje*, *agentes* con diferentes *comportamientos* se agrupan en el *espacio de aprendizaje ubicuo* formando

toda una *organización* (B.7). La *organización* define toda la estructura social y de comunicación, estableciendo elementos para facilitar la cooperación entre *agentes* consistentes en *mensajes y eventos* (B.8). La comunicación entre los distintos *nodos* se realiza por tanto a través de *mensajes y eventos* mediante los cuales los *agentes* pueden sincronizarse e intercambiar información acerca del *contexto* codificada mediante el *modelo de conocimiento*.

Por otro lado, las *herramientas de aprendizaje* que se emplean en el *espacio de aprendizaje ubicuo* se corresponden con pequeñas *aplicaciones software* instalables en los *dispositivos* (B.9). Mediante estas *aplicaciones* de los *dispositivos*, los estudiantes generan *productos* durante el aprendizaje que son materializados en la *infraestructura* como *objetos de aprendizaje* (B.10). Los *objetos de aprendizaje*, con las connotaciones que introdujimos en el apartado 2.3.1, se representan como una estructura computacional consistente en el contenido del *producto*, como por ejemplo una foto de un arbusto observado, un mapa conceptual de un modelo de ecosistema, etc., y metadatos asociados que hacen posible la búsqueda y reutilización del *producto* en distintas momentos durante los *procesos pedagógicos*.

En torno a todos estos elementos genéricos de la Figura 4.1 caracterizados a partir del concepto de *Espacio de Aprendizaje Ubicuo*, definiremos a continuación la arquitectura genérica de DAEDALUS.

4.2 Arquitectura Genérica

Nuestra solución se formula a partir de la arquitectura genérica que se presenta en este apartado y se plantea como un conjunto de niveles apilados verticalmente. Cada uno de estos niveles define de forma incremental una serie de conceptos y elementos fundamentales que se exportan a los niveles superiores, incorporándose de modo sucesivo al conjunto global de la arquitectura, y que constituirán los utensilios necesarios para modelar y construir sistemas educativos que puedan dar soporte a un aprendizaje ubicuo basado en agentes software. Los pilares básicos que sustentan nuestra arquitectura genérica son por un lado el tratamiento de las características distribuidas inherentes a los *espacios de aprendizaje ubicuos* y por otro lado las características contextuales de los mismos.

La arquitectura genérica aporta desde sus estratos inferiores mecanismos para manejar y abstraer la distribución, promoviendo una división del

sistema en componentes que podrán repartirse entre el amplio rango de dispositivos heterogéneos presentes en los escenarios pedagógicos. La búsqueda de un diseño altamente componentizado de la arquitectura reportará en beneficios que posteriormente se verán plasmados en un efectivo despliegue de los sistemas educativos distribuidos y una gestión eficaz de los propios componentes que los formen. Todos los componentes presentan estructuras prefijadas a modo de armazones predefinidos que faciliten a los diseñadores y desarrolladores las tareas de diseño e implementación abstrayéndolos de los elementos de bajo nivel. La estrella de estos componentes y que tiene mayor peso en la definición de la arquitectura genérica y por ende de toda nuestra propuesta es el concepto de *agente software*. Los agentes son modelados dentro de nuestra arquitectura como los componentes más importantes y con las características más avanzadas del sistema. Bajo el concepto de agente software se empaquetarán los mecanismos esenciales para dotar de servicios inteligentes a los *espacios de aprendizaje ubicuos*. Gracias a esta aproximación los elementos distribuidos presentes en estos espacios pueden responder al entorno de forma autónoma, presentar habilidades para comunicarse entre ellos y ser proactivos en sus acciones.

Por otro lado, de la misma forma que la arquitectura genérica trata los aspectos distribuidos de los escenarios, la arquitectura también busca soportar desde su base la manipulación efectiva de los elementos contextuales presentes en la implantación de *espacios de aprendizaje ubicuos*. De esta forma la aproximación empleada consiste en la incorporación de un *modelo de conocimiento* que los componentes inteligentes, es decir los agentes, manejarán para manipular información acerca de los elementos presentes en el entorno ubicuo, posibilitando el uso de este conocimiento para ofrecer servicios pedagógicos adaptados a los usuarios y a sus situaciones actuales. Principalmente, en este *modelo de conocimiento* se incluye información acerca de los implicados en los procesos pedagógicos, tanto desde un punto de vista individual como de comunidad, sus actividades, junto con los dispositivos, herramientas y artefactos o productos que emplean. La arquitectura establece estructuras para la explicitación formal de todos los conceptos contextuales útiles y necesarios para el soporte personalizado y adaptado de los procesos pedagógicos. Estas estructuras se concretan como una ontología (Gruber, 1995) que definirá una conceptualización compartida del contexto para la utilización por parte de los componentes de la arquitectura.

Utilizando como bloques elementales los componentes distribuidos y las estructuras de conocimiento contextual, en los niveles superiores de la archi-

itectura genérica se articulan los *servicios inteligentes* ofrecidos para soportar y promover el aprendizaje en los *espacios de aprendizaje ubicuos*. Dependiendo de la complejidad del escenario, los requisitos pedagógicos y las necesidades de soporte, adaptación y personalización del sistema tecnológico a los *procesos pedagógicos*, la provisión de distintos *servicios inteligentes* será conveniente y en consecuencia adecuados conjuntos de componentes y representaciones de conocimiento habrán de ser empleadas para modelarlos.

No obstante, la capacidad de reaprovechamiento de estos *servicios inteligentes* para otros escenarios se postula como uno de los objetivos y puntos clave de nuestra arquitectura. Esto redundará en un imprescindible enfoque hacia la reutilización, plasmado por un lado en su clara división en componentes distribuidos inteligentes, o *agentes software*, y por otro lado basado también en cierta medida en la perspectiva extensible con la que han sido concebidos los *modelos de conocimiento* de la arquitectura. En resumidas cuentas, dentro de la arquitectura genérica se persigue que quede reflejada una visión marcada por tecnologías basadas en la Inteligencia Artificial pero siempre con una mirada en la Ingeniería del Software que promueva la extensibilidad y la reutilización.

De modo específico, DAEDALUS es una materialización concreta de esta arquitectura genérica que se presenta en este apartado y que constituye una prueba de su viabilidad como herramienta de desarrollo para poder implementar sistemas para el aprendizaje colaborativo, móvil y ubicuo. En el siguiente apartado se detallarán los distintos niveles en que se estructura la arquitectura genérica y se describirán sus características internas.

4.2.1 Estratificación de la Arquitectura Genérica

Se propone una división de la arquitectura genérica en cuatro niveles, tal y como se representa en la Figura 4.2. Cada uno de los niveles inferiores define nuevos elementos que son utilizados en los niveles superiores.

El primero de estos niveles trata de manera general todos los conceptos necesarios para integrar la tecnología de agentes a la arquitectura, sin entrar en detalles pedagógicos, y constituye por tanto la Plataforma Genérica. Los otros tres niveles se centran ya en aspectos educativos, por lo que juntos forman la llamada Plataforma Educativa.

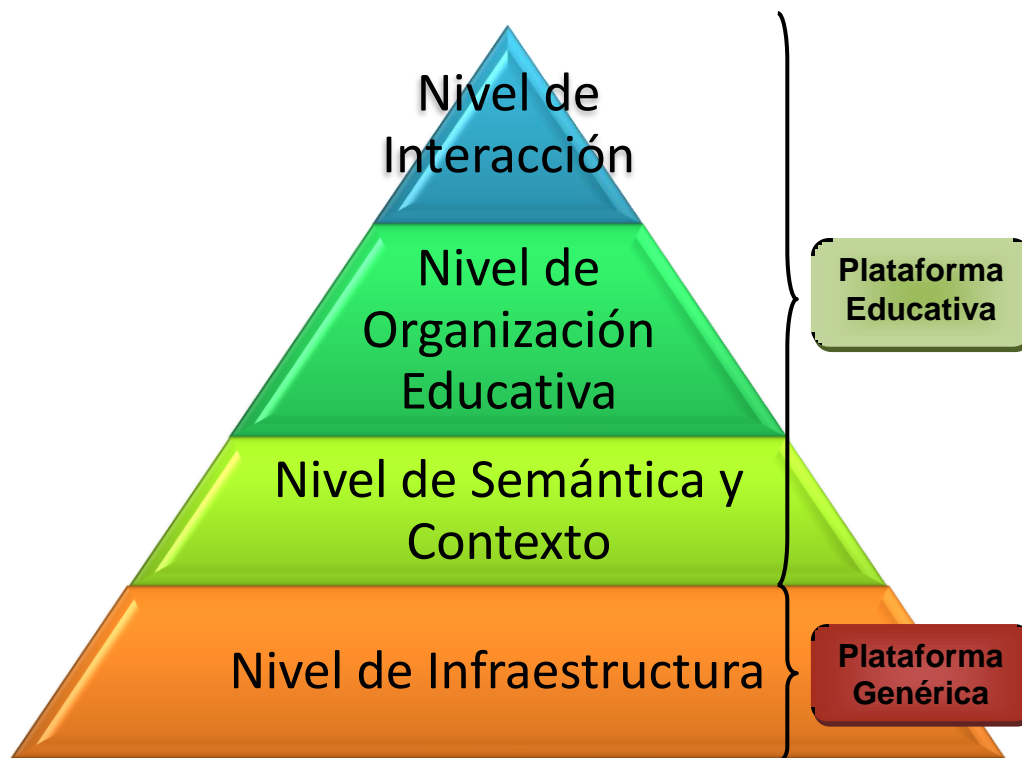


Figura 4.2 Niveles de la Arquitectura Genérica

PLATAFORMA GENÉRICA

- **NIVEL DE INFRAESTRUCTURA.** Dentro de este nivel se sientan las bases sobre las que se articulan el resto de niveles de la arquitectura. En este nivel se definen los conceptos clave de la *infraestructura* tecnológica a modo de patrones estructurales con el fin de dar un enfoque basado en componentes distribuidos a la arquitectura en el que la reusabilidad, extensibilidad y escalabilidad se exporte a los niveles superiores. Uno de estos patrones estructurales característico de este nivel y que goza de mayor importancia en toda la arquitectura es el de *agente*.

PLATAFORMA EDUCATIVA

- **NIVEL DE SEMÁNTICA Y CONTEXTO.** En este nivel se modelan los utensilios básicos necesarios para poder aportar coherencia semántica a la arquitectura y ofrecer un vocabulario común a todos los componentes con el que estructurar el conocimiento que el sistema

tiene sobre el entorno ubicuo. Se aportan por tanto mecanismos para articular el *modelo del conocimiento* que será utilizado por los *servicios inteligentes* y se definen entidades de contexto, sus relaciones y las estructuras para poder transmitirlos y permitir una comunicación eficaz entre los elementos distribuidos de la infraestructura tecnológica.

- **NIVEL DE ORGANIZACIÓN EDUCATIVA.** Este es el nivel más importante desde el punto de vista pedagógico y el más completo de la arquitectura. En este nivel se definen de forma concreta todos los componentes distribuidos que serán empleados para prestar los *servicios inteligentes* de la *infraestructura* que dan soporte a los escenarios de aprendizaje. Es aquí donde se especifican los distintos *comportamientos* que establecen las responsabilidades de los *agentes* dentro de la *organización* y cómo será su forma de interactuar con otros agentes y recursos de organización.
- **NIVEL DE INTERACCIÓN.** Este último nivel es el que está expuesto y en contacto con el usuario. En este nivel se define el modo de interacción con los usuarios del sistema (estudiantes, profesores, etc.), las interfaces gráficas que éstos podrán manipular y cómo interactuarán los servicios inteligentes con dichos usuarios. De modo particular, en este nivel se estructuran los flujos de los *procesos pedagógicos* en términos referentes a la interacción de las *personas* con el *entorno ubicuo* a través de los *dispositivos*, sus *herramientas* o *aplicaciones software* que deberán utilizar para manipular los objetos de aprendizaje.

En resumidas cuentas dentro de la Plataforma Educativa, el Nivel de Semántica y Contexto captura todo el modelo de dominio del sistema, el nivel de Organización Educativa configura todos los servicios y elementos controladores de la arquitectura y el Nivel de Interacción se encarga de la parte relacionada con la interfaz de usuario. Esta estructuración de la Plataforma Educativa persigue un claro diseño que separe distintos cometidos dentro de la arquitectura entendida ésta como sistema global, inspirándose y ajustándose esta separación en cierto modo al patrón de ingeniería conocido como Modelo-Vista-Controlador (Krasner & Pope, 1988).

4.3 Conceptualización Global de la Arquitectura DAEDALUS

Una vez identificados los elementos característicos de los *espacios de aprendizaje ubicuos* en el apartado 4.1.2 y, en base a éstos, perfilado en la sección 4.2 una arquitectura genérica dividida en niveles, nuestra intención es refinar progresivamente todas las unidades de que se compone la arquitectura DAEDALUS con el fin de poder ofrecer una perspectiva global de ésta. En la Figura 4.3 es posible apreciar un resumen de los elementos principales que se tratan en la arquitectura ordenados por niveles. A continuación introduciremos una breve descripción conceptual de los elementos localizados en cada nivel, para que el lector pueda obtener una primera noción concreta de conjunto de toda la arquitectura DAEDALUS. En los posteriores capítulos 5, 6, 7 y 8 desgranaremos sucesivamente cada uno de estos niveles.

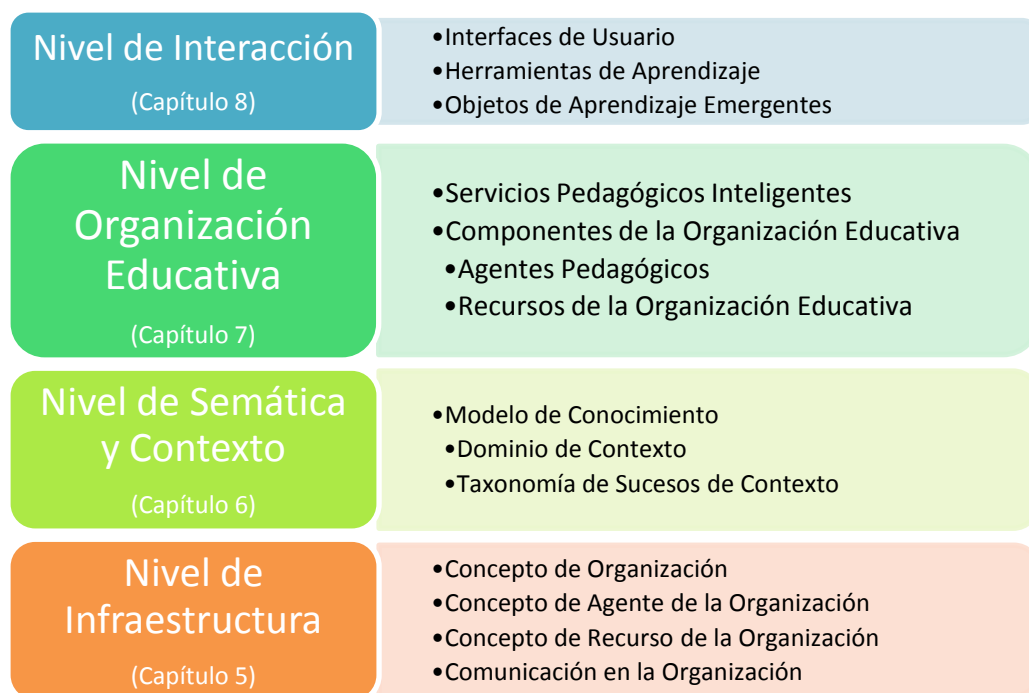


Figura 4.3 Elementos principales de cada nivel de la arquitectura

4.3.1 Elementos del Nivel de Infraestructura

El Nivel de Infraestructura fija tres conceptos fundamentales que serán la piedra angular de toda la arquitectura de DAEDALUS: organización, agen-

te y recurso. Transversalmente a todos estos se sitúa un cuarto elemento imprescindible para la arquitectura: la comunicación. Se hablará más extensamente del Nivel de Infraestructura en el capítulo 5, pero a continuación presentamos las características principales de estos elementos:

- **CONCEPTO DE ORGANIZACIÓN.** El concepto de organización representa la estructura que engloba todos los componentes software de la arquitectura y las relaciones que se establecen entre éstos. La organización de DAEDALUS se encuentra distribuida en nodos separados físicamente. Los nodos son elementos lógicos que sirven para particionar la organización y distribuirla a lo largo de la infraestructura. Cada uno de los nodos de la organización se corresponde con un dispositivo del entorno ubicuo con capacidad de proceso para albergar algún tipo de componente software. Es decir, los nodos de la organización están localizados en dispositivos de distintos tipos como ordenadores personales, portátiles, servidores, dispositivos móviles, etc. La organización está compuesta por dos capas distintas: la capa de control, que está formada por ‘componentes agente’, y la capa de recursos, formada por ‘componentes recurso’.
- **CONCEPTO DE AGENTE.** Los agentes en DAEDALUS son componentes software en los que se delega la funcionalidad de los servicios inteligentes y que son encargados de realizar tareas para garantizar la provisión de éstos. Los agentes son los controladores del sistema, capaces de cooperar con otros agentes y compartir su conocimiento acerca del entorno ubicuo de aprendizaje. Todos los agentes son entidades autónomas independientes que están gobernados por el Principio de Racionalidad (Newell, 1982). Para llevar a cabo sus objetivos, los agentes emplean otro tipo de componentes de la organización, los recursos.
- **CONCEPTO DE RECURSO.** Los recursos de organización son componentes software que ofrecen acceso a funcionalidades definidas para los agentes. Se diferencian de los agentes por ser componentes con características pasivas dentro de la organización, comparado con el comportamiento activo de los agentes. No obstante, permiten realizar operaciones a los agentes como por ejemplo acceder a bases de datos, mostrar información en las interfaces de usuario, realizar búsquedas sobre el repositorio de objetos de aprendizaje, etc.

- **COMUNICACIÓN EN LA ORGANIZACIÓN.** La comunicación dentro de la organización es una necesidad que tiene que aportarse a nivel de infraestructura. La comunicación se estructura como un recurso de organización especial presente en cada nodo y que permite retransmitir a los demás nodos mensajes y eventos a todos los componentes software, que les permite intercambiar información con el fin de coordinarse o de compartir información relativa al contexto educativo. Dado el dinamismo de la infraestructura, la organización está preparada para una eventual escisión de nodos, fruto de la movilidad de los dispositivos, siendo el recurso de comunicación el que gestiona esta conexión/desconexión de nodos.

4.3.2 Elementos del Nivel de Semántica y Contexto

El Nivel de Semántica y Contexto, que se describe más detalladamente en el capítulo 6, aporta principalmente el modelo de conocimiento que resulta de vital importancia para establecer representaciones comunes que comparten los componentes de la organización en su comunicación. He aquí una breve introducción a los elementos que DAEDALUS trata en este nivel:

- **MODELO DE CONOCIMIENTO.** El modelo de conocimiento de DAEDALUS define cómo se estructura la información en la organización y qué mecanismos existen para representar y diseminar el conocimiento acerca del entorno ubicuo por parte de los componentes software. Por ello se definen mecanismos para poder modelar por un lado la representación estática de dicho conocimiento sobre el entorno, y por otro lado el comportamiento dinámico para gestionar y distribuir los cambios y variaciones de este entorno, respectivamente: el dominio de contexto y la taxonomía de sucesos de contexto.
 - **DOMINIO DE CONTEXTO.** El dominio de contexto incluye toda la información que la organización necesita manejar acerca del *espacio de aprendizaje ubicuo*, incluyendo personas, procesos pedagógicos, elementos de la infraestructura, etc. Se describe mediante una ontología, que incluye todas las entidades y las relaciones de contexto necesarias para que la organización pueda prestar los servicios inteligentes. De esta forma, la ontología define todo el conocimiento del dominio de contexto que se precisa modelar dentro de la arquitectura.

- **TAXONOMÍA DE SUCESOS DE CONTEXTO.** El contexto de un entorno ubicuo es un ente predominantemente dinámico por lo que resulta práctico para ofrecer un comportamiento adaptado el poder capturar ciertas variaciones en su estado. Las variaciones en el contexto son notificadas y distribuidas en la arquitectura DAEDALUS como sucesos a lo largo de la organización. Los sucesos de contexto reflejan situaciones dinámicas que ocurren en un preciso momento y que pueden dar pie a una reacción de los agentes e implicar un cambio en el conocimiento sobre el contexto por parte de éstos. La taxonomía de sucesos de contexto explicita y estructura estos cambios para que los componentes de la organización puedan tanto generarlos como reaccionar ante ellos.

4.3.3 Elementos del Nivel de Organización Educativa

Dentro del Nivel de Organización Educativa, que se expone en el capítulo 7, se sitúan como elemento central los servicios pedagógicos inteligentes que la infraestructura busca proveer dentro del *espacio de aprendizaje ubicuo*. Los elementos singulares de la arquitectura DAEDALUS en este nivel son por tanto, junto con los servicios inteligentes, los componentes de la organización educativa responsables de la provisión de estos servicios. Esto incluye tanto los agentes como los recursos que se diseñan en este nivel expresamente para tal propósito:

- **SERVICIOS PEDAGÓGICOS INTELIGENTES.** Todas las funcionalidades de la infraestructura se agrupan como conjuntos de servicios que son definidos en la organización y que se despliegan en el entorno ubicuo cuyo fin es ofrecer un soporte personalizado y adaptado de los procesos pedagógicos. Los distintos servicios pedagógicos inteligentes se basan en el análisis del contexto para administrar el espacio de aprendizaje ubicuo, gestionando la infraestructura, ayudando a preparar las actividades y personalizando los dispositivos, y para respaldar los procesos pedagógicos, coordinando las actividades, recomendando tareas y contenidos, dirigiendo las relaciones sociales, proponiendo colaboraciones, etc., por citar algunas posibilidades.
- **COMPONENTES DE LA ORGANIZACIÓN EDUCATIVA.** La Organización Educativa integra todos los agentes y recursos de la infraestructura que se emplean para favorecer las funcionalidades

necesarias que buscan dar apoyo y soporte a los distintos usuarios presentes en dicho espacio para la consecución eficaz de sus actividades de aprendizaje. Estas funcionalidades se encapsulan en componentes agente que a su vez utilizan componentes recursos de la organización educativa.

- **AGENTES PEDAGÓGICOS.** Los agentes pedagógicos son definidos siguiendo los patrones definidos en el Nivel de Infraestructura y entre ellos se dividen las responsabilidades para satisfacer la consecución de los servicios pedagógicos inteligentes. Dependiendo de la complejidad de los servicios, estas responsabilidades tendrán que repartirse entre uno o varios agentes, que compartirán objetivos y tendrán que cooperar comunicando entre sí.
- **RECURSOS DE LA ORGANIZACIÓN EDUCATIVA.** También definidos por los patrones, los recursos de la organización educativa incluyen repositorios de aprendizaje, donde los agentes pueden obtener artefactos para los procesos pedagógicos, plataformas de gestión de aprendizaje, donde se puede efectuar un minado de personas y actividades, hasta las propias interfaces de usuario, recursos que por su influencia son tratados en un nivel propio.

4.3.4 Elementos del Nivel de Interacción

Al ser la infraestructura de un espacio de aprendizaje ubicuo un sistema eminentemente interactivo, los elementos con los que interactúan los usuarios reciben una atención especial. En concreto en el Nivel de Interacción, que se desgana en el capítulo 8, esta importancia la reciben en general las interfaces de usuario que estarán disponibles en los dispositivos, junto con las aplicaciones software y objetos de aprendizaje que deberán manejar las personas:

- **INTERFACES DE USUARIO.** Las personas entran en contacto con los dispositivos a través de las interfaces de usuario presentes en estos. Las interfaces de usuario articulan el espacio de trabajo que pone en contacto a las personas con el entorno ubicuo, articulando de qué forma se accede a las actividades, tareas y artefactos educativos. Los recursos de interfaz de usuario poseen operaciones de transformación, para mostrar u ordenar elementos, que ofrecen la capacidad

a los agentes de poder adaptar las experiencias de las personas, y ofrecerles acceso a determinada información, herramientas y objetos de aprendizaje para llevar a cabo los procesos pedagógicos.

- **HERRAMIENTAS.** Las herramientas son aplicaciones software con las cuales se llevan a cabo las actividades pedagógicas dentro de los dispositivos. Son las que exponen una serie de funcionalidades a los usuarios para poder realizar las tareas que se les pide y para poder manipular en las actividades los artefactos u objetos de aprendizaje.
- **OBJETOS DE APRENDIZAJE.** Los objetos de aprendizaje tienen un lugar destacado dentro de la arquitectura DAEDALUS ya que representan tanto los artefactos necesarios para realizar las actividades, como los resultados y productos del aprendizaje generados por los estudiantes durante éstas. Los estudiantes crean y editan de forma colaborativa estos objetos de aprendizaje a través de las herramientas, almacenándose con metadatos para poder ser buscados por personas o agentes posteriormente.

5

NIVEL DE INFRAESTRUCTURA

Dentro de este capítulo expondremos el nivel más fundamental de la arquitectura de DAEDALUS, el Nivel de Infraestructura. Éste es el nivel que sustenta el resto de los niveles de nuestra arquitectura y el que aporta las nociones necesarias para poder llevar a cabo desarrollos e implementaciones de *espacios de aprendizaje ubicuos* basándose en una aproximación orientada a agentes software.

Apoyándose ampliamente en conceptos introducidos por la plataforma multi-agente ICARO-T, la cual fue presentada previamente en el apartado 2.8, el Nivel de Infraestructura es el encargado de integrar estos conceptos en la arquitectura de DAEDALUS con el fin de fundamentar una base sólida para hacer uso de la tecnología de agentes, pero siempre y en todo momento, mediante una perspectiva de ingeniería del software. Esta perspectiva presentará vital importancia a la hora de poder modelar la infraestructura tecnológica y establecer los fundamentos de una arquitectura distribuida

estructurada en componentes inteligentes. Cabe destacar que esta arquitectura no sólo focaliza en la comunicación de estos componentes inteligentes, es decir en la comunicación entre agentes, sino que los define tanto a escala individual como a escala de organización, incorporando armazones software a modo de patrones para guiar los desarrollos. Se establece dentro de este Nivel de Infraestructura por tanto, y como se verá a lo largo este capítulo, la idea de patrones de diseño introducida por ICARO-T para poder representar una serie de principios de comportamiento predefinidos para los agentes y una estructuración del sistema con forma de organización jerárquica. No obstante, la incorporación de ICARO-T a los requisitos que presenta DAEDALUS no ha sido directa, sino que ha sido necesario solventar problemas de integración que también comentaremos a continuación.

5.1 Arquitectura Orientada a Agentes

Los Espacios de Aprendizaje Ubicuos en DAEDALUS se modelan como sistemas orientados a agentes software. Los agentes son la piedra angular de la arquitectura DAEDALUS que hace posible distribuir todas las funcionalidades del sistema a lo largo de cualquiera de los dispositivos, tanto si son móviles como no móviles, en el entorno ubicuo. Mediante la utilización del paradigma de agentes se posibilita la dotación de autonomía y proactividad a los elementos tecnológicos para dar soporte inteligente a los usuarios de los dispositivos del espacio de aprendizaje. Además se proporciona capacidades de comunicación a la infraestructura para interconectar todos los elementos presentes en dicho espacio.

Dentro del Nivel de Infraestructura se engloban las nociones comunes al resto de niveles de la Plataforma Educativa que posibilitarán la descripción de éstos en términos orientados a agentes. Entre estas nociones fundamentales se encuentran tres conceptos que marcarán el diseño y la implementación tecnológica de los *espacios de aprendizaje ubicuos*. Estos son el *concepto de organización*, el *concepto de agente* y el *concepto de recurso*. La visión con la que se arroja luz sobre estos conceptos claves para conseguir aplicar una orientación a agentes software mediante mecanismos aptos y probados para la construcción de sistemas complejos se toma prestada de la plataforma multiagentes ICARO-T. Esta es una visión que está dirigida hacia un desarrollo con una perspectiva de ingeniería del software y que se ha ido refinando a lo

largo de la aplicación e implementación de diversos sistemas complejos multi-agente (Pavon et al., 2008).

5.1.1 Sistemas complejos multi-agente distribuidos

De la misma forma, los sistemas a los que está enfocado el empleo de la arquitectura que presentamos en nuestra propuesta son en principio entornos de aprendizaje tecnológicamente complejos, compuestos teóricamente por multitud de elementos distribuidos móviles que pueden cambiar de contexto, tanto físico como educativo, y que por tanto deberán ser suficientemente inteligentes para adaptarse a dichos cambios y responder en consecuencia para continuar ofreciendo servicios pedagógicos personalizados a los usuarios que interactúan con ellos. Los agentes son los encargados de garantizar este último fin.

Los agentes software se despliegan por los distintos dispositivos de la infraestructura, formando pequeñas agrupaciones situadas dentro de ellos y constituyendo en conjunto toda una organización global. Cada una de estas agrupaciones se define como un nodo de la organización, y cada nodo ofrece facilidades de comunicación con los agentes de otros nodos. De esta manera, dentro de un dispositivo móvil cohabitan varios agentes, cada uno con distintas metas y objetivos, pero que pueden colaborar entre ellos y con agentes de otros dispositivos para lograr alcanzar dichas metas. Por ejemplo, dentro de una PDA perteneciente a un alumno haciendo una visita a un museo interactivo de ciencia puede existir un *agente personal* cuya meta sea mostrar actividades contextualizadas y personalizadas al alumno. Para lograr este objetivo, el *agente personal* colabora con un *agente de recursos pedagógicos*, también dentro de la PDA, que evalúa y le informa al primero acerca de qué herramientas y qué objetos de aprendizaje disponibles dentro del dispositivo son adecuados para el contexto pedagógico actual del alumno, que le ha sido compartido por dicho *agente personal*. No obstante, en su labor por ofrecer recursos pedagógicos interesantes y contextualizados, este *agente de recursos pedagógicos* de la PDA puede interactuar, si hay conexión, con otros agentes situados en el servidor del museo y que pueden comunicarle información sobre varios artefactos del museo que pueden ser interesantes para el usuario.

Otro ejemplo similar puede darse durante una visita al campo, en donde en la PDA de cada alumno existe un posible *agente de colaboración*. Este agente es capaz de analizar el contexto actual del alumno que se encuentra utilizando el dispositivo con el contexto de usuarios cercanos, aportado por

otros *agentes de colaboración* que habitan en otras PDAs. De esta forma cuando el agente encuentra otro *agente de colaboración*, éstos son capaces de intercambiar información contextual acerca de los intereses, necesidades y las actividades que están realizando los alumnos en los dispositivos que emplean. Cuando los *agentes de colaboración* encuentran pautas predeterminadas de colaboraciones potenciales, estos agentes informan de ello a los respectivos *agentes personales* de las PDAs que en función del estado del alumno (ocupado, libre, no quiere que se le moleste...) le propondrán, a través de recursos de interfaz de usuario, colaborar en ciertas tareas con otros alumnos.

En definitiva, una de las características principales de la arquitectura DAEDALUS es que incorpora un enfoque para el modelado y desarrollo de infraestructuras educativas tecnológicamente complejas que es soportado desde su raíz por una perspectiva de sistema multi-agente, buscando de manera intuitiva proporcionar características inherentemente distribuidas. Además, otra aportación singular para este desarrollo de sistemas complejos multi-agente distribuidos consiste en la aproximación de establecer patrones de diseño que modelan las ideas clave de la arquitectura DAEDALUS con el fin de guiar los desarrollos orientados a agentes. En términos generales, los patrones de diseño explicitan la experiencia de diseño que una comunidad ha refinado y asimilado, y aportan soluciones depuradas a problemas recurrentes, habiendo sido utilizados con éxito en campos como la Arquitectura (Alexander, Ishikawa, & Silverstein, 1977) o la Ingeniería del Software (Gamma, Helm, R. Johnson, & Vlissides, 1993). Los patrones son un buen método para describir un marco de desarrollo (Ralph E. Johnson, 1992). La arquitectura DAEDALUS sigue este principio desde su base integrándolo en el Nivel de Infraestructura. DAEDALUS incorpora patrones e ideas haciendo uso de la experiencia generada y refinada por la plataforma multi-agente ICARO-T (Pavon et al., 2008), aportando estructuras a modo de armazones software documentados para agilizar los procesos de desarrollo.

Comenzaremos desgranando las distintas ideas y patrones que se articulan en este Nivel de Infraestructura, estando una de las principales de ellas representadas bajo el concepto de *organización*.

5.2 La Organización en DAEDALUS

El concepto de organización dentro de la arquitectura DAEDALUS hace referencia a cómo se distribuyen e interrelacionan los componentes software dentro de la infraestructura tecnológica. La organización de DAEDALUS sigue en líneas generales la estructuración definida por ICARO-T. Esta estructuración articula la organización en dos clases de componentes: *agentes* y *recursos*. Cada una de estas clases de componentes establece sendas capas dentro de la organización, a saber, la *capa de control* y la *capa de recursos*. Los *agentes* por tanto son componentes situados en la *capa de control* y encargados de dirigir todo lo que ocurre dentro del sistema, mientras que los *recursos* son componentes que aportan funcionalidades a los *agentes* para que éstos puedan alcanzar sus objetivos. Entre las funcionalidades que aportan los *recursos de la organización* pueden aparecer recursos para el suministro de datos, que permiten el acceso a bases de datos o repositorios, recursos para el manejo de visualizaciones, que dan acceso a interfaces de usuario, y recursos de gestión como la persistencia de configuraciones o recursos para la comunicación de los agentes, entre otros muchos. En definitiva, cualquier elemento que tenga un comportamiento estático definido por una interfaz de servicio se modela como un recurso dentro de la organización, mientras que los agentes por el contrario muestran un comportamiento más dinámico y hacen uso de estos recursos.

Una cuestión importante y que normalmente se infravalora en la definición de sistemas multi-agente es la necesidad de tratar detalles referentes a la propia gestión de los componentes dentro de la organización. A este respecto, la *capa de control* contiene dos categorías de agentes: *gestores* y *especialistas*. Los *gestores* activan, administran y comprueban el estado del resto de componentes de la organización, es decir, de los *recursos* y *agentes especialistas*. El Nivel de Infraestructura incluye tres tipos de *agentes gestores* a la infraestructura, de los mismos tipos y la misma funcionalidad que existen en ICARO-T: *Gestor de Agentes*, *Gestor de Recursos* y *Gestor de la Organización*.

El *Gestor de la Organización* es el encargado de poner en funcionamiento toda la organización y de terminarla en caso de encontrar alguna dificultad o error irrecuperable. Para controlar toda la organización este gestor delega en el *Gestor de Recursos* y el *Gestor de Agentes*. El *Gestor de Recursos* configura y arranca los componentes de tipo recurso del sistema, su-

pervisándolos constantemente y reportando cualquier incidencia al *Gestor de la Organización*. De la misma forma, el *Gestor de Agentes* instancia los agentes especialistas de la aplicación y comprueba periódicamente sus estados.

El Nivel de Infraestructura establece de este modo un esquema de organización jerárquico (que puede observarse en la Figura 5.1) que sirve como patrón de diseño para modelar los sistemas educativos distribuidos. Es más, DAEDALUS incorpora estos agentes gestores en el Nivel de Infraestructura con implementaciones por defecto de ICARO-T, para poder ser empleados directamente en los desarrollos de sistemas multi-agente y hacer uso de este patrón de organización gestionada de forma inmediata.

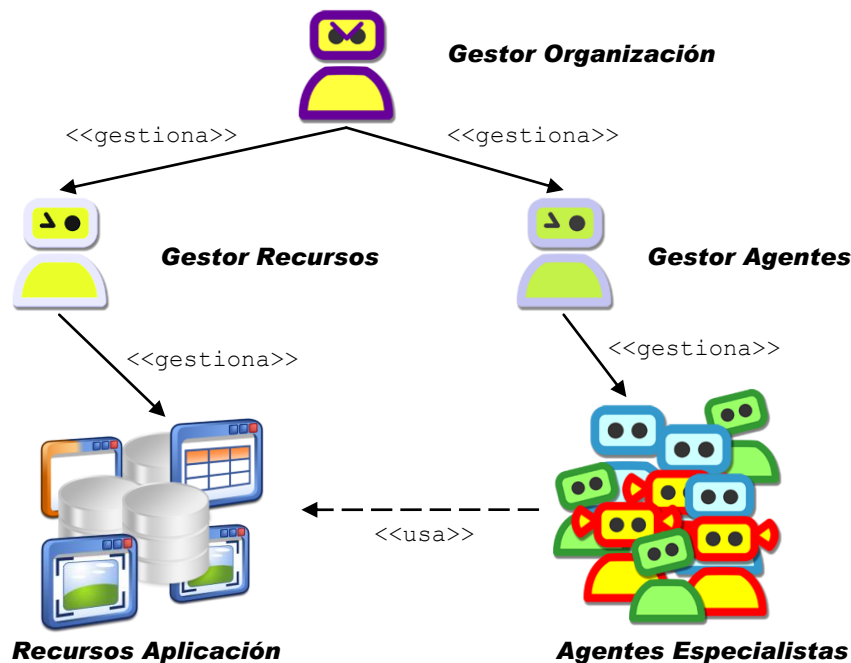


Figura 5.1 Esquema de la organización

Sin embargo, aunque DAEDALUS incorpora de manera nativa en general la mayoría de conceptos y patrones de la plataforma ICARO-T dentro del Nivel de Infraestructura, la incorporación directa de ICARO-T para nuestros requisitos no es posible como veremos a continuación, haciendo necesario extender y adaptar ICARO-T a nuestras necesidades.

5.2.1 Acercando ICARO-T a los requerimientos de DAEDALUS

En primer lugar, aunque la plataforma ICARO-T está pensada para aplicaciones distribuidas en varios nodos, realmente tiene una administración centralizada. Si bien los componentes de la organización ICARO-T se pueden distribuir en nodos separados físicamente, toda esta organización está gestionada por un único trío de gestores. Esto puede ser conveniente ya que aporta un punto centralizado de control y gestión de toda la plataforma, pero uno de los requerimientos de DAEDALUS es que cualquiera de los nodos, es decir dispositivos que contengan agentes y que potencialmente sean utilizados por usuarios, pueda ofrecer características intermitentes en la conexión pero a pesar de ello seguir ofreciendo servicios contextuales y personalizados a dichos usuarios. Esto es un problema desde el punto de vista de nuestros requisitos y que tendremos que resolver dentro del Nivel de Infraestructura de DAEDALUS.

Otro punto a tener en cuenta y derivado de lo anterior es la comunicación en la organización. DAEDALUS necesita de una comunicación flexible que posibilite distintas topologías de conexión entre dispositivos y que se adapte a cualquier tipo de espacio de aprendizaje. Debido a esto resulta necesario ampliar las capacidades de comunicación disponibles en ICARO-T para hacer flexible nuestra arquitectura a distintas configuraciones dinámicas de conexión entre dispositivos.

Finalmente, y no menos importante, un requerimiento necesario para desarrollar tecnología que de soporte a un aprendizaje móvil es que el sistema de agentes funcione obviamente en dispositivos móviles. Los patrones de comportamiento de los agentes de ICARO-T no están enfocados a desarrollos en dispositivos móviles. Esto hace necesario adaptar y articular patrones con características restringidas creando nuevos tipos de comportamiento de agentes capaces de funcionar en dispositivos móviles.

ORGANIZACIÓN COMPLETAMENTE DISTRIBUIDA Y AUTÓNOMA

La organización en la arquitectura DAEDALUS se encuentra distribuida en nodos separados físicamente. Cada uno de los nodos de la arquitectura reside en un dispositivo con capacidad de proceso para albergar algún agente o recurso de la organización. Es decir, los nodos de la organización se corresponden con dispositivos de distintos tipos, como ordenadores personales,

portátiles, servidores, dispositivos móviles, etc., que pueden tener diferentes modos de conexión dependiendo del momento.

Un cambio importante y necesario respecto a ICARO-T es que cada nodo pasa a ser totalmente independiente y auto-gestionado. Dado que en ICARO-T existen únicamente tres agentes gestores para toda la organización que se pueden situar en cualquier nodo, se crean dependencias de conexión hacia los nodos con dichos gestores. En caso de una probable desconexión de un nodo con dependencias del resto de la organización se anularía toda la gestión de los componentes de dicho nodo. En el peor de los casos, si los tres gestores se encuentran en nodos distintos, la mayoría de los nodos muestran dependencias (como se aprecia en el ejemplo de comparación de la Figura 5.2). Sin embargo, en DAEDALUS se flexibiliza este modelo haciendo que cada nodo contenga sus propios gestores. Cada nodo en DAEDALUS es una organización completa auto-gestionada en el sentido estricto que se puede unir y desconectar de forma dinámica a otros nodos, no necesitando de ningún otro agente gestor externo para funcionar. En caso de desconectarse un nodo de la red, los agentes dentro de este nodo pueden seguir funcionando de forma autónoma, ofreciendo total soporte al usuario, y volver a comunicarse con los agentes de otros nodos cuando haya una reconexión.

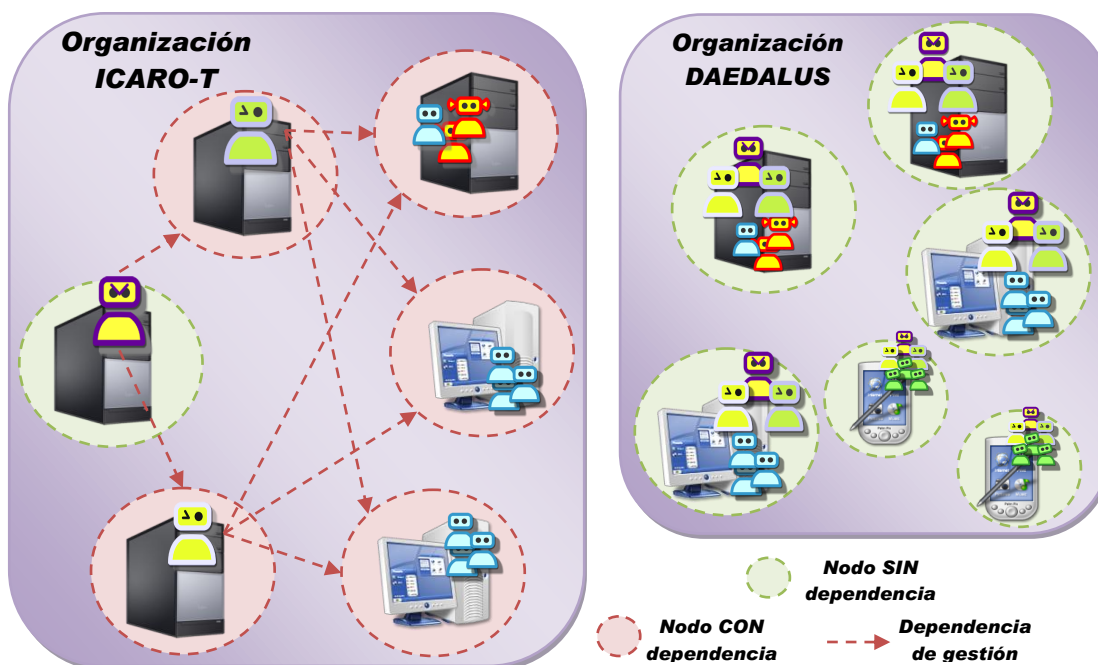


Figura 5.2 Diferencias entre la organización en ICARO-T y la organización en DAEDALUS

El resultado es que DAEDALUS puede considerarse constituida por un conjunto de mini-organizaciones con capacidad de auto-gestionarse y funcionar por separado. Sin embargo, también pueden interactuar conjuntamente formando espontáneamente una organización global. De este modo la organización en la arquitectura DAEDALUS pueden extenderse a lo largo de todo el *espacio de aprendizaje ubicuo*, aunque éste esté distribuido por varios escenarios con distintos tipos de conexión (Figura 5.3). Esta flexibilidad en la forma de conexión, como se verá a continuación, presenta también repercusiones en algo primordial: la comunicación en la organización.

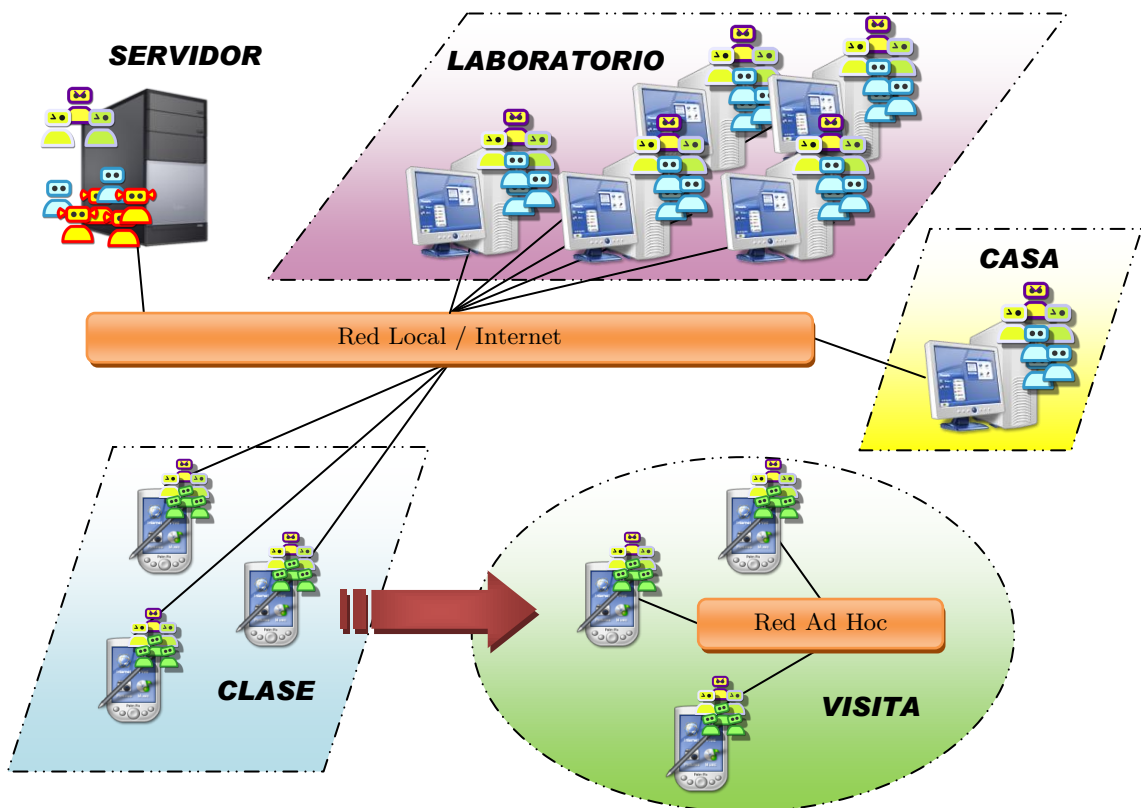


Figura 5.3 Ejemplo de organización distribuida por distintos escenarios

ORGANIZACIÓN CON COMUNICACIÓN FLEXIBLE

Como se ha visto la arquitectura DAEDALUS está pensada para funcionar en diferentes tipos de escenarios (Figura 5.3), lo que establece varios esquemas entre nodos de comunicación. Una típica configuración es la cliente-servidor (Figura 5.4 A), como por ejemplo cuando se utilizan PDAs en clase o en algún edificio como un museo que posee conexión WIFI para comunicar los

agentes de los dispositivos móviles con los de un servidor. Otra configuración se da cuando un nodo de la plataforma ha de poder funcionar por sí solo (Figura 5.4 B), permitiendo dar una experiencia aun así completa por ejemplo en lugares donde no hay conexión o cobertura de datos móvil, como una salida al campo. Por último, se debe permitir un modo de trabajo entre iguales (en inglés ‘*peer to peer*’) (Figura 5.4 C) con el fin de poder dar soporte al aprendizaje colaborativo cuando se esté nuevamente en escenarios sin conexión a Internet pero los dispositivos admitan una comunicación directa a pares, por ejemplo mediante una *red ad-hoc* Bluetooth o WIFI.

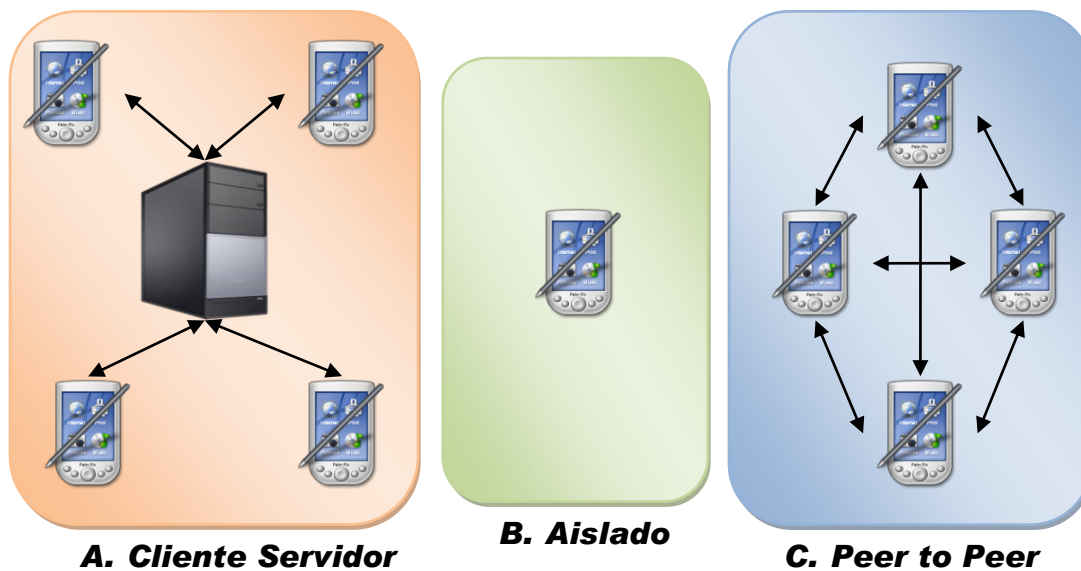


Figura 5.4 Topologías de funcionamiento de comunicación

Los agentes pueden comunicar con los agentes de otros nodos siempre y cuando haya conexión, estando preparados para una eventual desconexión y falta de conexión con el exterior del dispositivo. Una consecuencia de este funcionamiento desacoplado es que DAEDALUS incluye funcionalidades no presentes en ICARO-T relativas a la comunicación, como por ejemplo la capacidad de poder descubrir nodos de forma dinámica.

En el apartado 5.4 se analizará con más detenimiento la comunicación dentro de la arquitectura DAEDALUS y cuáles son las funcionalidades de comunicación necesarias para poder dar soporte a la distribución dentro de la organización.

TIPOS DE COMPORTAMIENTO DE AGENTES DE LA ORGANIZACIÓN

De la misma forma que en la plataforma ICARO-T, los agentes de la arquitectura DAEDALUS se diseñan e implementan siguiendo una serie de patrones de diseño software que establecen sus estructuras internas y que permiten definir los comportamientos de éstos. ICARO-T ofrece patrones para dos tipos de comportamiento de agente: *reactivo* y *cognitivo*. Estos patrones dependen en gran medida de la capacidad de proceso del dispositivo. Por ejemplo, el *patrón agente cognitivo* puede llegar a consumir muchos recursos por lo que es recomendable utilizar servidores dedicados con alta capacidad de proceso y memoria para albergar cierta cantidad de agentes de este tipo. Por el contrario, el modelo de *patrón de agente reactivo* es mucho más ligero lo que posibilita hospedar un buen conjunto de estos agentes en un ordenador personal de potencia media con incluso algún agente cognitivo. Sin embargo, este *patrón de agente reactivo* sigue siendo pesado para poder funcionar en dispositivos móviles. Por supuesto, una plataforma de agentes, para dar soporte a tecnologías de aprendizaje móvil, ha de funcionar obviamente en dispositivos móviles. Dado que tanto el *patrón agente cognitivo*, como el *patrón agente reactivo* no están diseñados para funcionar en dispositivos móviles, DAEDALUS añade un nuevo tipo de patrón que servirá para crear agentes en dispositivos móviles y que se denomina *patrón agente micro-reactivo*. Se trata pues de un nuevo patrón basado en el *patrón agente reactivo* pero algo menos pesado, ya que se intenta adaptar sus estructuras internas manteniendo el mismo modelo reactivo, modificándolo únicamente para funcionar de forma más adecuada en dispositivos móviles.



Figura 5.5 Tipos de Patrones de Agente

Por lo tanto los patrones de tipo de comportamiento de agentes disponibles en DAEDALUS son (Figura 5.5):

1. Patrón Agente Cognitivo
2. Patrón Agente Reactivo
3. Patrón Agente Micro-Reactivo

Los patrones de comportamiento cognitivo y reactivos, incluyendo el micro-reactivo, son presentados con mayor profundidad en el siguiente apartado.

5.3 Los Agentes en DAEDALUS

Como hemos comentado anteriormente, los agentes son el componente software más importante dentro de la arquitectura DAEDALUS y el cual tendrá un rol de controlador dentro de ésta. Los agentes de DAEDALUS son entidades autónomas que perciben el entorno obteniendo información acerca del contexto del *espacio de aprendizaje ubicuo*. A modo de estímulos reaccionan frente a la información recibida, por un lado alterando su estado mental, que es una representación de su conocimiento sobre el mundo y por otro, en función de este estado mental y de los propósitos que ostenten, comunicándose entre sí y efectuando operaciones sobre los elementos del espacio de aprendizaje. Dependiendo del modo en que razonen sobre el entorno y manipulen su estado mental, los agentes mostrarán distintos tipos de capacidades cognitivas y deliberativas en su comportamiento.

Un punto importante ya destacado es que DAEDALUS emplea los patrones de comportamiento de agente que define ICARO-T, los cuales aportan tanto una descripción en términos abstractos de cómo tienen que ser estos comportamientos, junto con un marco de desarrollo software a modo de armazón para diseñar y programar agentes de estos tipos. Los patrones de comportamiento definen cómo los agentes de cada tipo perciben el entorno, de qué forma se representarían éste en su estado mental, qué capacidades cognitivas tienen y qué posibilidades tienen para interactuar con el entorno y otros agentes. De modo general, los patrones de agentes de DAEDALUS se pueden estructurar internamente en tres partes:

- *Un modulo de percepción*. Se encarga de procesar la información que recibe el agente del ‘exterior’, filtrarla y asimilarla. A menudo un

agente únicamente estará interesado en la información relativa a los objetivos y tareas para los que está diseñado, por eso el módulo de percepción se encarga de seleccionar la información adecuada y representarla como evidencias que podrán ser manipuladas por el siguiente módulo.

- *Un módulo de control y razonamiento.* Es el encargado del procesamiento del estado mental del agente. Dependiendo del tipo de comportamiento del agente, este módulo ofrecerá distintos grados de complejidad, integrando en éste todos los procesos que rigen el comportamiento del agente y pudiendo integrar mecanismos de razonamiento, de aprendizaje, etc.
- *Un módulo de actuación.* Se responsabiliza de llevar a cabo las acciones que son ordenadas o planificadas por el módulo de control.

Básicamente, se identifican dos tipos de comportamiento diferentes de agentes en función del *módulo de control y razonamiento*: reactivos y cognitivos. Por un lado, el *patrón agente cognitivo* aporta a la arquitectura un tipo de comportamiento muy potente y abstracto. Los agentes cognitivos de DAEDALUS tienen mecanismos efectivos para manejar e interpretar conocimiento, y se definen en términos abstractos que incluyen la definición de tareas y objetivos junto con mecanismos de inferencia para tratarlos.

Por otro lado, el *patrón agente reactivo* puede verse como un patrón de comportamiento reactivo puro. Los agentes reactivos reciben impulsos del exterior que les hacen cambiar de estado y efectuar acciones para avanzar hacia estados objetivo. Como se ha comentado en el apartado anterior, la arquitectura DAEDALUS ha adaptado este patrón de comportamiento reactivo para poder ser empleado en dispositivos con características restringidas, definiendo lo que se ha denominado el *patrón agente micro-reactivo* para ser utilizado en PDAs y teléfonos móviles. Al ser patrones, tanto el reactivo como el micro-reactivo, que aunque con distintas capacidades en términos de potencia ambos comparten las mismas estructuras y habilidades, variando únicamente en su implementación, la forma de definirlos se mantiene igual y se tratarán consecuentemente en un mismo apartado.

En las siguientes secciones se exponen con más detenimiento los detalles de los tipos de patrones de comportamiento de agentes de DAEDALUS y sus estructuras internas.

5.3.1 Patrón Agente (Micro-)Reactivo

Los patrones reactivo y micro-reactivo ofrecen mecanismos relativamente simples, comparados con el patrón cognitivo, pero a su vez extremadamente flexibles para definir agentes en DAEDALUS. Esto los hace idóneos cuando se necesita definir servicios inteligentes sencillos dentro de la infraestructura o se desea estructurar un servicio más complejo en responsabilidades simples repartidas entre varios *agentes (micro-)reactivos*.

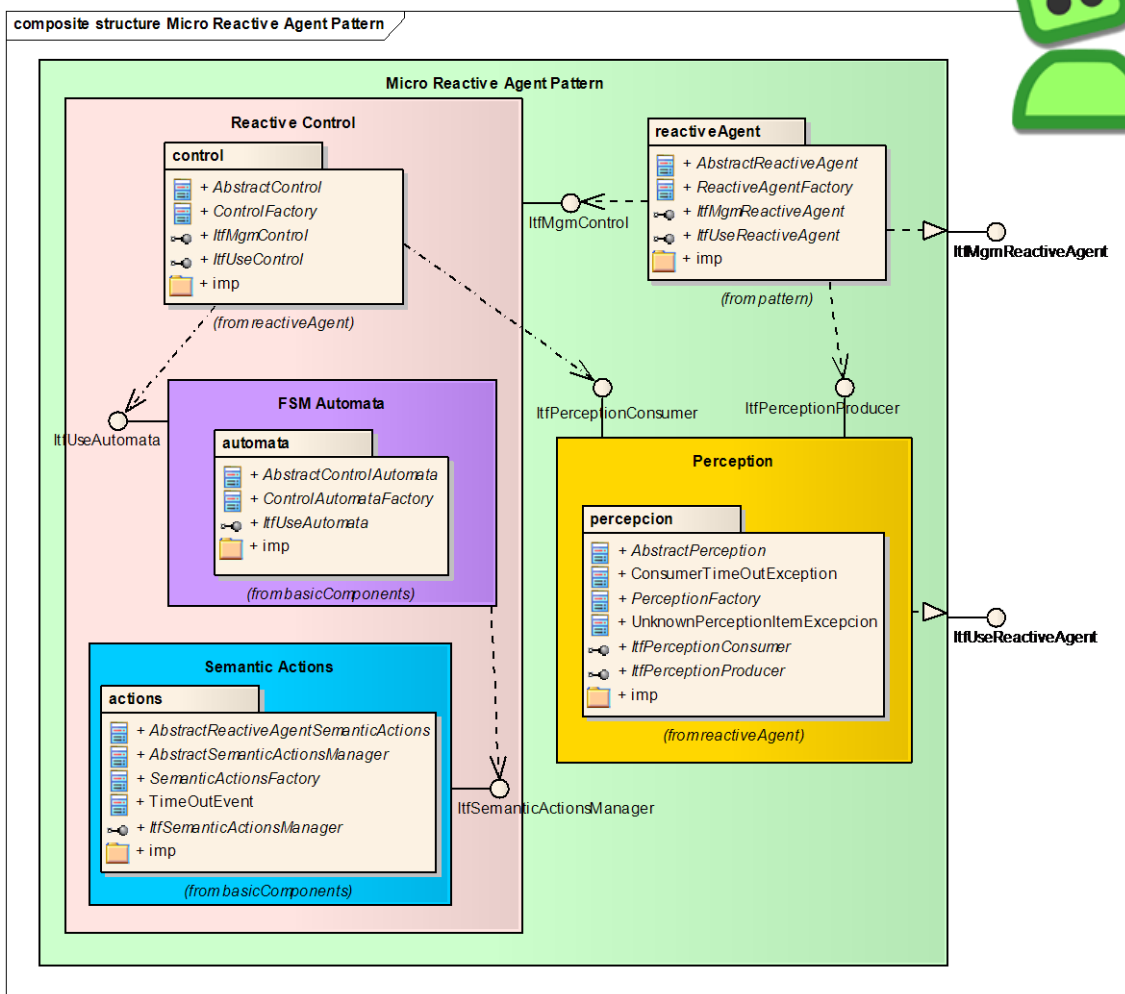


Figura 5.6 Patrón Agente (Micro-)Reactivo

En la Figura 5.6 puede apreciarse un diagrama de la estructuración interna de este patrón de diseño. Todos los *agentes (micro-)reactivos* poseen dos interfaces externas claramente diferenciadas. Por un lado la *interfaz de uso* (*ItfUseReactiveAgent*), a través de la cual reciben mensajes y eventos

del exterior, y por otro lado la *interfaz de administración* (ItfMgmReactiveAgent), con operaciones para que los agentes gestores puedan controlar e inspeccionar el estado interno del agente. La *interfaz de uso* está directamente ligada al *módulo de percepción* del agente (Perception en la Figura 5.6). El *módulo de percepción* del agente (*micro-*)reactivo filtra todos los mensajes y eventos, y los interpreta como entradas para su *módulo de control* (Reactive Control en la Figura 5.6). El *módulo de control* se encarga de consumir estas entradas provenientes de la percepción y las utiliza para alimentar un *autómata* implementado mediante una máquina de estados finitos (FSM Automata en la Figura 5.6). Este *autómata* transita a través de estados debido a las entradas que va recibiendo y tiene registradas una serie de *acciones semánticas* asociadas a cada transición. Las *acciones semánticas*, son listados de operaciones que el agente puede realizar. El *gestor de acciones semánticas* (Semantic Actions en la Figura 5.6) controla la ejecución de las acciones y si su finalización debe bloquear al agente o no, dependiendo del tipo de transición. Por tanto, en definitiva, el *módulo de control* pone en marcha el *autómata* cada vez que se reciben nuevos impulsos del exterior y que a su vez ejecuta *acciones semánticas* en consecuencia, otorgando de esta manera un comportamiento puramente reactivo al patrón. La definición de *agentes (micro-)reactivos* mediante este tipo de patrón de comportamiento se realiza con la especificación tanto de la *máquina de estados*, para el *módulo de control*, como la de las *acciones semánticas* asociadas (Figura 5.7).

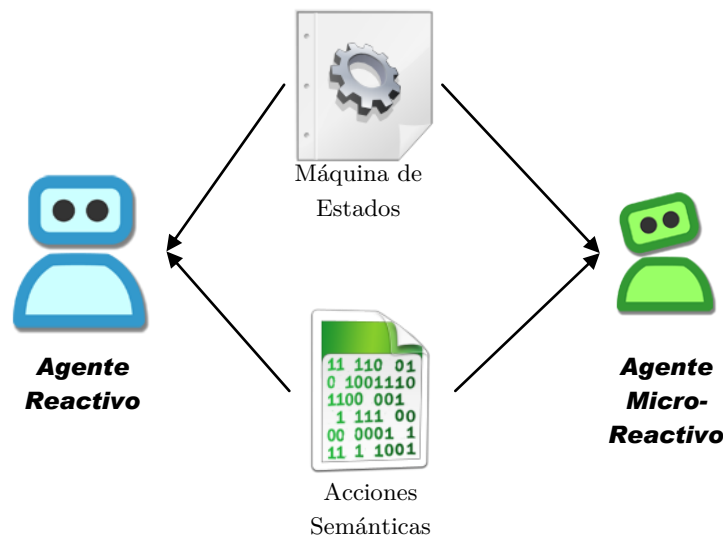


Figura 5.7 Elementos para la definición de agentes (*micro-*)reactivos

La *máquina de estados* establece los estados del *autómata* del *agente (micro-)reactivo* junto con las transiciones que indican qué entrada está asociada a qué *acción semántica*. Las *acciones semánticas* se describen mediante instrucciones en código. De forma concreta, el desarrollador que utiliza el *patrón agente (micro-)reactivo* para instanciar agentes en la infraestructura únicamente ha de aportar un archivo con la *máquina de estados* codificada en formato XML, junto con las clases de código Java de las *acciones semánticas*.

5.3.2 Patrón Agente Cognitivo

El *patrón agente cognitivo* exhibe un comportamiento mucho más complejo que los patrones reactivos ya que incluye representaciones mentales internas, tales como *evidencias*, *creencias*, *objetivos* y *tareas*, que le permiten trabajar a niveles de abstracción mayores.

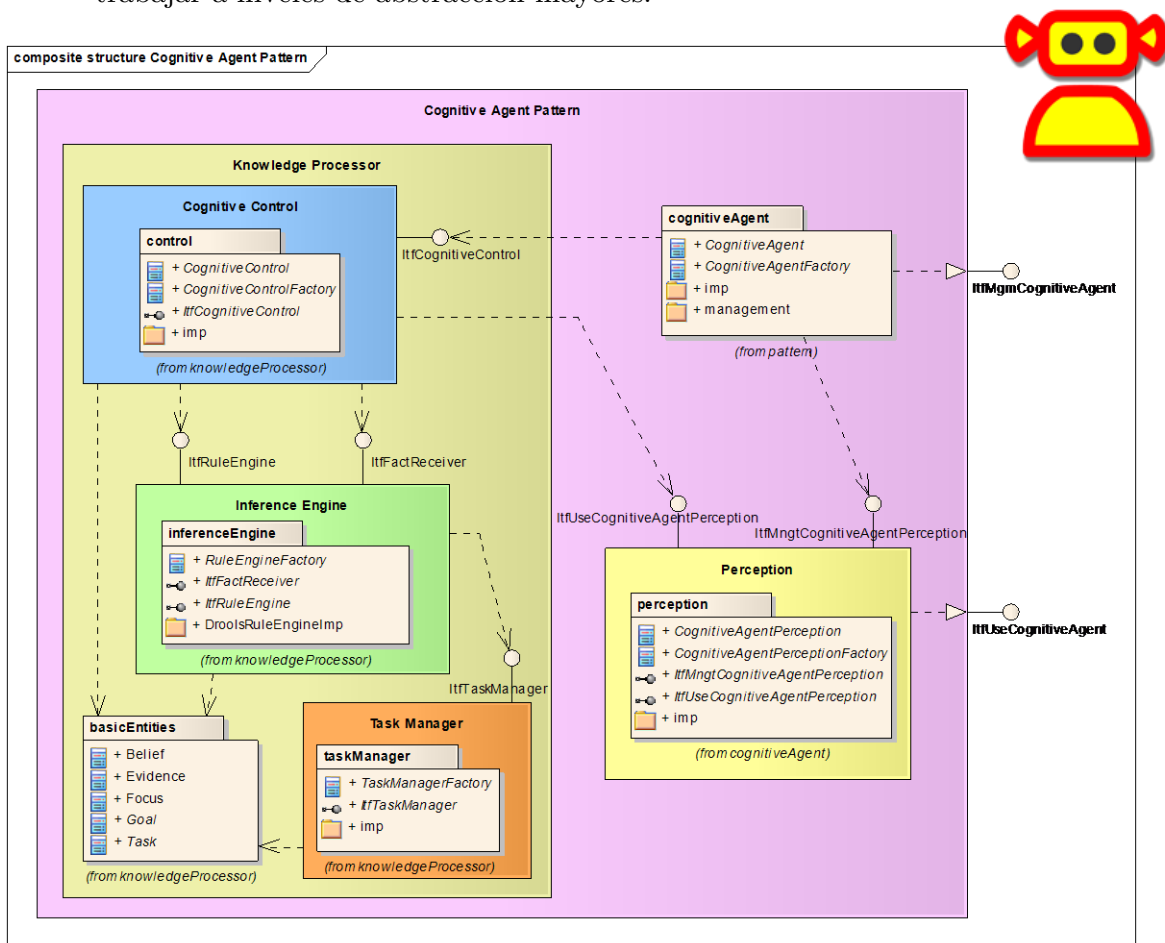


Figura 5.8 Patrón Agente Cognitivo

En la Figura 5.8 se muestra un diagrama de la estructura interna de éste patrón de diseño. El *patrón agente cognitivo* posee, al igual que su homólogo reactivo, tanto una interfaz de uso (*ItfUseCognitiveAgent*) que le permite recibir información del exterior en forma de mensajes y eventos, junto con una interfaz de administración (*ItfMgmCognitiveAgent*) para ser gestionado. El *módulo de percepción* del patrón cognitivo (*Perception*) resulta ligeramente más complejo que el del reactivo y es donde la información recibida por el agente es interpretada y puesta a disposición para el *módulo de control* (*Cognitive Control*). No obstante, este *módulo de control* está integrado dentro de lo que se denomina el *procesador de conocimiento* del agente cognitivo (*Knowledge Processor*). El *procesador de conocimiento* es donde se manipulan las entidades propias del estado mental del agente cognitivo y se realizan razonamientos sobre ellas. Para este fin, el *procesador de conocimiento* incluye un *motor de inferencia*, que ofrece mecanismos para razonar acerca de los *objetivos* y *tareas* del agente, en base a las *evidencias* que recibe y *creencias* que genera.

Como se muestra en la Figura 5.9, los mensajes y eventos recibidos por el agente son en un primer término filtrados por el *módulo de percepción*, y posteriormente interpretados y convertidos en *evidencias*. Una *evidencia* representa información del mundo exterior que ha sido extraída a partir de los eventos y mensajes. El *módulo de control* del *procesador de conocimiento* acepta estas *evidencias* y realiza un proceso de asimilación, transformándolas en *creencias* del agente que sirven para alimentar el *motor de inferencia* del patrón.

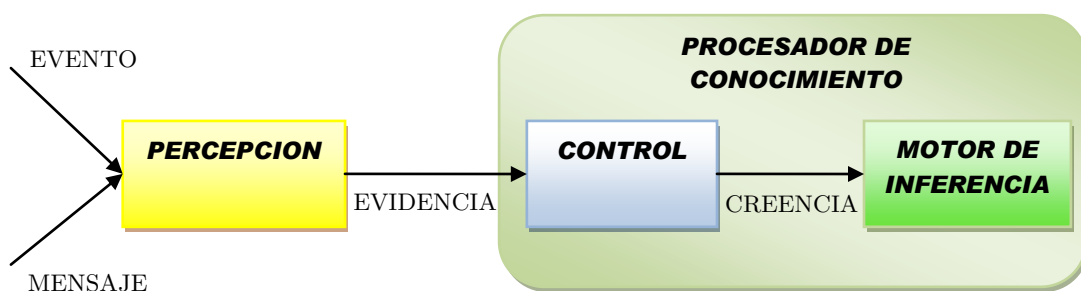


Figura 5.9 Procesamiento de la información del agente cognitivo

El conjunto de *creencias* que posee en un determinado momento un agente cognitivo le sirven para seleccionar las *tareas* a realizar más adecuadas para poder satisfacer sus propios *objetivos*. De forma consecuente, a la hora de definir un agente cognitivo dentro de la arquitectura DAEDALUS, es neces-

rio precisar los procesos mentales que asocian *creencias* con las *tareas* a efectuar para conseguir sus *objetivos*. Esto se traduce de forma concreta para el *patrón agente cognitivo* en que el comportamiento se hace explícito mediante tres elementos diferenciados (Figura 5.10): (1) un conjunto de objetivos, (2) un conjunto de tareas y (3) los procesos de gestión de los objetivos. Los procesos de gestión de objetivos sirven para guiar los razonamientos del agente, definiendo cuándo y en qué orden se ejecutan las tareas en base a las creencias conocidas por el agente.

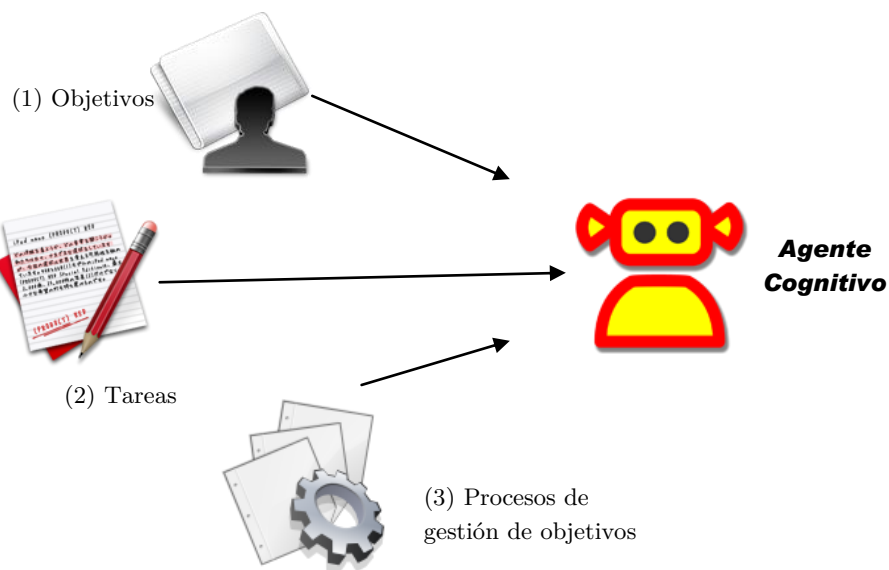


Figura 5.10 Elementos para la definición de agentes cognitivos

Estos procesos se modelan dentro del *motor de inferencia* del patrón mediante reglas de producción. Las reglas que definen los procesos de gestión de objetivos se dividen en:

- *Creación de Objetivos*. Establecen unos objetivos iniciales, además de permitir la creación de nuevos objetivos según se presenten nuevas metas y condiciones de forma dinámica.
- *Focalización de Objetivos*. Seleccionan objetivos en función de las prioridades y necesidades del agente. El agente puede saltar de objetivo a objetivo, atendiendo a qué algún objetivo sea más prometedor, el que se encuentra en resolución sea inalcanzable o haya que postergarlo.
- *Resolución de Objetivos*. La resolución de objetivos se traduce en la ejecución de tareas por parte del agente para poder influir en su en-

torno y de forma indirecta en su propio estado mental. Las tareas que realiza el agente tienen con fin poder avanzar hacia la resolución del objetivo focalizado.

5.4 Comunicación en DAEDALUS

La comunicación en la arquitectura DAEDALUS define de qué modo se transmitirá la información a través de los nodos de toda la infraestructura tecnológica. Así pues, la comunicación es un servicio fundamental dentro del Nivel de Infraestructura que habilita que sus componentes software puedan intercambiar información, aun estando en nodos diferentes o en definitiva en dispositivos separados físicamente. La comunicación entre estos componentes en DAEDALUS se puede efectuar de dos maneras según las necesidades: mediante *mensajes* y mediante *eventos*. Una consideración importante que afecta a esta comunicación es nuevamente el carácter distribuido de la infraestructura, junto con la posibilidad eventual de conexión/desconexión dinámica entre nodos. De esta forma, las facilidades de comunicación que estén presentes en cada nodo deben de gestionar además el descubrimiento de los demás nodos accesibles junto con la localización y registro de agentes y recursos remotos. En resumen, la comunicación en DAEDALUS se divide básicamente en cuatro funcionalidades:

- Envío y recepción de mensajes.
- Suscripción y distribución de eventos.
- Descubrimiento dinámico de nodos.
- Registro de agentes y recursos.

Dentro del Nivel de Infraestructura se aportan uno o varios recursos de comunicación que se encargarán de proveer de estas cuatro funcionalidades. Estos recursos están presentes en cada uno de los nodos, para poder ofrecer funcionalidades de comunicación entre los agentes del mismo nodo incluso ante una eventual desconexión del resto de nodos, manteniéndose así el carácter autónomo de cada dispositivo del entorno ubicuo. Todos los agentes en DAEDALUS poseen las mismas capacidades de comunicación sea cual sea su patrón de comportamiento, de tal forma que por ejemplo un agente micro-reactivo puede enviar mensajes de la misma manera que un agente cognitivo y puede suscribirse y recibir eventos al igual que haría éste. Por tanto lo que

diferenciará a los agentes con distintos patrones de comportamiento no será su capacidad de comunicación, sino la manera de tratar y reaccionar ante los mensajes y eventos provistos por esta comunicación.

Cabe destacar que aunque con gran relevancia dentro de la infraestructura, el concepto de comunicación se encuentra al mismo nivel que el concepto de organización o el concepto de agente. Esto representa una diferencia importante con respecto a otras plataformas que, como por ejemplo JADE y su equivalente móvil JADE-LEAP (Fabio Bellifemine et al., 2008), articulan todo un marco de desarrollo multi-agente basado principalmente en la comunicación entre agentes mediante FIPA, sin aportar pautas a los desarrolladores para diseñar los propios agentes ni estructurar sus organizaciones.

5.4.1 Envío y recepción de mensajes

Un mensaje es un elemento de comunicación que se envía de un agente emisor a uno o varios agentes receptores. Cuando un agente manda un mensaje hace explícito quién o quiénes serán sus destinatarios. El envío de mensajes en DAEDALUS está controlado por los recursos de comunicación presentes en los nodos del agente emisor y del agente receptor, como se muestra en el diagrama de secuencia de la Figura 5.11.

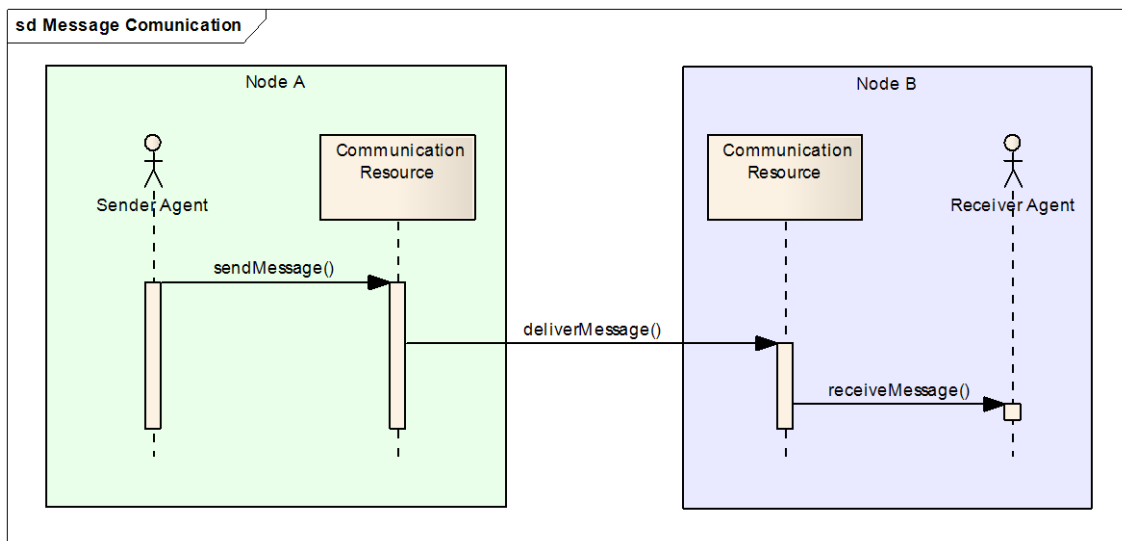


Figura 5.11 Envío y recepción de mensajes

El recurso de comunicación (**Communication Resource**) recibe del agente emisor (**Sender Agent**) el mensaje y se encarga de retransmitirlo a su

homólogo del nodo del agente receptor (*Receiver Agent*), haciéndolo llegar en última instancia al agente receptor. De esta forma, los recursos de comunicación abstraen a los agentes de los detalles de bajo nivel relativos a la transmisión de mensajes, tales como tecnologías, canales, protocolos, seguridad y encriptación, etc.

Los mensajes están básicamente constituidos por un emisor, unos receptores y un contenido. No obstante, los mensajes pueden presentar cualquier estructura si es necesario para permitir la interoperabilidad con otras plataformas de agentes, como por ejemplo ACL del estándar FIPA (FIPA, 2002d). El contenido es la información real que se transmite entre dos agentes. Este contenido está codificado mediante algún lenguaje cuya semántica ha de ser reconocida por ambos agentes. A tal respecto, se trata en el capítulo 6, dentro del Nivel de Semántica y Contexto, todo el *modelo de conocimiento* que los agentes en DAEDALUS pueden intercambiar y manejar.

5.4.2 Suscripción y distribución de eventos.

Mientras que los *mensajes* son transferidos entre agentes especificando sus destinatarios, los *eventos* son diseminados por todos los nodos accesibles y distribuidos a los agentes en función de su *tipo*.

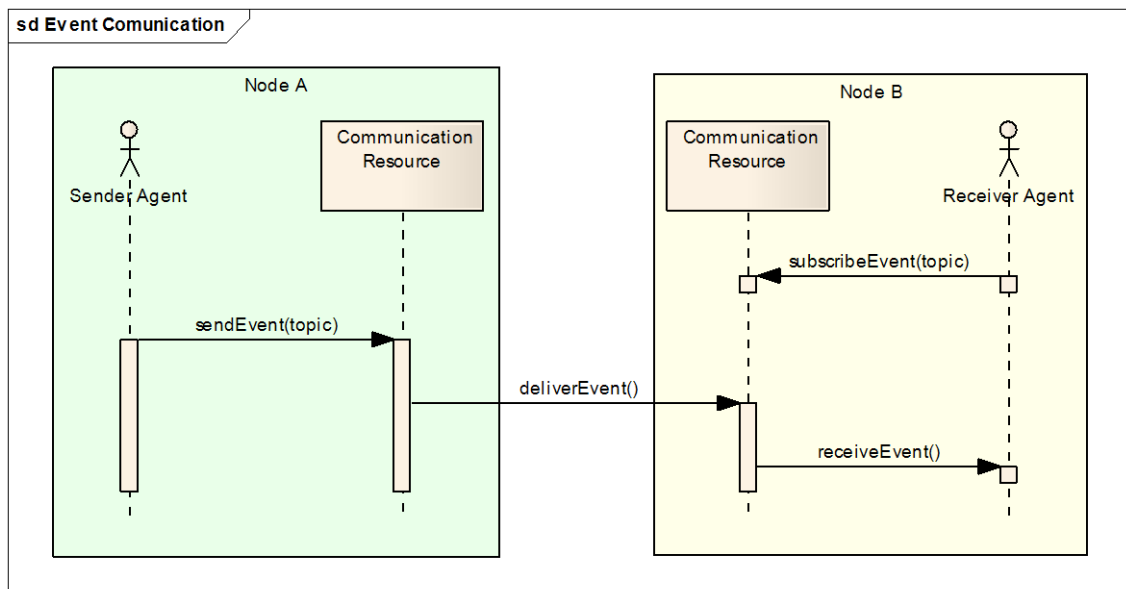


Figura 5.12 Suscripción y recepción de eventos

Mediante un mecanismo de suscripción, cada agente puede registrar en qué *tipos de eventos* está interesado, recibiendo únicamente éstos.

Como se muestra en el diagrama de secuencia de la Figura 5.12, los agentes realizan un paso previo de suscripción a eventos. El agente receptor (*Receiver Agent*) indica al recurso de comunicación (*Communication Resource*) de su nodo a qué tipo de eventos se suscribe. Todos los eventos que son enviados por un agente emisor (*Sender Agent*), son distribuidos a los demás recursos de comunicación de otros nodos, que los filtran y reparten entre los agentes interesados.

Los eventos son un método de comunicación muy potente para la disseminación de información dentro de sistemas multi-agente (Tarkoma, 2003), además de un mecanismo clave para transmitir cambios en el contexto dentro de la arquitectura DAEDALUS, utilizando la taxonomía de tipos de sucesos de contexto que se presenta en el apartado 6.3.

5.4.3 Descubrimiento dinámico de nodos

Algo totalmente necesario dentro de un entorno distribuido con múltiples dispositivos, los cuales se corresponden con nodos en la infraestructura que pueden añadirse de forma dinámica, es un servicio de descubrimiento dinámico de nodos. Cuando un nuevo dispositivo se conecta o entra en la red, los recursos de comunicación detectan este suceso y se genera un evento de conexión que avisa a los agentes interesados y les posibilita establecer contacto con los agentes del nodo recién descubierto u operar con los recursos contenidos en éste. De la misma forma cuando se detecta que un nodo ya no es accesible, se genera el consecuente evento que indica la desconexión de dicho nodo.

Así pues nuevamente, los recursos de comunicación abstraen al resto de la infraestructura de los detalles tecnológicos de la red, escondiendo las tecnologías y protocolos de detección de nodos remotos, sustituyéndolos por mecanismos basados en el sistema nativo de eventos de la arquitectura.

5.4.4 Registro de agentes y recursos

Los recursos de comunicación llevan la cuenta de cuáles son los agentes y recursos disponibles en el conjunto total de nodos accesibles. Es similar a los componentes de directorio que se establecen en la especificación de arquitec-

tura abstracta de FIPA (FIPA, 2002a) y presentes en otras plataformas, como el *Directory Facilitator* presente en JADE (F. Bellifemine, G. Caire, & G. Rimassa, 2001). Esto permite a los agentes disponer un registro con las direcciones de los agentes y recursos de la organización. Cada vez que se produce una conexión o desconexión de un nodo remoto, se actualiza consecuentemente este registro de manera local en cada nodo. Los agentes pueden consultar en cualquier momento esta funcionalidad de registro a través de los recursos de comunicación de la arquitectura.

6

NIVEL DE SEMÁNTICA Y CONTEXTO

Si bien el Nivel de Infraestructura de la arquitectura de DAEDALUS sienta una base para incorporar los instrumentos necesarios con los que poder distribuir el sistema a lo largo de la infraestructura, el Nivel de Semántica y Contexto es el encargado de habilitar la cohesión semántica a lo largo de toda esta distribución en dicha infraestructura. Este nivel representa el primer bloque de la Plataforma Educativa de la arquitectura genérica de DAEDALUS, que se presentó en el apartado 4.2.1. El objetivo principal de este nivel es aportar el “pegamento semántico” con el que unir los distintos componentes de la organización y que permita su interoperabilidad en términos de un vocabulario compartido común de dominio educativo. Para este fin es necesario establecer técnicas para la representación

y compartición del conocimiento en DAEDALUS, las cuales usarán los agentes para hacer referencia a las entidades del contexto presentes en el entorno ubicuo. Mediante un *modelo de conocimiento* se estructuran todas las entidades y relaciones presentes en el contexto del *espacio de aprendizaje ubicuo* y se representan los distintos eventos que establecen cambios en dicho contexto. Este conocimiento enfocado principalmente al tratamiento de los elementos contextuales será imprescindible para poder articular funcionalidades y servicios personalizados a los usuarios y adaptados a sus situaciones dentro del *espacio de aprendizaje ubicuo*.

6.1 Representación del Conocimiento de DAEDALUS

La arquitectura DAEDALUS requiere de mecanismos para poder caracterizar y estructurar la información que sus componentes software conocen acerca de los elementos presentes en el *espacio de aprendizaje ubicuo*. Toda esta información estructurada articula el conocimiento que el sistema posee sobre el entorno y su contexto, incluyendo por tanto las personas que hay en él, sus relaciones sociales, los procesos pedagógicos que están realizando, los dispositivos que emplean, etc. Este conocimiento acerca del contexto del entorno ubicuo resulta de suma importancia para permitir que los agentes puedan ofrecer servicios pedagógicos inteligentes que estén personalizados a los usuarios y adaptados a los procesos que realizan. No obstante, para habilitar su procesamiento por los elementos tecnológicos de la infraestructura, este conocimiento necesita ser representado de forma tratable computacionalmente. Es por tanto que se precisa de la definición de un *modelo de conocimiento* en DAEDALUS que estructure toda la información conocida en la organización y que establezca mecanismos para representar, manipular y diseminar el conocimiento acerca del entorno ubicuo a lo largo de los componentes software de ésta. Se trata pues de definir las técnicas de representación que permitan especificar los elementos de dominio de contexto compartidos y proporcionar, gracias a éstas, un vocabulario común que haga posible el entendimiento en la comunicación y cooperación entre los componentes de la organización. Es importante destacar que, dado que el contexto es una entidad de por sí cambiante con respecto del tiempo, será necesario modelar tanto la representación estática que sirva para captar las particularidades de los elementos del con-

texto en un momento dado, como las características dinámicas propias del contexto, que reflejen los cambios que acontecen en el entorno ubicuo.

6.1.1 Ontologías

El *modelo de conocimiento* de DAEDALUS se describe mediante una ontología. Una ontología es una especificación explícita formal de una conceptualización compartida (Borst, 1997; Gruber, 1993). Las ontologías capturan y especifican el conocimiento de dominio junto con su semántica inherente a través de terminologías consensuadas con axiomas y restricciones formales. Por tanto, las ontologías ofrecen un mecanismo estándar y formal, cuya fuerza de conceptualización se consigue mediante la normalización y formalización. La normalización queda reflejada gracias a un acuerdo semántico en el significado de los términos, mientras que la formalización queda reflejada en los lenguajes ontológicos formales que son usados para codificar las ontologías (Bachimont, Isaac, & Troncy, 2002). Entre los lenguajes ontológicos actuales destaca OWL (*Web Ontology Language*), lenguaje auspiciado por la W3C y cuya versión más reciente OWL 2 (W3C, 2009) es un estándar de facto para la explicitación de ontologías.

Técnicamente, las ontologías deben tener asociado un vocabulario bien formado junto con relaciones claramente definidas entre los distintos términos. De acuerdo a su nivel de expresividad podemos clasificar las ontologías en ligeras y pesadas. Las ontologías ligeras buscan principalmente una conceptualización consensuada, incluyendo conceptos, taxonomías y relaciones entre conceptos y sus propiedades (Corcho, Fernández-López, & Gómez-Pérez, 2003). Las ontologías pesadas aplican axiomas y restricciones, que son los bloques fundamentales para hacer una interpretación semántica de los conceptos y las relaciones de las ontologías (Gruber, 1995).

El objetivo dentro del Nivel de Semántica y Contexto es el de obtener una ontología ligera que sirva de *modelo de conocimiento* y que permita su utilización en cualquier dispositivo. Esta ontología incluirá básicamente los elementos contextuales que son necesarios representar del *espacio de aprendizaje ubicuo* para habilitar las funcionalidades que permitan que los agentes coordinen los servicios personalizados a los usuarios y adaptados al contexto. En consecuencia, toda la interpretación semántica, razonamiento y deducción a partir del *modelo de conocimiento* es delegada en las capacidades de los agentes de DAEDALUS, cuya definición se corresponde dentro del Nivel de Organización Educativa que se expone en el capítulo 7. Cabe adelan-

tar, sin embargo, que los agentes con comportamientos más potentes, funcionando en dispositivos más potentes, serán los encargados de efectuar inferencias y razonamientos más complejos, preparando y generando en cierto modo información de utilidad a agentes con menos capacidades, habitualmente localizados en dispositivos móviles.

La intención de la arquitectura DAEDALUS no es especificar una ontología universalmente completa y que abarque cualquier posible caso pedagógico. Ya existen autores y diversas iniciativas que buscan esta aproximación (Mizoguchi & Bourdeau, 2000; Berri, Benlamri, & Atif, 2006). En vez de buscar la completitud, la aproximación que se sigue es mucho más pragmática y consiste en lograr una ontología apta para las necesidades particulares que presenten los *espacios de aprendizaje ubicuos* a desarrollar. En definitiva, la ontología ha de ser suficientemente simple para permitir su utilización en dispositivos móviles con menores capacidades, pero a la vez ser lo convenientemente expresiva y potente para este fin, pudiendo servir de base extensible que permita añadir nuevos elementos para nuevos dominios educativos.

6.1.2 Integración con otras ontologías y extensibilidad

La ontología del *modelo de conocimiento* de DAEDALUS busca describir todos los elementos del entorno ubicuo que son necesarios para los agentes pero siempre teniendo en cuenta y reutilizando a ser posible modelos contextuales ontológicos presentes en la literatura (Ye, Coyle, Dobson, & Nixon, 2007).

Aparte de la necesidad de obtener una ontología ligera para poder ser utilizada dentro de dispositivos con potencia y características limitadas, otra preocupación central en el desarrollo de la ontología de nuestra infraestructura es la extensibilidad. Mediante la extensibilidad será posible añadir conceptos y relaciones de manera que se pueda ampliar la capacidad expresiva de la ontología en función de las necesidades de nuevos espacios de aprendizaje. Como recomendación general, todas las ontologías han de estar diseñadas para ofrecer una base conceptual (Gruber, 1995). Debe ser fácil poder añadir nuevos términos a las ontologías sin afectar a los presentes, esto quiere decir que los nuevos elementos no deben causar modificación, ni confusión ni ambigüedad sobre el significado de los preexistentes.

Una forma que tiene la arquitectura de DAEDALUS de garantizar esta extensibilidad en el *modelo de conocimiento* es ofrecer conceptos clave que

puedan servir como puntos de anclaje a la hora de añadir los nuevos términos. Este es el enfoque utilizado por las denominadas *ontologías comunes superiores* (Mascardi, Cordi, & Rosso, 2007). Este tipo de ontologías están diseñadas para aportar una serie de conceptos clave y necesarios que sirven como cimiento para otras ontologías de nivel más específico. Un ejemplo de esto se observa dentro de la ontología de contexto CONON (Gu, Pung, & Zhang, 2004). CONON define una ontología común superior para los conceptos generales presentes en la computación pervasiva (tales como persona, localización, entidad computacional, actividad) que utiliza de base para ontologías de dominio específico (como por ejemplo la domótica).

Otra forma de fomentar la extensibilidad es la utilización de conceptos presentes en múltiples ontologías de dominio. Por ejemplo, la ontología de contexto SOUPA del sistema CoBrA (H. Chen, Finin, & Joshi, 2005) toma prestados términos de distintas ontologías de dominio estándar como FOAF (Friend Of A Friend) (Brickley & Miller, 2007), DAML-Time (Hobbs & Pan, 2004), OpenCyc (Matuszek, Cabral, Witbrock, & DeOliveira, 2006), RCC (Region Connection Calculus) (Borgo, Guarino, & Masolo, 1996), y Rei Policy Ontology (Kagal, Finin, & Joshi, 2003). Esta misma aproximación es usada en DAEDALUS, tomando como referencia otras ontologías además de distintas taxonomías de elementos pedagógicos y estándares como IMS LIP (IMS GLC, 2005), IMS LD (IMS GLC, 2003), IEEE LOM (IEEE Standards Department, 2002), etc., como fuente que pueda servir de términos base a partir de los cuales ampliar la ontología del modelo de conocimiento.

6.1.3 Agentes como manipuladores de contexto distribuido

La mayoría de aproximaciones basadas en ontologías que manejan información de contexto lo realizan de manera centralizada (Ye et al., 2007), inclusive las que hacen uso de la tecnología de agentes, como por ejemplo el sistema multi-agente CoBrA y su Context Broker (H. Chen et al., 2004a). En contraposición a esto, la arquitectura DAEDALUS se construye alrededor de un tratamiento descentralizado y distribuido de la información contextual. De esta forma en vez de utilizar un componente central en el sistema para tratar, almacenar y repartir el contexto, como ocurre en la amplia mayoría de arquitecturas propuestas para dar soporte al aprendizaje móvil mediante el contexto (Schmidt, 2007; Drira et al., 2006; Wang, 2004; Trifonova & Ronchetti, 2004), DAEDALUS se constituye por múltiples componentes autónomos que se dedican a esta tarea, es decir, los agentes. Esto significa que todos los agen-

tes definidos dentro de los niveles de la Plataforma Educativa son considerados en mayor o menor medida como *manipuladores de contexto*, definiéndose en este Nivel de Semántica y Contexto los modelos y mecanismos necesarios para habilitarles como tales y aportándoles la facultad de poder extraer, compartir, razonar y actuar en base a la información de contexto del *espacio de aprendizaje ubicuo*.

Cada agente de DAEDALUS es capaz de interpretar y usar el contexto en base a sus necesidades y capacidades. Habitualmente los agentes se diseñan para estar únicamente interesados en los elementos de contexto que son precisos para realizar sus tareas y cumplir sus responsabilidades.

Ciertos agentes con menor capacidad de proceso, en concreto los agentes con comportamientos reactivos situados en dispositivos móviles, dependerán de otros agentes más potentes que les provean de información útil del contexto previamente analizada. Debido a esto último, a pesar que se establece a todos los agentes de DAEDALUS como manipuladores de contexto, de forma común aparecerán dentro de la organización de DAEDALUS componentes cuya responsabilidades principales estén en mayor medida asociadas a la manipulación de la información de contexto para provisionar a otros agentes. Éstos reciben el nombre de *agentes de contexto* y pueden establecerse distintos tipos de funciones respecto a su relación primaria con el modo de procesamiento de la información contextual. Dentro de la arquitectura DAEDALUS se observan, de forma general y desde la perspectiva del Nivel de Semántica y Contexto, agentes que efectúan extracción de información del contexto, derivación de nuevo conocimiento de contexto, y agregación y persistencia de toda esta información (ver Tabla 6.1).

Función del Agente de Contexto	Descripción	Modo General de Actuación
Extracción	Obtener información de contexto directamente de una fuente.	Efectúa un minado de datos en una base de datos o repositorio. Recibe datos de un sensor. Traduce estos datos a información del modelo de conocimiento.
Derivación	Inferir nueva información de contexto.	Aplica reglas o cualquier método de deducción sobre los elementos del contexto con el fin de conseguir nueva información útil para otros componentes a partir del contexto con el que se cuenta.
Agregación/Persistencia	Recopilar y almacenar información de contexto previamente extraída para su posterior reutilización.	Empaqueta la información extraída de diversas fuentes para difundirla entre nodos de la organización y compartirla entre distintos componentes distribuidos. Guarda la información de contexto en una base de datos para permitir realizar búsquedas y recuperaciones por parte de cualquier otro componente.

Tabla 6.1 Funciones de agentes de contexto

La extracción de contexto implica obtener información directa a partir de fuentes como sensores en los dispositivos, bases de datos, repositorios o interfaces de usuario. Este proceso implica la conversión de la información al *modelo de conocimiento*, que permitirá ser entendida y procesada por todos los componentes de la arquitectura. La derivación de contexto suele ser una función que implica la inferencia y deducción de conocimiento mediante técnicas de razonamiento. Es por ello que suele ser realizada por agentes cognitivos que son capaces de generar nuevas creencias a partir del conocimiento que disponen, descubrir evidencias contradictorias en el modelo de conocimiento, y habitualmente compartir esta información con los agentes más simples. Por último, la agregación y persistencia de contexto permite recopilar y reunir in-

formación sobre una entidad de contexto, como por ejemplo un dispositivo o una persona, con el fin de ser enviada y compartida con otros componentes, o almacenarla en alguno de los dispositivos de la infraestructura para habilitar búsquedas, recuperaciones y posteriores análisis del contexto por parte de cualquier componente.

6.1.4 Difusión del conocimiento: modelo estático y modelo dinámico

Dado que los diferentes componentes distribuidos de la arquitectura DAEDALUS son capaces de procesar en mayor o menor medida el conocimiento de contexto, queda por concretar qué mecanismos tienen disponibles estos componentes para la transmisión y difusión del conocimiento. El modelo de conocimiento estructura el contexto en dos vistas, una estática y otra dinámica. Dividimos por tanto el *modelo de conocimiento* en dos:

- **DOMINIO DE CONTEXTO.** Que trata el modelo estático del contexto.
- **TAXONOMÍA DE SUCESOS DE CONTEXTO.** Que trata el modelo dinámico del contexto.

El *dominio de contexto* establece una formalización de todos los elementos de contexto procesables por la organización, indicando atributos y relaciones entre elementos. Por otro lado, la *taxonomía de sucesos de contexto* es la responsable de sistematizar los distintos sucesos que ocurren en el *espacio de aprendizaje ubicuo*, y que sirven para difundir por toda la red del sistema las actualizaciones incrementales de los elementos del dominio de contexto.



Figura 6.1 Ontología del modelo de conocimiento modelada con Protégé

Todos los elementos presentes en el *modelo de conocimiento* se describen en una ontología OWL 2 modelada con Protégé (Figura 6.1) y cuya representación en RDF/XML, la sintaxis obligatoria prescrita por la recomendación W3C (W3C, 2009), está incluida en el Apéndice B.

La visión estática del contexto se representa mediante *entidades de contexto* (`ContextEntity`), que se definen dentro del *dominio de contexto* en la sección 6.2. Este modelo estático se transmite entre agentes por medio de mensajes que contienen las denominadas *declaraciones de contexto* (`ContextStatement`), que se detallan en la sección 6.2.1, mientras que la difusión del carácter dinámico se logra mediante los *sucesos de contexto* (`ContextEvent`), que se especifican dentro de la *taxonomía de sucesos de contexto* en la sección 6.3. Así pues, las *declaraciones de contexto* empaquetan información sobre las *entidades de contexto* en un preciso instante, mientras que los *sucesos de contexto* indican cambios que afectan al modelo estático.

6.2 Dominio de Contexto

Dentro del *dominio de contexto* se hacen explícitos todos los elementos que se desean modelar del *espacio de aprendizaje ubicuo* y que van a ser procesables computacionalmente por los distintos componentes software de la organización. El *dominio de contexto* en consecuencia define las posibilidades de adaptación en nuestro sistema ya que determinará las entidades y relaciones de contexto disponibles que manejarán los agentes para ofrecer servicios pedagógicos inteligentes. A modo de ejemplo y siguiendo el escenario de ENLACE descrito en el apartado 3.1.1, con el fin de configurar los dispositivos para las actividades es necesario que el modelo pueda especificar entre otros las dependencias entre procesos pedagógicos y herramientas, las características técnicas de los dispositivos de los usuarios y su configuración actual, los tipos de objetos de aprendizaje tratables por las herramientas, etc. Para poder seleccionar contenidos y objetos de aprendizaje será necesario conocer las relaciones entre las unidades sociales, tanto personas como grupos, con las actividades y los dispositivos que disponen. Con el objetivo de dar soporte adaptado social y colaborativo también tienen que estar presentes relaciones entre personas, pertenencia de usuarios a grupos, localización de todos estos, disponibilidad para la comunicación, etc., etc.

No obstante, algo esencial resulta en primera instancia determinar cómo se compartirá el conocimiento del *dominio de contexto* entre los agentes de la organización y qué estructuras estarán disponibles para tal fin. Es por ello que se introducirá el concepto de *declaración de contexto*.

6.2.1 Declaraciones de contexto

Las *declaraciones de contexto* (`ContextStatement`, en Figura 6.1) representan descripciones parciales del conocimiento que los agentes tienen acerca del contexto en un determinado momento. Una *declaración de contexto* aporta una suposición acerca del *espacio de aprendizaje ubicuo* codificada de forma atómica y que sirve de mecanismo de empaquetamiento para la difusión del conocimiento entre distintos componentes de la arquitectura DAEDALUS. De este modo, una agregación de *declaraciones de contexto* puede ser transferida entre agentes, por medio de la comunicación vía mensajes que se comentó en el apartado 5.4.1, para compartir el conocimiento que tienen éstos sobre las *entidades de contexto*.

Una *declaración de contexto* tiene básicamente una estructura similar a una declaración RDF (`rdf:Statement`) (W3C, 2004a), ya que incluye un *sujeto*, un *predicado* y un *objeto*. Tanto el *sujeto* como el *objeto* representan una entidad del contexto, mientras que el *predicado* es una relación que les asocia. No obstante, en determinadas ocasiones a la hora de difundir información de contexto a lo largo de una aplicación distribuida, hay que tomar en consideración y modelar detalles como la intimidad de la información que se maneja (H. Chen, Finin, & Joshi, 2004b), que determinará en qué modo se puede distribuir, junto con la fidelidad que se le otorga a dicha información (Karen Henricksen & Jadwiga Indulska, 2006). A este respecto, DAEDALUS permite la extensión de una *declaración de contexto* básica con restricciones acerca de los términos de privacidad, la proveniencia, su validez temporal y la confianza asociada a tal declaración. Para ello se sigue un mecanismo similar al utilizado en (Heckmann, 2005, pág. 57) para expresar declaraciones de situación en un modelo de usuario ubicuo, efectuando una reificación de la tripleta RDF para añadirle información adicional (Figura 6.2).

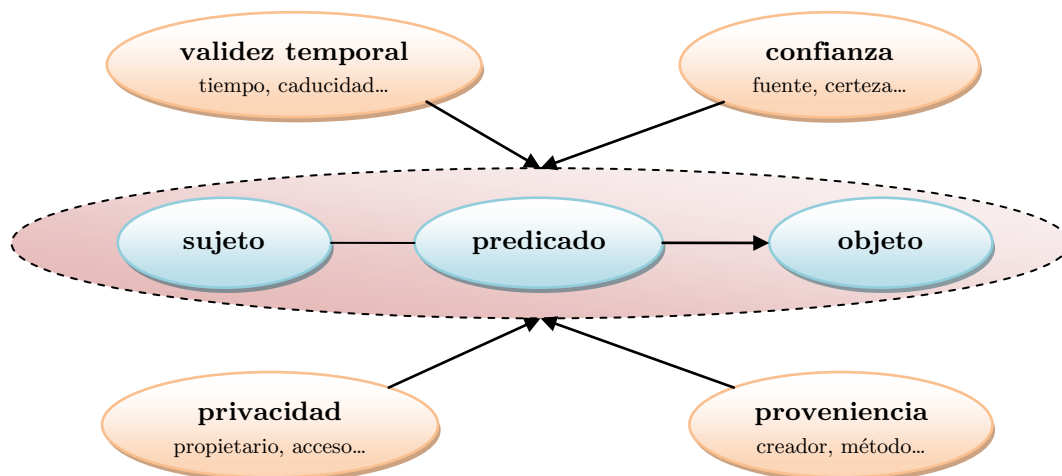


Figura 6.2 Reificación de la declaración de contexto

A efectos prácticos estos metadatos se modelan como propiedades asociadas a la clase de la ontología `ContextStatement`, y pueden ser procesados por *componentes manipuladores de contexto*, comentados en el apartado 6.1.3, para poder descubrir conflictos o informaciones contradictorias dentro del *modelo de conocimiento* y limitar el uso a componentes autorizados. Por otro lado, `ContextStatement` también incluye obviamente propiedades para la triplete sujeto-predicado-objeto (`subject`, `predicate` y `object`, respectivamente), que hacen referencia a las entidades y relaciones del contexto, las cuales se pasan a detallar a continuación.

6.2.2 Entidades y relaciones de contexto

Las *entidades de contexto* identifican todos los elementos que el *modelo de conocimiento* es capaz de representar y por extensión los conceptos que los agentes pueden manipular. Una *entidad de contexto* puede ser una Persona, una Actividad o incluso un determinado Lapso de Tiempo o Localización. Las *entidades de contexto* están caracterizadas por una serie de propiedades que las definen, que llamaremos *atributos*, y presentan asociaciones entre sí mediante una serie de propiedades, que llamaremos *relaciones*.

Dentro del lenguaje de representación que utilizaremos para describir el dominio de contexto, asociaremos las *entidades de contexto* con clases (`owl:Class`) en la ontología descendientes de la clase `ContextEntity` (Figura 6.1) y sus atributos y relaciones con propiedades (`owl:DataProperty` y `owl:ObjectProperty`, respectivamente) en la ontología. Todas las *entidades*

de contexto se estructuran jerárquicamente mediante relaciones de subsunción (`rdfs:subClassOf`). Para facilitar la explicación en adelante acompañaremos diagramas que incluirán todos estos elementos siguiendo la nomenclatura gráfica que se muestra en la Figura 6.3.

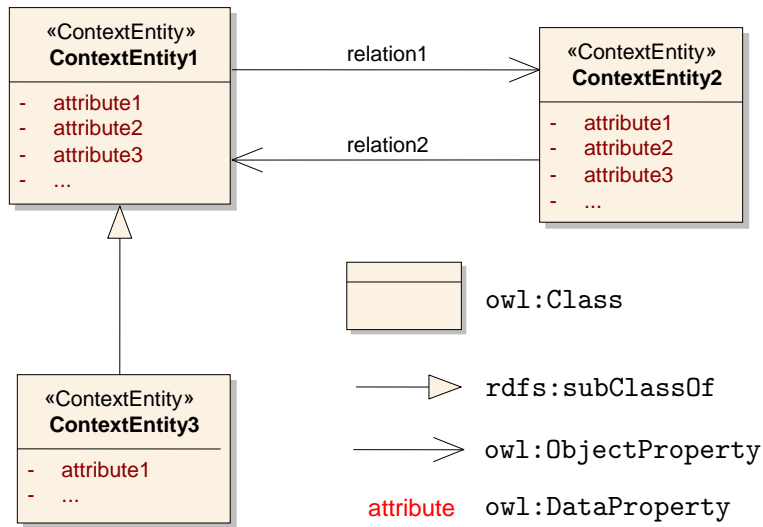


Figura 6.3 Nomenclatura del lenguaje gráfico de representación

6.2.3 Dimensiones del contexto

De forma similar a otras propuestas de estructuración del contexto (Wang, 2004), DAEDALUS divide el *dominio de contexto* en distintas dimensiones, como se muestra en la Figura 6.4.



Figura 6.4 Partición del dominio de contexto en DAEDALUS

El *dominio de contexto* de DAEDALUS se vertebra a lo largo de seis sub-contextos o dimensiones interrelacionadas entre sí. Las dimensiones del dominio de contexto integran todos los elementos del *espacio de aprendizaje ubicuo* identificados en el apartado 4.1.2 y se estructuran asociándose a éstos tal y como se muestra en la Tabla 6.2.

DOMINIO DE CONTEXTO	ELEMENTOS DEL ESPACIO DE APRENDIZAJE UBICUO
Dimensión de Artefacto Educativo	Herramientas/Aplicaciones Software, Productos/Objetos de Aprendizaje
Dimensión de Individuo	Persona, Agente
Dimensión de Comunidad	Comunidad, Organización, Rol, Comportamiento, Colaboración
Dimensión de Proceso Pedagógico	Procesos Pedagógicos (Proyectos, Actividades, Tareas)
Dimensión de Espacio-Tiempo	Entorno Ubicuo (Tiempo y Localización)
Dimensión de Infraestructura	Infraestructura, Dispositivos, Nodos, Mensajes, Eventos

Tabla 6.2 Elementos del espacio de aprendizaje ubicuo asociados a las dimensiones del dominio de contexto

A continuación se analizará cada una de estas dimensiones más detenidamente.

DIMENSIÓN DE ARTEFACTO EDUCATIVO

El concepto central de la Dimensión de Artefacto Educativo (Figura 6.5) es el concepto de *Artefacto de Aprendizaje* (LearningArtefact). Un *Artefacto de Aprendizaje* engloba tanto el concepto de *Objeto de Aprendizaje* (LearningObject) como el concepto de *Herramienta* (LearningTool). En este dominio se definen pues todas las relaciones que posee un *Objeto de Aprendizaje* con respecto a las demás entidades de contexto y se le asocia con un *Tipo* (LearningObjectType). El concepto de *Tipo de Objeto de Aprendizaje* centraliza toda la información acerca de una clase de *Objeto de Aprendizaje*, incluyendo su relación con las *Herramientas*. Mediante una reificación de las relaciones de metadatos de los *Objetos de Aprendizaje* es posible vertebrar el conocimiento para poder añadir a la ontología distintas relaciones con las demás entidades de contexto (Mayorga et al., 2007).

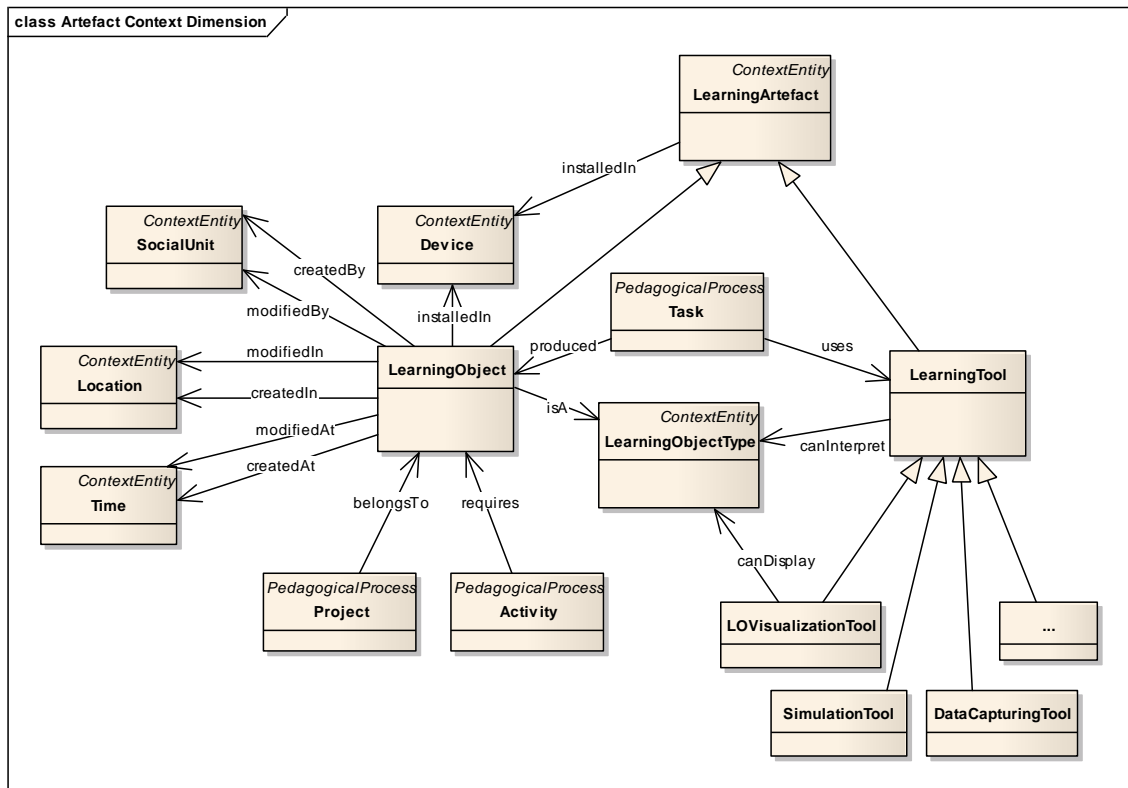


Figura 6.5 Dimensión de Artefacto Educativo

Un *Dispositivo* puede tener instalado (*installedIn*) una serie de *Artefactos de Aprendizaje*, incluidas *Herramientas*. El concepto *Herramienta* se relaciona con el concepto *Tarea* (*Task*) mediante una relación de *Uso* (*uses*). Las *Herramientas* pueden refinarse en toda una clasificación que sirva para estructurar estas herramientas en función del uso que pueden hacer de ellas los usuarios para las *Tareas*, como capturar datos (*DataCapturingTool*), hacer simulaciones (*SimulationTool*), visualizar objetos de aprendizaje (*LOVisualizationTool*), etc., etc.

DIMENSIÓN DE INDIVIDUO

La Dimensión de Individuo trata las características individuales relativas a la gente implicada en el *espacio de aprendizaje ubicuo*. Este tratamiento se hace por un lado a un nivel general, caracterizando las propiedades de las personas como usuarios del sistema, y por otro lado a un nivel educativo, con-

templando a estas personas de forma más específica como estudiantes con sus objetivos pedagógicos.

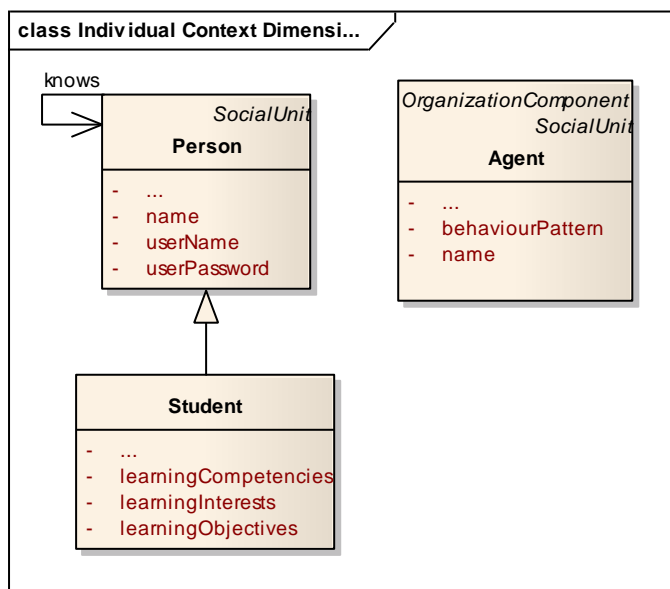


Figura 6.6 Dimensión de Individuo

De esta forma, como se muestra en la Figura 6.6, la entidad *Persona* (*Person*) está destinada a contener información de carácter general de un usuario, como su identificación (nombre, dirección...), credenciales (claves, contraseñas...), preferencias (idioma, uso de letra grande...), etc. Es equivalente a la clase `foaf:Person` de FOAF (Brickley & Miller, 2007) con lo que aprovecha la relación de conocimiento (`foaf:know`) entre personas. En definitiva, es el punto de extensión que se articula para definir todo un modelo de usuario con las características necesarias dentro del dominio de contexto. La entidad *Estudiante* (*Student*) que descende de *Persona*, de forma semejante es el punto en la ontología para extenderla e incorporar un modelo de estudiante que incluya por ejemplo preferencias de aprendizaje, incluidas preferencias cognitivas (por ejemplo estilo de aprendizaje), conocimientos adquiridos por el estudiante, etc., recomendándose emplear y basarse en conceptos definidos en perfiles de estudiante estándares como por ejemplo IMS LIP (IMS GLC, 2005). Nuevamente, todas las extensiones que se incluyan responderán en gran medida a las necesidades de modelado del usuario y estudiante que se requieran para las funcionalidades de los componentes de la organización. Dentro de esta dimensión también se incluye el concepto *Agente*

(Agent), entendido este último como una entidad computacional de la organización (es decir, un agente software).

DIMENSIÓN DE COMUNIDAD

Una entidad *Persona* (Person) está situada con un contexto social, el cual la Dimensión de Comunidad toma en consideración. Dentro de esta dimensión (Figura 6.7) se articula el concepto de *Unidad Social* (SocialUnit). Una *Unidad Social* puede ser o bien una *Persona*, o bien un *Grupo* (Group), o bien un *Agente* (Agent).

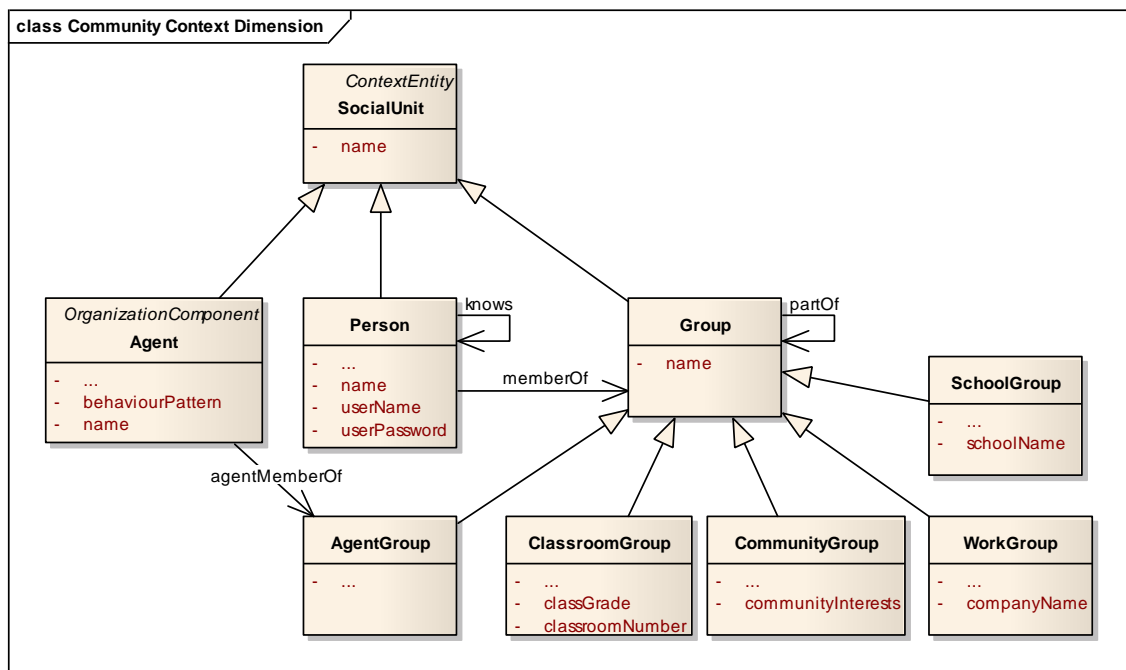


Figura 6.7 Dimensión de Comunidad

El concepto *Grupo* permite estructurar toda esta comunidad virtual de aprendizaje mediante la definición de distintos tipos de grupos relativos a una *Comunidad* (CommunityGroup), una *Escuela* (SchoolGroup), una *Clase* (ClassGroup), un *Grupo del Trabajo* (WorkGroup), etc. La relación *Parte De* (partOf) permite incluir unos grupos dentro de otros, y la relación *Miembro De* (memberOf) regula la inclusión de una *Persona* en un *Grupo*.

La razón de incluir el concepto *Agente* como *Unidad Social* es permitir incorporarlo dentro de la descripción de la comunidad virtual de aprendizaje.

También se establece un *Grupo de Agentes* (AgentGroup) cuyos miembros son únicamente agentes.

DIMENSIÓN DE PROCESO PEDAGÓGICO

La Dimensión de Proceso Pedagógico trata en consideración todos los procesos dinámicos que suceden en un *espacio de aprendizaje ubicuo*. El concepto de *Proceso Pedagógico* (PedagogicalProcess) es el principal de esta dimensión (Figura 6.8), concretándose en los conceptos *Proyecto* (Project), *Actividad* (Activity) y *Tarea* (Task). Un *Proceso Pedagógico* puede estar contextualizado en una *Localización* y en un *Tiempo* (mediante las relaciones *locatedIn* y *happensAt*). Las *Actividades* se engloban en *Proyectos* y se dividen a su vez en *Tareas*, estableciéndose toda esta composición mediante la relación *Parte De* (*partOf*), que permite incluso anidar una *Actividad* dentro de otra. Una *Actividad* puede diferenciarse entre *Actividades Programadas* (ScheduledActivity) frente a *Actividades Espontaneas* (SpontaneousActivity), y *Actividades de Aprendizaje* (LearningActivity) frente a *Actividades de Apoyo* a este aprendizaje (SupportActivity).

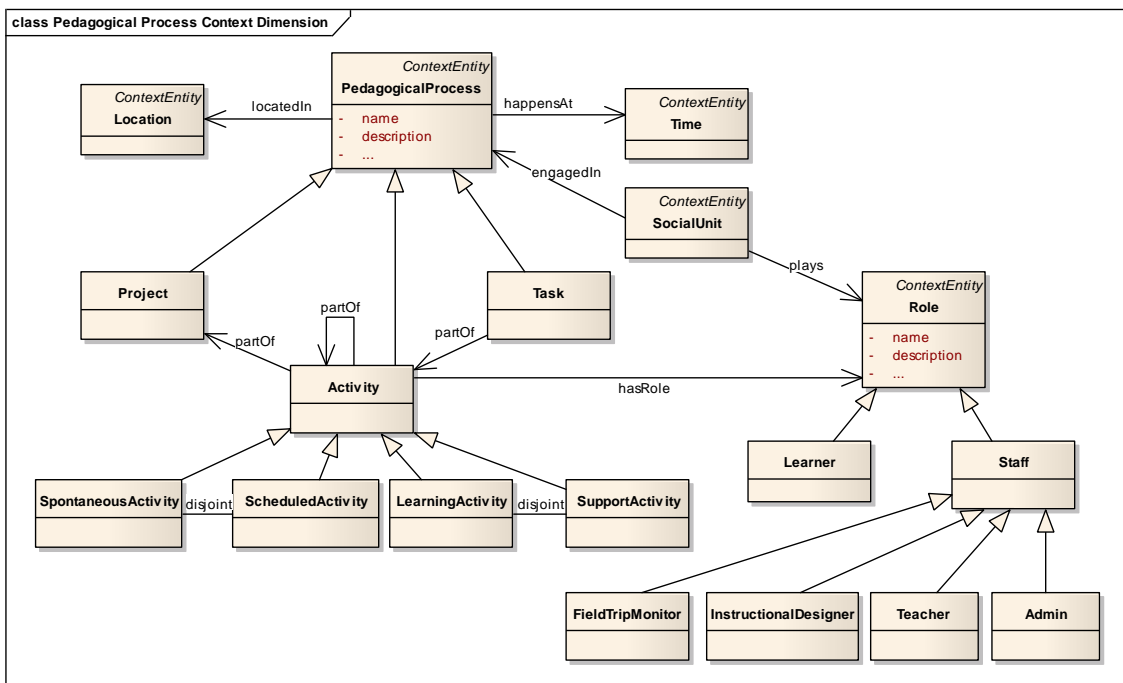


Figura 6.8 Dimensión de Proceso Pedagógico

En esta dimensión también se define el concepto de Rol (Role), que se refina en el rol de aprendiz (*Learner*) y personal docente (*Staff*) de soporte a este aprendizaje. Una *Unidad Social* puede estar involucrada en un *Proceso Pedagógico* (*engagedIn*) y jugar un *Rol* dentro de éste (*plays*). Tanto el concepto *Proceso Pedagógico* como el concepto *Rol* pueden servir de punto de extensión para modelar diseños pedagógicos más complejos, como por ejemplo basándose el trabajo desarrollado en (Vélez, 2009) o en el estándar *IMS Learning Design* (IMS GLC, 2003), ya que tanto el primero como modelos ontológicos que parten del segundo (Amorim, Lama, Sánchez, Riera, & Vila, 2006) han sido una inspiración para obtener los conceptos básicos presentes en esta dimensión.

DIMENSIÓN DE ESPACIO-TIEMPO

Dos propiedades importantes y consideradas desde los inicios de la conceptualización de los sistemas basados en contexto son la noción de *Localización* (*Location*) y la noción de *Tiempo* (*Time*).

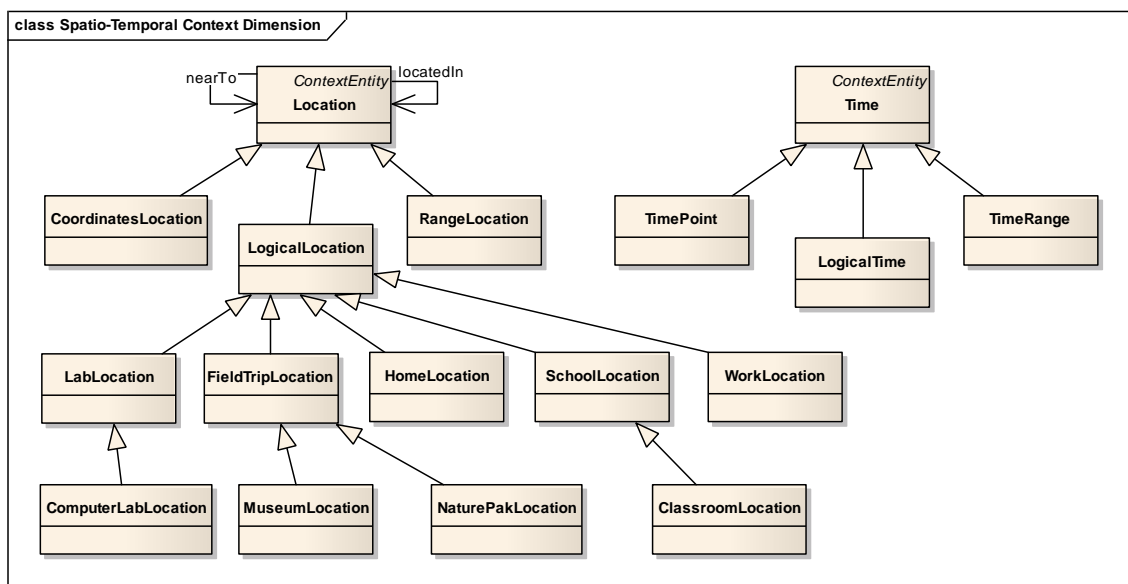


Figura 6.9 Dimensión de Espacio-Tiempo

Basándose en el análisis de distintas ontologías de contexto, (Ye et al., 2007) sugieren una serie de características que una ontología general debería aportar y que se incluyen en la Dimensión de Espacio-Tiempo (Figura 6.9):

- La posibilidad de representar distintos tipos de localizaciones y tiempos incluyendo:
 - Localizaciones basadas en coordenadas (`CoordinatesLocation`) y momentos de tiempo (`TimePoint`).
 - Rangos de regiones (`RangeLocation`) y de tiempos (`TimeRange`).
 - Localizaciones simbólicas (`LogicalLocation`), como escuela, excursión, laboratorio, etc., y tiempos simbólicos (`LogicalTime`).
 - La posibilidad de explotar relaciones espaciales. Lo que incluye
 - Relaciones de jerarquía (`locatedIn`).
- Relaciones de adyacencia (`nearTo`).

DIMENSIÓN DE INFRAESTRUCTURA

La Dimensión de Infraestructura contempla los conceptos necesarios para representar los elementos tecnológicos del *espacio de aprendizaje ubicuo*, incluyendo la organización y sus componentes (Figura 6.10). Dos son los conceptos principales: *Dispositivo* (`Device`) y *Nodo* (`OrganizationNode`). Un *Nodo* forma una estructura de red mediante la relación *Conectado A* (`connectedTo`) y contiene desplegados (`deployedIn`) *Componentes de Organización* (`OrganizationComponent`), incluyendo *Agentes* (`Agent`) y *Recursos de Organización* (`OrganizationResource`). El concepto de *Dispositivo* se descompone en *Móvil* (`MobileDevice`) y *Estático* (`StaticDevice`), para refinarse en los distintos tipos de dispositivos que se contemplan en el entorno ubicuo.

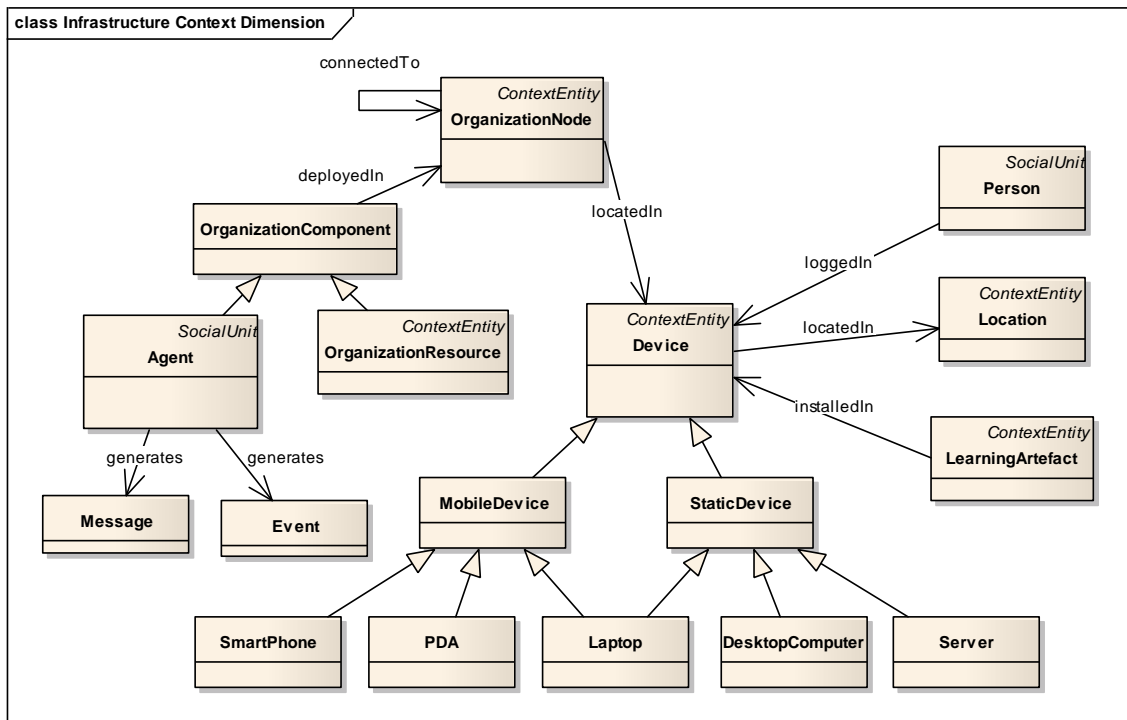


Figura 6.10 Dimensión de Infraestructura

Un *Dispositivo* puede presentar asociaciones con distintos elementos del contexto como por ejemplo con *Persona*, mediante la relación *Registrado En* (*loggedIn*), o con *Localización*, empleando la relación *Localizado En* (*locatedIn*). También un *Dispositivo* puede tener instalados (relación *installedIn*) distintos *Artefactos de Aprendizaje* (*LearningArtefact*), que son el elemento principal de la siguiente dimensión.

6.3 Taxonomía de sucesos de contexto

Los *sucesos de contexto* (*ContextEvent*, Figura 6.1) reflejan situaciones que ocurren en un preciso momento y que pueden dar pie a una reacción de los agentes o un cambio en el *modelo de contexto*. Las variaciones en el contexto son pues notificadas y distribuidas en la arquitectura DAEDALUS como sucesos a lo largo de la organización. Los *sucesos de contexto* reflejan situaciones dinámicas que tienen lugar en un determinado instante en el tiempo. Estas situaciones pueden dar pie a una reacción de los agentes e implicar un cambio en el conocimiento sobre el contexto por parte de éstos. La *taxonomía de sucesos de contexto* explicita y estructura todos estos posibles cambios pa-

ra que los componentes de la organización puedan tanto generarlos y compararlos como entenderlos y reaccionar ante ellos.

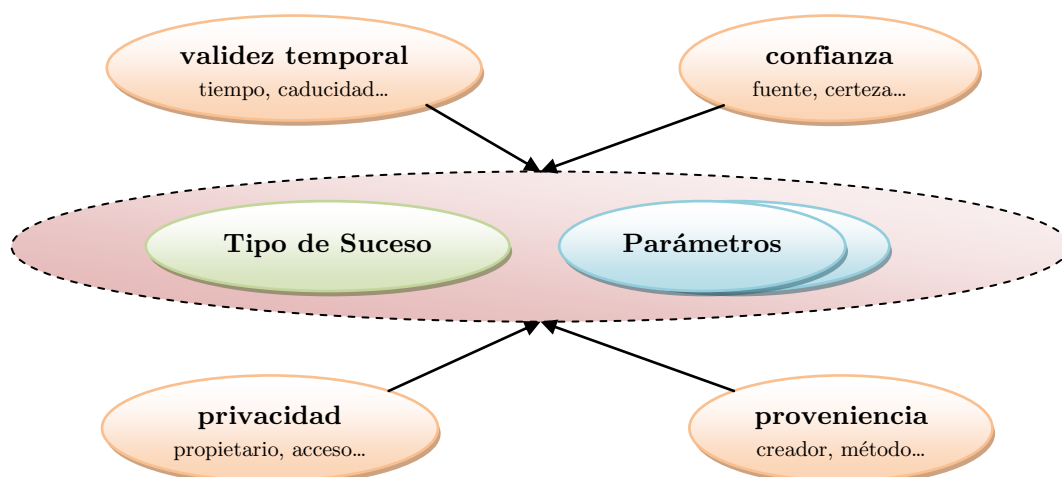


Figura 6.11 Reificación del suceso de contexto

Como se muestra en la Figura 6.11, un suceso de contexto se reifica de forma similar a una declaración de contexto (Figura 6.2). Mediante la definición de una *taxonomía de sucesos de contexto* compartida, los agentes pueden intercambiar información tipificada que puede ser de utilidad para cualquier componente de la arquitectura. Los *sucesos de contexto* se caracterizan por poseer un tipo y unos parámetros que representan *entidades de contexto*. De este modo sucesos como “el usuario U se ha identificado en el dispositivo D”, “un nuevo dispositivo D se ha descubierto”, “el usuario U inicia actividad A”, pueden servir para hacer reaccionar a los agentes y actualizar su representación interna del contexto. De forma implícita, cada suceso lleva asociado una marca temporal que sitúa en un determinado momento su ocurrencia

6.3.1 Clasificación de sucesos de contexto

Se pretende crear una taxonomía jerárquica de todos los sucesos que pueden aparecer en el desarrollo de los escenarios pedagógicos, pero siempre dejando entreabierto la posibilidad de extender el conjunto de sucesos en función de las necesidades para atender los *servicios pedagógicos inteligentes* que se deseen desarrollar. Establecemos una división de sucesos en función de la dimensión de contexto a la que estén asociados (Figura 6.12):

- Sucesos de Artefacto Educativo

- Sucesos de Individuo
- Sucesos de Comunidad
- Sucesos de Proceso Pedagógico
- Sucesos de Espacio-Tiempo
- Sucesos de Infraestructura

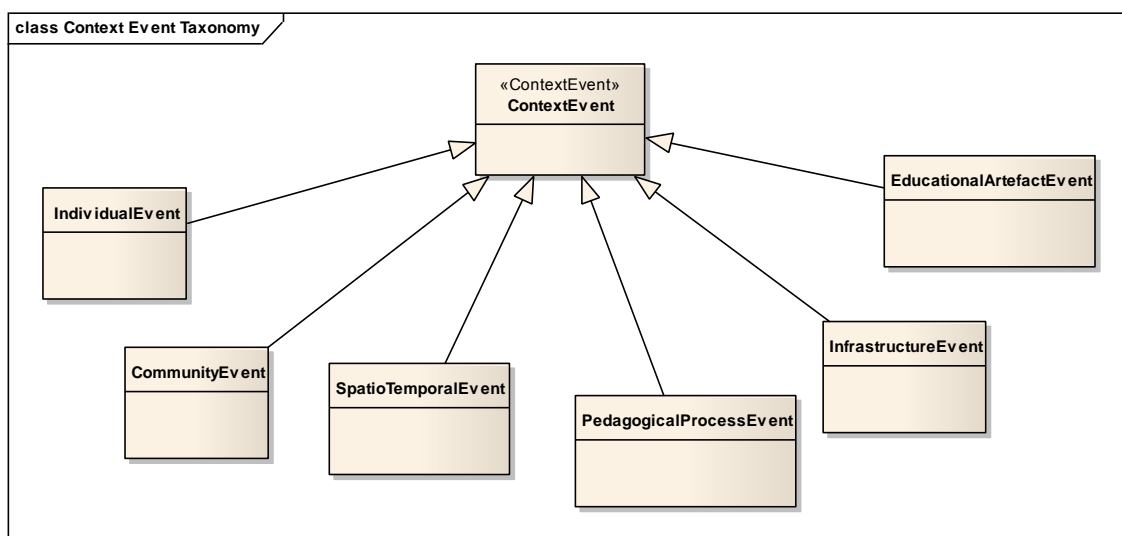


Figura 6.12 Clasificación de Sucesos de Contexto

A continuación mostraremos los *sucesos de contexto* clasificados, especificando su nombre, parámetros y una breve explicación de cada uno. Los *sucesos de contexto* son habitualmente transmitidos dentro de la arquitectura DAEDALUS como *eventos*, los cuales fueron descritos a Nivel de Infraestructura en el apartado 5.4.2. De este modo los agentes pueden suscribirse en base a los nombres de los eventos, para recibir únicamente los cambios en el contexto en los que estén interesados.

SUCESOS DE ARTEFACTO EDUCATIVO

Los Sucesos de Artefacto Educativo contemplan los cambios que repercuten sobre Objetos de Aprendizaje y Herramientas.

En la Tabla 6.3 se pueden ver distintos sucesos pertenecientes a esta categoría.

Tipo del Suceso	Explicación	Parámetros
daedalus/learningObject/new	Un nuevo objeto de aprendizaje es añadido al sistema.	LO
daedalus/ learningObject/modified	Un objeto de aprendizaje conocido ha sido modificado	LO
daedalus/ learningObject/deleted	Un objeto de aprendizaje conocido ha sido eliminado.	LO
daedalus/person/launchTool	Un usuario ha arrancado una herramienta.	Person Tool
daedalus/person/quitTool	Un usuario ha cerrado una herramienta.	Person Tool

Tabla 6.3 Sucesos de Artefacto Educativo

SUCESOS DE INDIVIDUO

Los Sucesos de Individuo tratan los cambios en el contexto relativos a una Persona o Estudiante. Básicamente sirven para comunicar los cambios en el estado de las personas que hay en el sistema o en la información pedagógica conocida acerca de los estudiantes.

En la Tabla 6.4 se listan una serie de sucesos pertenecientes a esta categoría.

Tipo del Suceso	Explicación	Parámetros
daedalus/person/loggedIn	Una persona se conecta al sistema a través de un dispositivo	Person Device
daedalus/person/loggedOut	Una persona se desconecta del sistema a través de un dispositivo	Person Device
daedalus/person/updated	La información perteneciente al perfil de un usuario se ha actualizado o modificado.	Person
daedalus/student/learningGoalReached	Un estudiante ha alcanzado un objetivo pedagógico.	Student
daedalus/student/competenciesGained	Un estudiante ha logrado nuevas competencias pedagógicas.	Student

Tabla 6.4 Sucesos de Individuo

SUCESOS DE COMUNIDAD

Los Sucesos de Comunidad permiten transmitir cambios sobre la formación de grupos e información relativa a la interacción dentro de la comuni-

dad. Esto incluye de forma importante sucesos para el tratamiento de la colaboración y los conflictos cognitivos que pueden ocurrir entre miembros de la comunidad.

Tipo del Suceso	Explicación	Parámetros
daedalus/group/join	Una unidad social (usuario, agente o miembros de todo un grupo) se une a un grupo.	Group SocialUnit
daedalus/group/quit	Una unidad social (usuario, agente o miembros de todo un grupo) abandona un grupo.	Group SocialUnit
daedalus/group/updated	La información perteneciente a un grupo se ha actualizado o modificado.	Group
daedalus/person/agreesWith	Una persona está de acuerdo con otra.	Person1 Person2
daedalus/person/disagreesWith	Una persona está en desacuerdo con otra.	Person1 Person2
daedalus/person/asksTo	Una persona ha realizado una petición a otra.	Person1 Person2
daedalus/collaboration/propose	Se propone una colaboración en la realización de un proceso pedagógico entre dos unidades sociales.	PedagogicalProcess Social Unit1 Social Unit2
daedalus/collaboration/accept	Una unidad social está de acuerdo con una proposición de colaboración en un proceso pedagógico.	PedagogicalProcess Social Unit1 Social Unit2
daedalus/collaboration/refuse	Una unidad social rechaza una proposición de colaboración en un proceso pedagógico.	PedagogicalProcess Social Unit1 Social Unit2

Tabla 6.5 Sucesos de Comunidad

La Tabla 6.5 muestra una serie de sucesos pertenecientes a esta categoría. No obstante, los sucesos de comunidad que aparecen en este listado desde el punto de vista de la colaboración se encuentran muy ligados a los sucesos de la categoría que se describe a continuación.

SUCESOS DE PROCESO PEDAGÓGICO

Los Sucesos de Proceso Pedagógico (Tabla 6.6) son relativos a los cambios que se producen en relación a las actividades que realizan las personas. De forma especial se contemplan dentro de esta categoría los sucesos que tienen relación con la colaboración para iniciar actividades conjuntas o tareas que involucren a distintos usuarios.

Tipo del Suceso	Explicación	Parámetros
daedalus/activity/created	Se crea una nueva actividad emergente.	Activity
daedalus/person/changeActivity	Un usuario ha decidido dedicarse a la realización de una actividad.	Person Activity
daedalus/person/changeTask	Un usuario ha decidido dedicarse a la realización de una tarea.	Person Task
daedalus/activity/start	Se da comienzo a una actividad.	Activity
daedalus/activity/pause	Una actividad se congela y no se permite trabajar a los usuarios en ella momentáneamente	Activity
daedalus/activity/continue	Se reanuda una actividad.	Activity
daedalus/activity/stop	Se finaliza una actividad y no se permite trabajar a los usuarios en ella	Activity
daedalus/project/start	Se inicia un proyecto, permitiéndose trabajar en él.	Project
daedalus/project/stop	Un proyecto es cerrado y no se permite trabajar en él.	Project

Tabla 6.6 Sucesos de Proceso Pedagógico

SUCESOS DE ESPACIO-TIEMPO

Los Sucesos de Espacio-Tiempo tratan las variaciones que suceden sobre ciertas entidades en función del lugar y momento. Esto incluye los cambios en su localización absoluta como en su localización relativa a otras entidades, o su contextualización temporal.

Cabe recordar que de forma implícita, cada suceso lleva asociado por definición una marca temporal que sitúa en un determinado momento su ocurrencia, por lo que lo que se muestran aquí son sucesos relacionados en mayor medida con las propiedades espaciales.

La Tabla 6.7 contempla una lista de sucesos pertenecientes a esta categoría.

Tipo del Suceso	Explicación	Parámetros
daedalus/person/hasApproached	Un usuario está cerca de otra entidad	Person Entity Location
daedalus/person/changesLocation	Un usuario cambia de localización.	Person Location
daedalus/device/hasApproached	Un dispositivo está cerca de otra entidad	Device Entity Location
daedalus/device/changesLocation	Un dispositivo cambia de localización.	Device Location

Tabla 6.7 Sucesos de Espacio-Tiempo

SUCESOS DE INFRAESTRUCTURA

Los Sucesos de Infraestructura (Tabla 6.8) se utilizan para comunicar cambios en los nodos y dispositivos, incluyendo de forma general la configuración tecnológica dentro de la arquitectura.

Tipo del Suceso	Explicación	Parámetros
daedalus/node/connected	Un nodo ha sido conectado a la infraestructura.	Node
daedalus/node/disconnected	Un nodo ha sido desconectado de la infraestructura.	Node
daedalus/device/found	Un dispositivo ha sido encontrado en un nodo.	Device Node
daedalus/device/lost	Un dispositivo ha sido perdido de un nodo.	Device Node
daedalus/device/resourceInstalled	Un recurso de aprendizaje ha sido instalado en un dispositivo	Device Learning Resource

Tabla 6.8 Sucesos de Infraestructura

7

NIVEL DE ORGANIZACIÓN EDUCATIVA

Dentro de este capítulo se analizarán los elementos que se encuentran situados dentro del Nivel de Organización Educativa. Como se ha comentado anteriormente, la *organización educativa* en DAEDALUS es la encargada de definir y provisionar los servicios pedagógicos que se articulan dentro del *espacio de aprendizaje ubicuo*. En primer lugar analizaremos y estructuraremos los distintos servicios que pueden darse en escenarios con tecnologías móviles. A continuación discutiremos acerca de los distintos tipos de agentes pedagógicos necesarios para dar estos servicios, siempre con una mirada puesta en la generalización y la reusabilidad. Finalmente, se tratará el diseño de Organizaciones Educativas mediante

agentes pedagógicos, discutiéndose cómo crear puentes de unión entre este nivel con el Nivel de Infraestructura.

7.1 Servicios Pedagógicos Inteligentes y agentes de la Organización Educativa

La *organización educativa* de un *espacio de aprendizaje ubicuo*, tal y como se introdujo en el apartado 4.3.3, engloba todos los componentes software, tanto agentes como recursos, directamente relacionados con las funcionalidades necesarias para ofrecer apoyo inteligente a los distintos usuarios presentes en dicho espacio, con el fin de facilitarles un soporte eficaz a las actividades de aprendizaje. Todas estas funcionalidades se agrupan en torno a lo que es denominado como conjuntos de *servicios pedagógicos inteligentes* (SPIs). Los SPIs reciben este nombre por ser servicios para el desarrollo de procesos pedagógicos caracterizados principalmente tanto por un profuso empleo de la información contextual educativa disponible en el sistema, como por el uso de técnicas de razonamiento sobre dicha información utilizando componentes software inteligentes. La arquitectura DAEDALUS aporta mecanismos para enfocar la definición, estructuración y diseño de estos servicios pedagógicos en términos concretos a un alto nivel de abstracción, encapsulándolos dentro de componentes inteligentes de la *organización educativa*. Es decir, los SPIs se definen y despliegan en el escenario de aprendizaje dentro del Nivel de Organización Educativa de DAEDALUS mediante lo que se establece como *agentes pedagógicos*. De modo convenido, se empleará el término *agente pedagógico* para hacer referencia a aquellos agentes de la arquitectura que en cierta medida se encuentran relacionados con las tareas necesarias para facilitar SPIs dentro del *espacio de aprendizaje ubicuo*.

Los *agentes pedagógicos* presentan comportamientos cuyos objetivos y tareas se circunscriben a la planificación y prestación eficaz de servicios adaptados y más adecuados a todo momento y situación, buscando en última instancia contribuir a un eficaz funcionamiento de la comunidad de estudiantes y profesores (y posiblemente de otros agentes) del *espacio de aprendizaje ubicuo*. Esto implica necesariamente la utilización de información del entorno actual para lograr este cometido e incluso una previa planificación para lograr la correcta prestación de servicios a posteriori. Así por ejemplo, para prestar servicios que promuevan la colaboración entre estudiantes en actividades de

aprendizaje móvil es necesario que en todo momento se mantengan actualizados mapas sociales con información acerca de potenciales usuarios colaboradores que incluya su localización y estados actuales, o por ejemplo para prestar servicios de recomendación de nuevas actividades en escenarios posiblemente aislados o sin capacidades de comunicación, como salidas al campo, puede ser necesaria una planificación y selección previa de artefactos educativos de etapas anteriores que puedan encajar con los futuros escenarios de los estudiantes.

7.1.1 Sistematización de SPIs en DAEDALUS

Una conveniente manera de sistematización de los posibles SPIs existentes dentro del Nivel de Organización Educativa de la arquitectura DAEDALUS se basa en los niveles inferiores. Dado que los *servicios pedagógicos inteligentes* (Nivel de Organización Educativa) se caracterizan en primer lugar por el empleo del *contexto* (Nivel de Semántica y Contexto) y en segundo lugar por su encapsulación dentro de *agentes* (Nivel de Infraestructura), una forma apropiada de estructuración de éstos ha de tener en cuenta esta dicotomía.

Es por ello que se establece una división de los SPIs en función del tipo primario de información contextual que es manipulada para su prestación y las capacidades predominantes mostradas por los agentes responsables de tal prestación. Siguiendo esta propuesta de clasificación, en la Tabla 7.1 se muestra una relación de distintos ejemplos de SPIs, clasificados por un lado en base a las dimensiones de contexto (Individuo, Comunidad, Espacio-Tiempo, Proceso Pedagógico, Infraestructura Educativa y Artefacto Educativo), que se presentaron en el apartado 6.2.3, y a las capacidades principales modeladas en los patrones de comportamiento de los agentes que se describieron en el apartado 5.3 (Percepción, Razonamiento y Actuación).

Esta clasificación de SPIs nos permite acercarnos también hacia una cierta sistematización de los diferentes *agentes pedagógicos* de DAEDALUS, según el tipo de servicios que buscan proveer.

		Capacidades del Agente (Nivel de Infraestructura)		
		Percepción	Razonamiento	Actuación
Dimensión del Contexto (Nivel de Semántica y Contexto)	Artefacto Educativo	Extracción de información en repositorios de objetos de aprendizaje, del estado actual de las herramientas software...	Inferencia de conocimiento empleando metadatos de objetos de aprendizaje, características de herramientas...	Distribución de recursos educativos, recomendación de objetos de aprendizaje y herramientas...
	Individuo	Extracción de información sobre usuarios y minado de modelos de estudiantes, profesores...	Gestión, inferencia y actualización de datos del modelo de usuario, estudiante...	Provisión y filtrado de información personalizada basada en perfiles de usuarios, modelos de estudiante...
	Comunidad	Minado de información social relativa a grupos, relaciones sociales entre estudiantes, roles...	Deducción de relaciones entre miembros de la comunidad, búsqueda de patrones de colaboración...	Organización y gestión de grupos de usuarios, establecimiento de posibles colaboraciones entre estudiantes...
	Proceso Pedagógico	Minado de información sobre proyectos, actividades, tanto futuras, actuales, previas, previstas, espontáneas...	Creación y estructuración de actividades según necesidades y objetivos pedagógicos...	Reparto de actividades, guía y regulación el desarrollo de las mismas...
	Espacio-Temporal	Detección de localización de los elementos del contexto y su ocurrencia en el tiempo...	Inferencia de relaciones espacio-temporales, asociación de elementos a distintos momentos o lugares...	Recomendación de elementos educativos en función de la localización y el momento actual...
	Infraestructura Educativa	Descubrimiento de recursos distribuidos y componentes de la infraestructura, como dispositivos, extracción de características de estos componentes, recursos...	Selección y adaptación de contenidos educativos a los elementos disponibles de la infraestructura...	Instalación de herramientas, configurar dispositivos, sincronización de información distribuida...

Tabla 7.1 Sistematización y ejemplos de SPIs en función de los niveles inferiores

7.1.2 Clasificación de agentes pedagógicos en DAEDALUS

Tomando una visión transversal y vertical de la Tabla 7.1 podemos agrupar toda la primera columna en servicios relativos a la obtención de información contextual, dando lugar a los Agentes de Minado y Extracción de Contexto. La segunda columna engloba servicios que derivan nueva información o conocimiento sobre el contexto actual utilizable por el resto de elemen-

tos del espacio de aprendizaje, estableciendo los llamados Agentes de Derivación de Contexto.

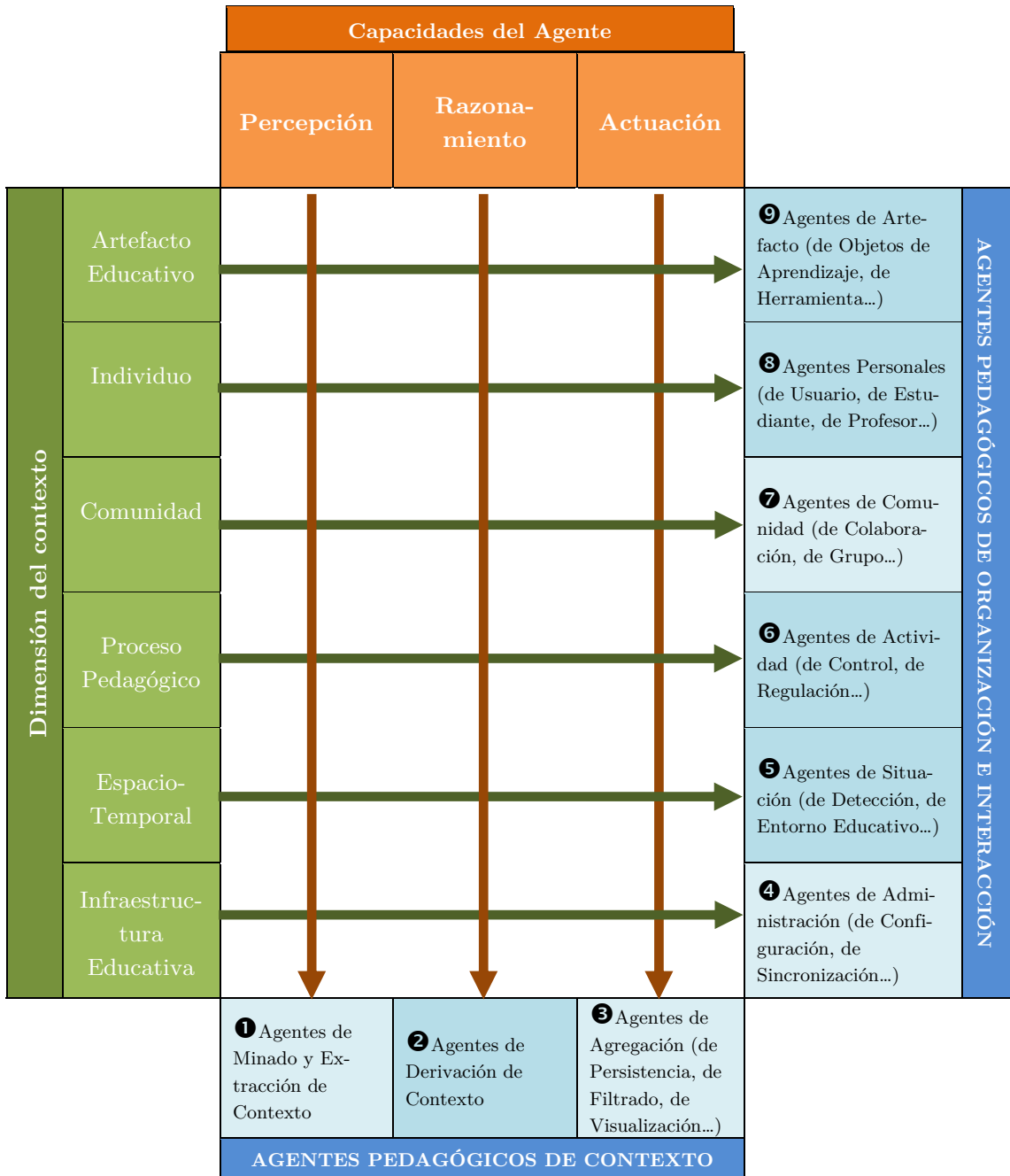


Figura 7.1 Clasificación de agentes pedagógicos

Y finalmente la tercera columna contiene servicios que agregan y hacen un uso en sí del conocimiento de contexto, ya sea para almacenarla, mostrarla

o utilizarla como recomendaciones. Esta triple división, por lo tanto, nos aporta una primera estructuración de *agentes pedagógicos* constituida en tres tipos de agentes que concuerda con los previamente denominados Agentes Pedagógicos de Contexto que fueron introducidos en el Nivel de Semántica y Contexto dentro del apartado 6.1.3.

Además, tomando una visión transversal pero esta vez horizontal de la Tabla 7.1, se establece otra perspectiva complementaria que da pie a una división de los *agentes pedagógicos* en consonancia con el Nivel de Organización Educativa y en la que se profundizará a continuación.

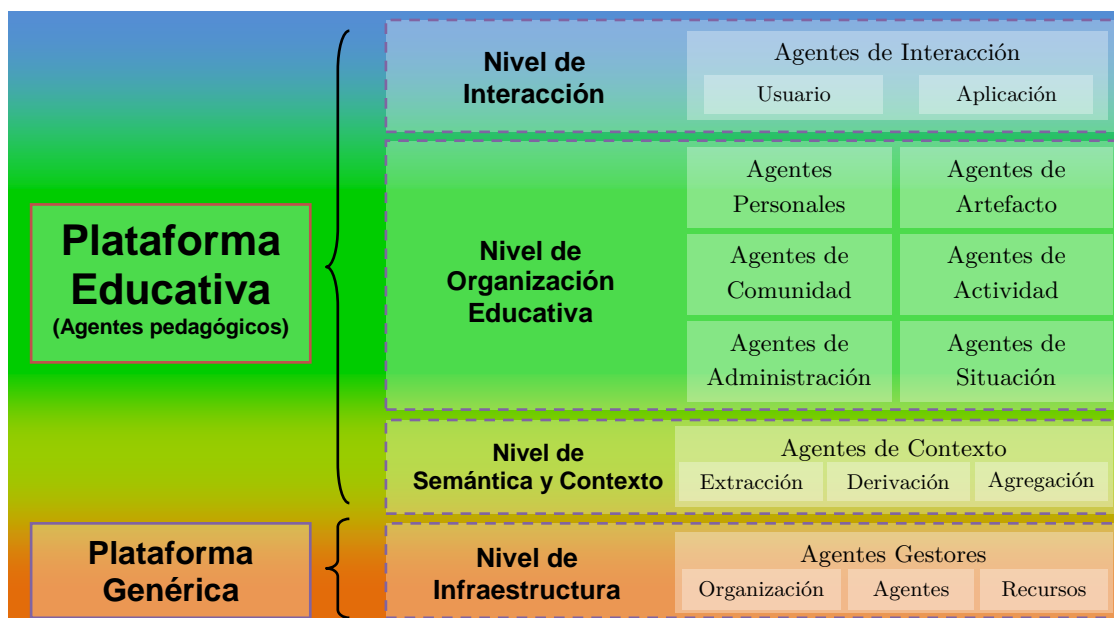


Figura 7.2 Tipos de agentes pedagógicos en DAEDALUS

En la Figura 7.2 se ofrece una visión estructurada según los niveles de la Arquitectura Genérica de DAEDALUS y que agrega a los *agentes pedagógicos* en distintos grupos según estos niveles. Como se puede apreciar en la figura, no se tiene en cuenta a los Agentes Gestores del Nivel de Infraestructura, introducidos en la sección 5.2, como estrictamente 'pedagógicos' debido a su carácter genérico y posible aplicación a cualquier ámbito, no únicamente al contexto educativo.

Es importante matizar que todos estos grupos de *agentes pedagógicos* no son disjuntos entre sí y en la práctica pueden surgir diseños de *espacios de aprendizaje ubicuos* que incluyan distintos agentes especialistas cuya implicación en responsabilidades pedagógicas dentro de varios servicios permita que

puedan ser considerados como pertenecientes a múltiples de estos grupos a la vez. Así por ejemplo, un mismo *agente pedagógico* (Figura 7.3) que se dedique a obtener información de perfil de un estudiante puede considerarse dependiendo la perspectiva un Agente de Extracción a Nivel de Semántica y Contexto, un Agente Personal a Nivel de Organización o un Agente de Usuario, o Nivel de Interacción.

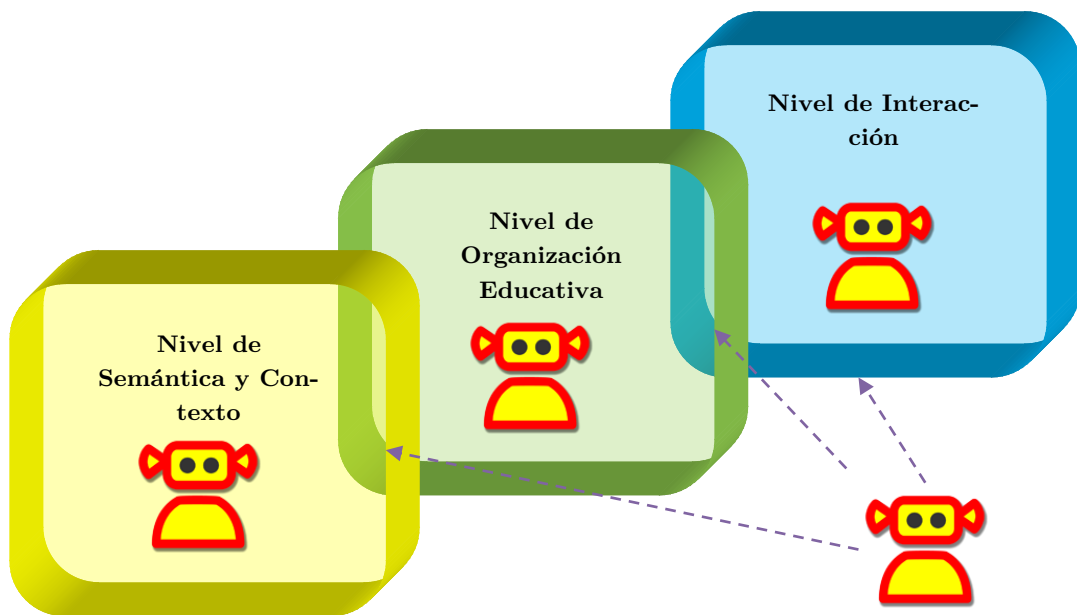


Figura 7.3 Mismo agente, distintas perspectivas

Pasamos a describir en los siguientes apartados cada uno de estos grupos de *agentes pedagógicos* situados en la Plataforma Educativa según se muestra en la Figura 7.1 y Figura 7.2:

- Agentes de Contexto (❶Extracción, ❷Derivación, ❸Agregación)
- Agentes de Administración ❹
- Agentes de Situación ❺
- Agentes de Actividad ❻
- Agentes de Comunidad ❼
- Agentes Personales ❸
- Agentes de Artefacto ❹
- Agentes de Interacción (❸Usuario, ❹Aplicación)

AGENTES DE CONTEXTO

Los Agentes de Contexto cumplen una función primordial dentro de la arquitectura DAEDALUS y en general en cualquier sistema que esté adaptado al contexto ('context-awareness'). Esta clase de agentes focalizan sus responsabilidades en tareas relacionadas con la gestión del modelo contextual: desde la obtención, recopilación y almacenamiento de datos de contexto hasta la deducción e inferencia de nueva información contextual a partir de la obtenida previamente. A menudo esta clase de agentes actúan dentro de la *organización educativa* como productores de información relevante y necesaria para el desempeño de las responsabilidades de otros agentes. Aunque en mayor o menor medida todos los agentes de la arquitectura DAEDALUS manipulan información de contexto de una u otra manera, ya que utilizan el mismo modelo de conocimiento para comunicarse y entenderse, en sentido estricto, únicamente son Agentes de Contexto aquellos que entran en una de estas tres categorías:

- *Agentes de Minado y Extracción de Contexto.* Efectúan un mapeo entre una fuente externa (sensor, base de datos, o incluso un agente de extracción...) y el lenguaje ontológico de la arquitectura. Bajo el nombre de *agentes de información* son habituales en muchos tipos de sistemas, y dentro del dominio educativo son los encargados de extraer datos de fuentes pedagógicas, tales como repositorios de objetos de aprendizaje, plataformas de aprendizaje, bases de datos de estudiantes, etc. Y recopilar toda esta información, posiblemente traduciéndola al lenguaje ontológico de la arquitectura, para que pueda ser empleada por otros agentes
- *Agentes de Derivación de Contexto.* Son capaces de inferir nuevo conocimiento aplicando deducciones sobre el conocimiento conocido. Aplica reglas o cualquier método de deducción sobre los elementos del contexto con el fin de conseguir nueva información útil para otros componentes a partir del contexto con el que se cuenta
- *Agentes de Agregación de Contexto.* Empaquetan la información extraída de diversas fuentes para difundirla entre nodos de la organización y compartirla entre distintos componentes distribuidos. Guardan la información de contexto en una base de datos para permitir realizar búsquedas y recuperaciones por parte de cualquier otro componente.

AGENTES DE ADMINISTRACIÓN

Los Agentes de Administración manejan los aspectos, habitualmente distribuidos, relativos a la configuración y empleo de los elementos tecnológicos presentes en infraestructura educativa. De forma proactiva deben mantener dispuesta la infraestructura tecnológica para eventuales actividades de los usuarios, persiguiendo que todos los elementos necesarios para el funcionamiento de los dispositivos, tanto información contextual necesaria como artefactos educativos, se encuentren disponibles y distribuidos correctamente. Así pues se concretan dos sub-tipos específicos de Agentes de Administración:

- *Agentes de Sincronización.* Se encargan de sincronizar la información contextual disponible en el sistema a través de los distintos nodos con el fin de que el modelo de contexto se encuentre actualizado y disponible en dispositivos móviles que tengan que funcionar en modo desconectado. También realizan tareas de distribución de artefactos como herramientas u objetos de aprendizaje a lo largo de los distintos dispositivos.
- *Agentes de Configuración de Dispositivo.* Se encargan de la instalación de herramientas en los dispositivos además de su configuración con objetos de aprendizaje. Normalmente son los encargados también de proveer a otros agentes acerca de las capacidades y el estado en que se encuentra dicho dispositivo.

AGENTES DE SITUACIÓN

Los Agentes de Situación se especializan fundamentalmente en el tratamiento de la información de espacio y tiempo con el fin de establecer relaciones entre los distintos elementos presentes dentro del *espacio de aprendizaje ubicuo*. Estos agentes aportan servicios a la infraestructura para el reconocimiento de situaciones pedagógicas emergentes o entornos de aprendizaje que permitirán a los usuarios disfrutar de experiencias adaptadas a la localización o al momento. Destacamos dos sub-tipos concretos de Agentes de Situación en función de su dinamismo, denotado éste como su capacidad de reconocer distintos escenarios de aprendizaje:

- *Agentes de Estancia o Entorno Educativo.* Son agentes encargados de coordinar los elementos situados en un espacio físico determinado. Es por ello que normalmente se encuentran circunscritos a un escenario de aprendizaje concreto, como por ejemplo un aula, la sala de

un museo o un determinado parque natural. Pueden estar especializados en la detección de usuarios situados en dicho espacio y comenzar servicios que promuevan o ayuden a éstos en el desarrollo de sus actividades de aprendizaje, pudiendo por ejemplo relacionar los procesos que realizan con elementos presentes en el escenario.

- *Agentes de Detección de Situaciones Emergentes.* Son agentes más generales que los anteriores en tanto y cuanto no se limitan a un determinado escenario de aprendizaje sino que son capaces de descubrir potenciales escenarios en función de la información espacio-temporal. En base a información espacio-temporal del contexto se encargan de coordinar y manipular elementos educativos que se circunscriben a las características actuales de los usuarios, como por ejemplo proponer echar un vistazo a un determinado artefacto cuando se está en cierta localización de un museo o recomendar un estudiante realizar los deberes por la tarde.

AGENTES DE ACTIVIDAD

Este tipo de agentes se responsabilizan de ofrecer servicios relacionados con los procesos pedagógicos a los usuarios antes, durante y tras la finalización de éstos. Por un lado antes pueden ofrecer servicios como por ejemplo soporte a los diseñadores didácticos durante la autoría de las actividades y sus recursos pedagógicos necesarios, o soporte a los estudiantes en la preparación de una salida al campo o al museo. Durante pueden ofrecer ayuda a los usuarios, estructurando y recomendando cadenas de actividades adaptadas. Al final de los procesos pueden analizar el desarrollo de éstos para por ejemplo dar información a los estudiantes acerca de sus fallos en las actividades, evaluar y agregar información para los profesores, etc. Podemos establecer varios sub-tipos:

- *Agentes de Secuenciación de Actividades.* Son capaces de construir y adaptar procesos pedagógicos a los usuarios y sus situaciones, creando flujos de actividad adaptados.
- *Agentes de Recomendación de Actividades.* Van ligados a la categoría de Agentes Detectores de Situaciones Pedagógicas, pudiendo verlos como una especialización de estos.
- *Agentes de Control de Actividades.* Se preocupan de que el desarrollo de las actividades ocurra de la manera más efectiva posible, detec-

tando bloqueos o efectuando acciones para focalizar a los usuarios en sus tareas. Fácilmente este tipo de agentes puede entrar dentro de la categoría de *Agentes de Colaboración* (ver más adelante) si entre sus objetivos se contempla la búsqueda efectiva de la colaboración entre usuarios durante los procesos pedagógicos.

- *Agentes de Análisis de Actividades*. Son capaces de procesar a posteriori los datos disponibles sobre los procesos para poder generar información para evaluar.

AGENTES DE COMUNIDAD

Los Agentes de Comunidad se encargan de responsabilidades relacionadas con grupos de usuarios, incluyendo el tratamiento de las relaciones que existen entre éstos. Habitualmente trabajan con Agentes de Proceso Pedagógico o Agentes de Administración para poder dar soporte a la actividad y la administración de la arquitectura. Según esto podemos identificar dos subtipos de esta clase de agentes:

- *Agentes de Grupo*. Este tipo de agentes se especializan en el tratamiento de la información relativa a un grupo o comunidad dentro del espacio de aprendizaje ubicuo. Entre sus posibles funciones se encuentra el seguimiento del grupo y el minado de información relativa a él. Esto puede ser utilizado para estructurar roles dentro del grupo o crear nuevos grupos, atendiendo a requisitos pedagógicos.
- *Agentes de Colaboración*. Analizan las relaciones que existen entre los usuarios y fomenta la interacción entre ellos. Buscan patrones de colaboración con el fin de detectar afinidades entre estos usuarios que puedan ser útiles para el desarrollo de sus procesos pedagógicos y en base a estos patrones activan mecanismos para hacer efectiva la comunicación entre los usuarios implicados.

AGENTES PERSONALES

Los Agentes Personales guardan y controlan toda la información personal del modelo de usuario o, de forma más concreta, del modelo de estudiante, de las personas a las que representan. Son capaces de ponerse en contacto con otros agentes de la organización educativa para coordinar servicios personalizados gracias a la información que controlan acerca de los usua-

rios, y especialmente de los estudiantes. No obstante también pueden ofrecer servicios que ayuden a cualquier otra clase de usuarios, como por ejemplo profesores o administradores de una plataforma educativa con el fin de ayudarles a estructurar grupos, roles o realizarles tareas administrativas proactivamente como mandar emails, funciones según un calendario, etcétera.

A menudo los Agentes Personales suelen entrar en contacto con las personas a las que representan para preguntarles información, recomendarles actividades, colaboraciones o artefactos, o simplemente ofrecerles motivación. Es por ello que esta categoría de agentes pueden considerarse como una clase de Agentes de Interacción, pero como veremos en seguida la definición de Agente de Interacción puede abarcar más que únicamente Agentes Personales.

AGENTES DE ARTEFACTO

Los Agentes de Artefacto son aquellos capaces de realizar manipulaciones de los objetos de aprendizaje, como por ejemplo su búsqueda, tratamiento de contenidos, y por otro lado son capaces de comunicarse con las distintas herramientas software, por ejemplo para iniciar en éstas tareas de forma proactiva para los usuarios de forma contextualizada. Así pues diferenciamos entre dos tipos de Agentes de Artefactos:

- *Agentes de Objetos de Aprendizaje.* Esta clase de agentes extraen y manipulan información de repositorios, accediendo a metadatos de los objetos de aprendizaje, accediendo a éstos y realizando una interpretación dinámica de los distintos contenidos educativos y productos generados por los usuarios. Son capaces de manipular el contenido de objetos de aprendizaje para adaptarlos tanto en formato, para hacerlos accesibles a cualquier tipo de dispositivo, como en fondo, atendiendo a las necesidades particulares de los estudiantes a los que vayan los contenidos.
- *Agentes de Herramientas.* Esta clase de agentes son capaces de manipular las herramientas o aplicaciones software para poder extraer información de ellas o comunicarse con ellas para poder ejecutar acciones dentro ellas, que pueden dar lugar a un soporte a los usuarios que emplean dichas herramientas.

AGENTES DE INTERACCIÓN

Los Agentes de Interacción son todos aquellos que en mayor o menor medida establecen una comunicación directa con los usuarios. En esta clase de agentes se incluyen a menudo muchos Agentes Personales e incluso Agentes de Artefactos, ya que pueden interactuar indirectamente con el usuario a través de herramientas.

Los Agentes de Interacción más básicos pueden limitarse a manipular la interfaz gráfica para resaltar elementos u ordenarlos, mientras que otros más avanzados pueden ser capaces de mostrar representaciones gráficas complejas como por ejemplo avatares con forma humana o incluir interacciones multimodales como el procesamiento de voz. Por otro lado, como se ha comentado previamente, puede existir una interacción indirecta mediante otros elementos que se consideran pertenecientes al Nivel de Interacción de DAEDALUS, como los objetos de aprendizaje o las herramientas. Los Agentes de Interacción pueden mostrar en la interfaz objetos de aprendizaje, o arrancar funcionalidades de herramientas o comunicarse con los usuarios a través de éstas. Estas interacciones se detallarán más en profundidad en el capítulo 8.

7.2 Diseño de Servicios Pedagógicos Inteligentes como Agentes

Para facilitar el desarrollo de sistemas complejos enfocados al aprendizaje móvil y colaborativo, la arquitectura DAEDALUS persigue en todo momento una correspondencia lo más directa posible entre el diseño de los *servicios pedagógicos inteligentes* (SPIs) y la propia implementación de éstos. A este respecto, como se ha comentado anteriormente, los SPIs que se diseñan dentro de la arquitectura DAEDALUS son modelados e implementados como *agentes pedagógicos*. Es por ello que el diseño de la *organización educativa* se basa en los patrones y modelos computacionales establecidos en los niveles inmediatamente inferiores de la arquitectura. De esta forma, los *agentes pedagógicos* y recursos asociados que constituyen la *organización educativa* hacen uso de todos los modelos descritos en el Nivel de Infraestructura, detallados en el Capítulo 5, para definir tanto la estructura de la organización en sí como la de las propias unidades internas que la componen. Además, los componentes de la *organización educativa* se articulan alrededor de todo el

modelo de conocimiento que es definido en el Nivel de Semántica y Contexto del Capítulo 6, permitiendo tener presente en todo momento un modelo estructural de la información contextual a la hora de diseñar SPIs y contar tanto a la hora de implementación como de ejecución con modelos computacionales para poder hacer frente a la manipulación y tratamiento de esta información.

Dado que una de las características principales de la organización de DAEDALUS es su topología distribuida, esto supone de forma implícita que los SPIs que se estructuren dentro de ella tendrán que repartirse a menudo a lo largo de distintos nodos o dispositivos de la infraestructura. Aparece por tanto la necesidad de una descomposición más fina de los servicios para poder ser repartidos entre varios *agentes pedagógicos* que pueblen diferentes nodos implicados. Las características naturales de distribución de los componentes de DAEDALUS a lo largo de la organización permiten que existan agentes en distintos nodos de la infraestructura que busquen de forma conjunta, coordinada y cooperativa lograr la prestación de los SPIs. Consecuentemente ha de producirse un reparto parcial de las ocupaciones entre componentes de los distintos nodos involucrados en la provisión de un determinado servicio y una obligada cooperación entre estos elementos distribuidos con el fin de prestar dicho servicio. Surge de esta forma y de manera intuitiva la posibilidad de modelar servicios distribuidos dentro de la arquitectura DAEDALUS como un conjunto de *responsabilidades* que se asignan a *agentes pedagógicos* de diferentes nodos. Así pues definimos las *responsabilidades* como los compromisos y obligaciones de un *agente pedagógico* en un determinado nodo hacia la organización para la provisión compartida de un SPI. Estas *responsabilidades* se desgranán en las etapas de diseño del *espacio de aprendizaje ubicuo* a partir de los SPIs a implementar como casos de uso del sistema que con mucha seguridad relacionarán varios componentes de la organización en una búsqueda de la prestación de un determinado servicio (Figura 7.4, ❶). Esta es una manera muy práctica de diseñar entornos de aprendizaje distribuidos ya que el diseño se realiza a un alto nivel tratando de identificar en primera instancia los *servicios pedagógicos inteligentes* con los que se pretende dotar al sistema. A continuación se realizará un refinamiento de cada uno de ellos en una lista de *responsabilidades*, asociando cada una de estas *responsabilidades* a los nodos implicados en ofrecer los servicios distribuidos y posteriormente repartiéndolas entre diferentes *agentes pedagógicos* que residan en dichos nodos.

7.2.1 Modelando responsabilidades dentro de los agentes pedagógicos

Una vez reconocidas las *responsabilidades* de los nodos e identificados los agentes necesarios en cada nodo responsables de prestar los servicios pedagógicos, se procede a estructurar las *responsabilidades* como elementos básicos de sus modelos internos. Cada responsabilidad de un agente pedagógico ha de pasar a formar parte de su estructura interna y se modelará como una serie de objetivos y tareas (Figura 7.4, ❷). Los objetivos son los fines últimos de las responsabilidades y por tanto los agentes tratan de perseguir su cumplimiento de forma satisfactoria para la organización educativa. Posiblemente y dependiendo de la complejidad de cada responsabilidad, sea necesario configurar distintos objetivos dentro del agente para modelar dicha responsabilidad o incluso se necesite refinar aún más la responsabilidad de varios objetivos repartidos entre distintos agentes y que éstos colaboren para cumplirlos.

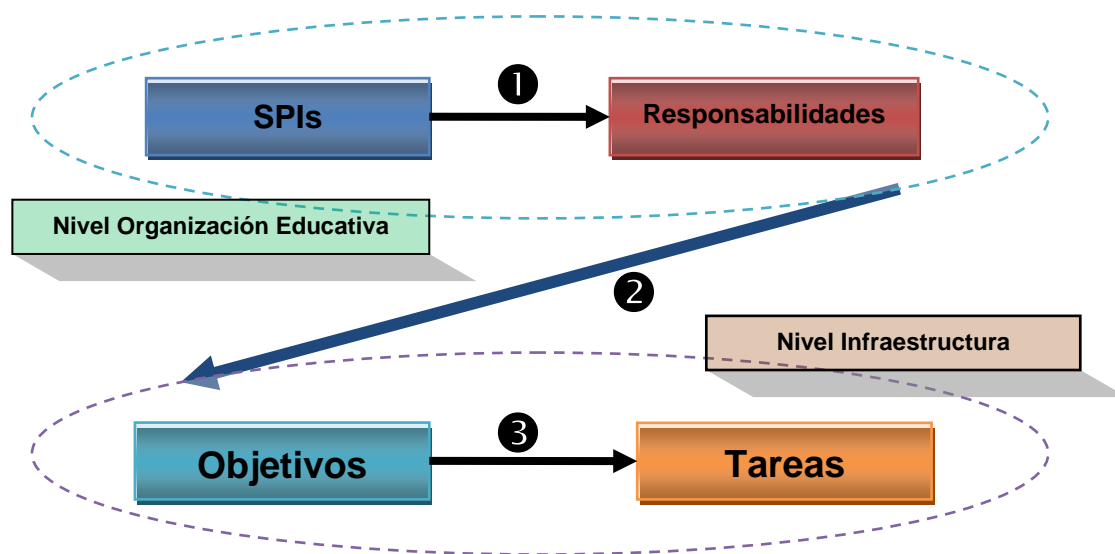


Figura 7.4 Del diseño educativo, al diseño con agentes

Por otro lado, las tareas son los elementos activos de dichas responsabilidades y representan las acciones que los agentes pueden llevar a cabo para conseguir el cumplimiento de sus objetivos y por extensión la provisión de los servicios pedagógicos de la infraestructura. Las tareas se agrupan dentro de los objetivos del agente y se estructuran ordenadamente en forma de planes que le permiten avanzar hacia la resolución de sus objetivos (Figura 7.4, ❸).

Cada tarea efectúa cambios en el estado del entorno, incluidos otros agentes, recursos, dispositivos, usuarios o el propio agente, lo que provoca ya sea de forma directa o indirecta a través de su percepción del entorno, un cambio en el estado mental del agente. Así pues las tareas en definitiva permiten a los agentes buscar el cumplimiento de sus responsabilidades dentro del entorno educativo.

Cada agente hace uso de un determinado conjunto de elementos de dominio definidos dentro del modelo de conocimiento de DAEDALUS. La percepción del agente absorbe estos elementos que pasan a formar parte de su estado mental, condicionando de un modo u otro la resolución de los objetivos de dicho agente.

Dependiendo del modelo de comportamiento concreto del agente, el mapeo de las responsabilidades como entes abstractos de diseño a los elementos concretos de implementación definidos en su patrón de comportamiento será más o menos inmediato. De este modo, los agentes cognitivos ofrecen un mayor grado de abstracción de sus elementos básicos internos, lo que permite un mapeo explícito hacia los objetivos y tareas de los agentes. Sin embargo, los agentes reactivos poseen un patrón de comportamiento de más bajo nivel de abstracción por lo cual la transferencia de *responsabilidades* identificadas en el diseño no es trasladable de forma tan directa. A su favor, los agentes reactivos ofrecen una manera más sencilla de modelar responsabilidades poco complejas o incluso dividir responsabilidades complejas en varias más simples y asignarlas a varios agentes reactivos.

RESPONSABILIDADES EN AGENTES PEDAGÓGICOS REACTIVOS

El patrón de comportamiento de los agentes (micro-)reactivos, tal y como fue descrito en el apartado 5.3.1, se basa en un autómata de estados finitos. De esta manera, las responsabilidades asociadas a este tipo de agentes se modelan como una sucesión de estados a través de los cuales el agente va transitando según recibe ciertas entradas y realizando acciones para cumplir con dicha responsabilidad.

En la Figura 7.5 se puede apreciar un esquema genérico de autómata para un agente pedagógico (micro-)reactivo. De forma recomendable, es importante que un agente pedagógico (micro-)reactivo contenga una inicialización (Init), una terminación (Termination) y una gestión de errores (Error Management), cada una de estas representada mediante una sub-maquina de

estados dentro del autómata. De la misma manera, cada responsabilidad pedagógica (Pedagogic Responsibility) a diseñar dentro del agente se estructura como otra sub-máquina de estado que modelará todos los pasos para realizar. Cada uno de estos pasos se corresponde con una transición entre estados que queda establecida mediante pares entrada/acción semántica ([input]/SemanticAction()). Una transición es activada por una entrada generada por el módulo de percepción del agente y que disparará una acción semántica a realizar por éste en búsqueda de la resolución de la responsabilidad. Las entradas pueden contener parámetros que hacen referencia a elementos del *espacio de aprendizaje ubicuo*, que se especifican según el modelo de conocimiento de DAEDALUS, permitiendo manipularse esta información dentro de las acciones semánticas.

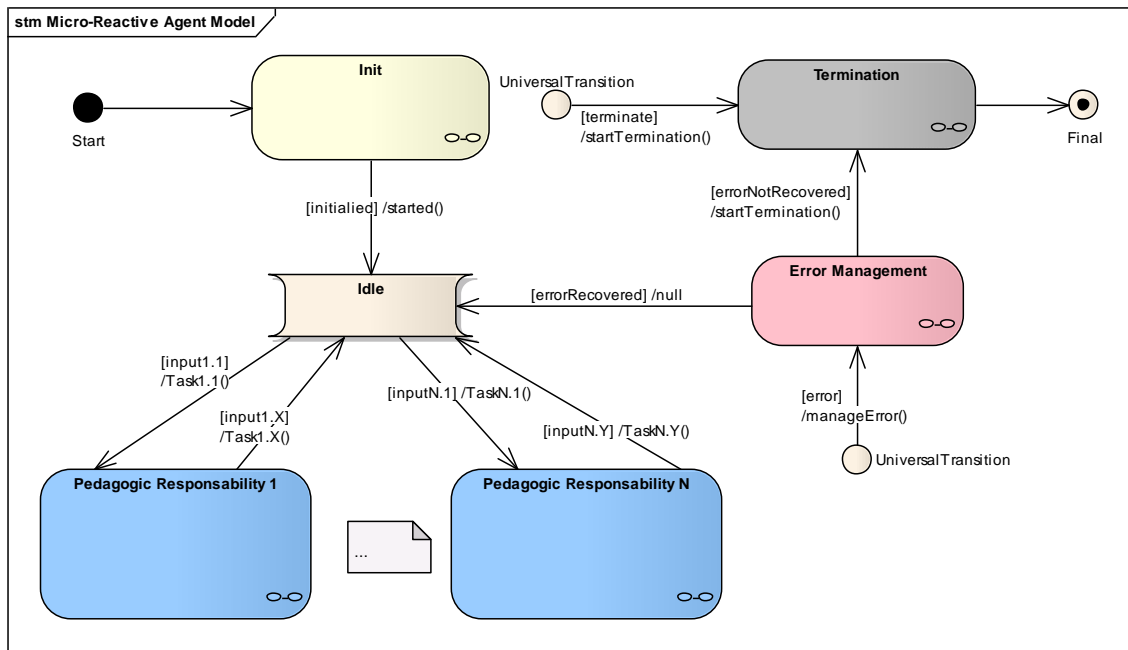


Figura 7.5 Esquema general de autómata de agente pedagógico (micro-)reactivo

Algo habitual en el diseño de un agente reactivo es que contenga un estado de espera (*idle*) desde el cual se inicia cada una de estas responsabilidades, 'reaccionando' ante una activación externa. Durante la realización de una responsabilidad, el agente ha de estar preparado para cualquier error, que será manejado en su gestión de errores, o para una inmediata terminación por parte de una orden a través de su interfaz de gestión.

RESPONSABILIDADES EN AGENTES PEDAGÓGICOS COGNITIVOS

Las responsabilidades en los agentes pedagógicos cognitivos se mapean directamente como objetivos de éstos y un conjunto de tareas asociadas. Una responsabilidad para un agente pedagógico cognitivo es un objetivo que el agente ha de resolver mediante una selección apropiada de tareas. La selección de las tareas se hace de acuerdo a un plan de resolución de objetivos codificado como reglas. En la Figura 7.6 se puede observar el esquema general de una regla de un agente pedagógico cognitivo en notación de diagrama gráfico (izquierda), junto con su correspondiente sintaxis programática (derecha). Toda regla se modela mediante un lado izquierdo (*when*), que define la precondition que de cumplirse para que se active la regla, y un lado derecho (*then*) que describe las tareas que el agente realizará en caso de activarse dicha regla. Habitualmente el lado izquierdo contendrá referencias al objetivo (*Goal*), al foco (*Focus*) y a una serie de creencias (*Belief*). Mientras que el lado derecho, aparte de especificar las tareas a ejecutar por el agente, puede contener modificaciones directas de los elementos del estado mental del agente añadiendo (*insert*), actualizando (*update*) o retirando (*retract*) creencias u otros elementos del estado mental del agente.

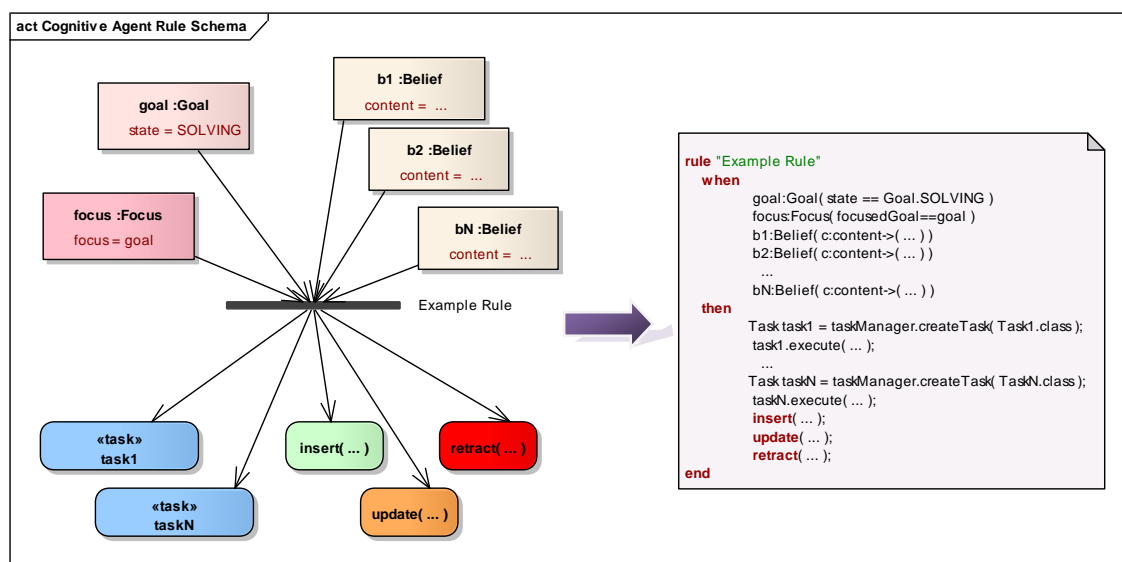


Figura 7.6 Esquema general de una regla de un agente pedagógico cognitivo

Existen tres tipos de reglas: de creación, de control y de resolución de objetivos, respectivamente.

En primer lugar, las reglas de creación de objetivos se emplean para hacer consciente al agente de que tiene que cumplir una determinada responsabilidad. En la Figura 7.7 se puede apreciar el esquema habitual de una regla de creación de objetivos. Suele contar con una precondition dispuesta mediante una serie de creencias que activa la inserción de un nuevo objetivo en estado pendiente (PENDING).

Los objetivos de un agente pueden atravesar por diferentes estados durante su ciclo de vida, como pendiente (PENDING), resolviéndose (SOLVING), resuelto (SOLVED) o fallido (FAILED).

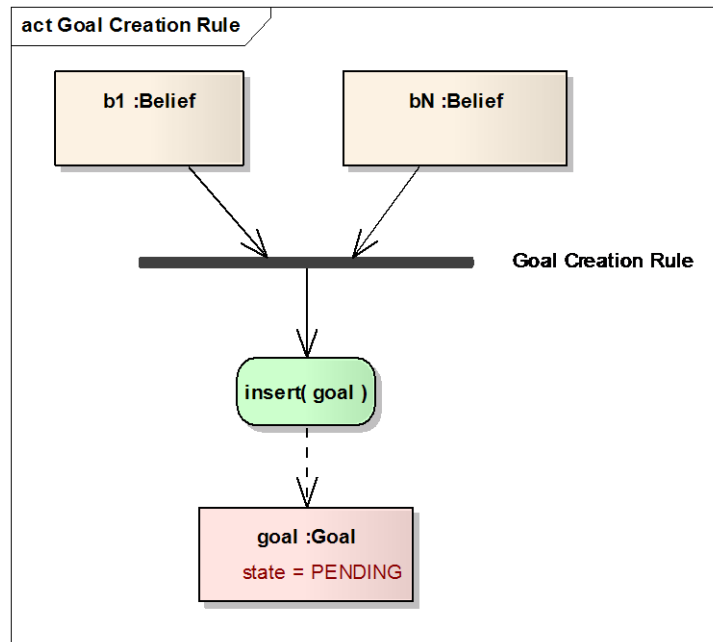


Figura 7.7 Esquema de regla de creación de objetivos

El control del objetivo al que el agente está prestando atención se realiza mediante un mecanismo de foco (Focus). El foco determina en cada momento el objetivo que el agente va a intentar resolver. Puede darse el caso durante la resolución de un objetivo que sea necesario atender a una responsabilidad más prioritaria, o simplemente que no sea posible avanzar en la resolución de la responsabilidad actual. Es por tanto que se utilizan reglas de cambio de foco como mecanismo para poder realizar este cambio. Una regla de cambio de foco (Figura 7.8) tiene un precondition que especifica el estado del foco actual y el estado de ciertos objetivos. Como resultado de la activación de una regla de este tipo, se produce una actualización del foco, que apuntará a otro objetivo diferente.

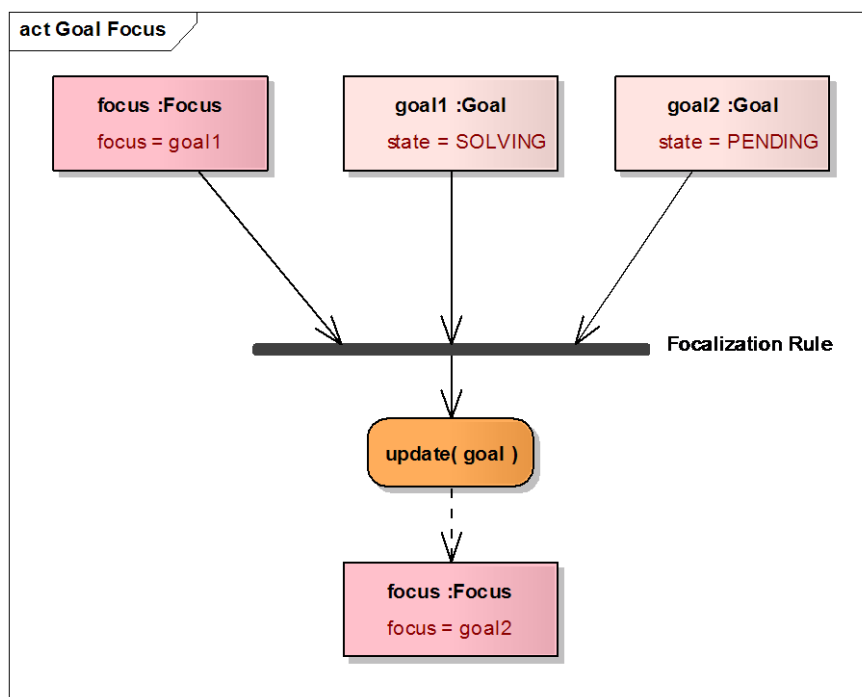


Figura 7.8 Esquema de regla de focalización de objetivos

Finalmente, el tercer tipo de reglas se corresponde con las reglas de resolución de objetivos. Estas son las reglas que guiarán al agente hacia la consecución de un determinado objetivo y las que estructurarán la ejecución de las tareas que deberá realizar para lograrlo. Todas las reglas de resolución de objetivo tienen como precondition el objetivo en cuestión junto con el foco que apunta a dicho objetivo. No obstante, podemos dividir en tres subtipos de reglas (Figura 7.9). Primeramente tenemos las reglas de inicialización (Init Rule), las cuales básicamente son las encargadas de arrancar la resolución de un objetivo, pasando el estado de éste de pendiente (PENDING) a resolviéndose (SOLVING), y aprovechando a realizar cualquier tarea necesaria con la inicialización del objetivo. Por otro lado están las reglas de desarrollo de la resolución del objetivo (Development Rule). Y finalmente se encuentran las reglas de conclusión de la resolución de un objetivo (Conclusión Rule), que determinan que cuando ciertos hechos se dan lugar, el estado del objetivo pasa a estar resuelto (SOLVED) si todo ha salido con éxito, o fallido (FAILED) en caso contrario.

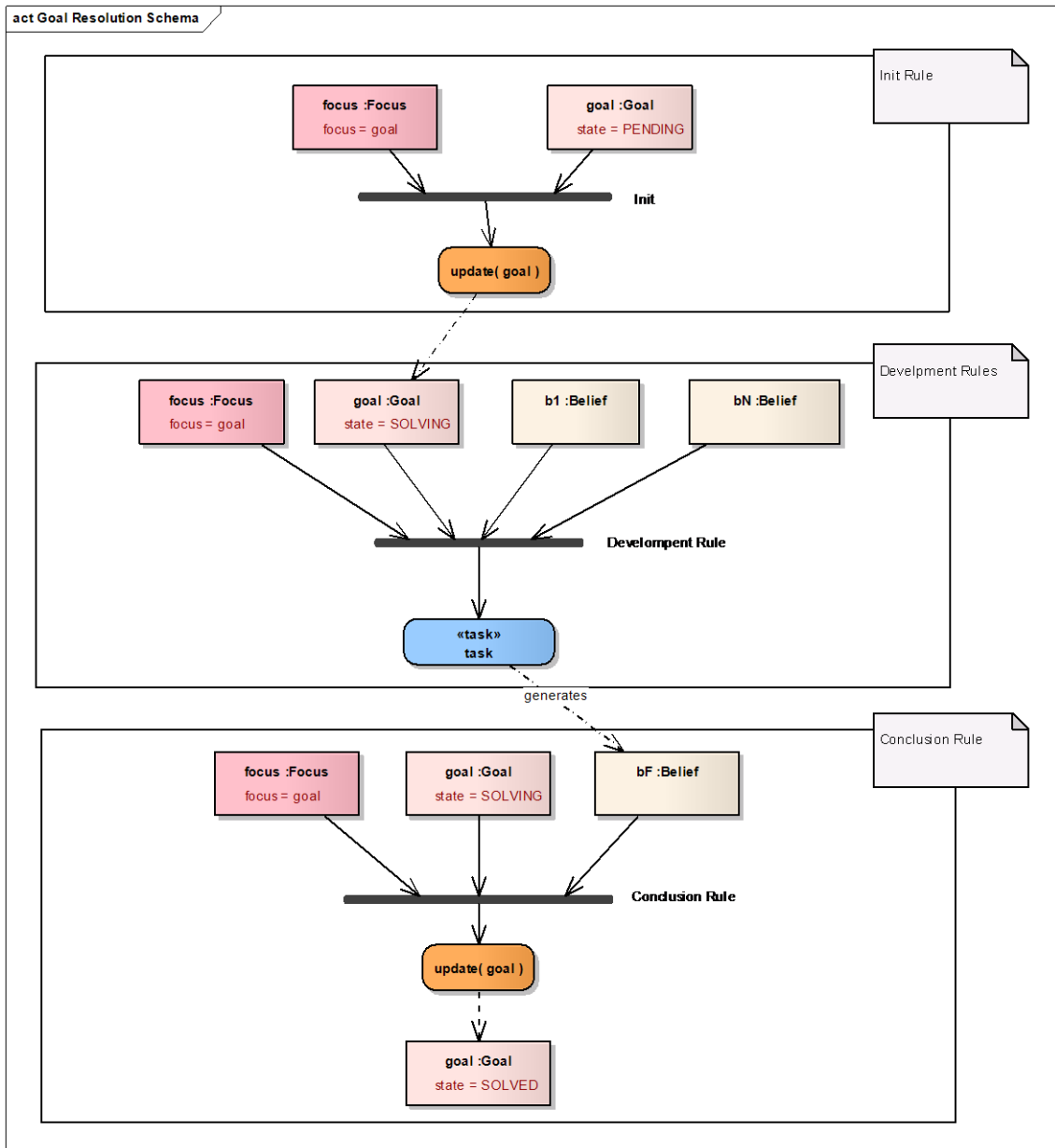


Figura 7.9 Esquema general de resolución de objetivos

Las tareas de los agentes cognitivos se especifican dentro de las reglas de desarrollo del objetivo. Estas tareas son acciones que se disparan al cumplirse las precondiciones de este tipo de reglas obteniéndose como resultado una modificación del estado mental del agente al generarse una serie de creencias.

La modificación del estado mental del agente por parte de la ejecución de una tarea puede ser *directa*, si las tareas modifican directamente los ele-

mentos del estado mental del propio agente, o *indirecta*, si las tareas sobre el entorno del agente, como por ejemplo los demás agentes o recursos de la organización educativa, y estos cambios son posteriormente captados por la percepción del agente y asimilados en última estancia como nuevas creencias.

8

NIVEL DE INTERACCIÓN

Dentro del Nivel de Interacción se sitúan todos los elementos necesarios para poder ofrecer la interactividad entre el sistema y el usuario. Al ser un espacio de aprendizaje ubicuo un sistema eminentemente interactivo para los usuarios, los elementos con los que interactúan los usuarios reciben una atención especial. A continuación en este capítulo se discutirán los elementos presentes en este Nivel de Interacción, incluyendo las interfaces de usuario, las herramientas software, que permiten a los usuarios manipular los productos que deberán generar, junto con los agentes y recursos específicos que los gestionan.

8.1 Elementos del Nivel de Interacción

Dentro del Nivel de Interacción se sitúan todos los elementos que están relacionados con la comunicación con el usuario. Esta comunicación se realiza de dos maneras diferentes:

- Directamente, mediante interfaces de usuario disponibles en los dispositivos del espacio de aprendizaje ubicuo.
- Indirectamente, mediante el acceso a las herramientas de aprendizaje que residen en los dispositivos y que los alumnos manipulan durante sus actividades.



Figura 8.1 Elementos del Nivel de Interacción

Los agentes y recursos que se definen en este nivel de la arquitectura DAEDALUS son los encargados de gestionar tanto las interfaces de usuario como las funcionalidades de las herramientas de aprendizaje en función de las necesidades de los agentes del Nivel de Organización Educativa.

8.2 Recursos de Interacción

Los recursos de interacción cumplen la misión de dar una forma de acceso y de control para los agentes de interacción a las interfaces de usuario y herramientas.

Estos recursos cumplen básicamente una función de recubrimiento de recursos externos a la organización para que puedan ser empleados por los agentes. Los recursos de interacción exponen por tanto a los agentes las funcionalidades y características accesibles y modificables de las interfaces de usuario y herramientas.

8.2.1 Recursos de Interfaces de Usuario

Las personas entran en contacto con los dispositivos a través de las interfaces de usuario presentes en estos. Las interfaces de usuario articulan el espacio de trabajo que pone en contacto a las personas con el entorno ubicuo, organizando de qué forma se accede y visualizan las actividades, tareas y artefactos educativos. Los recursos de interfaz de usuario poseen operaciones de transformación, para mostrar u ordenar elementos, que ofrecen la capacidad a los agentes de poder adaptar las experiencias de las personas, y ofrecerles acceso a determinada información, herramientas y objetos de aprendizaje para llevar a cabo los procesos pedagógicos.

No se pone ningún tipo de restricción acerca de cómo ha de ser la interfaz ni qué tecnología ha de emplearse (programática, web, ...). Lo que sí que se plantea dentro de este nivel es que la interfaz de usuario ha de estar en consonancia con el resto de la arquitectura y por lo tanto se modela como un componente más. Este componente será un recurso que los agentes serán capaces de utilizar. Por tanto la interfaz se expone como un servicio con una API para poder ser manejada por los agentes.

Esto se alinea claramente con patrones como Modelo-Vista-Controlador, definiendo los recursos de interfaz de usuario como elementos de la vista y los agentes como los elementos controladores dentro del sistema, permitiendo una reestructuración de éste fácilmente y promoviendo la reusabilidad. De esta forma por ejemplo, un sistema de agentes que exponga su interfaz mediante una serie de páginas web, puede ser posteriormente reestructurado para mostrarse dentro de un dispositivo móvil como una inter-

faz gráfica programática de cualquier sistema operativo, sin tener que hacer modificaciones en absoluto de la capa de control, es decir de los agentes.

8.2.2 Recursos de Herramientas de Aprendizaje

Las herramientas de aprendizaje son las piezas angulares para el diseño de los procesos pedagógicos y la generación de sus productos, es decir, los objetos de aprendizaje. Los usuarios, principalmente estudiantes, han de tener acceso directo a ellas mediante objetos manipulables que se representan dentro del Nivel de Interfaz de Usuario.

Como se ha comentado en anteriores capítulos, los objetos de aprendizaje tienen un lugar destacado dentro de la arquitectura DAEDALUS ya que representan tanto los artefactos necesarios para realizar las actividades, como los resultados y productos del aprendizaje generados por los estudiantes durante estas. Los estudiantes crean y editan de forma colaborativa estos objetos de aprendizaje a través de las herramientas, almacenándose con metadatos para poder ser buscados por personas o agentes posteriormente.

Las herramientas son aplicaciones software con las cuales se llevan a cabo las actividades pedagógicas dentro de los dispositivos. Son las que exponen una serie de funcionalidades a los usuarios poder realizar las tareas que se les pide y para poder manipular en las actividades los artefactos u objetos de aprendizaje. Los recursos de herramientas de aprendizaje son los que exponen a los agentes de interacción y les permiten acceder y controlar dichas funcionalidades.

8.3 Agentes de Interacción

Los Agentes de Interacción son todos aquellos agentes que son capaces de controlar los distintos elementos de interacción descritos, es decir interfaces de usuario y herramientas, para gestionar la comunicación de la Organización Educativa con los usuarios del espacio de aprendizaje ubicuo. Habitualmente, agentes de la Organización Educativa pueden considerarse agentes de Interacción como suele ser el caso de muchos Agentes Personales, que hacen uso de las interfaces de usuario, e incluso Agentes de Artefactos, ya que suelen interactuar indirectamente con el usuario a través de herramientas.

8.3.1 Agentes de Usuario

Este tipo de agente es el que está en contacto con el usuario a través de una interfaz gráfica o de texto. Los Agentes de Usuario más básicos pueden limitarse a manipulaciones sencillas de la interfaz gráfica, como por ejemplo para resaltar elementos u ordenarlos, mientras que otros más avanzados pueden ser capaces de mostrar representaciones gráficas complejas como por ejemplo avatares con forma humana o incluir interacciones multimodales como el procesamiento de voz.

A veces se encuentra representado por un avatar que sirve como personaje virtual para hacer de puente entre el estudiante y la aplicación, humanizándola en cierta medida. Las responsabilidades del Agente de Usuario y el Agente Personal son difusas, habiendo veces que se solapan, como en términos de autenticación y autorización de un usuario al sistema y al ser el Agente de Usuario el encargado de personalizar y adaptar la interfaz a los gustos del usuario.

8.3.2 Agentes de Aplicación

Por otro lado, como se ha comentado previamente, puede existir una interacción indirecta mediante otros elementos que se consideran pertenecientes al Nivel de Interacción de DAEDALUS, como los objetos de aprendizaje o las herramientas. Los Agentes de Interacción pueden mostrar en la interfaz objetos de aprendizaje, o arrancar funcionalidades de herramientas o comunicarse con los usuarios a través de éstas.

PARTE III

Aplicación y

Resultados

9

LA ARQUITECTURA DAEDALUS Y EL ESCENARIO ENLACE

La arquitectura DAEDALUS se ha ido poniendo a prueba en una serie de escenarios pedagógicos con el fin de realizar una evaluación formativa del sistema. En este capítulo retomaremos nuestro escenario de diseño planteado en el apartado 3.1 para tratar de definir los componentes, tanto agentes como recursos, necesarios en una *organización educativa* con el fin de definir los *servicios pedagógicos inteligentes* concretos que den respuesta a las necesidades planteadas en la sección 3.1.3.

Para abordar el diseño de la *organización educativa* hay que especificar cuáles van a ser los agentes y los recursos necesarios para poder poner en marcha cada escenario. La forma que proponemos, en un primer paso, consiste en obtener una lista de las responsabilidades asociadas a cada nodo. Poste-

riormente se tratará de agrupar dichas responsabilidades con el fin de identificar los agentes implicados en nuestro escenario. Al escenario descrito en el apartado 3.1.1, con el que hemos enfocado el diseño de DAEDALUS y al que dedicamos este capítulo, se le une otro más que describimos en el capítulo 10 con el que se ha testado la idoneidad de nuestra arquitectura frente a otros escenarios reales y su adaptación y extensibilidad a nuevos escenarios.

9.1 Infraestructura para el Escenario

ENLACE

Como elección de diseño para el escenario de salida al campo de ENLACE se estructurará la organización en dos tipos distintos de nodos:

- Nodo Servidor
- Nodo Móvil

Los Nodos Móviles se corresponden con los dispositivos PDAs que utilizan los estudiantes durante su visita al parque natural. Los Nodos Servidores serán nodos estáticos que se corresponden con ordenadores situados en el colegio y servirán de almacén de los artefactos educativos y de los datos de los usuarios, mediante un Repositorio de Objetos de Aprendizaje, específicamente E-LOR, y un LMS, instalados en dichas máquinas. Para simplificar la explicación tomaremos en cuenta que sólo existe un Nodo Servidor, aunque potencialmente se podría distribuir sus componentes en varias máquinas.

9.2 SPIs y Responsabilidades en el Escenario

ENLACE

Retomando el escenario de diseño del apartado 3.1.1, se procede a elaborar los distintos elementos necesarios correspondientes al Nivel de Organización Educativa. En un primer refinamiento, en función de la descripción del escenario del apartado 3.1.1 y de los casos de uso extraídos en el apartado 3.2.2, podemos enumerar los siguientes servicios pedagógicos inteligentes (S) y dividirlos en responsabilidades (R) asociados a los tipos de nodos según se muestra en la Tabla 9.1:

Servicios Pedagógicos Inteligentes	Responsabilidades
<p>S1. Sincronización de información contextual entre dispositivos antes y después de las actividades.</p> <p>(Extraído del Caso de Uso 1)</p>	<p>R1. Recopilación global de información presente en los dispositivos (Nodo Servidor)</p> <p>R2. Gestión global de información presente en los dispositivos (Nodo Servidor)</p> <p>R3. Gestión local de la información del dispositivo (Nodo Móvil)</p>
<p>S2. Administración de herramientas en dispositivos para que estén listas para las actividades.</p> <p>(Extraído del Caso de Uso 1)</p>	<p>R1. Selección y envío de herramientas a los dispositivos. (Nodo Servidor)</p> <p>R2. Gestión local de la instalación de herramientas en dispositivo. (Nodo Móvil)</p> <p>R3. Informar al usuario de la instalación de nuevas herramientas. (Nodo Móvil)</p>
<p>S3. Configuración y personalización de dispositivos con objetos de aprendizaje relativos a la actividad y al alumno.</p> <p>(Extraído del Caso de Uso 1)</p>	<p>R1. Selección y envío de objetos de aprendizaje a los dispositivos (Nodo Servidor)</p> <p>R2. Gestión local de la configuración de objetos de aprendizaje en dispositivo. (Nodo Móvil)</p>
<p>S4. Identificación y recomendación de posibles oportunidades de colaboración con otros miembros de la comunidad.</p> <p>(Extraído del Caso de Uso 2)</p>	<p>R1. Mantener actualizada en tiempo real información del contexto en cada dispositivo. (Nodo Móvil)</p> <p>R2. Analizar patrones de colaboración frente a información de contexto. (Nodo Móvil)</p> <p>R3. Enviar peticiones de colaboración y habilitar recursos necesarios. (Nodo Móvil)</p>
<p>S5. Control del desarrollo de la actividad</p> <p>(Extraído del Caso de Uso 3)</p>	<p>R1. Mantener actualizada en tiempo real información del contexto en cada dispositivo. (Nodo Móvil)</p> <p>R2. Interpretar peticiones de control sobre el desarrollo de las actividades. (Nodo Móvil)</p>
<p>S6. Reconfiguración de la infraestructura y creación de nuevas actividades en función de eventos imprevistos.</p> <p>(Extraído del Caso de Uso 4)</p>	<p>R1. Crear y distribuir nuevas actividades a dispositivos. (Nodo Móvil)</p> <p>R2. Enviar recursos necesarios para la nueva actividad. (Nodo Móvil)</p> <p>R3. Informar a los usuarios de nuevas actividades disponibles. (Nodo Móvil)</p>

Tabla 9.1 SPIs y Responsabilidades para el escenario ENLACE

Ciertas responsabilidades necesitarán ser refinadas nuevamente para poder ser repartidas de manera homogénea según funciones a distintos agentes. Así pues, S1-R2 plantea la necesidad de gestionar la información que hay en cada uno de los dispositivos desde el nodo servidor. Esto incluirá sub-responsabilidades referentes por un lado a la gestión de la información de los usuarios, que tendrá que ser extraída de un LMS junto con información sobre sus actividades, y por otro a la gestión de los artefactos de los usuarios, que incluye el almacenamiento central en un LOR de los objetos de aprendizaje generados en los dispositivos. A partir de las responsabilidades planteadas en la Tabla 9.1, vamos a continuar avanzando en el análisis de estas responsabilidades con el fin de refinarlas y asignarlas a agentes especialistas que modelarán los servicios pedagógicos inteligentes en el escenario de ENLACE.

9.2.1 Agentes Pedagógicos para el escenario ENLACE

Los agentes de DAEDALUS para el escenario de ENLACE persiguen el cumplimiento de los requisitos pedagógicos del sistema. Están diseñados con el fin de aportar una funcionalidad concreta y parcial de la aplicación. Por un lado, agentes cognitivos situados en el Nodo Servidor son responsables de obtener toda la información necesaria para configurar, instalar, seleccionar herramientas y artefactos de aprendizaje, y sincronizar la información adecuada a los dispositivos móviles. Por otro lado, agentes micro-reactivos en los Nodos Móviles utilizan la información suministrada por los cognitivos para desarrollar las actividades de aprendizaje y prestar servicios a los alumnos.

La organización de agentes del escenario de ENLACE (Figura 9.1) destinada a prestar los servicios inteligentes para este escenario contendrá:

- Agentes de Contexto
 - Agentes de Extracción y Minado de Datos (`LearningDataMinerAgent`)
 - Agentes de Derivación de Contexto (`GlobalContextAgent`)
- Agentes de Administración
 - Agentes de Sincronización de Dispositivos (`DeviceManagerAgent`)
 - Agentes de Configuración de Dispositivo (`DeviceAgent`)
- Agentes de Comunidad
 - Agentes de Colaboración (`CollaborationAgent`)
- Agentes Personales / Usuario (`PersonalAgent`)

Los Agentes Personales serán también los Agentes de Interacción dentro de nuestro escenario, ya tomarán el rol de Agentes de Usuario para comunicarse con los usuarios a través de la interfaz gráfica de los dispositivos móviles, y poder hacer llegar toda la información procedente del resto de la organización de agentes.

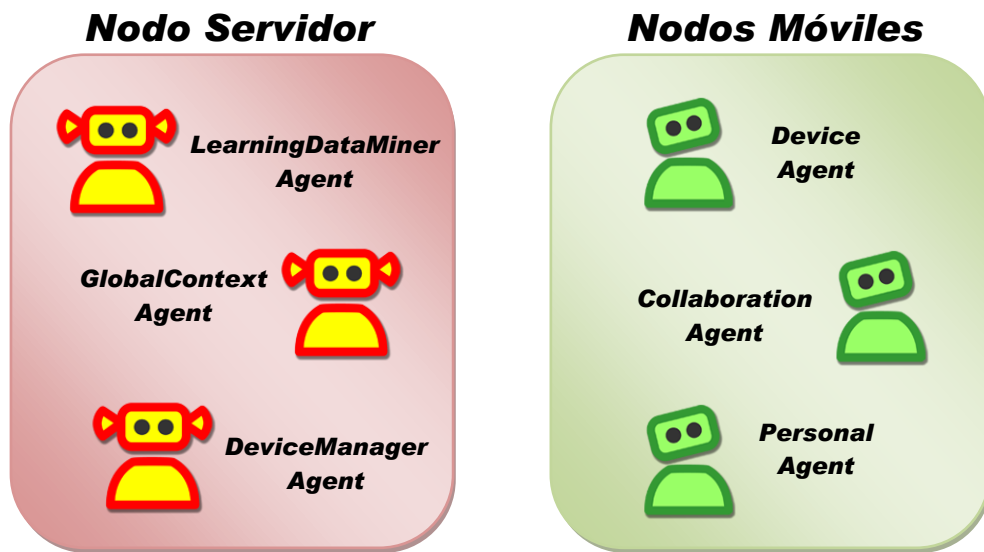


Figura 9.1 Agentes Pedagógicos para el escenario ENLACE

La elección de tipo de nodo para cada uno de estos agentes pedagógicos se corresponde con las responsabilidades que llevarán asociadas por decisiones de diseño (ver Tabla 9.1). Los agentes pedagógicos situados dentro del Nodo Servidor serán diseñados como *agentes cognitivos* y poseerán por tanto gran capacidad de razonamiento para poder coordinar todo el *espacio de aprendizaje ubicuo*. Los agentes pedagógicos que se incluyen en los Nodos Móviles serán *agentes micro-reactivos* con menor potencia de razonamiento pero con la capacidad de acompañar a los usuarios y habitar las PDAs que se utilizarán durante las salidas al campo. A continuación se desarrollarán las responsabilidades de los agentes pedagógicos de ENLACE de cada nodo.

NODO SERVIDOR

En el Nodo Servidor residen tres agentes cognitivos, *LearningDataMinerAgent*, *GlobalContextAgent* y *DeviceManagerAgent*, que se reparten las responsabilidades de la Tabla 9.1 referentes a este tipo de nodo:

- *LearningDataMinerAgent*. Este agente tiene como función el minado y la extracción de datos que serán necesarios para la configuración y el desarrollo de los procesos pedagógicos. Es el responsable directo de S1-R2 y S3-R1 de la Tabla 9.1. Se encarga de obtener información social, pedagógica y de contenidos educativos, con el fin de clasificarla y de que esté disponible para el resto de agentes de la organización educativa. Es el responsable de realizar búsquedas en repositorios de objetos de aprendizaje y en sistemas de gestión educativos para estructurar todos los datos acerca de los grupos y usuarios, sus actividades y tareas, y sus productos y resultados.

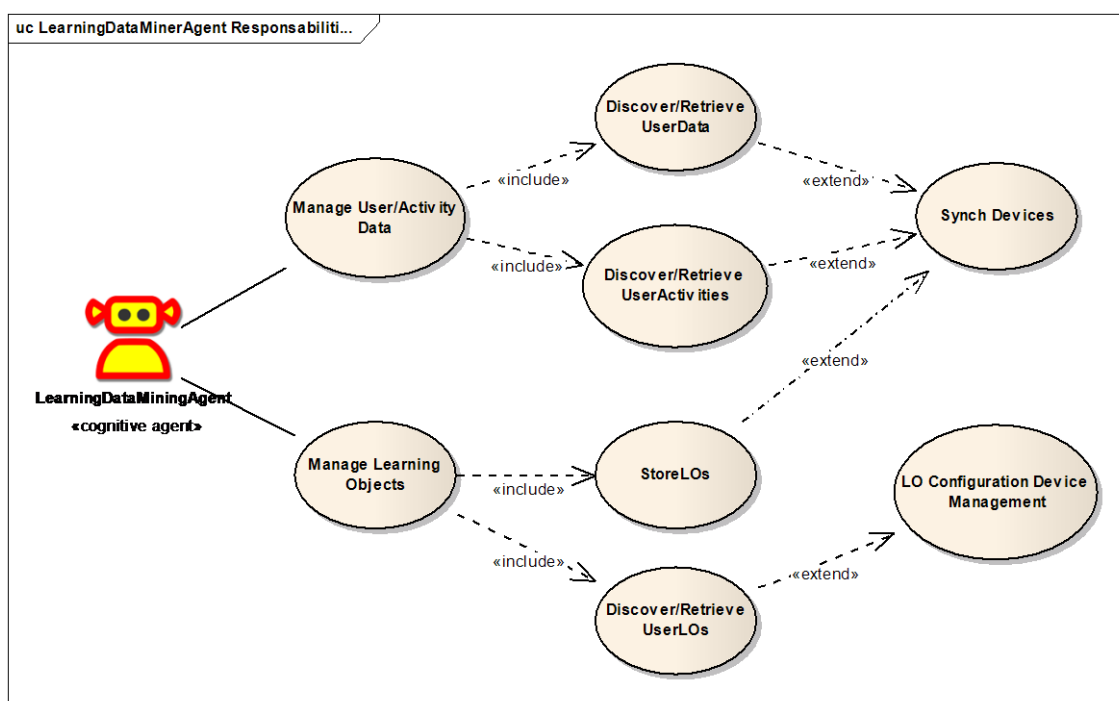


Figura 9.2 Responsabilidades de LearningDataMiningAgent

En el diagrama de responsabilidades de la Figura 9.2, se puede observar que sus dos responsabilidades principales se desglosan en la gestión global dentro de la infraestructura de los datos de usuarios (*Manage User/Activity Data*), que incluye el descubrimiento y recuperación del LMS de los datos sobre estos usuarios y sus actividades (*Discover/Retrive UserData* y *Discover/Retrive UserActivities*), y que será parte necesaria para poder sincronizar los dispositivos (*Synch Devices*). Por otro lado se encargará de la gestión global de los objetos de

aprendizaje (Manage Learning Objects), que incluye su descubrimiento y recuperación (Discover/Retrieve UserLOs) para poder personalizar los dispositivos de los usuarios (LO Configuration Device Management) y su almacenamiento en el LOR (Store LOs).

- *GlobalContextAgent*. Este agente se encarga de mantener un modelo contextual global, guardándolo en una base de datos para mantener un historial, y es capaz de inferir información contextual para generar eventos de interés para los otros agentes, estando relacionado con las responsabilidades S1-R2, S2-R1 y S3-R1 de la Tabla 9.1.

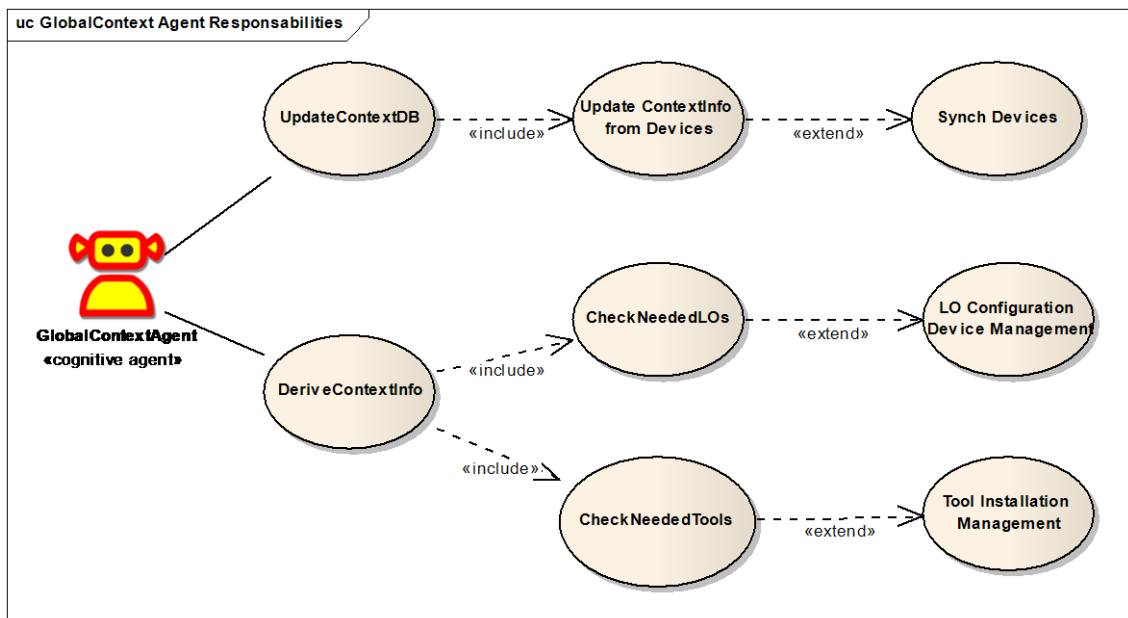


Figura 9.3 Responsabilidades de GlobalContextAgent

Concretamente dentro de nuestro escenario y como se muestra en el diagrama de responsabilidades de la Figura 9.3, el agente *GlobalContextAgent* se encarga de mantener actualizada en el sistema la información contextual extraída de los dispositivos, y es el encargado de comprobar analizando esta información si es necesario algún recurso de aprendizaje, ya sean herramientas u objetos de aprendizaje, en algún dispositivo.

- *DeviceManagerAgent*. Este agente es el encargado de gestionar todos los dispositivos móviles del escenario, con la ayuda de los dos agentes anteriores. Se encarga de contactar con cada dispositivo y extra-

er toda la información acerca del dispositivo y el usuario que lo está utilizando, y transmitirle información para su configuración, que se corresponde con S1-R1, S2-R1 y S3-R1 de la Tabla 9.1.

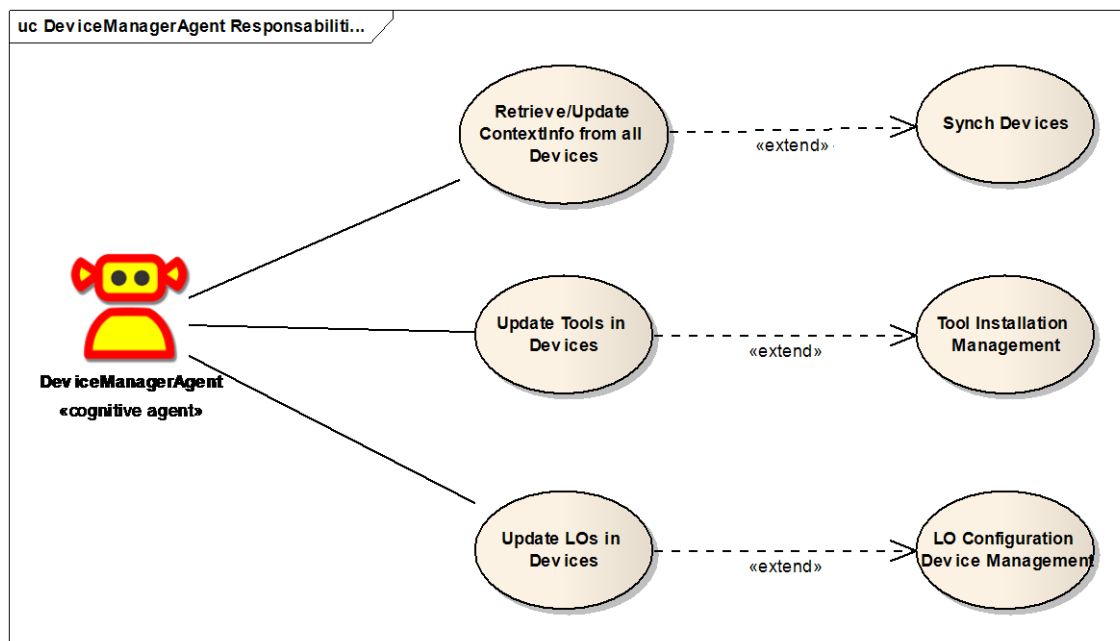


Figura 9.4 Responsabilidades de DeviceManagerAgent

En el diagrama de responsabilidades del agente *DeviceManagerAgent* de la Figura 9.4, que incluyen obtener y actualizar la información de contexto de los dispositivos móviles (Retrieve/Update ContextInfo), actualizar las herramientas de los dispositivos (Update Tools in Devices) y actualizar los objetos de aprendizaje de los dispositivos (Update Los in Devices) para poder personalizar los dispositivos de los usuarios (LO Configuration Device Management).

NODO MÓVIL

En los Nodos Móviles residirán los siguientes agentes micro-reactivos:

- *DeviceAgent*. Mantiene la información local del dispositivo acerca del usuario y sus actividades, y se encarga de instalar y desinstalar artefactos, como herramientas u objetos de aprendizaje.

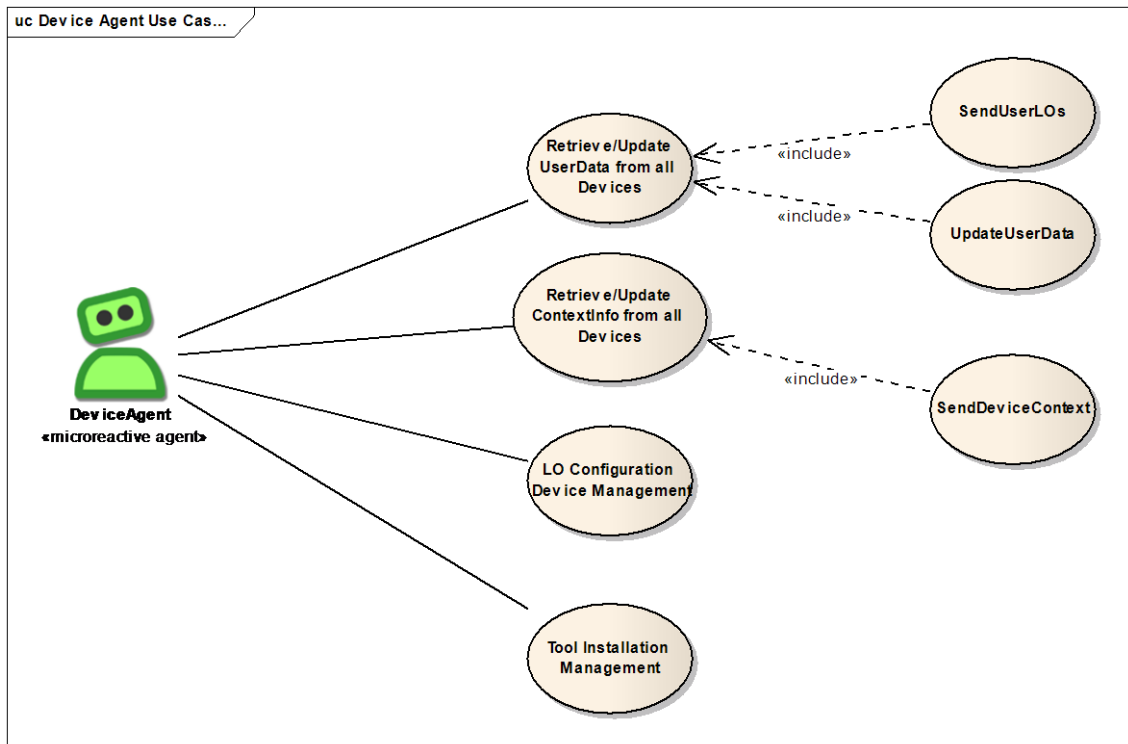


Figura 9.5 Responsabilidades de DeviceAgent

Se encarga de contactar con cada dispositivo y extraer toda la información acerca del dispositivo y el usuario que lo está utilizando, y transmitirle información para su configuración, que se corresponde con S1-R3, S2-R2 y S3-R2 de la Tabla 9.1.

- *CollaborationAgent*. Mantiene información acerca de los usuarios que se encuentran en el entorno y las actividades que éstos están desarrollando, con el propósito de buscar posibles colaboraciones para el usuario del dispositivo.

Se encarga de contactar con otros Agentes de Colaboración y extraer toda la información acerca de los usuarios relevantes o del entorno, para poder proponer colaboraciones para participar en actividades con otros usuarios, regular las actividades y configurar actividades espontáneas. Todo esto se corresponde con S4-R1, S4-R2, S5-R1, S6-R1 y S6-R2 de la Tabla 9.1.

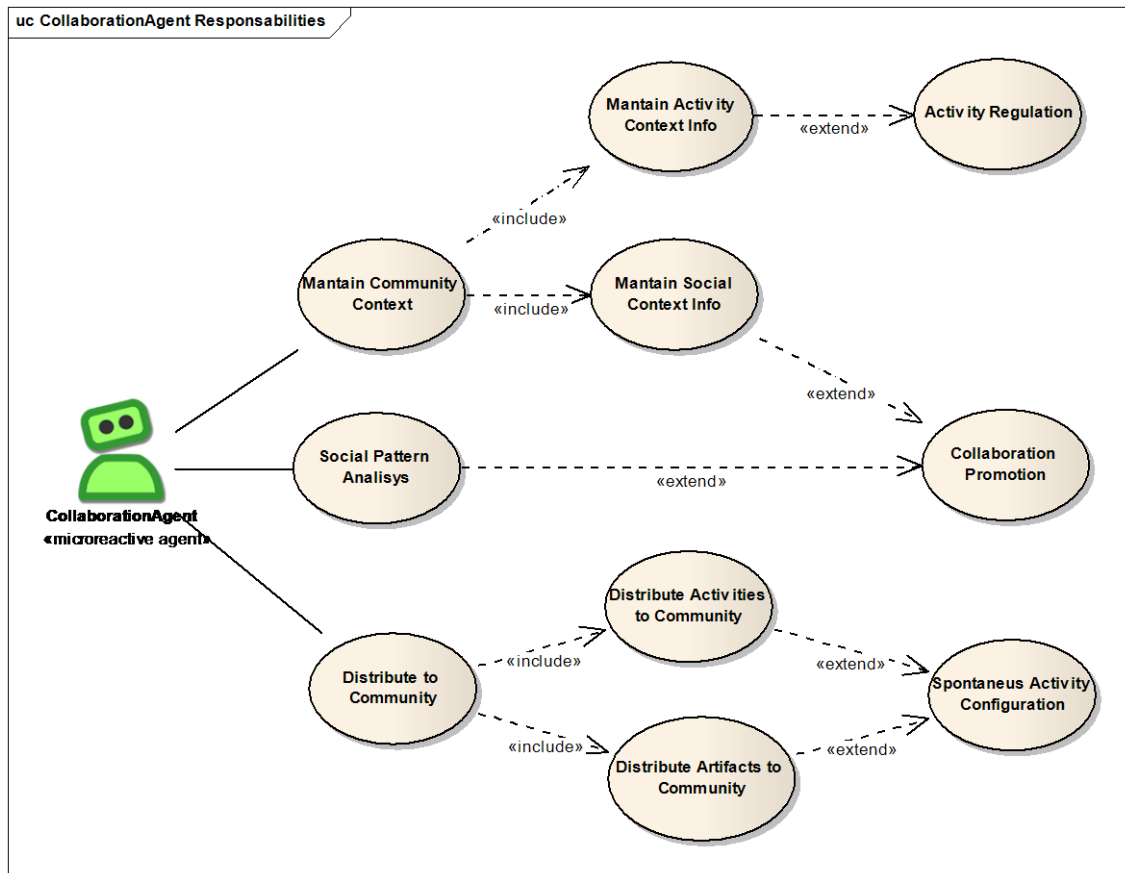


Figura 9.6 Responsabilidades de CollaborationAgent

- *PersonalAgent*. Se encarga de conocer el estado del alumno y decidir cuándo interrumpirle con preguntas, recomendaciones o mensajes proporcionados por otros agentes. Además se encarga de la comunicación con el usuario, mandándole mensajes a través de una interfaz de usuario. Este tipo de agente se corresponde por tanto también con lo que se ha denominado Agente de Interacción.

Sus distintas responsabilidades se corresponden con S2-R3, S4-R3, S5-R2 y S6-R3 de la Tabla 9.1.

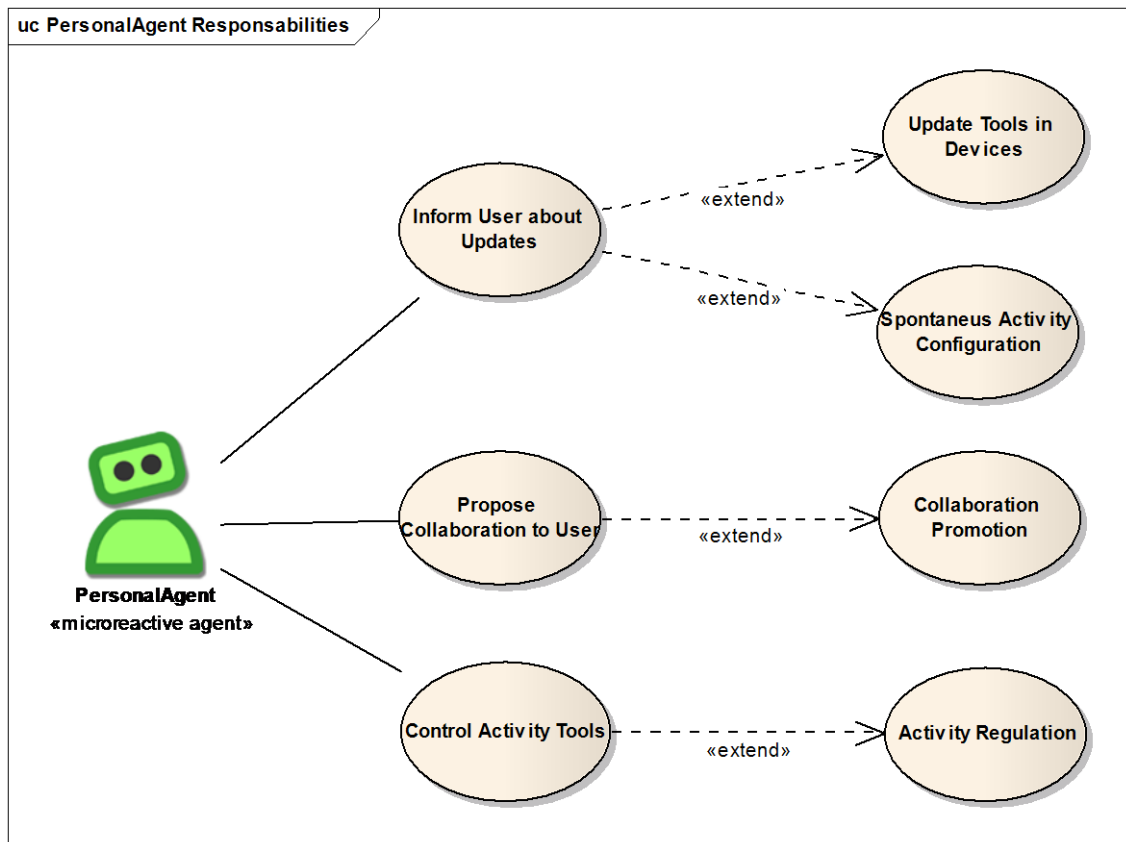


Figura 9.7 Responsabilidades de Personal Agent

9.2.2 Recursos para el escenario de ENLACE

Como se ha comentado anteriormente, los recursos de organización son entidades computacionales que ofrecen funcionalidades a los agentes para poder realizar acciones en busca de sus objetivos. Los recursos de organización ofrecen acceso a los agentes a recursos propios del sistema, como bases de datos locales, interfaces de usuario, etc., y también a recursos externos que no pertenecen al sistema. Si el recurso es externo, como por ejemplo un repositorio de objetos de aprendizaje, se realiza una aproximación de recubrimiento (*wrapper*) que por un lado ofrece comunicación con éste y por otro desacopla la lógica interna del recurso de los agentes.

NODO SERVIDOR

Dentro del Nodo Servidor se localizan:

- *LOR Wrapper*. Es utilizado por los agentes pedagógicos para acceder un repositorio de objetos de aprendizaje externo, concretamente dentro del escenario ENLACE al repositorio E-LOR. Aporta información sobre los objetos de aprendizaje disponibles, permitiendo su almacenamiento, búsqueda y recuperación.
- *LMS Wrapper*. Es utilizado por los agentes pedagógicos para acceder un LMS externo, concretamente dentro del escenario ENLACE al LMS Pelican. Aporta información social y de actividad sobre usuarios, grupos y tareas.
- *Context Database*. Sirve para almacenar la información contextual que va recabando el agente *GlobalContextAgent*.

NODO MÓVIL

Dentro del dispositivo móvil:

- *Tool Database*. Se trata de un recurso propio del dispositivo móvil para poder gestionar las herramientas instaladas en el dispositivo.
- *LO Database*. Otro recurso propio del dispositivo móvil pero esta vez para poder gestionar los objetos de aprendizaje instalados en el dispositivo.
- *Local Context Database*. Sirve para almacenar la información contextual de forma local en el dispositivo, para poder recuperarla por los agentes micro-reactivos en cualquier momento, aunque no haya comunicación con otros nodos.
- *Mobile User Interface*. Aporta mecanismos para controlar la interfaz de usuario de los dispositivos móviles, permitiendo a los agentes *Personal Agent* interactuar con los usuarios.

Por otro lado, un recurso común que está presente en todos los nodos de la arquitectura, sin importar qué tipo de nodo, es el recurso de comunicación, anteriormente comentado en el apartado 5.4 dentro del Nivel de Infraestructura. El recurso de comunicación habilitará el envío y recepción de mensajes y eventos a todos los agentes de los distintos nodos del escenario ENLACE.

9.3 Puesta en Marcha: Salida al Monte del Pardo

El prototipo correspondiente al escenario de ENLACE comentado en este capítulo ha sido implementado y probado en dispositivos reales. En concreto, para el Nodo Servidor se ha empleado una máquina Linux que alberga tanto los recursos externos de LOR y LMS, E-LOR y Pelican en este caso, respectivamente. Por otro lado, las PDAs (Nodos Móvil) que se han utilizado para las pruebas son del modelo HP 310, con 128Mb de memoria RAM y una versión de la máquina virtual Java llamada J9 desarrollada por IBM para dispositivos basados en Windows Mobile.

En términos de implementación el lenguaje de programación empleado ha sido Java, tanto para los Nodos Móvil como el Nodo Servidor, y una característica notable ha sido la adaptación de ICARO-T para poder funcionar sobre el estándar OSGi (OSGi, 1999). Para obtener más detalles acerca de la implementación de DAEDALUS y de los diagramas de reglas y máquinas de estado de los agentes, es posible consultarlos en el Apéndice A y el Apéndice C.

9.3.1 Arranque del Nodo Servidor

El proceso de arranque del Nodo Servidor se realiza de la manera estándar de una organización de DAEDALUS y se gestiona tal y como se explicó en el apartado 5.2. Es decir, una vez que el usuario (administrador) arranca la plataforma se generan los recursos de infraestructura esenciales, como por ejemplo el de comunicación, para posteriormente iniciar los agentes gestores. Los agentes gestores serán los encargados de poner en marcha a los agentes cognitivos y los recursos de la organización presentes en el Nodo Servidor.

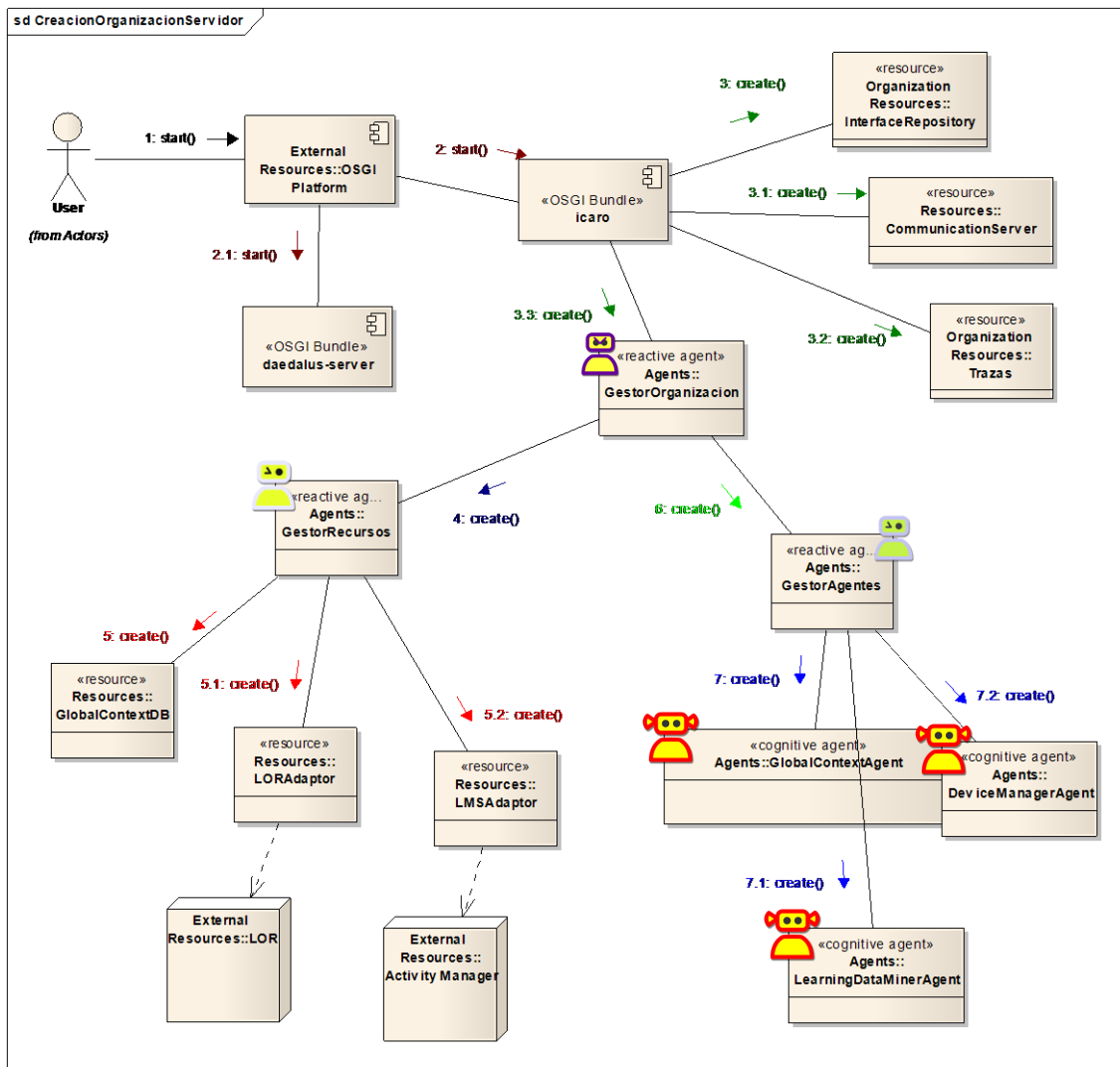


Figura 9.8 Arranque del Nodo Servidor

9.3.2 Arranque del Nodo Móvil

De la misma forma, cuando se arranca DAEDALUS desde una PDA, se produce el mismo proceso en el cual los agentes gestores crean a los agentes micro-reactivos especialistas y a los recursos de organización locales de la PDA.

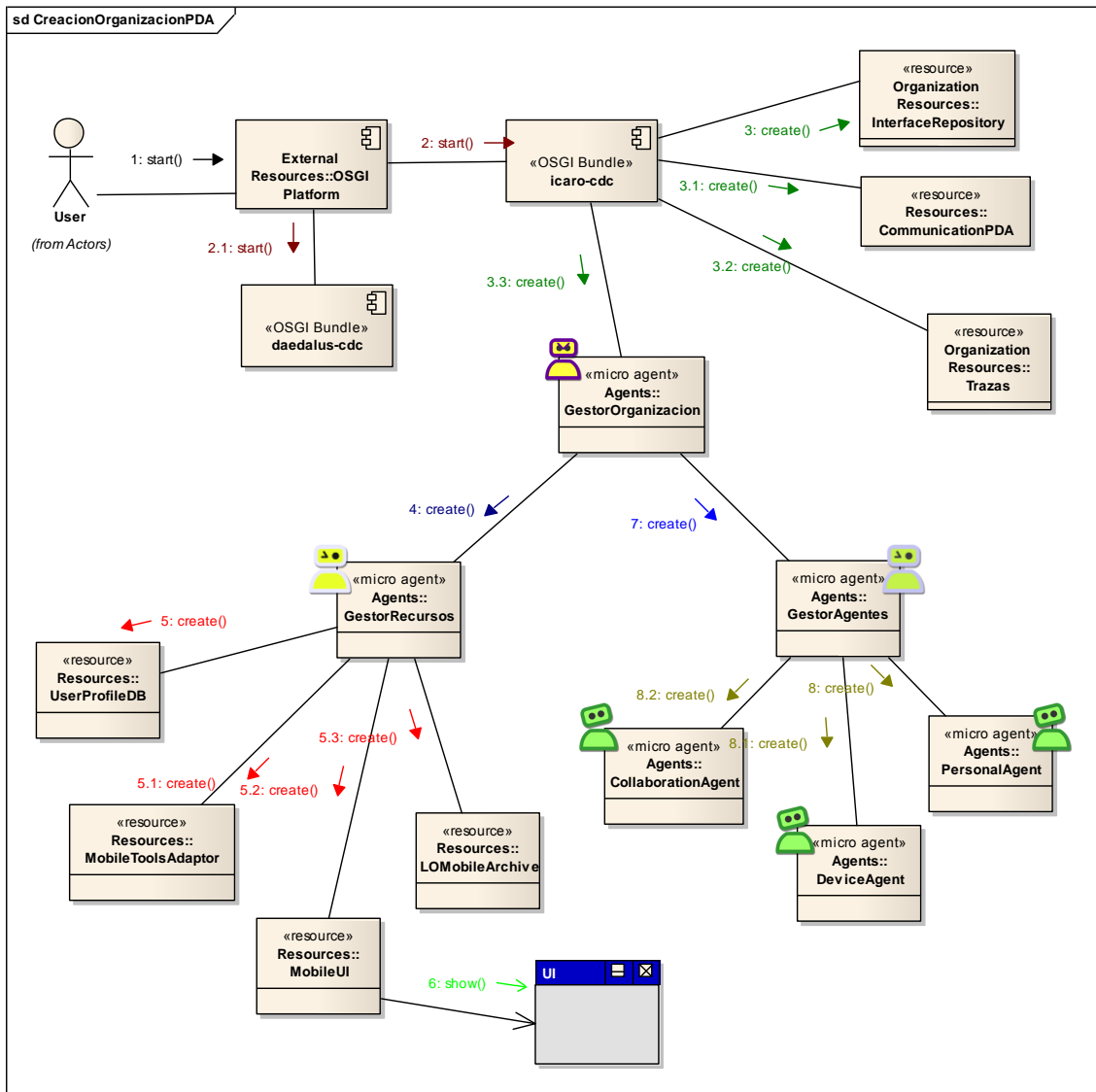


Figura 9.9 Arranque del Nodo Móvil

No obstante, lo que verá el usuario es la interfaz gráfica que se le presenta fruto de la creación del recurso de interfaz de usuario por parte de los agentes de gestión. De esta forma, el sistema permite una serie de funcionalidades básicas a los estudiantes como son la autenticación dentro del sistema mediante una pantalla de identificación (Figura 9.10).

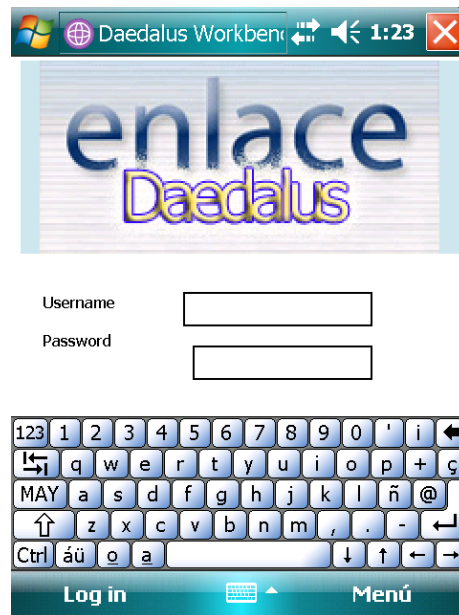


Figura 9.10 Interfaz de Usuario de Inicio

La interfaz gráfica esta generada por el recurso *MobileUserInterface* que es controlado por el agente *PersonalAgent* de la PDA. Este agente presentará mediante esta interfaz gráfica toda la información necesaria al usuario, como por ejemplo acerca de las actividades que tiene disponibles en función del contexto (Figura 9.11).

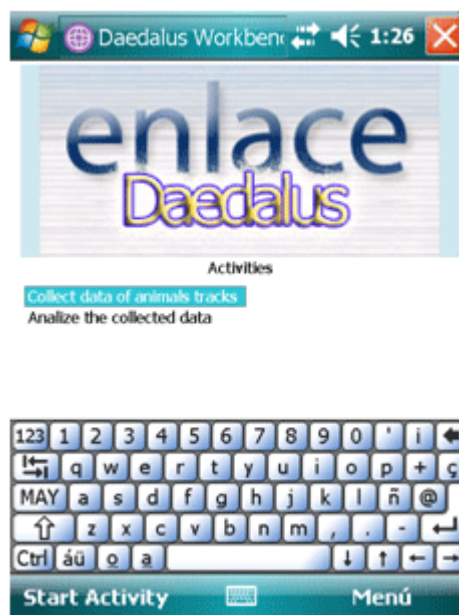


Figura 9.11 Interfaz de Información al Usuario

Una vez que el usuario selecciona una actividad se le presenta la información de esta incluidas, como por ejemplo dar acceso a las herramientas que puede utilizar en un determinado momento (Figura 9.12).

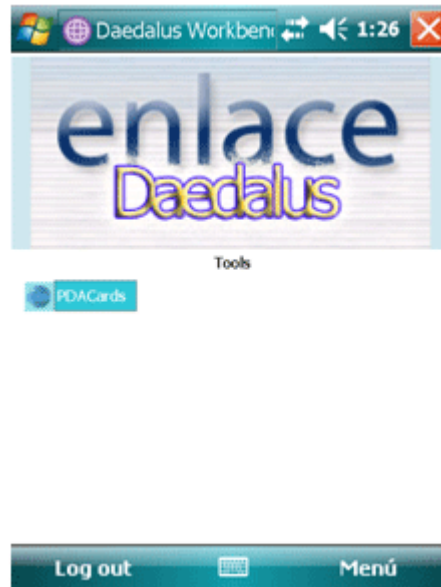


Figura 9.12 Interfaz de Información de recursos para la actividad

El recurso *MobileUserInterface* dispone de todo un conjunto de operaciones para poder mostrar las distintas pantallas de interfaz según sean requeridas por el agente *PersonalAgent*. Este recurso también dispone de operaciones para mostrar mensajes de información y poder pedir la introducción de datos por parte del usuario.

9.3.3 Caso de Prueba 1: Administración de la infraestructura distribuida

Tal y cómo han sido diseñados los agentes pedagógicos buscarán mantener la sincronización de información entre dispositivos, gestionar la instalación de herramientas pedagógicas para poder realizar las actividades y la configuración personalizada de recursos educativos. A continuación mostraremos cómo se realizan estos procesos en este escenario.

INSTALACIÓN DE HERRAMIENTAS EN DISPOSITIVOS

El proceso de instalación de herramientas comienza cuando el agente *GlobalContextAgent* descubre que es necesaria la instalación de una herramienta en un determinado dispositivo. Esto se produce cuando dicho agente cognitivo posee un conjunto determinado de creencias. En concreto tiene que existir para el agente un Nodo (Node) asociado a un Dispositivo (Device), una Persona (Person) registrada al Dispositivo, una Actividad (Activity) relacionada con la Persona, y finalmente una Herramienta (Tool) requerida para la Actividad y que no esté ya instalada. Toda esta cadena lógica dará como resultado la realización de una serie de tareas para procesar el descubrimiento de la necesidad de instalación.

A modo ilustrativo, en la Tabla 9.2 se muestra una de las reglas de desarrollo del objetivo *DeriveContextInfo* del agente *GlobalContextAgent* que concretamente contiene la lógica que determina esta ejecución de tareas.

```

rule "Resolucion Obj Derive ContextInfo - Check device needs install tool"
when
  obj:DeriveContextInfoGoal(state==Goal.SOLVING)
  Focus(foco:foco->(obj==foco))
  $d:Device( $dId:id )
  $n:Node( $nId:id )
  ContextStatement( subject.id==$dId, predicate.label=="locatedIn", object.id==$nId )
  $p:Person( $pId:id )
  ContextStatement( subject.id==$pId, predicate.label=="loggedIn", object.id==$dId )
  $a:Activity( $aId:id )
  ContextStatement( subject.id==$uId, predicate.label=="engagedIn", object.id==$aId )
  $t:Tool( $tId:id )
  ContextStatement( subject.id==$aId, predicate.label=="requires", object.id==$tId )
  not( (exists ContextStatement( subject.id==$tId, predicate.label=="installedIn",
                                object.id==$dId ))
        or (exists ContextStatement( subject.id==$dId,
                                    predicate.label=="needs", object.id==$tId )) )
then
  Task tarea = gestorTareas.crearTarea(ProcessDeviceNeedsTool.class);
  tarea.ejecutar($d, $t, $n);
  Task tarea2 = gestorTareas.crearTarea(StoreRDF.class);
  tarea2.ejecutar($d, "needs", $t);
end

```

Tabla 9.2 Regla de Desarrollo del objetivo *DeriveContextInfo* del *GlobalContextAgent*

La tarea *ProcessDeviceNeedsTool* será la que provoque la comunicación con el agente *DeviceManagerAgent* y se inicie el proceso que se muestra en la Figura 9.13.

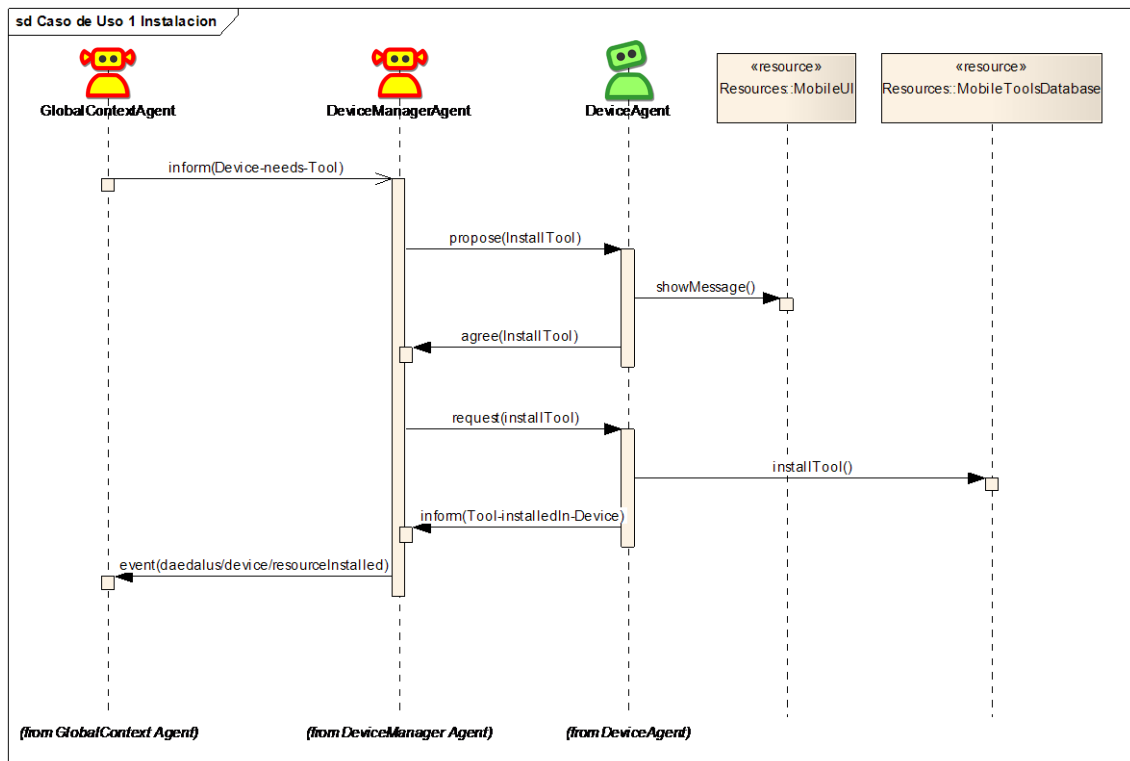


Figura 9.13 Instalación de herramientas en dispositivos

El agente *DeviceManagerAgent* se pondrá en contacto con el agente *DeviceAgent* del dispositivo en cuestión, quien propondrá al usuario a través de un mensaje a la interfaz de usuario *MobileUI* si está de acuerdo con la instalación. De ser así, nuevamente el agente *DeviceManagerAgent* se comunicará con el agente *DeviceAgent* para enviarle el código de la herramienta a instalar.

Este último agente realizará la instalación local de la herramienta haciendo empleo del recurso *MobileToolsDatabase*. Si todo resulta correcto, se informará al *DeviceManagerAgent* de ello, que a su vez finalmente enviará un evento `daedalus/device/resourceInstalled` a todos los agentes interesados de mantener el estado del contexto.

CONFIGURACIÓN PERSONALIZADA DE DISPOSITIVOS

El proceso de configuración de los dispositivos (Figura 9.14) resulta análogo al anterior de instalación de herramientas. Nuevamente el agente de contexto *GlobalContextAgent* será el encargado de derivar información a partir de su conocimiento de contexto para descubrir si puede resultar necesario el

tener disponible en algún dispositivo determinados artefactos, atendiendo por ejemplo a las anteriores y futuras actividades a realizar por parte del alumno.

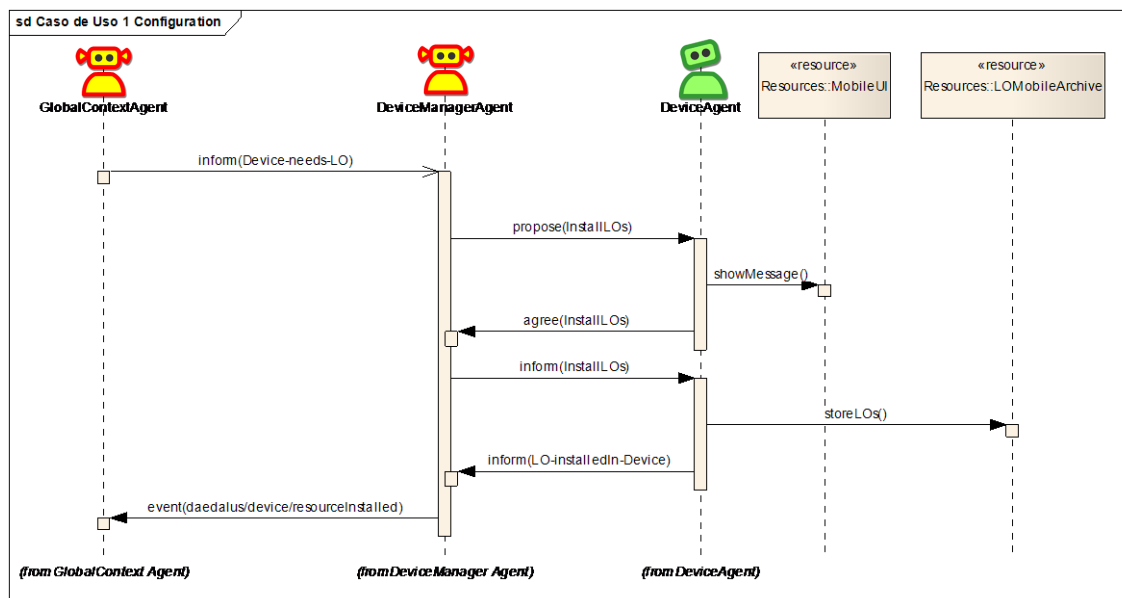


Figura 9.14 Configuración personalizada de dispositivos

El agente *GlobalContextAgent* informa al agente *DeviceManagerAgent* de qué objetos de aprendizaje son necesarios en qué dispositivo. A partir de ahí se vuelve a establecer comunicación con el agente *DeviceAgent* del dispositivo concreto que tramitará la confirmación de configuración con el usuario a través del recurso *MobileUI*, y realizará también el almacenamiento en el repositorio local del dispositivo móvil de los distintos objetos de aprendizaje, para que se encuentren disponibles para las actividades.

Finalmente se enviará también en este caso por parte del agente *DeviceManagerAgent* un evento `daedalus/device/resourceInstalled` a todos los agentes interesados de mantener el estado del contexto.

SINCRONIZACIÓN DE INFORMACIÓN DE DISPOSITIVOS

El proceso de sincronización de información de dispositivos (Figura 9.15) permite la recuperación de información de los alumnos en las actividades tan pronto como un Nodo Móvil entra en contacto con el Nodo Servidor cuando existe conexión entre ellos. Este proceso se pone en marcha al recibir un evento de tipo `daedalus/node/connected` el agente *DeviceManagerAgent*.

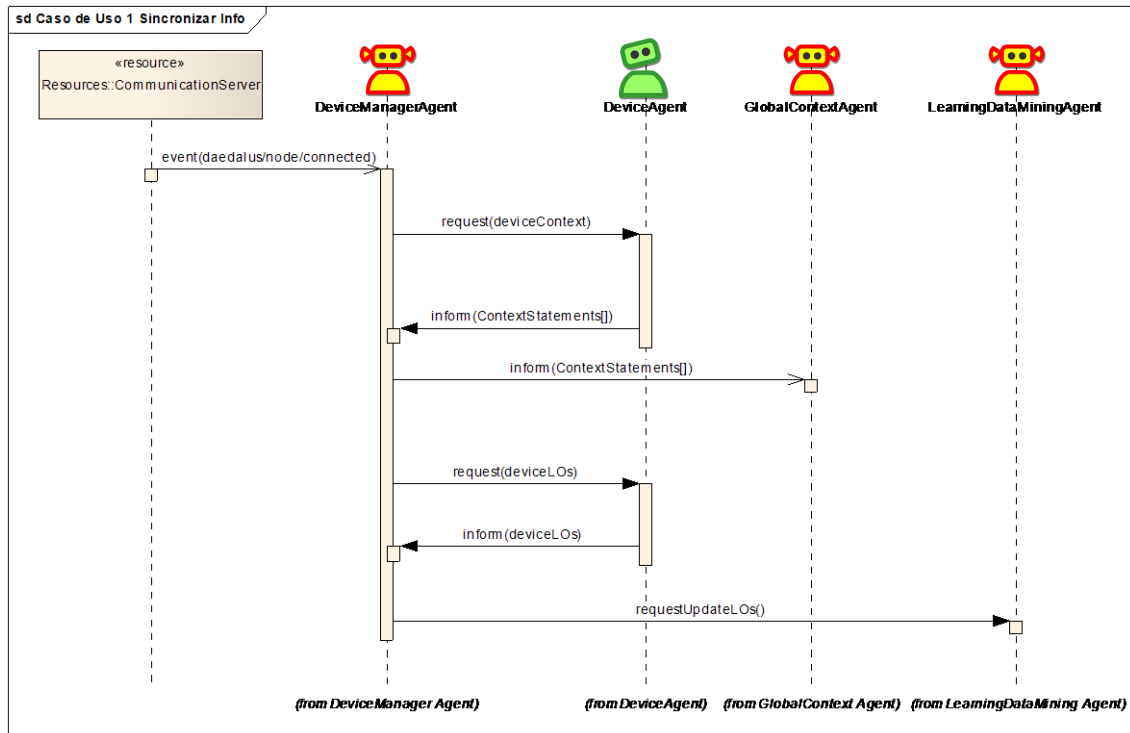


Figura 9.15 Sincronización de Información de Dispositivos

Este agente *DeviceManagerAgent* establecerá un dialogo con agente *DeviceAgent* para primeramente recuperar la información de contexto existente dentro del dispositivo, con el fin de poder trasmitírsela al agente *GlobalContextAgent* para sincronizarla con el contexto global que mantiene dicho agente, y posteriormente recuperar los objetos de aprendizaje creados por los alumnos durante las actividades. Estos objetos de aprendizaje serán transmitidos al agente *LearningDataMiningAgent* para almacenarlos dentro del LOR.

9.3.4 Caso de Prueba 2: Fomento contextualizado de la colaboración

Tal y como ha sido diseñado, el principal responsable de fomentar la colaboración durante las actividades es el agente *CollaborationAgent*. Este agente de forma reactiva a través de los eventos que reciba, se preocupará de estar en contacto con agentes homólogos suyos en otros dispositivos para mantener información actualizada de los alumnos de alrededor (Figura 9.16).

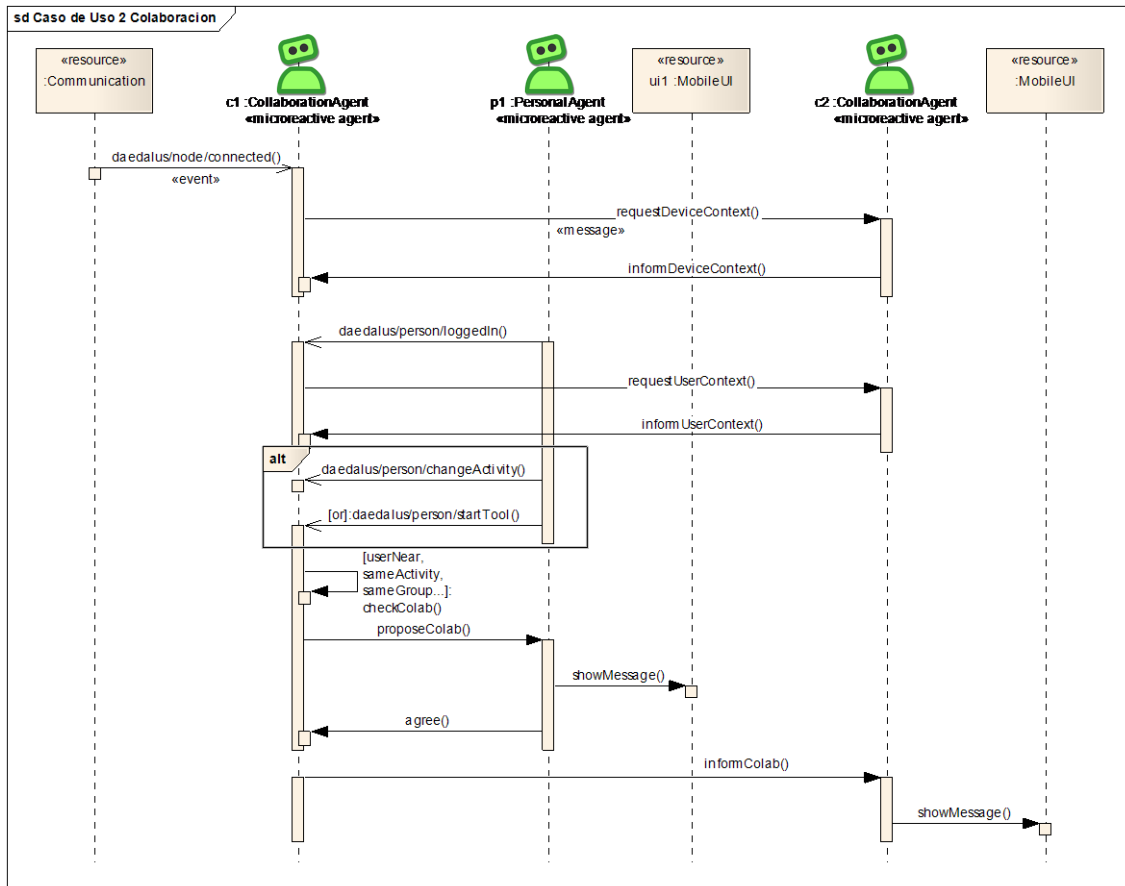


Figura 9.16 Fomento contextualizado de la colaboración

En cuanto un agente *CollaborationAgent* recibe un evento de nodo descubierto `daedalus/node/connected`, de forma reactiva pedirá información de contexto acerca del dispositivo asociado en dicho nodo al agente *CollaborationAgent*. De la misma forma, en cuanto recibe un evento `daedalus/person/loggedIn`, el agente vuelve a pedir información pero esta vez referente al usuario recién registrado. Esta información se irá actualizando al recibir más eventos relativos a los usuarios. Cada vez que esto sucede el agente *CollaborationAgent* realiza una serie de comprobaciones predefinidas, que pueden establecer que por ejemplo cuando otro usuario se encuentre cerca o que sea de su mismo grupo y tenga que realizar una misma actividad se informe al usuario local con un mensaje de proposición de colaboración a través del recurso *MobileUI*. Si éste acepta, se comunicará esta intención de colaboración al *CollaborationAgent* remoto que hará lo propio mostrando esta proposición de colaboración al otro alumno.

9.3.5 Caso de Prueba 3: Regulación y Control sobre la Actividad

Otro servicio que se presta dentro del escenario ENLACE es la regulación y control de las actividades. Cuando un usuario con privilegios, como por ejemplo un monitor durante una salida al campo, lo desea puede controlar el desarrollo de la actividad en los dispositivos de los alumnos para evitar por ejemplo que éstos se despisten con sus PDAs cuando el monitor está hablando y quiere que se le preste atención.

A tal efecto, según se muestra en la Figura 9.17, el agente *PersonalAgent* del monitor es capaz de enviar un evento de pausa de la actividad `daedalus/activity/pause` que será recibido por todos los agentes *PersonalAgent* de los alumnos. En cuanto esto ocurra, se mostrará un mensaje a través de la interfaz de usuario que indica que la actividad ha sido parada por el monitor. En cuanto se vuelva a recibir un evento de reanudación `daedalus/activity/continue` o pase un tiempo preestablecido, el agente *PersonalAgent* del alumno volverá a mostrar la interfaz de trabajo y las herramientas para poder seguir trabajando.

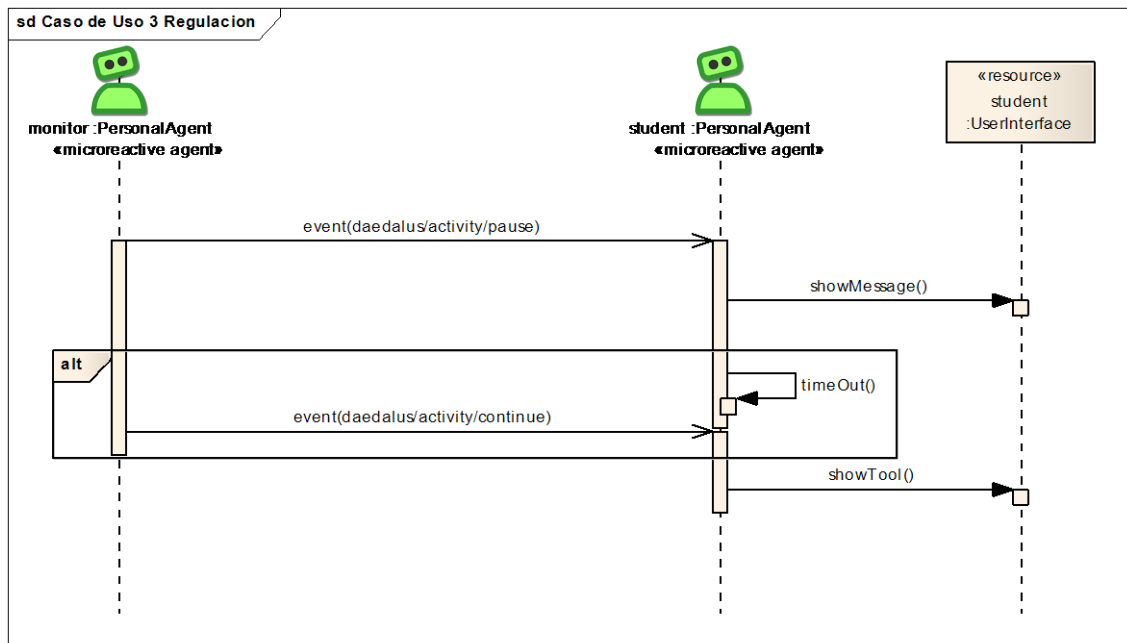


Figura 9.17 Regulación y Control sobre la Actividad

9.3.6 Caso de Prueba 4: Reconfiguración eventual durante Actividades

Por último, el agente *CollaborationAgent* también será responsable de la reconfiguración debido a la creación de una actividad espontanea, como por ejemplo si el monitor decide crear durante una salida al campo una nueva actividad de búsqueda de restos de aves carroñeras al descubrir que hay muchas por la zona visitada.

Una vez que el monitor introduce la descripción de la nueva actividad, el agente *CollaborationAgent* se pondrá en contacto con los demás agentes *CollaborationAgent* de los dispositivos de los alumnos. En cuanto esto ocurre, se transmite la información de la nueva actividad y adicionalmente los recursos necesarios para su realización. Finalmente se informa al alumno y se le propone trabajar en el desarrollo de esta nueva actividad.

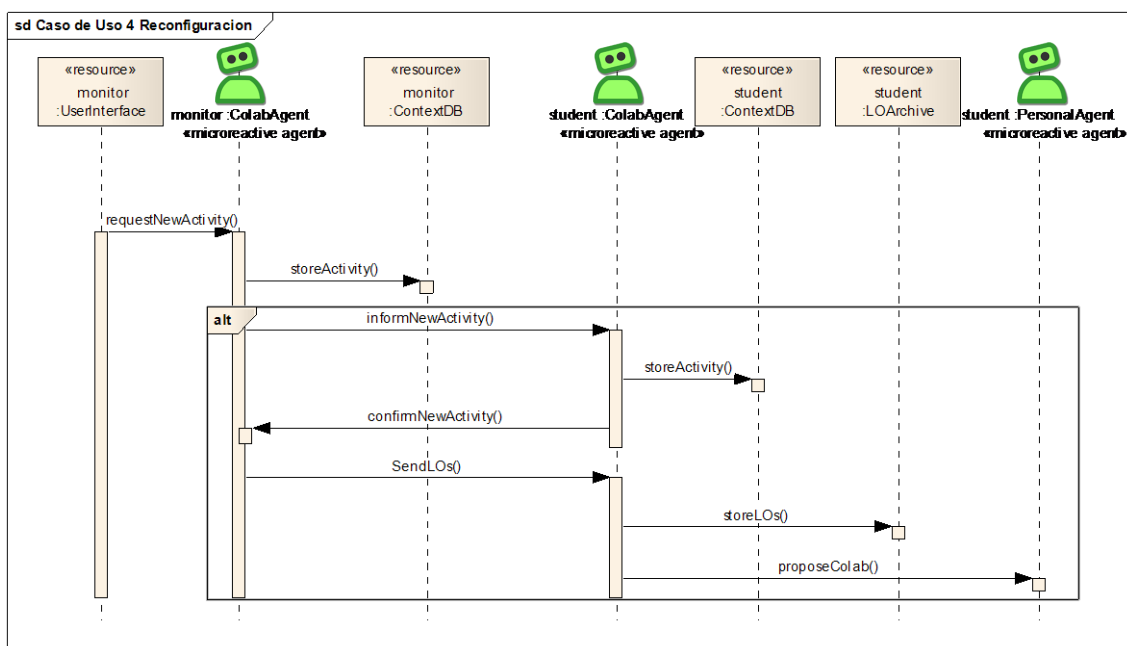


Figura 9.18 Reconfiguración eventual durante Actividades

10

LA

ARQUITECTURA

DAEDALUS FRENTE A UN

NUEVO ESCENARIO

Una vez vista la aplicación de la arquitectura DAEDALUS al escenario de ENLACE, pondremos nuevamente a prueba nuestra aproximación para definir a un nuevo espacio de aprendizaje ubicuo. Mediante un nuevo escenario de aplicación, pretendemos demostrar la adaptabilidad de nuestra arquitectura y probar su extensibilidad frente a nuevos requisitos. Esto resulta de gran importancia a la hora de poder aplicar nuestra solución propuesta a nuevos casos reales.

El escenario que se presenta en este capítulo es fruto de la estancia breve que tuvo lugar en la Universidad de Toronto. Para implantar este escenario se empleó la arquitectura DAEDALUS, permitiendo diseñar e implementar un aula inteligente compuesta por múltiples pantallas y dispositivos integrados y distribuidos con el fin de dar soporte a actividades pedagógicas en el dominio de las matemáticas. En base a la síntesis y abstracción de los procesos y metodología en la aplicación de DAEDALUS, en el capítulo final se presentará una discusión acerca de las recomendaciones de uso, adaptación y extensibilidad para la implantación de futuros escenarios con esta arquitectura.

10.1 Definición del Escenario 'Aula Inteligente'

En la primavera de 2009, el autor de esta Tesis realizó una estancia de tres meses en la Universidad de Toronto, dentro del OISE (*Ontario Institute for Studies in Education*). El grupo de investigación con el que allí se colaboró tenía mucho interés tanto en el estudio como en el desarrollo de nuevos escenarios pedagógicos dentro de situaciones educativas formales pero caracterizadas por una alta presencia de componentes tecnológicos con los cuales dar apoyo al aprendizaje. Más concretamente, el trabajo que se realizó allí fue el de dar forma a un entorno dentro de la escuela que contase con multitud de dispositivos, tanto móviles como embebidos en el espacio, y que pudiese reaccionar y adaptarse al contexto de los estudiantes de forma dinámica e inteligente. Todas estas eran premisas apropiadas para la aplicación de la arquitectura DAEDALUS dentro de este posible escenario pedagógico, con el fin de buscar un acercamiento que diese cohesión a los elementos distribuidos en el espacio, que hiciese un uso del contexto y emplease un diseño orientado a agentes para facilitar el desarrollo del mismo.

Aunque bien es cierto que este escenario no estaba caracterizado por una alta movilidad ni de sus dispositivos ni de sus usuarios, ya que se limitaba a un recinto cerrado de unos 40m², los rasgos principales constituidos por la necesidad de interconexión de componentes dependientes del contexto colocados por todo el recinto abogaban por una implantación distribuida. La arquitectura DAEDALUS podía brindar una solución distribuida práctica y fácil de implementar.

El *espacio de aprendizaje ubicuo* donde iban a tener lugar los procesos pedagógicos de los estudiantes fue apodada ‘Aula Inteligente’ (*Smart Classroom*). En concreto el objetivo propuesto para esta aula era conseguir un espacio inteligente, compuesto por multitud de dispositivos tecnológicos con interconexión unos con otros, donde poder albergar distintas áreas o contextos temáticos para el aprendizaje en grupos.

Durante los meses de estancia se trabajó tanto en el diseño de la actividad pedagógica como en sus detalles técnicos. Toda la instalación del aula, diseño de la actividad, de la infraestructura, e implementación tuvo lugar en apenas dos meses y medio entre Marzo y Mayo de 2009 (véase la Figura 10.1). Gracias a los mecanismos de reutilización y extensibilidad a los que ha sido orientado el desarrollo de la arquitectura DAEDALUS, todos estos procesos se vieron acelerados, pudiendo realizarse en el breve lapso de tiempo disponible un prototipo funcional y llevar a cabo varias pruebas de concepto con usuarios reales.

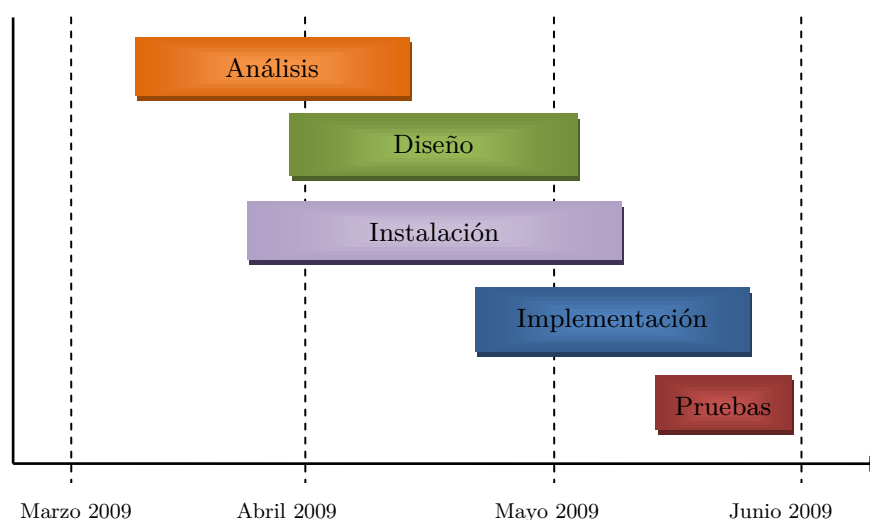


Figura 10.1 Evolución del Desarrollo del Escenario

Durante el análisis del escenario se plantearon diversas estrategias para abarcar el problema, a la vez que se emplearon técnicas de ‘*brainstorming*’ para buscar funcionalidades y siempre se mantuvo una negociación en todo momento entre los usuarios participantes incluyendo un profesor de matemáticas, los investigadores pedagógicos, los programadores y alumnos que colaboraron en la instalación de los elementos tecnológicos del aula. A lo largo de las distintas fases del desarrollo del escenario se produjo un solapamiento re-

iterativo de dichas fases que provocaba una retroalimentación en las fases solapadas. De esta forma, mientras que se concretaban las funcionalidades para el escenario en el análisis, se procedía paralelamente a la instalación de los dispositivos tecnológicos en el aula y de forma simultánea se iba dando forma a la actividad pedagógica. Cuando la infraestructura se encontró suficientemente definida y el diseño de la actividad estaba bastante maduro, se comenzó con la implementación de la Organización Educativa de agentes y recursos, ampliando de forma pertinente el Dominio de Contexto y la Taxonomía de Eventos según las necesidades específicas. Finalmente, se realizaron varias pruebas piloto con usuarios reales.

10.1.1 Descripción del escenario

Los alumnos de 14 años de la escuela UTS de Toronto van a realizar varias sesiones de trabajo con el fin de afianzar y profundizar en los conocimientos que han adquirido en sus clases de matemáticas. Estas sesiones las van a realizar de una nueva manera especial e innovadora, trabajando en un 'Aula Inteligente'.

El 'Aula Inteligente' que se ha instalado en su escuela es un aula dentro del edificio de su colegio y que está compuesta por un gran número de componentes interconectados que son capaces de dar apoyo al aprendizaje de forma contextual e inteligente. El aula cuenta con diversas pantallas y proyectores de gran tamaño cubriendo todas las paredes de la habitación, ordenadores servidores con *arrays* de tarjetas gráficas instaladas para controlar dichas pantallas, y una red inalámbrica que permite conectar dispositivos móviles y portátiles dentro del aula.

Los estudiantes entrarán en el 'Aula Inteligente' con el objetivo de trabajar con una serie de problemas matemáticos que tendrán que clasificar, resolver y relacionar sus soluciones de acuerdo a distintas categorías (geometría, polinomios, exponenciales...). Los problemas matemáticos son repartidos a los estudiantes de entre todo un conjunto al que previamente han contribuido distintos profesores, investigadores o los propios alumnos. Toda la actividad se realiza en grupo. Cada uno de los grupos cuenta con al menos tres miembros. Los grupos tienen que resolver los problemas en común, aunque es posible que vean distintos elementos en las pantallas de sus dispositivos personales. Los grupos también se encargan de realizar conexiones semánticas entre los problemas. De esta forma, nuevos problemas asociados aparecen a

otros grupos, que deben comentar y reflexionar sobre estas relaciones creadas por sus compañeros.

El sistema es el encargado de monitorizar y regular todo el proceso de la actividad desde distintos puntos de vista. Desde una perspectiva organizativa, el sistema busca orientar a los estudiantes en el aula, asignándoles grupos expertos, informándoles acerca de donde han de situarse, y coordinando los grupos de trabajo. Desde una perspectiva pedagógica, es el encargado de distribuir los problemas matemáticos, preguntar y hacer reflexionar a los alumnos acerca de las relaciones entre áreas funcionales, problemas y soluciones, además de identificar posibles conflictos cognitivos entre usuarios y entre grupos. Desde una perspectiva infraestructural, el sistema se encarga de la interconexión operativa y funcional de los distintos dispositivos, encamina y distribuye la información dependiendo de la situación física en el aula, habilita visualizaciones dependiendo de los dispositivos personales, de grupo y comunes a todos los usuarios del aula, controla la persistencia de los elementos educativos, tanto de los objetos de aprendizaje necesarios para la actividad, como la gestión de todos los resultados generados por los usuarios dentro de ésta.

10.1.2 Estructuración del Aula

El aula se encuentra fragmentada en varios espacios de trabajo físicos. Cada uno de estos espacios de trabajo posee una pantalla de gran tamaño en la pared con una visualización con información referente a todos los miembros pertenecientes a dicho espacio. De esta forma hay varios de estos espacios de trabajo de grupo configurados según la distribución que se observa en la Figura 10.2.

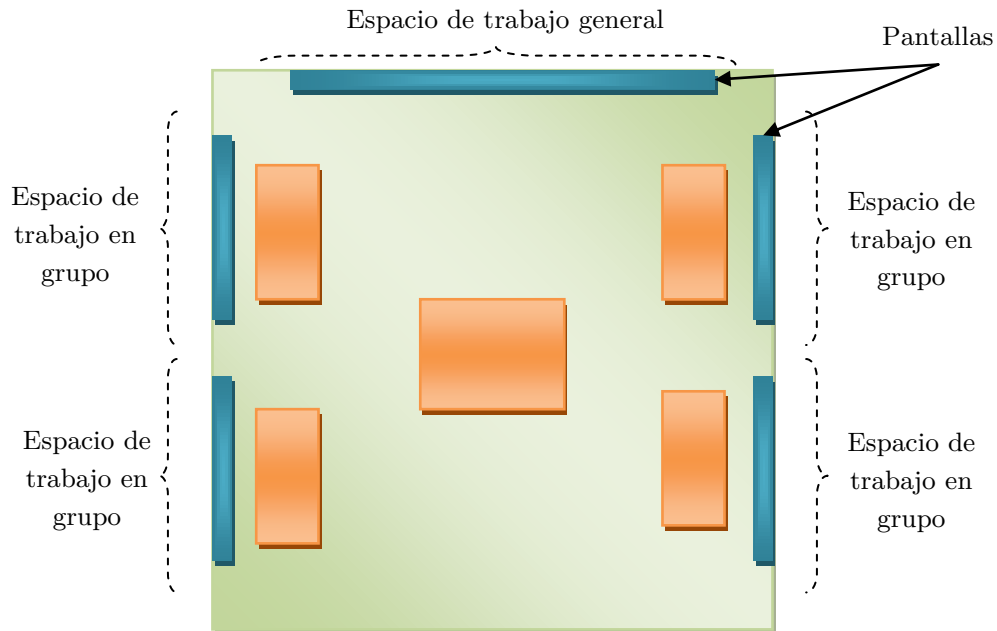


Figura 10.2 Distribución del aula

La sala se encuentra gobernada por una pantalla de mayor tamaño que representa el área de trabajo global de toda el aula y muestra en todo momento el mural de información referente a todas las tareas y sus resultados que se están produciendo en el aula. Las pantallas fueron instaladas como proyectores de corto alcance (Figura 10.3), que permiten a los usuarios acercarse a ellas sin crear sombra para así poder señalar los contenidos y en un futuro interactuar de forma táctil.



Figura 10.3 Instalación de los Projectores

Aparte de las pantallas, los alumnos cuentan con diversos dispositivos personales para la realización de las actividades. Estos dispositivos han de ser móviles, lo que posibilite el movimiento entre espacios de trabajo del aula de forma cómoda. Cada uno de estos dispositivos posee una interfaz de usuario en la que aparecerá toda la información que va dirigida al dueño del dispositivo.

Así pues, de esta forma se articulan tres niveles de trabajo en el aula mediante la tecnología: el nivel individual, soportado por los dispositivos personales, junto con los de grupo y de clase, soportados por sendos murales de información.

10.1.3 Diseño de los procesos pedagógicos e interacción

La actividad que realizarán los alumnos está dividida en distintas etapas (Figura 10.4). En cada una de estas etapas los alumnos trabajarán con distintos problemas matemáticos, tratando de clasificarlos, resolverlos y reflexionar sobre ellos.

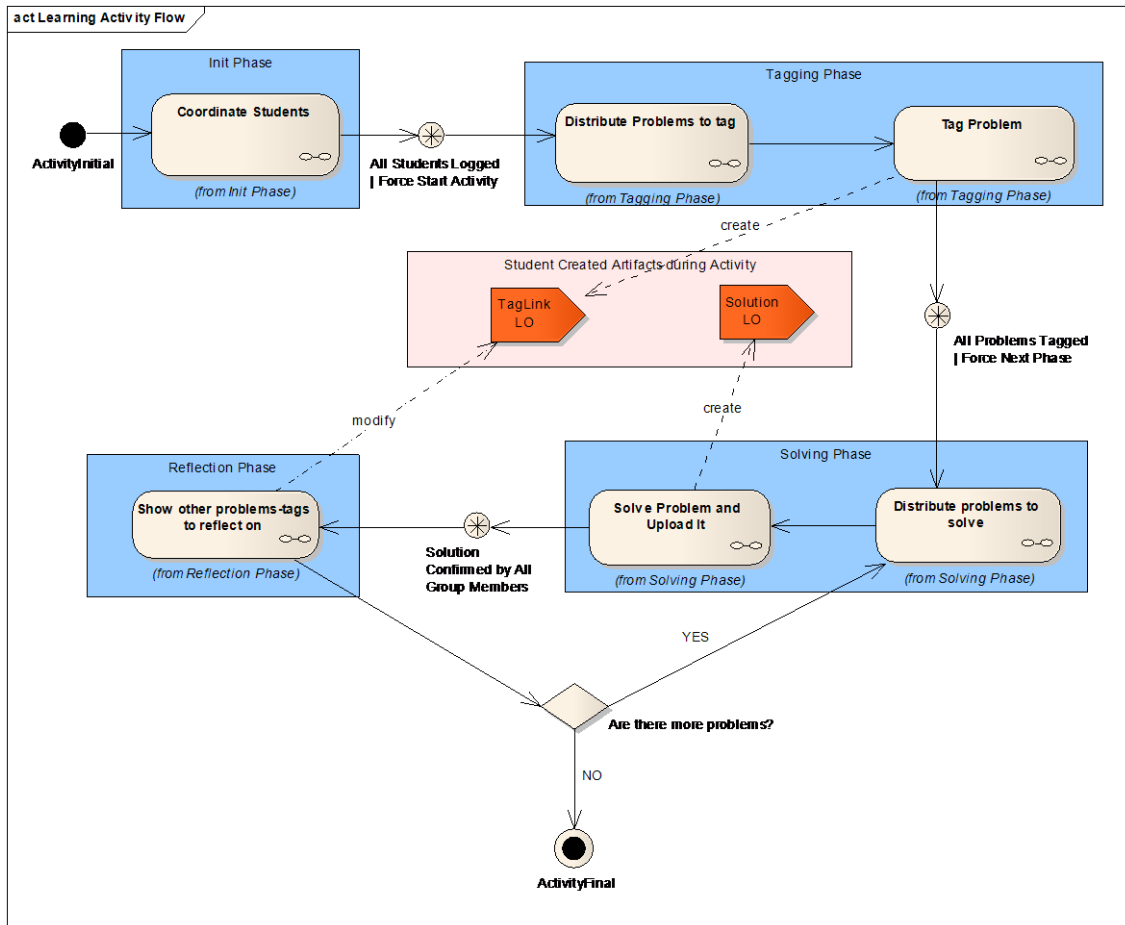
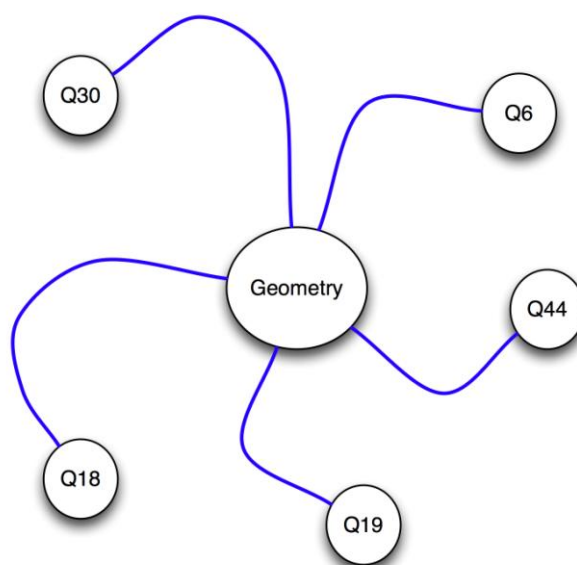


Figura 10.4 Flujo de la Actividad

La primera fase (Init Phase) comienza desde el momento en el que los usuarios entran en el aula y recogen un dispositivo personal de la mesa del centro de la habitación. El sistema es el encargado de coordinar la entrada de los alumnos en el aula. De esta forma el sistema ha de asignar a cada alumno un grupo de trabajo e informarle de su colocación dentro del aula. Cada grupo de trabajo tendrá asociado un área funcional de las matemáticas y sus integrantes asumirán el rol de expertos en esa área durante la actividad.

Una vez que todos los estudiantes están preparados y en sus puestos dentro del espacio de trabajo que les ha correspondido, el sistema comenzará a distribuir problemas matemáticos a los miembros de cada grupo para que en un primer término los identifiquen como pertenecientes a su área funcional o no. Esta fase de etiquetado (Tagging Phase) de problemas es realizada de forma cooperativa por los miembros del grupo, distribuyéndose los problemas de uno en uno a sendos dispositivos de cada alumno. Los alumnos de un mis-

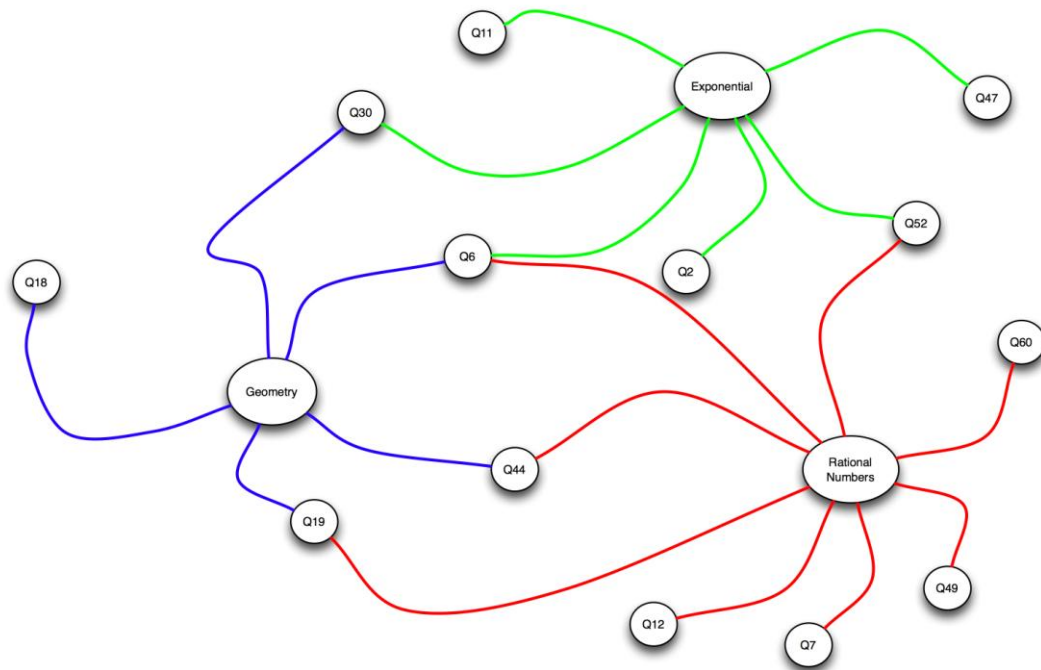
mo grupo etiquetan problemas diferentes entre ellos. La pantalla de sus dispositivos personales les mostrará el problema matemático, a modo de enunciado, junto con una pregunta similar a “¿Crees que este problema pertenece a tu grupo Geometría?”. Al responder afirmativamente a alguna pregunta el sistema establecerá un vínculo entre el área funcional y el problema, que se verá representado de forma gráfica y dinámica en la pantalla del grupo como aparece en boceto original de diseño presentado en la Figura 10.5.



Example of Connections Between Problems and Functional Area:
(In this case these questions have been tagged with "Geometry by students)

Figura 10.5 Boceto de Visualización de Grupo

A su vez, la pantalla principal también se irá actualizando en tiempo real y mostrará otra visualización en forma de grafo con las asociaciones entre problemas y categorías pero de toda la clase (Figura 10.6). De esta forma, en todo momento los usuarios, tanto alumnos como el profesor, pueden de un simple vistazo ver cuál es el estado del desarrollo de la actividad tanto a nivel de grupo como a nivel general de toda la clase.



Example of Connections Between Problems and Multiple Functional Areas:
(In this case three groups have all tagged the questions and several of questions have multiple connections - Q19, Q44, Q6, Q30, Q52)

Figura 10.6 Boceto de Visualización de Clase

Una vez que los problemas se han etiquetado por los alumnos comienza la fase de resolución (Solving Phase). El sistema buscará problemas para repartir a los grupos. Los problemas con mayor número de conexiones tendrán prioridad, por representar casos más interesantes de cara a la actividad. Los alumnos del grupo tendrán que resolver el problema por separado o en común, no obstante, tendrán que ponerse de acuerdo a la hora de responder con una solución aceptada por consenso. La interfaz de usuario de su dispositivo personal les permitirá introducir la solución, acto seguido dicha solución se mostrará a todos los miembros del grupo, que la tendrán que validar.

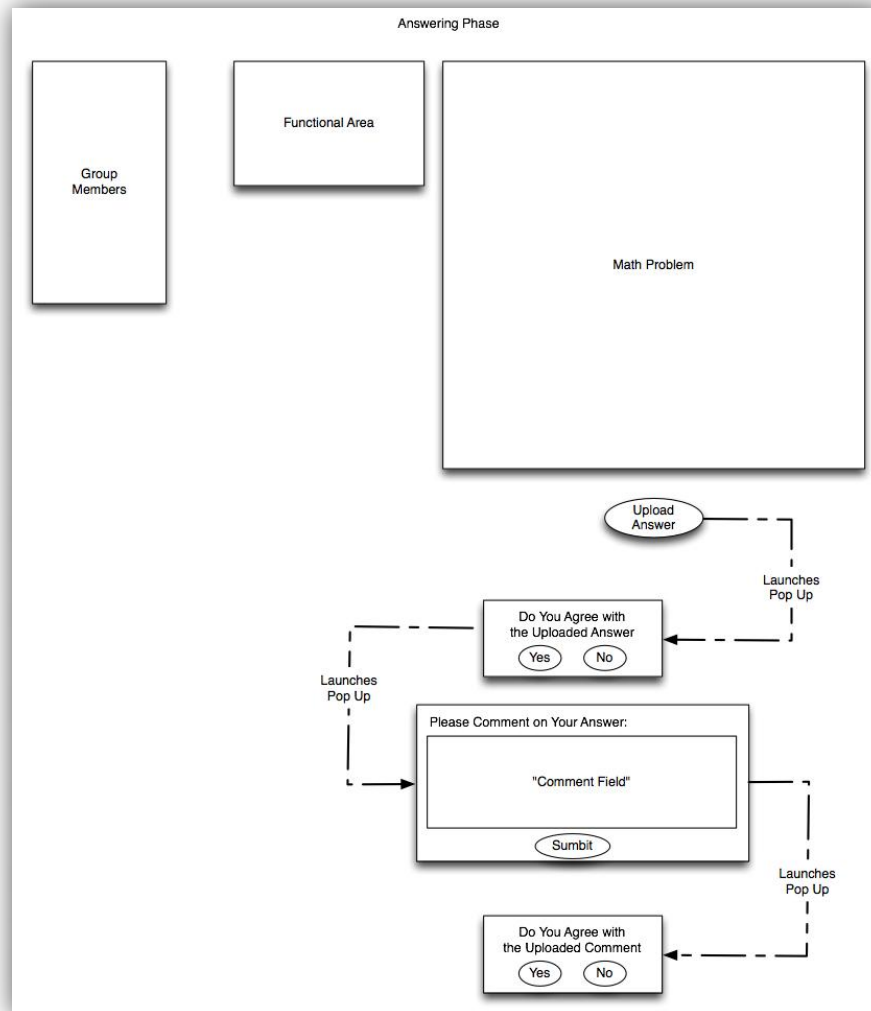


Figura 10.7 Boceto de la Interfaz de Usuario del Dispositivo Personal durante la Fase de Resolución de Problemas

En el boceto de diseño original mostrado en la Figura 10.7 se representa la distribución de elementos dentro de la interfaz de usuario, que incluye la lista dinámica de miembros del grupo (Group Members), el área funcional de matemáticas sobre la que se está trabajando (Functional Area) y un espacio reservado para mostrar el problema matemático (Math Problem). Asimismo, el boceto indica el proceso de confirmación de las imágenes subidas.

El sistema no permite continuar a la siguiente fase hasta que no exista consenso con la solución del grupo. El sistema almacenará la solución validada y la incorporará de forma gráfica a las visualizaciones de la clase y a la de los grupos que tengan definido el problema como perteneciente a su área fun-

cional. Nuevamente, los usuarios tendrán constancia en todo momento del grado de progreso de los demás miembros de otros grupos gracias a las visualizaciones. El sistema también pide al grupo un comentario acerca del proceso de resolución empleado para la solución, que se almacenará junto con la solución en sí.

Finalmente, una vez hayan contestado a un problema, el sistema les mostrará otras categorías que han sido asignadas al problema que acaban de resolver y tendrán que contestar si están de acuerdo o no y comentar. Esta nueva fase busca la reflexión por parte de los alumnos (*Reflection Phase*). Todas estas contestaciones se verán reflejadas en las visualizaciones de la clase. Eventualmente, si el sistema detecta que dos alumnos con distintas contestaciones, les puede pedir que discutan sobre ello.

Una vez que se haya terminado la reflexión sobre un problema se puede proceder a la resolución de uno nuevo o dar por terminada la actividad. Todos los resultados son almacenados como objetos de aprendizaje a lo largo de todo el proceso. Los productos llevan asociados metadatos para su posterior búsqueda y reutilización. Sin embargo, una parte importante dentro del diseño de la actividad es también la definición y especificación de estos productos.

ARTEFACTOS Y PRODUCTOS DEL ESCENARIO

Como se ha comentado anteriormente, tanto los artefactos necesarios para la realización de las actividades como los productos resultantes de estas se modelan en la arquitectura de DAEDALUS como objetos de aprendizaje. A la hora de atacar un nuevo escenario de implantación resulta imprescindible recapacitar acerca de qué tipos de artefactos educativos van a ser necesarios modelar y persistir. La definición de estos artefactos dentro del escenario tendrá repercusiones en dos planos distintos de la arquitectura. Por un lado afectará en el plano semántico, donde será necesario definir estos artefactos como conceptos dentro de la ontología y sus propiedades relacionadas, para poder permitir su empleo por parte de los agentes en tiempo de ejecución, posibilitando el intercambio de conocimiento y el razonamiento sobre estos objetos. Y por otro lado repercutirá en el plano de persistencia, siendo de vital importancia enfocar la definición de estos artefactos y sus metadatos pensando en una futura reutilización en actividades posteriores.

En concreto, dentro del escenario que nos atañe se han modelado los siguientes tipos de objetos de aprendizaje necesarios para la actividad:

- Problemas (Problem)
- Soluciones (Solution)
- Vínculos Problema-Área Funcional (Problem-To-Area)

Las Áreas Funcionales se han modelado dentro de los objetos de aprendizaje como metadatos propios de éstos. Es posible apreciar en la Figura 10.8 un ejemplo de relación entre estos tipos de objetos de aprendizaje entre sí.

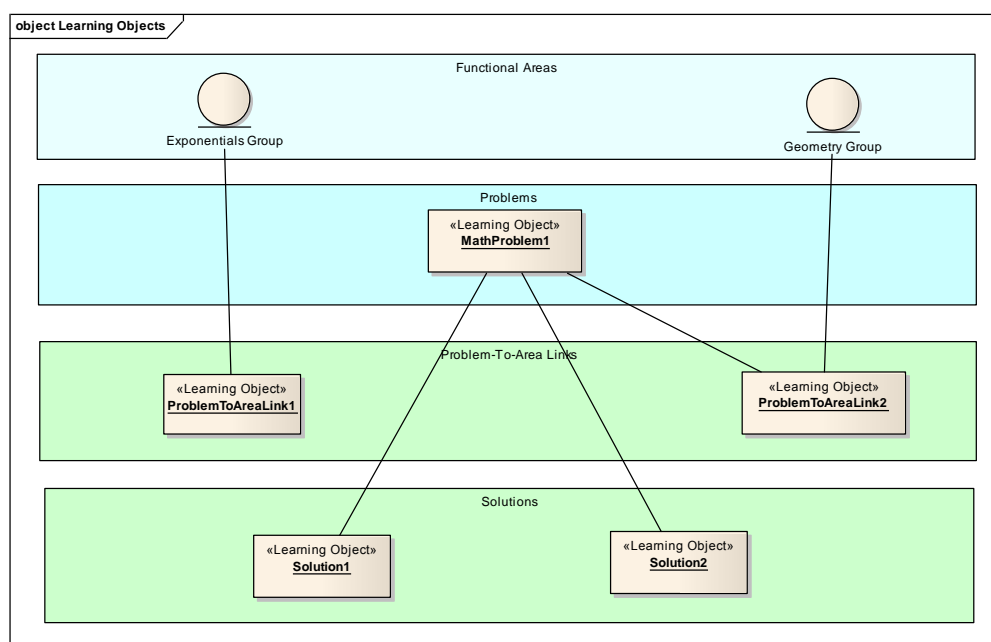


Figura 10.8 Esquema de Relaciones entre los Artefactos del Escenario

Una decisión de diseño ha sido el no modelar el vínculo entre problemas y áreas funcionales de las matemáticas como una propiedad o metadato dentro del artefacto 'Problema', sino como un objeto de aprendizaje en sí. Esto es debido a que el artefacto 'VínculoProblema-AreaFuncional' posee entidad propia desde el punto de vista de la actividad pedagógica, pues contiene información de utilidad para los usuarios involucrados en esta. Por tanto no es un mero enlace entre objetos sino que contendrá la información referente a dicho vínculo que incluye comentarios de los usuarios acerca de su corrección de este vínculo y el recuento de personas que están de acuerdo.

Los contenidos, junto con los metadatos más representativos de estos objetos, se presentan en la Tabla 10.1.

TIPO de LO	Problema	Solución	Vínculo Problema-Área Funcional
Contenido	Imagen JPG	Imagen JPG	Comentarios Votos
	Título Autor	Título Autor	Autor Contribuidores
Metadatos		ÁreaFuncional ProblemaRelacionado	Número de Votos ProblemaRelacionado ÁreaFuncional

Tabla 10.1 Contenido y Metadatos de los Objetos de Aprendizaje del Escenario

Los Problemas contienen un enunciado en forma de imagen y estarán presentes en el repositorio antes de la actividad. En principio es el profesor de matemáticas el encargado de suministrarlos, aunque es posible diseñar una actividad previa que incluya la creación y subida de problemas al repositorio por parte de los alumnos. Tanto las Soluciones como los VínculoProblema-AreaFuncional son generados como productos de la actividad por la interacción de los alumnos. Una Solución también será una imagen del problema resuelto. Los alumnos solucionarán el problema con papel y lápiz y subirán al sistema dicha solución como una fotografía tomada desde la cámara digital del dispositivo. Finalmente el objeto VínculoProblema-AreaFuncional tiene como contenido una estructura XML con los comentarios y votos a favor o en contra de dicho vínculo, junto con metadatos para realizar búsquedas acerca de quién es el autor de dicho vínculo, y/o qué usuarios han contribuido con comentarios.

10.2 DAEDALUS en el ‘Aula Inteligente’

El sistema que gestiona el escenario, y que será encargado de llevar a cabo la actividad, controlando los dispositivos y en definitiva toda la infraestructura esta implementado mediante la arquitectura DAEDALUS. En este nuevo escenario se puso a prueba la interoperabilidad de DAEDALUS para funcionar con otros sistemas previamente diseñados.

El grupo de investigación Encore de la Universidad de Toronto con el que se colaboró para desarrollar este escenario tenía varios requisitos con res-

pecto a la implementación. Por un lado, dado que este grupo de investigación es un miembro activo dentro de la comunidad SAIL, estaba interesado en incluir en el ‘Aula Inteligente’ varios de sus proyectos. SAIL (Scalable Architecture for Interactive Learning) (<http://sail.sourceforge.net/>) es una iniciativa que tiene como objetivo el crear una comunidad de desarrolladores de código abierto que, aun teniendo distintos intereses y opiniones dentro del campo de las tecnologías aplicadas a la enseñanza, todos puedan beneficiarse del acceso a una misma plataforma tecnológica central. Entre los proyectos que se están desarrollando en esta iniciativa está el repositorio de objetos de aprendizaje de código abierto llamado RoOLO (Repository of Open Learning Objects) (<http://code.google.com/p/roolo/>) y el Portal SAIL, un sistema gestor de aprendizaje. La implementación del ‘Aula Inteligente’ debía contar con estas dos aplicaciones para persistir los artefactos del escenario y para almacenar la información de los usuarios, respectivamente. Gracias a los mecanismos de envoltura de recursos externos presentes en DAEDALUS no fue complicado adaptar los componentes implicados como se comentará en el apartado 10.2.2.

Por otro lado, el grupo Encore también había estado trabajando previamente en las visualizaciones para las pantallas del aula. Concretamente, una estudiante de doctorado ya llevaba a medio camino una implementación en Flash de dichas visualizaciones que se quería utilizar. Además, debido a factores de tiempo, no fue posible implementar las interfaces de usuario en Java, sino que se optó por una implementación web, haciéndose necesaria otra solución de interoperabilidad.

A continuación se comentará el diseño que se usó para el modelo de conocimiento y el modelo de organización de este escenario.

10.2.1 Ampliación del Nivel de Semántica y Contexto

Un proceso común y habitualmente necesario a la hora de tratar nuevos escenarios con la arquitectura DAEDALUS es la necesidad de extensión de su modelo de conocimiento a Nivel de Semántica y Contexto para el tratamiento por parte de los agentes de nuevos elementos que aparecen en dichos escenarios. Es por ello que para este nuevo escenario de aula inteligente, el modelo de conocimiento de DAEDALUS tuvo que ser modificado para extender el *dominio de contexto* con nuevos conceptos y añadir nuevos elementos a la *taxonomía de sucesos de contexto*.

En el modelo de contexto se tuvo que añadir nuevas entidades subsumidas por el concepto de objeto de aprendizaje (*LO*) para representar los nuevos tipos de artefactos (Figura 10.9): Problema (*MathProblemLO*), Solución (*SolutionLO*) y VinculoProblema-AreaFuncional (*TagLinkLO*). Junto con estas entidades aparecieron nuevas relaciones para indicar los problemas que tiene que resolver un grupo (*hasToSolve*) o los que tiene que etiquetar un estudiante (*hasToTag*), y relaciones reificadas a partir de metadatos que indican si una solución resuelve un problema (*solves*) o si un problema está etiquetado con una determinada área funcional (*tags*).

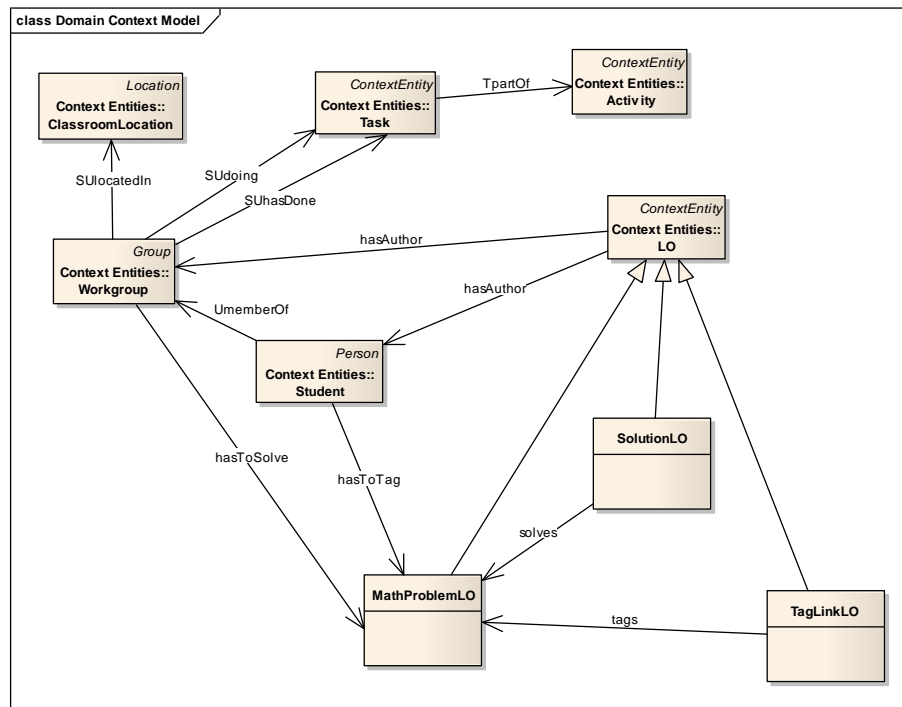


Figura 10.9 Ampliación del Dominio de Contexto

Dentro de la *taxonomía de sucesos de contexto*, fue necesario añadir sucesos que permitieran a los agentes compartir los cambios producidos en el contexto específico del aula inteligente y les permitiesen seguir el flujo de la actividad para poder coordinarse (Tabla 10.2):

Nombre del Evento	Explicación	Parámetros
daedalus/activity/userTagging	Un usuario está en la fase de etiquetando un problema.	Activity User ProblemLO
daedalus/activity/groupFinishedTagging	Los miembros de un grupo han terminado de etiquetar.	Activity Group
daedalus/activity/groupSolving	Un grupo está en la fase de resolución resolviendo un problema.	Activity Group ProblemLO
daedalus/activity/groupSolved	Un grupo ha resuelto un problema.	Activity Group ProblemLO SolutionLO
daedalus/activity/groupReflecting	Un grupo está reflexionando sobre un problema.	Activity Group ProblemLO

Tabla 10.2 Sucesos para los Procesos Pedagógicos en el Aula Inteligente

10.2.2 Diseño de la Organización Educativa

Dentro del Nivel de Organización Educativa es necesario especificar cuáles van a ser los agentes y los recursos necesarios para poder poner en marcha el escenario. La forma de hacer esto es en un primer término obtener una lista de las responsabilidades de los SPIs a desarrollar (apartado 10.1.1). Posteriormente se agrupan dichas responsabilidades con el fin de identificar los agentes implicados en nuestro escenario.

Las responsabilidades que podemos enumerar en función de la descripción del escenario del apartado 10.1.1 son:

- Obtención/Creación de la información de los grupos de trabajo.
- Asignación de una localización dentro del aula a un grupo de trabajo.
- Asignación de alumnos a los grupos según se conectan con su dispositivo.
- Orientar a los alumnos sobre dónde está su grupo de trabajo dentro del aula.
- Actualizar las visualizaciones de cada grupo y de clase.

- Buscar y Seleccionar Problemas para etiquetar
- Asignar Problemas para etiquetar a los grupos y usuarios
- Buscar y Seleccionar Problemas para resolver.
- Asignar Problemas para resolver a los grupos.
- Comprobar respuestas de los usuarios y que todos hayan respondido.
- Comprobar que todos los miembros de un grupo estén de acuerdo con una misma Solución.
- Buscar y preguntar sobre un vínculo entre pregunta y área.
- Crear objetos de aprendizaje de Soluciones y VinculosProblema-AreaFuncional.

En base a esta lista decidimos agrupar estas responsabilidades en tres nuevos agentes cognitivos diferentes que complementarán los ya existentes:

- Agente de Contexto
 - Agente SmartClassroom Coordinator (Coordinador del Aula Inteligente)
- Agente de Actividad
 - Agente MathActivity (Coordinador de la Actividad de Matemáticas)
- Agente de Comunidad
 - Agente MathGroup (Coordinador de los Grupos)

La Figura 10.10 muestra una descripción más detallada de la lista de responsabilidades y cómo se asocia a los agentes recién nombrados. Sin embargo, será necesario seguir el refinamiento de las responsabilidades de cada agente, que se expone a continuación.

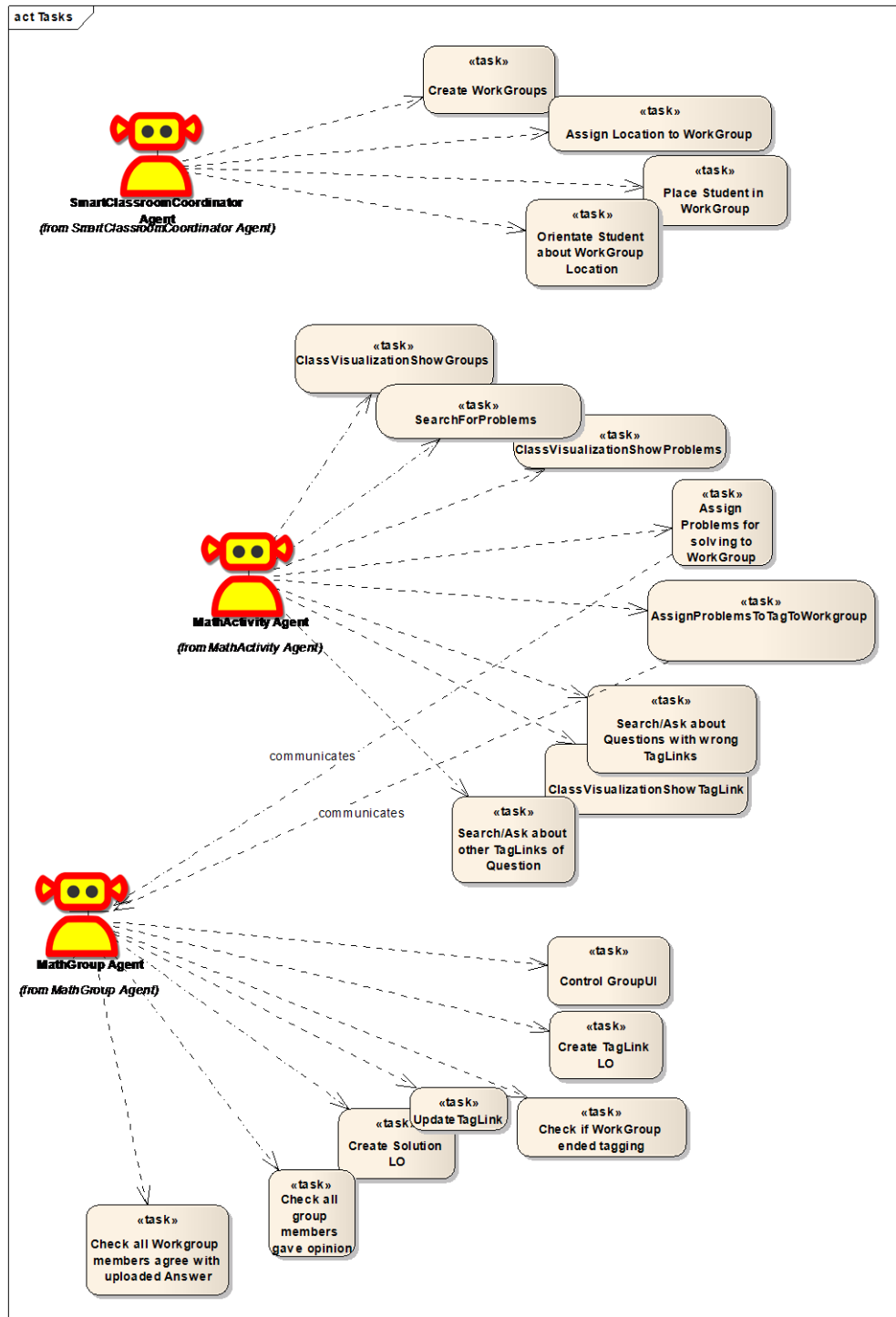


Figura 10.10 Responsabilidades de los Agentes

AGENTE SMARTCLASSROOMCOORDINATOR

Se puede considerar que el Agente *SmartClassroomCoordinator* desempeñará un papel más organizativo que pedagógico dentro del escenario. Este agente será el encargado de configurar toda el ‘Aula Inteligente’ y coordinar a los usuarios dentro de este entorno de aprendizaje (véase Figura 10.11). Más concretamente, el agente ha de preparar el entorno para que pueda desarrollarse la actividad.

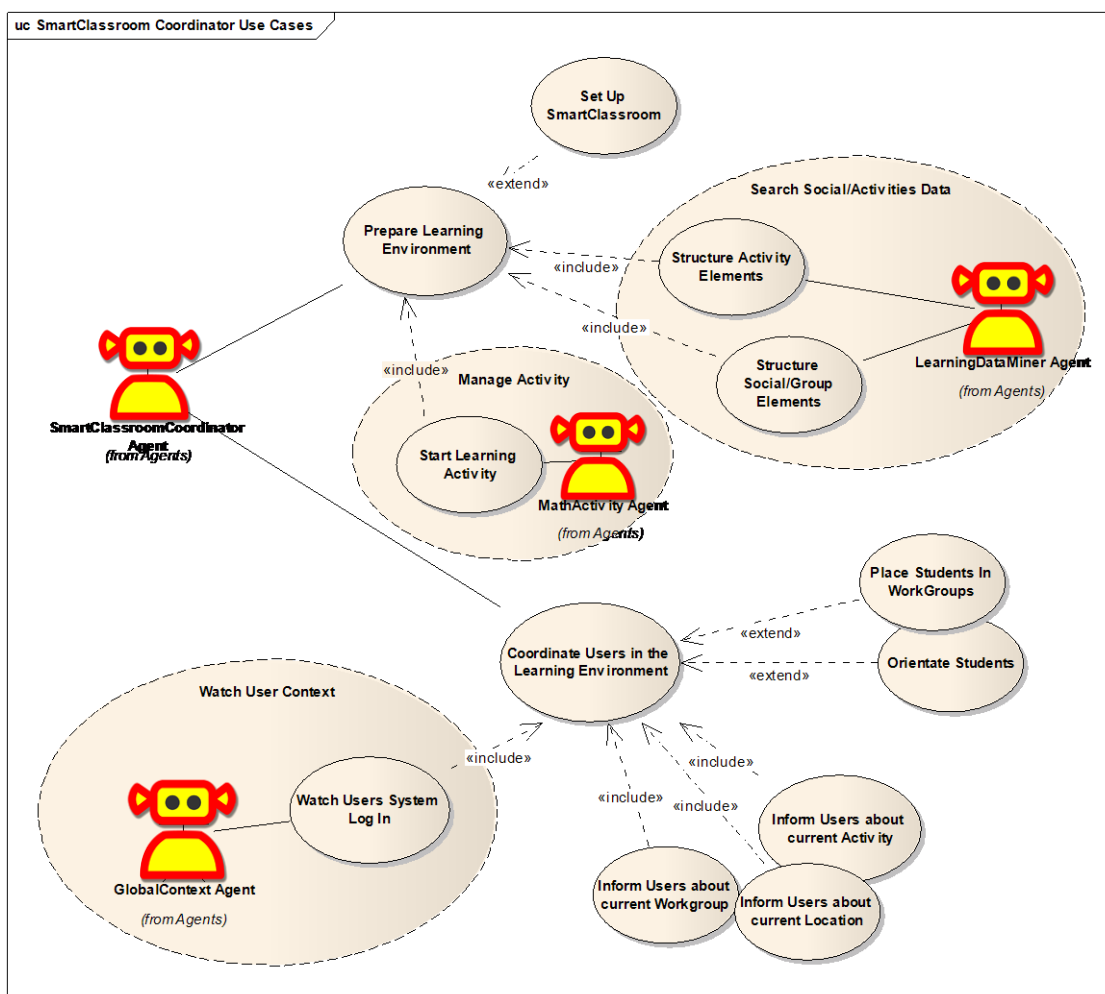


Figura 10.11 Responsabilidades del Agente *SmartClassroomCoordinator*

Para desarrollar esta gestión tendrá que coordinarse con el Agente *LearningDataMiner* (reutilizado del escenario de ENLACE) para obtener información sobre los usuarios y la actividad, además de ponerse en contacto con el Agente *MathActivity*.

En base a lo comentado, se diseña el Agente *SmartClassroomCoordinator* con dos objetivos como se muestra en la Figura 10.12. El agente tratará de resolver primero el objetivo de configuración

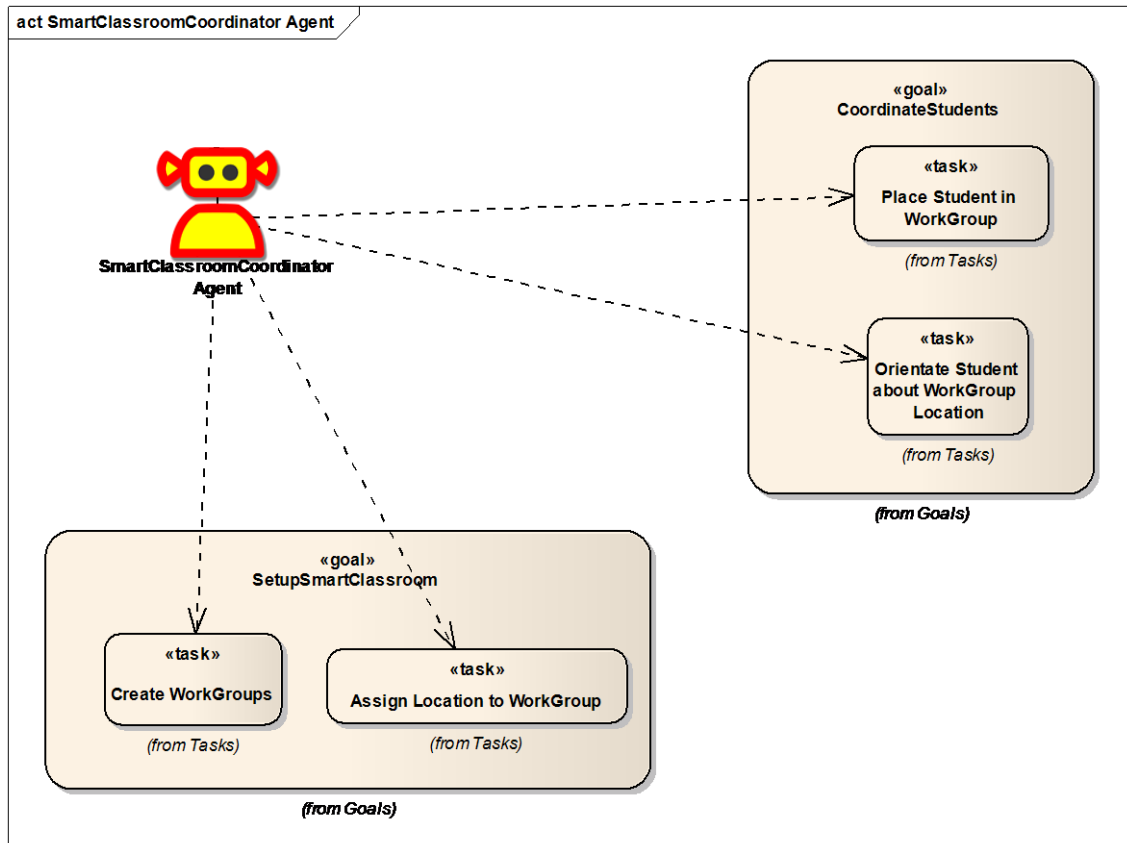


Figura 10.12 Objetivos y Tareas del Agente *SmartClassroomCoordinator*

AGENTE MATHACTIVITY

El Agente *MathActivity* tiene como papel dentro del escenario llevar a cabo la actividad de matemáticas. Esto incluye (Figura 10.13) realizar las tareas previas al inicio de ésta, cooperando con el Agente *LearningDataMiner* para recopilar los artefactos necesarios para la actividad (es decir, los objetos de aprendizaje Problemas), distribuir los problemas a los grupos y sus miembros, delegando para ello en las instancias del Agente *MathGroup*. Por otro lado, si ciertos patrones aparecen, como por ejemplo cuando hay dos alumnos que han etiquetado de forma distinta un problema o cuando existe un desacuerdo entre miembros del grupo, el agente ha de preguntar a los alumnos sobre de los Problemas y las Áreas Funcionales Matemáticas que están tra-

tando para guiar la reflexión en la actividad. También a lo largo de todo el escenario, el agente *MathActivity* será el encargado de ir actualizando las visualizaciones de la pantalla de la clase con toda la información del desarrollo de la actividad.

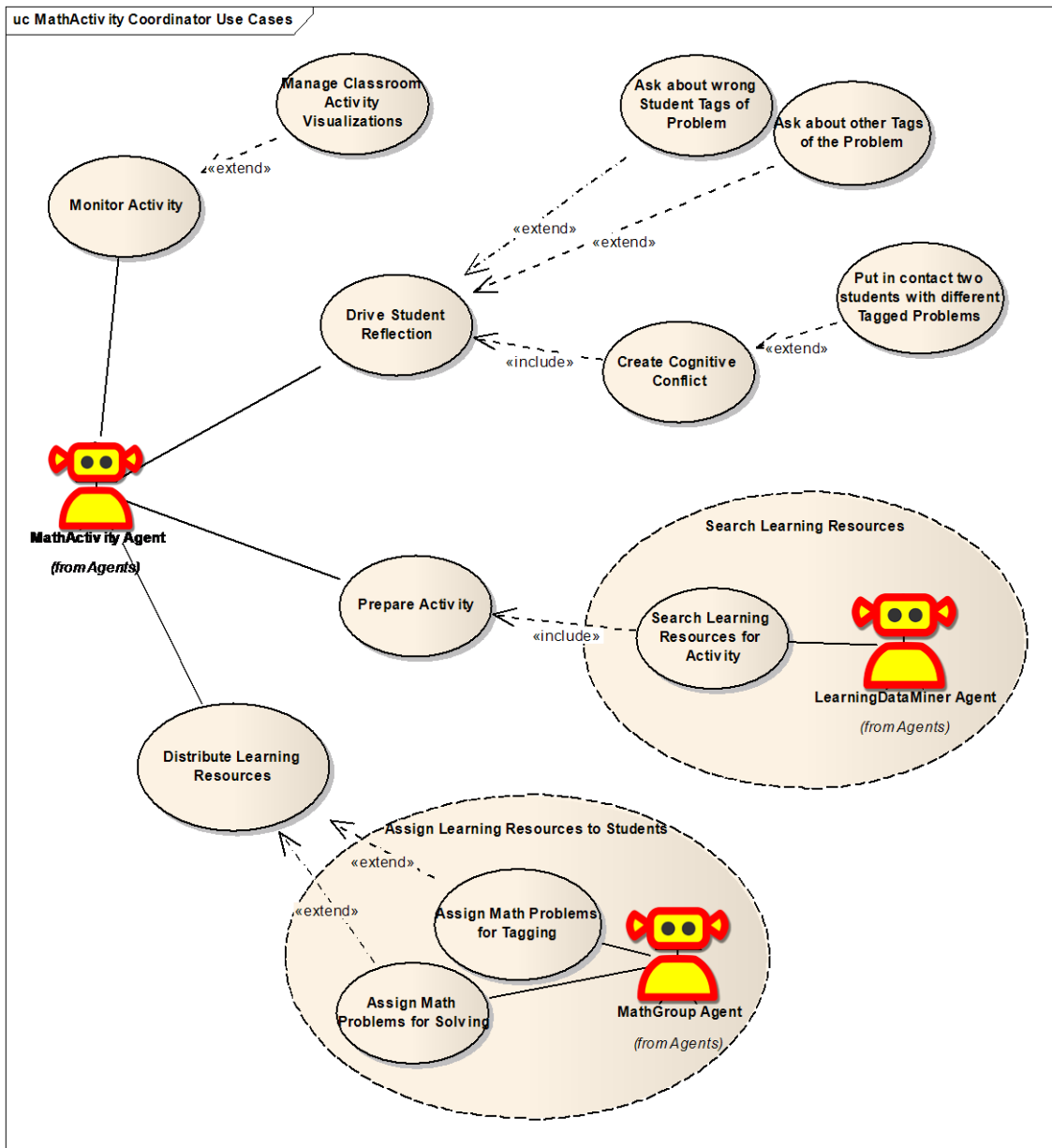


Figura 10.13 Responsabilidades del Agente *MathActivity*

Este agente ha sido descompuesto en tres objetivos (Figura 10.14): Preparar la Actividad, Distribuir Artefactos de la Actividad, Guiar la Reflexión del Alumno.

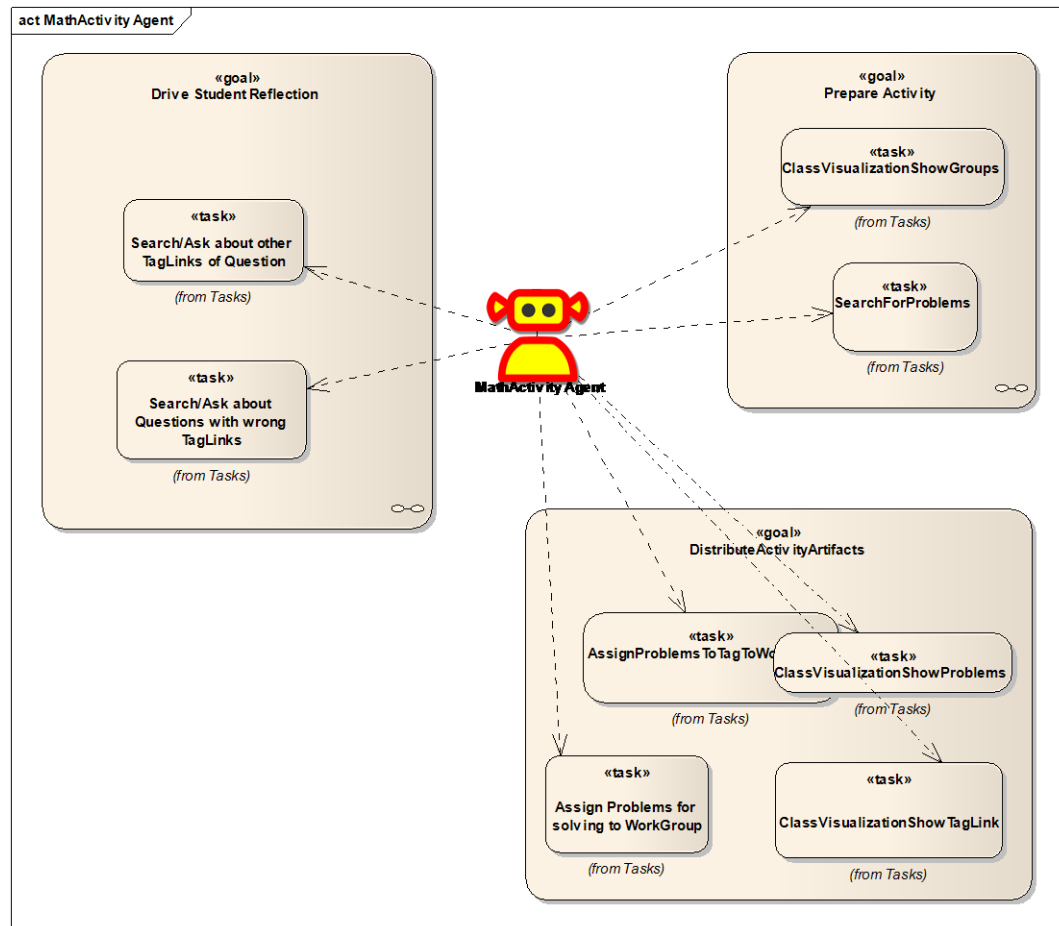


Figura 10.14 Objetivos y Tareas del Agente MathActivity

AGENTE MATHGROUP

El Agente *MathGroup* es el agente que está en contacto con el grupo y sus miembros, supervisando la actividad a nivel de grupo. Este agente es el único que posee varias instancias corriendo al mismo tiempo. En la organización existe un Agente *MathGroup* por cada uno de los grupos de trabajo, y todos ellos están coordinados por el Agente *MathActivity*.

El agente *MathGroup* supervisa a los miembros de su grupo asignado en cada una de las tres fases de la actividad, encargándose de almacenar los productos que se van generando en estas fases (Figura 10.15). Para realizar su

tarea el Agente MathGroup se comunica con los alumnos a través del recurso de interfaz gráfica.

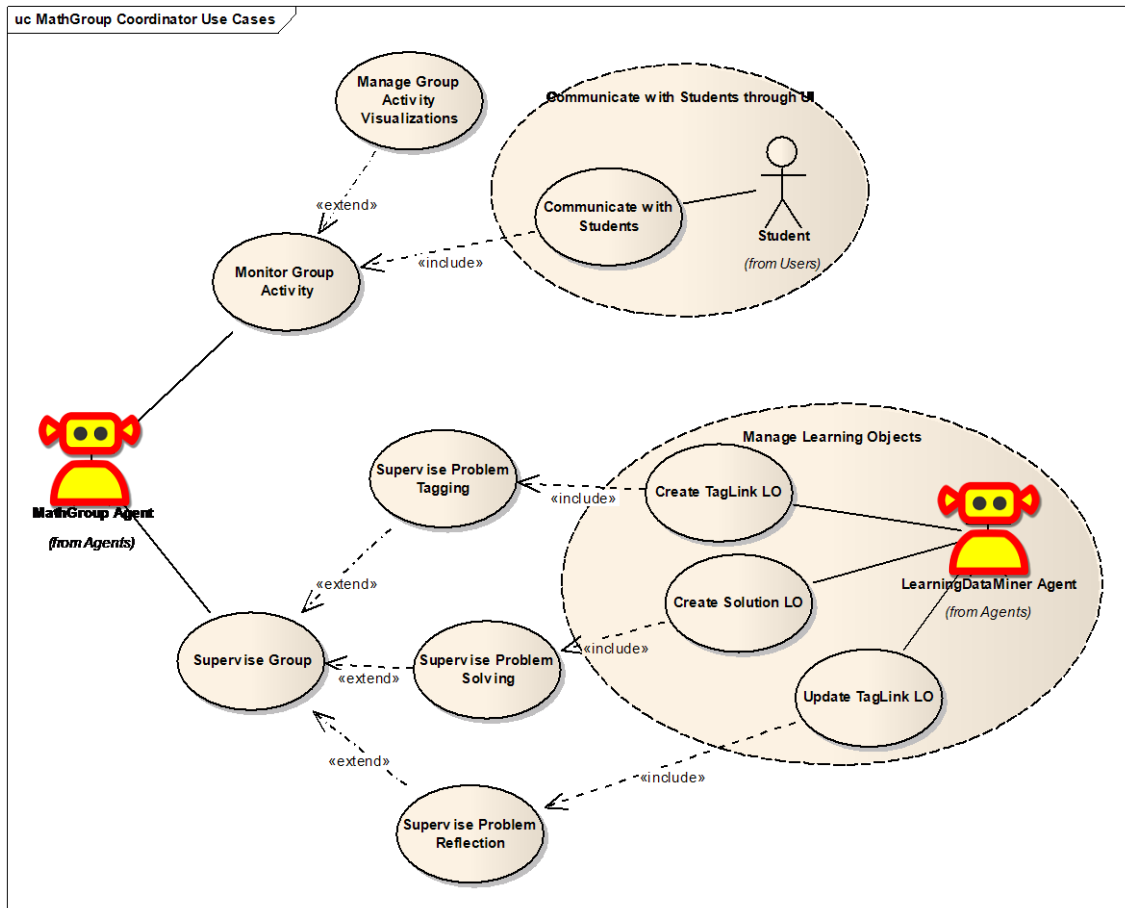


Figura 10.15 Responsabilidades del Agente MathGroup

Este agente se ha modelado a través de tres objetivos correspondientes a las fases de la actividad en las que están involucrados los alumnos: Supervisar Etiquetado de Problemas, Supervisar Resolución de Problemas y Supervisar Reflexión de Problemas. Los objetivos de este agente se van activando a petición del Agente *MathActivity* y cuando alguno de estos objetivos o tareas se han completado, se notifica a dicho agente mediante los eventos definidos en la Tabla 10.2.

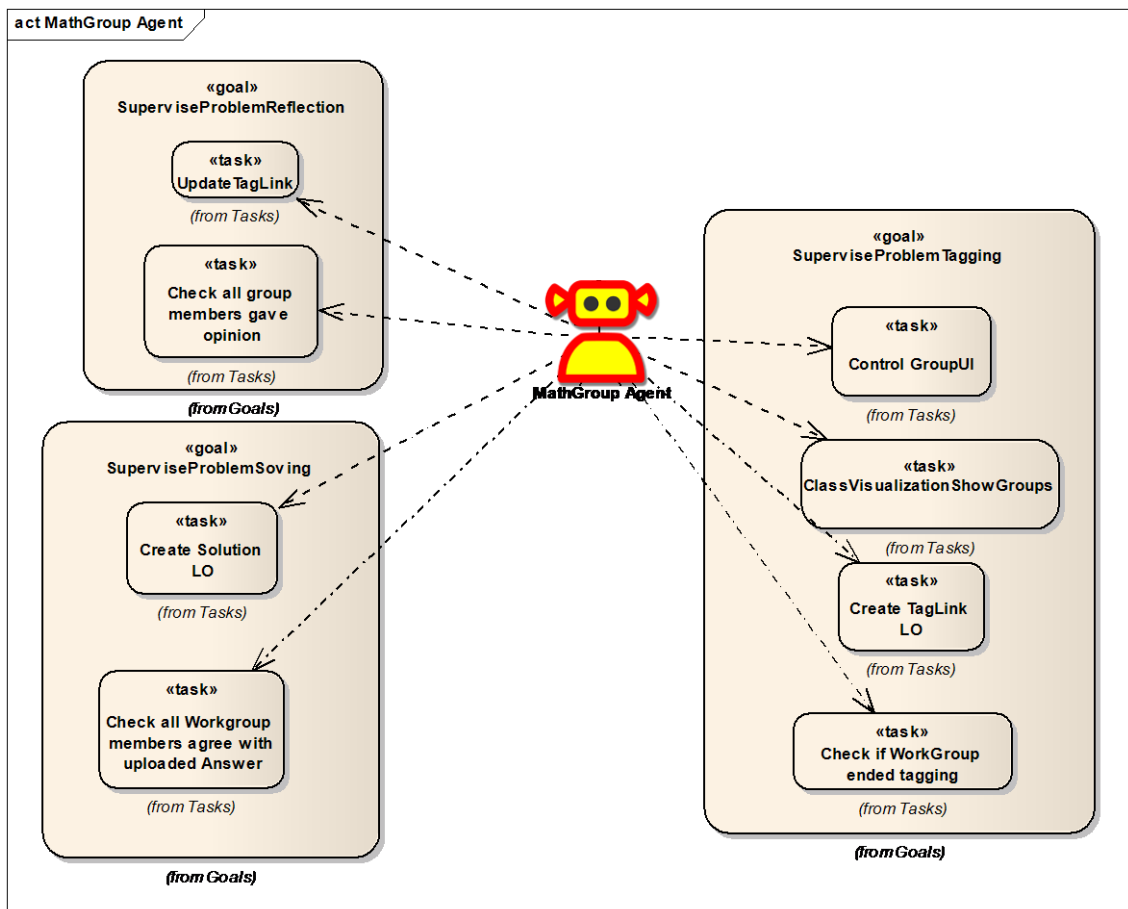


Figura 10.16 Objetivos y Tareas del Agente MathGroup

INTERACCIÓN ENTRE AGENTES

Entre los agentes *SmartClassroomCoordinator*, *MathActivity* y *MathGroup* se establece una jerarquía que se refleja en su interacción y en su protocolo de comunicación. Como puede apreciarse en la Figura 10.17, el Agente *SmartClassroomCoordinator* es el que da comienzo a la actividad comunicándose al Agente *MathActivity*. Una vez que éste ha recibido el mensaje, el Agente *MathActivity* será el controlador total del desarrollo de la actividad pedagógica. Básicamente dará las órdenes a las diferentes instancias de Agente *MathGroup* para cambiar entre las fases de etiquetado, resolución y reflexión.

Si se quisiese modificar el protocolo de desarrollo de la actividad simplemente habría que modificar las reglas de resolución de objetivos que gobiernan el comportamiento del Agente *MathActivity*.

Por otro lado, los distintos mecanismos de colaboración entre alumnos se encuentran repartidos entre los agentes *MathActivity* y *MathGroup*. Por un lado *MathActivity* es el encargado de buscar patrones que puedan promover la colaboración y el conflicto cognitivo. Una vez que encuentra alguno se lo comunica a la instancia de Agente *MathGroup* correspondiente que será el encargado de ponerlos en práctica interactuando con los estudiantes a través de las interfaces de usuario.

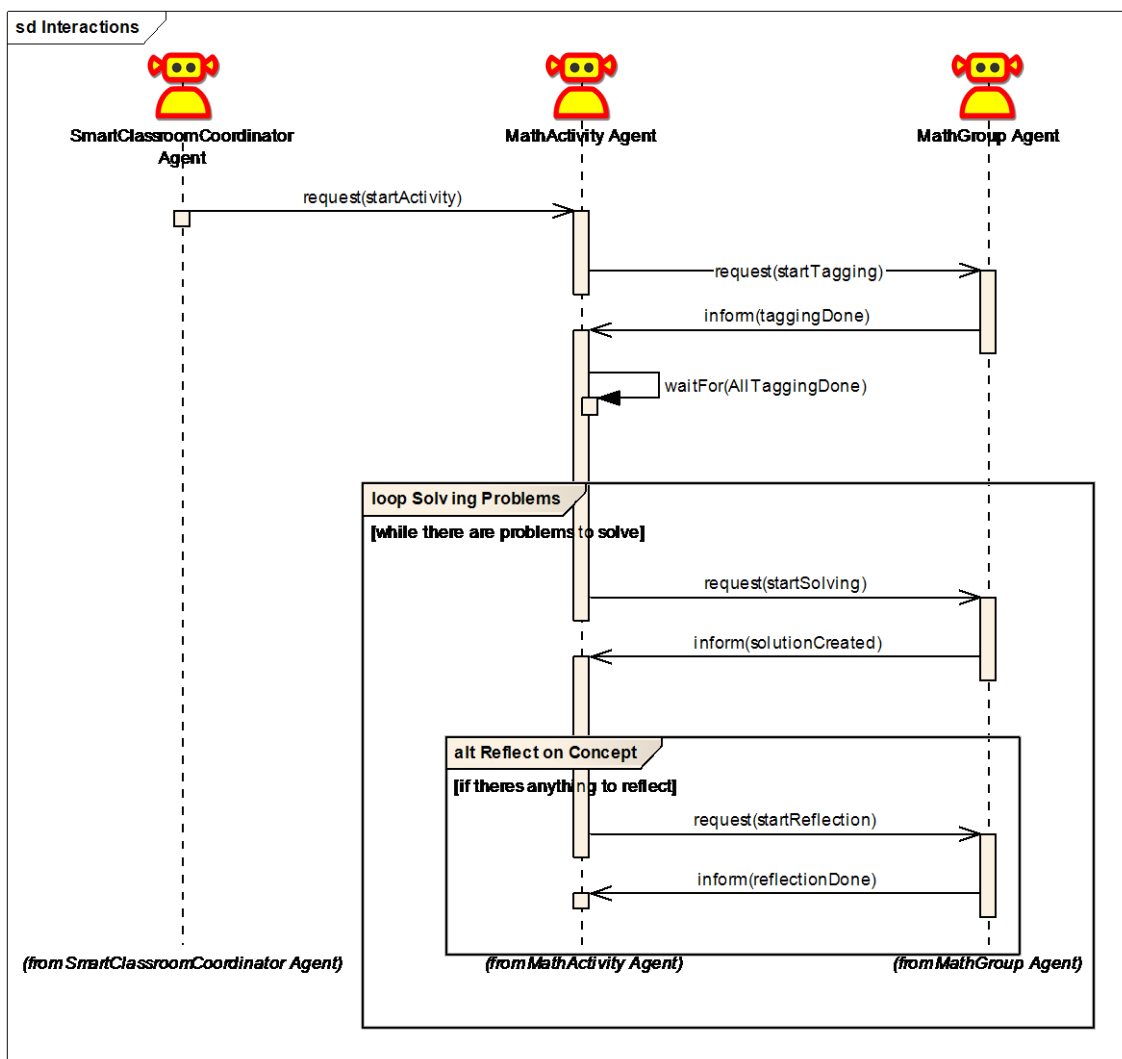


Figura 10.17 Interacción entre los agentes

10.2.3 Implementación de la Organización Educativa

El escenario está compuesto de un nodo servidor y distintos dispositivos móviles para los alumnos. El nodo servidor habitan los agentes *SmartClassroomCoordinator*, *MathActivity* y *MathGroup* recién descritos. Por otro lado, se reutilizan los agentes *GlobalContextAgent* y *LearningDataMinerAgent* del escenario ENLACE. El primero es empleado para mantener y distribuir la información de contexto entre todos los demás agentes, el segundo realiza búsquedas sobre datos para la actividad.

Aunque la idea original era la utilización en la implementación de Agentes Personales dentro de los dispositivos móviles, por los reducidos tiempos de desarrollo no fue posible, estableciéndose el control de los recursos de interfaz gráfica desde el nodo servidor, tal y como se muestra en la descripción de la organización presentada en la Figura 10.18.

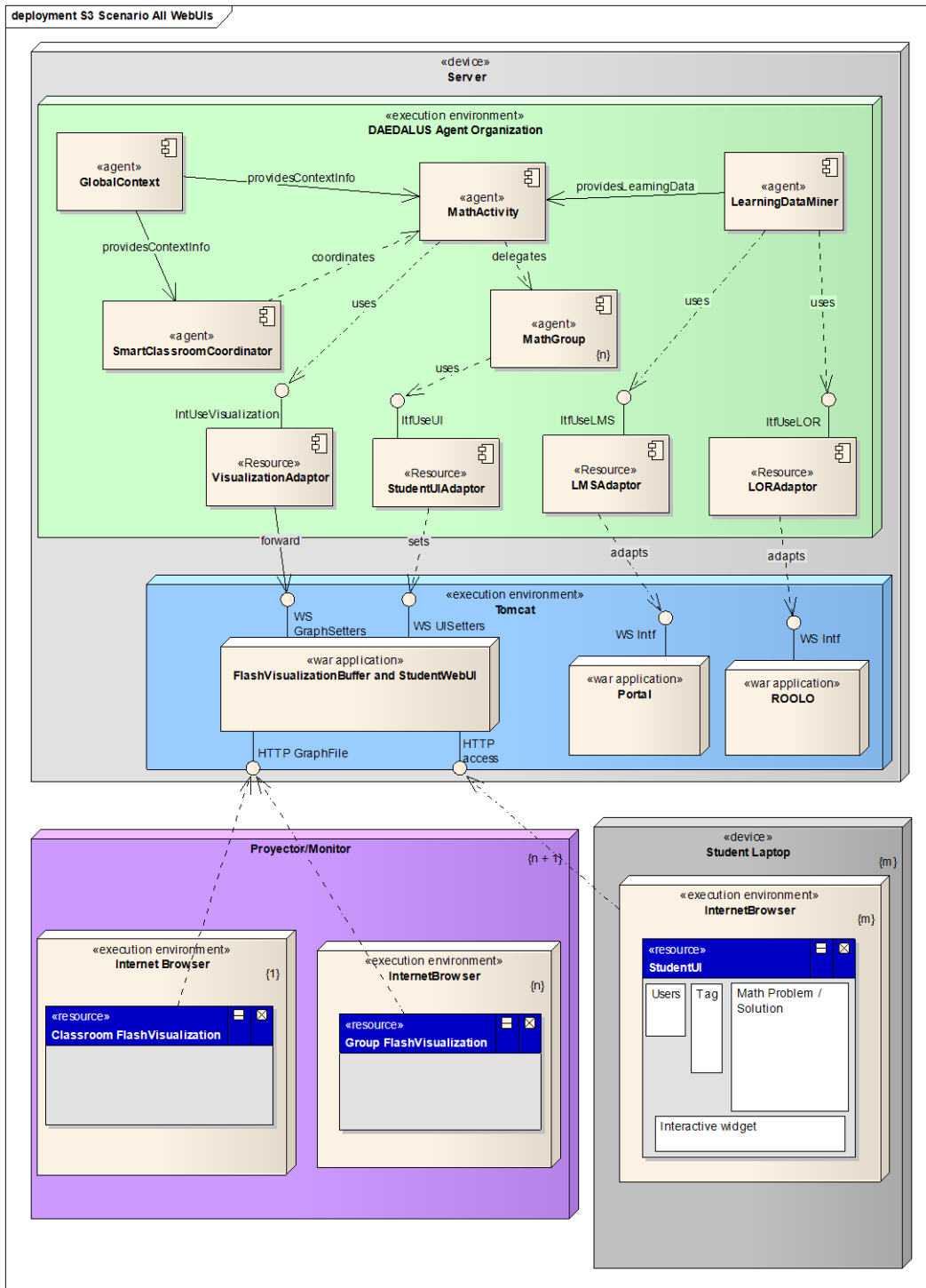


Figura 10.18 Descripción de la Organización para el Escenario

Fue necesario implementar nuevos recursos envoltorio para adaptar y hacer accesibles recursos externos a la organización de agentes. El recurso LORAdaptor se comunica mediante servicios web con el repositorio de objetos de aprendizaje ROOLO, al igual que el recurso LMSAdaptor hace con el Portal SAIL. También se implementaron recursos envoltorio para controlar las visualizaciones y las interfaces web. Los alumnos se conectaban al sistema a través de una interfaz web desde el navegador de sus dispositivos móviles.

Para el desarrollo y testeo del escenario y la organización de agentes se implementó una interfaz gráfica para el agente GlobalContext (Figura 10.19).

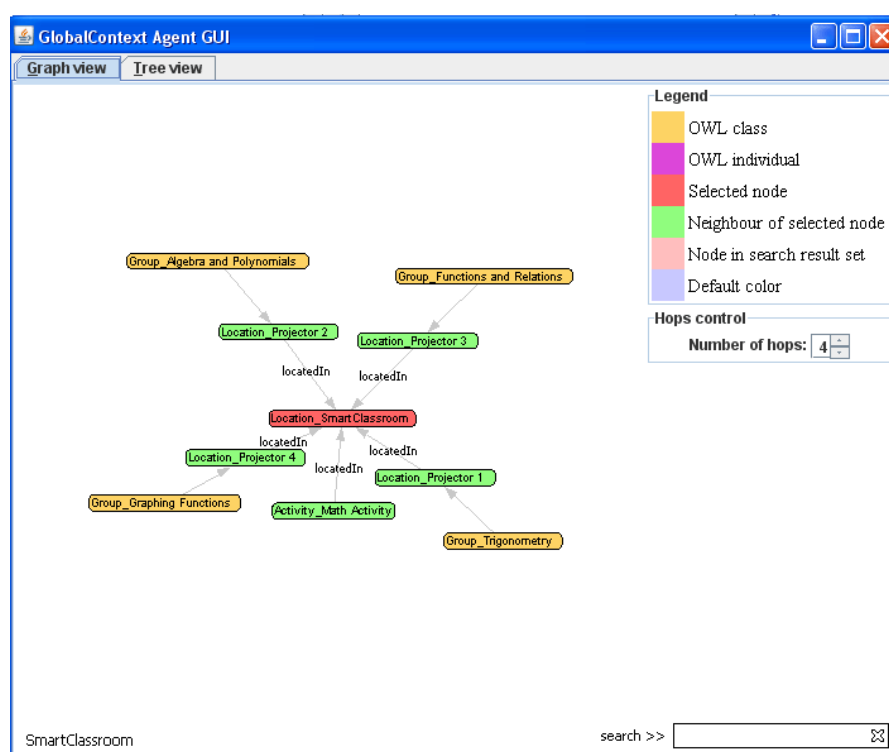


Figura 10.19 Recurso de Visualización del Agente GlobalContextAgent

10.3 Puesta en Marcha del escenario

El escenario se probó con usuarios reales pertenecientes a la escuela UTS de la Universidad de Toronto. Los alumnos emplearon ordenadores portátiles para el desarrollo de la actividad (Figura 10.20 y Figura 10.21).

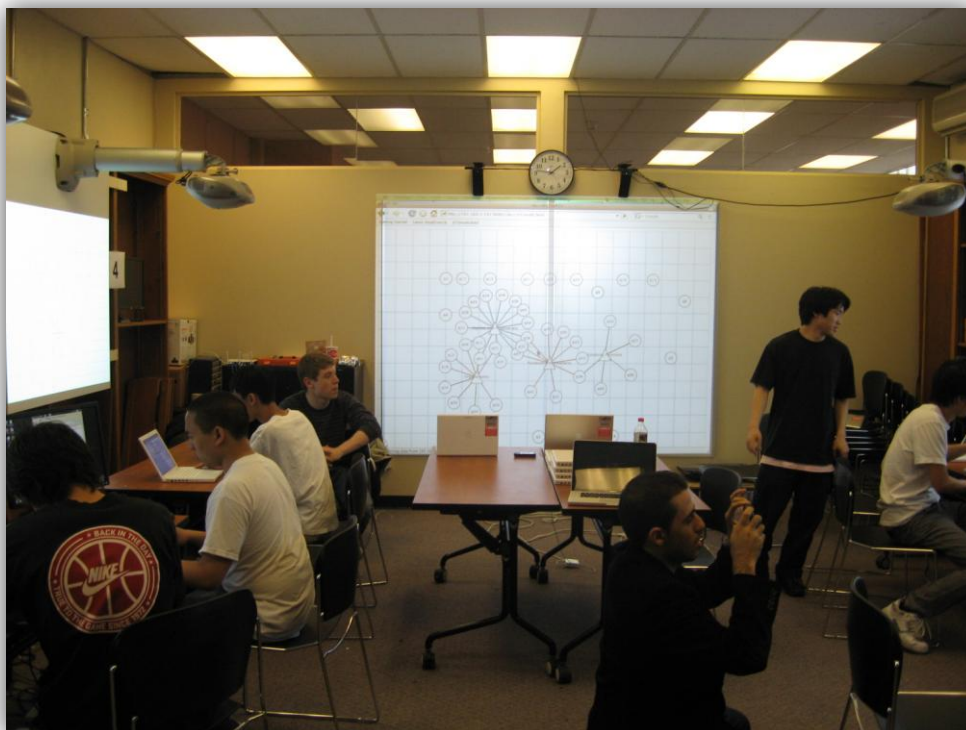


Figura 10.20 Desarrollo de la Actividad

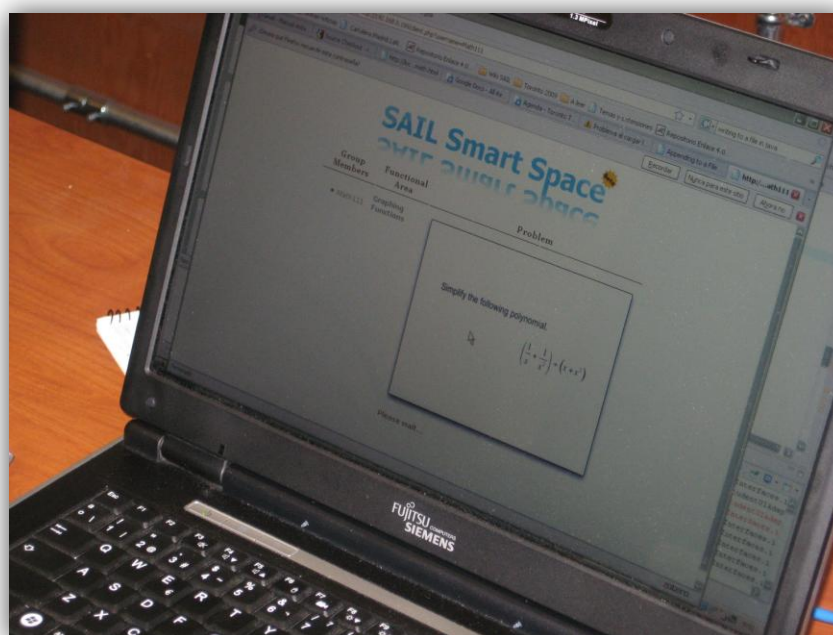


Figura 10.21 Interfaz de Usuario

Las pruebas del escenario fueron muy satisfactorias, tanto a nivel pedagógico como tecnológico. El sistema implantado mediante DAEDALUS para dar soporte a este escenario logró la coordinación dentro del aula inteligente de todos los elementos tecnológicos y permitió el desarrollo de los procesos pedagógicos planteados. En resumen, gracias a DAEDALUS se consiguió diseñar y crear un espacio de aprendizaje ubicuo con diferentes dispositivos móviles y embebidos, que a pesar de la complejidad tecnológica, ofreció nuevas e innovadoras experiencias educativas, y todo ello con unos tiempos de desarrollo muy ajustados.

11

DISCUSIÓN FINAL Y CONCLUSIONES

Para concluir con esta memoria de tesis, se procede a hacer una recapitulación que incluye en primer lugar una síntesis de recomendaciones de aplicación y extensibilidad para el uso de la arquitectura DAEDALUS en nuevos escenarios de aprendizaje, tratando de aprovechar y reutilizar todos los elementos básicos que aporta la arquitectura. A continuación, se procede a hacer una enumeración de las contribuciones de este trabajo de tesis, para completarlo finalmente con unas conclusiones propiamente dichas y una reseña acerca del trabajo futuro.

11.1 Recomendaciones de Aplicación y Extensibilidad

Como se ha demostrado en los escenarios de aplicación, la arquitectura DAEDALUS ofrece todos los elementos necesarios para poder llevar a cabo desarrollos en el ámbito de las tecnologías móviles educativas mediante agentes software. Basándonos en los escenarios descritos, se enuncian unas recomendaciones pragmáticas de uso, haciendo hincapié en cómo aplicar y extender la arquitectura para poder emplearla en nuevos escenarios.

No es nuestra intención aportar una metodología completa para el desarrollo de sistemas multi-agente, existiendo amplia literatura a tal respecto (J. Gómez-Sanz & J. Pavón, 2004), sino simplemente pretendemos extrapolar para el lector una serie de recomendaciones heurísticas basándonos en la experiencia de los escenarios desarrollados en este trabajo a modo de listado de recomendaciones a tener en cuenta a la hora de emprender el diseño e implementación de un escenario pedagógico utilizando la arquitectura DAEDALUS. No se trata de un listado de pasos lineal, sino que en muchas ocasiones hará falta tomar una aproximación iterativa con la que poco a poco se vaya refinando progresivamente el diseño de la organización educativa.

DISEÑO

- *Obtención de un escenario de diseño.* Este paso está ligado con la metodología de diseño basada en escenarios y mediante la cual es posible identificar todos los elementos que integrarán el escenario pedagógico. Esto implica partir de una descripción narrativa sobre cómo es el escenario y el modo de interacción que tendrán los usuarios dentro de él. Todos los siguientes pasos de diseño se basarán en cierta medida en este escenario de diseño.
- *Definición de la infraestructura tecnológica.* Desde un primer momento hace falta tener una idea acerca de qué dispositivos podrán utilizar los usuarios, tanto alumnos como profesores, monitores, etc., y en definitiva cuáles podrán ser las características tecnológicas del espacio (o espacios) de aprendizaje ubicuos que se quiere desarrollar.
- *Definición de Productos, LO y metadatos.* Dependiendo de las necesidades pedagógicas y las herramientas que los estudiantes tengan que

emplear hará falta definir nuevos tipos de objetos de aprendizaje, que representen los productos que los alumnos van generando a lo largo del desarrollo de sus actividades y que podrán reutilizar.

- *Extensión del modelo de contexto.* Hacer uso de los conceptos ancla del modelo de contexto, para añadir los elementos necesarios del espacio de aprendizaje ubicuo que se tengan que tratar computacionalmente.
- *Identificar servicios pedagógicos inteligentes del escenario.* A partir del escenario de diseño extraer los distintos servicios que serán soportados por la organización educativa. Es decir, identificar las funcionalidades necesarias para ofrecer apoyo inteligente y facilitar un soporte eficaz a las actividades de aprendizaje a los usuarios del escenario de diseño.
- *Identificar responsabilidades y establecer la organización educativa.* A partir de los servicios pedagógicos inteligentes extraídos, realizar una división de éstos en responsabilidades que permitirán una primera estructuración de los agentes y recursos necesarios en la organización educativa, y una primera consideración hacia los aspectos distribuidos de ésta.
- *Reutilización de componentes educativos.* Una recomendación importante es tener siempre en cuenta en el diseño de la organización educativa los agentes y recursos previamente disponibles de otros escenarios. Habitualmente pueden cubrir responsabilidades presentes en los nuevos escenarios directamente o simplemente modificarlos o extenderlos para incorporarlos a estos escenarios.
- *Asociar responsabilidades de cada agente.* Establecer un completo reparto de las responsabilidades sobre la organización educativa que cubra todas las responsabilidades necesarias.
- *Modelar interacciones entre agentes.* Tener presente las interacciones y dependencias que se establecen entre los agentes definidos para provisionar los servicios pedagógicos inteligentes, ya que pueden servir para identificar potenciales riesgos de funcionamiento y necesidades de representación del conocimiento para su intercambio en la comunicación.
- *Generación de Objetivos y Tareas.* Crear los distintos objetivos y establecer procedimientos y tareas de resolución. Se puede realizar

una aproximación de abajo a arriba (Bottom-Up), agrupando las tareas en objetivos, una aproximación de arriba abajo (Top-Down), refinando los objetivos en las acciones necesarias para su resolución, o bien una aproximación mixta.

IMPLEMENTACIÓN

- Creación de objetos de contexto nuevos a partir de la ontología.
- Seleccionar patrones para los agentes y efectuar mapeo de objetivos y tareas.
- Creación de los recursos de la organización educativa, incluyendo las interfaces de usuario que estarán disponibles en los dispositivos.

11.2 Contribuciones

Dentro de este trabajo de investigación se ha explorado y desarrollado una arquitectura que da soporte a nuevos paradigmas tecnológicos educativos emergentes de carácter colaborativo, ubicuo y distribuido, aportando a lo largo de todo el proceso una serie de modelos computacionales para sustentarlos. Esta nueva arquitectura ha sido basada en el paradigma de los sistemas *multi-agente*, con el fin de habilitar un soporte inteligente a escenarios de aprendizaje móvil y colaborativo.

El valor de las contribuciones que aporta este trabajo con respecto a lo existente en la literatura, tanto en lo referente a tecnología educativa como a la tecnología de agentes en general, representa un avance remarcable en cuanto a:

➤ Soporte distribuido al aprendizaje colaborativo y móvil

Este trabajo aporta una plataforma educativa software distribuida basada en agentes capaz de funcionar en un rango heterogéneo de dispositivos, incluidos dispositivos móviles. En todo momento se ha buscado una *solución distribuida*, en contraposición a soluciones de carácter centralizado, con el fin de aportar flexibilidad a la hora de definir y desarrollar espacios de aprendizaje ubicuos en los que los alumnos puedan colaborar. Este modelo de organización de agentes de carácter totalmente distribuido posee la capacidad de adaptarse dinámicamente a distintas topologías de red.

➤ **Arquitectura y su reusabilidad**

Una importante cuestión con respecto a otros frameworks o herramientas de desarrollo basadas en agentes es la consideración de un agente como un componente software que se instancia a partir de un patrón. Esta componen-tización promueve la reusabilidad de los agentes pedagógicos en nuevos esce-narios, además de facilitar la extensibilidad de la infraestructura.

➤ **Nivel de detalle en el diseño e implementación de casos**

Siendo desgraciadamente habitual el poco nivel de detalle técnico en la mayoría de las propuestas de desarrollo de sistemas basados en agentes pre-sentes en la literatura, una de las principales contribuciones de este trabajo de investigación es el aportar e implementar de forma detallada modelos y meca-nismos internos, tanto a nivel de agente como de organización.

Contribuciones importantes a destacar en cuanto a la implementación son la adaptación de estos modelos para funcionar en dispositivos móviles con menores capacidades junto con la definición de un recurso de comunicación que incorpora el estándar OSGi y que puede ser utilizado en diferentes áreas como por ejemplo la domótica, la automoción o el control industrial.

➤ **Modelización y sistematización de tipos de agentes nuevos imple-mentados**

Concretamente, este trabajo aporta dentro del campo del modelado de los agentes pedagógicos, una descripción detallada de sus funcionalidades y estructuras internas, elementos muy vagamente descritos en la literatura de los sistemas basados agentes para la educación y en concreto en los sistemas basados en agentes para el aprendizaje móvil.

Asimismo, se ha profundizado en la sistematización y clasificación de los distintos tipos de agentes pedagógicos además de trabajar en el modelado e implementación a detalle de un conjunto de nuevos agentes especializados que aportan servicios pedagógicos inteligentes para el desarrollo y el apoyo de actividades de aprendizaje.

➤ **Completitud del modelo de contexto**

Otra contribución notable de este trabajo consiste en ofrecer un com-pleto modelo de conocimiento para el tratamiento de la información contex-tual para los agentes pedagógicos de la organización educativa. Se ha definido una ontología de conceptos para el aprendizaje móvil y común a todos los

agentes de la organización educativa que incluye un dominio de contexto y una taxonomía de eventos.

Se aporta también una representación computacional de dichos modelos que permite el intercambio de información y conocimiento distribuido entre todos los agentes pedagógicos de la arquitectura, además de admitir su extensión para poder cubrir nuevos escenarios de aprendizaje.

➤ **Soporte a la continuidad espacial y temporal del aprendizaje**

Una necesidad planteada desde un principio en este trabajo ha sido el soporte a la continuidad del aprendizaje. Este trabajo contribuye al soporte, desarrollo y a la integración de actividades de aprendizaje colaborativas formales e informales que hacen uso de tecnologías móviles inalámbricas y de información contextual. Esta integración no se ha asociado únicamente a una dimensión espacial, es decir a lo largo de distintos escenarios localizados en diferentes entornos y contextos, sino también se da importancia al eje temporal, con flujos de actividades enlazados y diseñados para ser realizados a corto, medio y largo término.

➤ **Enfoque y evaluación basada en escenarios**

Una idea principal empleada a lo largo de este trabajo ha consistido en la integración en el proceso de desarrollo de un diseño basado en escenarios. Mediante el empleo de escenarios durante el desarrollo ha sido posible enfocar el diseño hacia detalles que precisan mayor atención desde el punto de vista de los usuarios.

➤ **Recomendaciones de uso futuro**

Finalmente, para promover el buen empleo y la extensión de los elementos de la arquitectura, con el fin de que sea posible un reaprovechamiento y un futuro empleo de éstos en nuevos espacios de aprendizaje ubicuos, este trabajo se complementa aportando una serie de recomendaciones de uso y extensibilidad.

11.3 Conclusiones

Los beneficios que aporta la tecnología de agentes son evidentes para poder implementar y dar soporte a escenarios educativos distribuidos y complejos: se puede conseguir una distribución real sin renunciar a un soporte inteligente, servicios adaptados y a un diseño realizado en alto nivel gracias al

concepto de agente. Es indudable que son necesarias herramientas y marcos de trabajo para poder llevar a cabo este tipo de desarrollos que permitan hacer efectivas y plasmar las distintas ideas pedagógicas y teorías del aprendizaje móvil y colaborativo. La arquitectura DAEDALUS es la primera que aporta unos primeros pasos hacia este objetivo, estableciendo modelos y una infraestructura distribuida que desarrolla las reglas y criterios a ser utilizados por los agentes cognitivos en los mecanismos de inferencia para determinar las acciones a realizar, así como la generación del modelo de contexto y la taxonomía de eventos.

Gracias a estar basada en ICARO-T, la arquitectura DAEDALUS considera a los agentes pedagógicos como componentes software que se instancian a partir de un patrón y permite definir nuevos patrones de agentes además de proporcionar normas de ingeniería sencillas para su diseño, su integración en la infraestructura y su posterior reutilización. Mediante el desarrollo de la arquitectura DAEDALUS no sólo se ha contribuido al mundo educativo sino también al mundo del desarrollo basado en agentes extendiendo ICARO-T y al mundo del desarrollo de sistemas distribuidos facilitando la comunicación y la distribución transparente de las entidades computacionales en una red de procesadores. Se simplifica el proceso de despliegue del modelo lógico en el modelo físico, utilizando componentes predefinidos y tecnologías estándar que se encargan de la creación y de la monitorización de los elementos desplegados en los nodos.

Esta infraestructura desarrollada se enfrenta además a aspectos de bajo nivel como la composición de los componentes, su distribución, coordinación y comunicación, para posibilitar definir en un nivel mayor de abstracción el sistema educativo dividiendo responsabilidades entre entidades agente y definiendo claramente cuáles son sus objetivos y tareas. Las ventajas son múltiples: una forma intuitiva de tratar el problema de la distribución, una forma eficaz de tratar y manipular información de contexto de forma distribuida, ya que cada entidad puede hacer uso de la información que necesita, dejando en manos de los agentes más potentes los tratamientos de conocimiento más complejos de la infraestructura. A pesar de esto, la arquitectura DAEDALUS todavía posee limitaciones y retos a superar, en su mayoría inherentes a la problemática del paso de modelos complejos de alto nivel a las estructuras computacionales de los agentes y que dificulta la creación de nuevos agentes pedagógicos de forma sencilla por cualquier diseñador educativo. Es por ello que surge la necesidad como líneas futuras tanto de generación de una mayor librería de componentes educativos disponibles para los diseñados

res educativos como el estudio en mayor profundidad de mecanismos para facilitar la creación de nuevos agentes pedagógicos a partir de modelos, como se comenta en el siguiente apartado de trabajo futuro. No obstante, DAEDALUS sienta las bases sobre las cuales poder llegar a alcanzar estos retos.

A nivel puramente más educativo, una plataforma tecnológica de estas características puede servir para desarrollar nuevas y más motivadoras experiencias de aprendizaje ubicuo mediante escenarios pedagógicos innovadores y nuevos dispositivos móviles. El tener disponibles elementos tecnológicos que habiliten diseños pedagógicos innovadores basados en el aprendizaje móvil y colaborativo puede catalizar la puesta en práctica y el refinamiento de las distintas nuevas teorías educativas recientes. El objetivo último de la tecnología educativa consistirá en poder ofrecer un aprendizaje continuo, ubicuo y de por vida, que esté adaptado a las necesidades de los alumnos y que sepa aprovecharse de todas las características contextuales presentes en el entorno que les rodea, promoviendo la interacción con ese entorno y la colaboración con los demás alumnos.

11.4 Trabajo y Líneas Futuras

Este trabajo aporta el marco inicial sobre el cual se podrá dibujar más adelante todo un conjunto de distintos cuadros. A raíz de este trabajo se plantea toda una línea futura de investigación y desarrollo dentro del campo de las tecnologías educativas con agentes pedagógicos inteligentes. Más específicamente se puede contemplar las siguientes ideas de trabajo futuro:

➤ **Nuevos agentes pedagógicos**

La Arquitectura DAEDALUS sienta las bases pero hace falta comenzar a investigar más posibilidades para los agentes pedagógicos, lo que incluye identificar y crear más agentes generales y específicos para una mayor variedad de problemas educativos.

➤ **Nuevas necesidades pedagógicas**

Una de las necesidades que se ha planteado por ejemplo en este trabajo ha sido la de dar soporte a la continuidad del aprendizaje. Una cuestión que podría ser interesante de profundizar sería cómo desarrollar agentes que garanticen no sólo la posibilidad de continuar tareas pendientes y disponibilidad de recursos previamente generados, sino también garantizar la continuidad de

las tareas colaborativas en caso de estar interactuando a través de herramientas de colaboración síncronas y producirse un cambio de contexto.

➤ **Razonamiento avanzado y nuevos patrones cognitivos**

Otra cuestión clave en la que se centra el marco de desarrollo software que se ha situado en este trabajo es la coordinación necesaria en los sistemas de agentes, principalmente entre los agentes capaces de razonar y tomar decisiones. Un punto interesante de profundización puede resultar del estudio de esta coordinación para evitar posibles casos en los que distintos agentes tomen decisiones contrarias y distribuyan el resultado de esas decisiones a todas las demás entidades.

➤ **Aproximación basada en modelos.**

Un siguiente paso lógico de mejora de la arquitectura puede ser la incorporación al marco de trabajo DAEDALUS de una aproximación basada en modelos (Model Driven Architecture) y de herramientas de generación automática de autómatas de agentes micro-reactivos y de reglas de agentes cognitivos a partir de los diseños de modelos gráficos. Esto sería de gran utilidad ya que simplificaría aún más los desarrollos de sistemas educativos distribuidos mediante agentes pedagógicos.

➤ **Adaptación a nuevos dispositivos de última generación**

A medida que los medios tecnológicos avanzan, las posibilidades para la creación de escenarios de aprendizaje aumentan. Mediante el empleo de nuevos dispositivos móviles de última generación no disponibles al inicio de esta tesis, es posible plantearse ampliar las capacidades de los agentes hospedados en nodos móviles, habilitándose la posibilidad de ofrecer agentes cognitivos dentro de dichos dispositivos que puedan ofrecer mejores experiencias al aprendizaje ubicuo de los alumnos.

REFERENCIAS

- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: towns, buildings, construction*. Oxford University Press, USA.
- Amorim, R. R., Lama, M., Sánchez, E., Riera, A., & Vila, X. A. (2006). A learning design ontology based on the IMS specification. *JOURNAL OF EDUCATIONAL TECHNOLOGY AND SOCIETY*, 9(1), 38.
- Anido, L. E., Fernández, M. J., Caeiro, M., Santos, J. M., Rodríguez, J. S., & Llamas, M. (2002). Educational Metadata and Brokerage for Learning Resources. *Computers & Education*, 38(4), 351–374.
- Bachimont, B., Isaac, A., & Troncy, R. (2002). Semantic Commitment for Designing Ontologies: A Proposal. En *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web* (págs. 114-121). Springer-Verlag. Recuperado a partir de <http://portal.acm.org/citation.cfm?id=650886>
- Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A Survey on Context-Aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), 263-277.
- Barbosa, D., Augustin, I., Barbosa, J., Yamim, A., da Silva, L., Fernando, C., & Geyer, R. (2006). Learning in a large-scale pervasive environment. En *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*. doi:10.1109/PERCOMW.2006.74
- Barros, B. (1999). *Aprendizaje Colaborativo en Enseñanza a Distancia: Entorno genérico para configurar, realizar y analizar actividades en grupo* (Ph. D. Dissertation). Universidad Politécnica de Madrid.

- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77. doi:10.1109/2.796139
- Bellifemine, F., Caire, G., & Rimassa, G. (2001). Developing multi-agent systems with JADE. *Intelligent Agents VII Agent Theories Architectures and Languages*, 42-47.
- Bellifemine, F., Poggi, A., & Rimassa, G. (1999). JADE—A FIPA-compliant agent framework. *Proceedings of PAAM*, 99, 97-108.
- Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology*, 50(1-2), 10-21. doi:10.1016/j.infsof.2007.10.008
- Berri, J., Benlamri, R., & Atif, Y. (2006). Ontology-based framework for context-aware mobile learning. En *Proceedings of the 2006 international conference on Wireless communications and mobile computing* (págs. 1307-1310). ACM New York, NY, USA.
- Borgo, S., Guarino, N., & Masolo, C. (1996). A Pointless Theory of Space Based on Strong Connection and Congruence. En *In Proceedings of Principles of Knowledge Representation and Reasoning (KR96)* (págs. 220-229). Morgan Kaufmann Publishers Inc.
- Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. University of Twente, Enschede, NL—Centre for Telematica and Information Technology.
- Brickley, D., & Miller, L. (2007, Noviembre 2). FOAF Vocabulary Specification. Recuperado Diciembre 14, 2009, a partir de <http://xmlns.com/foaf/spec/>
- Brooks, F. P. (1995). *The mythical man-month* (anniversary ed.). Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Brown, J. S., Collins, A., & Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher*, 18(1), 32-42. doi:10.3102/0013189X018001032
- Brown, P. J. (1996). The stick-e document: a framework for creating context-aware applications. *Proceedings of the Electronic Publishing*, 8, 259-272.
- Capuano, N., Marsella, M., & Salerno, S. (2000). ABITS: An agent based Intelligent Tutoring System for distance learning. En *Proceedings of the In-*

- ternational Workshop in Adaptive and Intelligent Web-based Educational Systems. Available at <http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000> (págs. 17-28).
- Celorrio, C., & Verdejo, M. F. (2007). Adapted Activity Deployment and Configuration in a Pervasive Learning Framework. *Pervasive Learning 2007: Design challenges and Requirements*, 51-58.
- Celorrio, C., & Verdejo, M. F. (2008). An Interoperable, Extensible and Configurable Service Architecture for an Integrated Educational Networking Infrastructure. En *Advanced Learning Technologies, 2008. ICAALT '08. Eighth IEEE International Conference on* (págs. 207-211). Presented at the *Advanced Learning Technologies, 2008. ICAALT '08. Eighth IEEE International Conference on*. doi:10.1109/ICALT.2008.248
- Celorrio, C., Verdejo, M. F., & Barros, B. (2005). Una Aproximación a la Distribución de Repositorios de Objetos de Aprendizaje basada en Servicios Web. En *Actas VI Congreso Nacional de Informática Educativa Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación*. Presented at the SINTICE, Granada, Spain.
- Celorrio, C., Verdejo, M. F., & Barros, B. (2006). Un Modelo de Distribución de Repositorios para un Portal de Comunidades de Aprendizaje Colaborativo. *Informática Educativa Comunicaciones*, 2(3).
- Chan, T., Roschelle, J., Hsi, S., Sharples, M., Brown, T., & Patton, C. (2006). One-to-One Technology-Enhanced Learning: An Opportunity for Global Research Collaboration. *Research and Practice in Technology Enhanced Learning*, 1(1), 3-29.
- Chen, G., & Kotz, D. (2000). A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.
- Chen, H., Finin, T., & Joshi, A. (2004a). An Ontology for Context-Aware Pervasive Computing Environments. *The Knowledge Engineering Review*, 18(03), 197-207.
- Chen, H., Finin, T., & Joshi, A. (2004b). A pervasive computing ontology for user privacy protection in the context broker architecture (Technical Report No. TR-CS-04-08). Baltimore County: University of Maryland.
- Chen, H., Finin, T., & Joshi, A. (2005). The SOUPA Ontology for Pervasive Computing. En B. Birkhäuser (Ed.), *Ontologies for Agents: Theory and*

- Experiences, Whitestein Series in Software Agent Technologies (págs. 233-258). Springer.
- Chen, Y., T. C. Kao, & J. P. Sheu. (2003). A mobile learning system for scaffolding bird watching learning. *Journal of Computer Assisted Learning*, 19(3), 347-359. doi:10.1046/j.0266-4909.2003.00036.x
- Chess, D., Harrison, C., & Kershenbaum, A. (1997). Mobile agents: Are they a good idea? En *Mobile Object Systems Towards the Programmable Internet* (págs. 25-45). Recuperado a partir de http://dx.doi.org/10.1007/3-540-62852-5_4
- COLDEX. (2001). Collaborative Learning and Distributed Experimentation (IST-2001-32327). IST-2001-32327. Recuperado a partir de <http://www.coldex.info>
- Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46(1), 41-64. doi:10.1016/S0169-023X(02)00195-7
- Cross, N. (2001). Design cognition: Results from protocol and other empirical studies of design activity. *Design knowing and learning: Cognition in design education*, 79-103.
- Dessinger, J. C., & Moseley, J. L. (2004). Full-Scope Evaluation: Raising the Bar. En *Confirmative evaluation: Practical strategies for valuing continuous improvement*. Pfeiffer.
- Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1), 4-7. doi:10.1007/s007790170019
- Dillenbourg, P. (1999). What do you mean by collaborative learning? En P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches* (págs. 1-19). Amsterdam, NL: Pergamon, Elsevier Science.
- Drira, R., Tirellil, I., Laroussi, M., Derycke, A., & Benghezala, H. (2006). What can we adapt in a Mobile Learning Systems. En *Conference IMCL2006*, April (págs. 19-21). Citeseer.
- DROOLS. (2009). Drools 5 - JBoss Community. Recuperado Junio 30, 2010, a partir de <http://www.jboss.org/drools>

- Duval, E., & Hodgins, W. (2003). A LOM research agenda. En Proceedings of the twelfth international conference on World Wide Web (págs. 1-9). CiteSeer.
- ENLACE. (2005). ENLACE (Proyecto subvencionado por CICYT-TIN 2004-04232). Recuperado a partir de <http://enlace.uned.es>
- Esmahi, L., & Badidi, E. (2004). An agent-based framework for adaptive M-learning. *Lecture notes in computer science*, 749-758.
- Faux, F., McFarlane, A., Roche, N., & Facer, K. (2006). *Futurelab: Learning with handheld technologies*.
- FIPA. (2002a). FIPA Abstract Architecture Specification (No. SC00001L). Recuperado a partir de <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>
- FIPA. (2002b). FIPA Agent Management Specification (No. SC00023K). Recuperado a partir de <http://www.fipa.org/specs/fipa00023/SC00023K.pdf>
- FIPA. (2002c). FIPA Agent Message Transport Service Specification (No. SC00067F). Recuperado a partir de <http://www.fipa.org/specs/fipa00067/SC00067F.pdf>
- FIPA. (2002d). FIPA ACL Message Structure Specification (No. XC00091C). Recuperado a partir de <http://www.fipa.org/specs/fipa00061/SC00061G.pdf>
- Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1), 17-37. doi:10.1016/0004-3702(82)90020-0
- Francisco J Garijo, Spina, D., & Polo, F. (2009). *ICARO Mini. Manual de Uso de Componentes*.
- Franklin, S., & Graesser, A. (1997). Is It an agent, or just a program?: A taxonomy for autonomous agents. En *Intelligent Agents III Agent Theories, Architectures, and Languages* (págs. 21-35). Recuperado Mayo 16, 2009, a partir de <http://dx.doi.org/10.1007/BFb0013570>.
- Friesen, N., Roberts, A., & Fisher, S. (2002). CanCore: Metadata for Learning Objects. *Canadian Journal of Learning and Technology*, 28(3), 43-53.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1993). Design patterns: Abstraction and reuse of object-oriented design. En *ECOOP'93-object-oriented programming, 7th European Conference, Kaiserslauten, Germany, July 26-30, 1993: proceedings* (pág. 406).

- Garijo, F. J., Bravo, S., Gonzalez, J., & Bobadilla, E. (2004). BOGAR_LN: An Agent Based Component Framework for Developing Multi-modal Services Using Natural Language. En *Current Topics in Artificial Intelligence* (Vol. 3040, págs. 207-220). Springer Berlin / Heidelberg. Recuperado a partir de <http://www.springerlink.com/content/kywadkmcqug8km99>
- Gascueña, J. M., & Fernández-Caballero, A. (2005). An Agent-based Intelligent Tutoring System for Enhancing E-learning/E-teaching. *International Journal of Instructional Technology and Distance Learning*, 2(11), 11-24.
- Goh, T., & Kinshuk. (2002). A discussion on mobile agent based mobile web-based ITS. En *Computers in Education, 2002. Proceedings. International Conference on* (págs. 1514-1515 vol.2). Presented at the *Computers in Education, 2002. Proceedings. International Conference on*.
- Gómez-Sanz, J., & Pavón, J. (2004). Methodologies for developing multi-agent systems. *Journal of Universal Computer Science*, 10(4), 359–374.
- Gómez-Sanz, J. (2002). MODELADO DE SISTEMAS MULTI-AGENTE. UCM. Recuperado a partir de <http://grasia.fdi.ucm.es/main/myfiles/tesisIngenias.pdf>
- Gómez-Sanz, J. J., Pavón, J., & Garijo, F. (2002). Meta-models for building multi-agent systems. En *Proceedings of the 2002 ACM symposium on Applied computing* (págs. 37-41). Madrid, Spain: ACM. doi:10.1145/508791.508801
- Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Commun. ACM*, 28(3), 300-311. doi:10.1145/3166.3170
- Graf, S. M. C., Liu, K., Chang, T. C., Wen, M., Tan, D., Dron, Q., Lin, J., et al. (2008). An Infrastructure for Developing Pervasive Learning Environments. En *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on* (págs. 389-394).
- Gregg, D. G. (2007). E-learning agents. *The Learning Organization*, 14(4), 300 - 312. doi:10.1108/09696470710749245
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5, 199–220.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5), 907–928.

- Gu, T., Pung, H., & Zhang, D. (2004). Toward an OSGi-based infrastructure for context-aware applications. *Pervasive Computing, IEEE*, 3(4), 66-74. doi:10.1109/MPRV.2004.19
- Hatala, M., Richards, G., Eap, T., & Willms, J. (2004). The interoperability of learning object repositories and services: standards, implementations and lessons learned. En *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (págs. 19-27). ACM New York, NY, USA.
- Heckmann, D. (2005). Ubiquitous user modeling. Universität des Saarlandes.
- Heery, R., & Patel, M. (2000). Application profiles: mixing and matching metadata schemas. *Ariadne*, 25, 2004-3.
- Henricksen, K., Indulska, J., & Rakotonirainy, A. (2003). Generating context management infrastructure from high-level context models. En *Proceedings of the 4th International Conference on Mobile Data Management* (págs. 1-6).
- Henricksen, K., & Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1), 37-64. doi:10.1016/j.pmcj.2005.07.003
- Hobbs, J. R., & Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1), 66-85.
- Hoppe, U., Pinkwart, N., Oelinger, M., Zeini, S., Verdejo, M. F., Barros, B., & Mayorga, J. I. (2005). Building bridges within learning communities through ontologies and "thematic objects". En *Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!* (págs. 211-220). International Society of the Learning Sciences.
- ICARO-T. (2008). Proyecto ICARO: Desarrollo de Aplicaciones Distribuidas con Organizaciones de Agentes. Recuperado Junio 30, 2010, a partir de <http://icaro.morfeo-project.org/>
- IEEE Standards Department. (2002, Julio 15). IEEE 1484.12.1-2002, Learning Object Metadata Standard. Recuperado Junio 22, 2009, a partir de http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
- IMS GLC. (2003, Febrero). IMS Learning Design Specification. Recuperado Diciembre 16, 2009, a partir de <http://www.imsglobal.org/learningdesign/>

- IMS GLC. (2005, Enero 4). IMS Learner Information Package Version 1.0.1 Final Specification. Recuperado Diciembre 16, 2009, a partir de <http://www.imsglobal.org/profiles/>
- Jafari, A. (2002). Conceptualizing Intelligent Agents for Teaching and Learning. *Educause Quarterly*, 25(3), 28-34.
- Johnson, R. E. (1992). Documenting frameworks using patterns. En conference proceedings on Object-oriented programming systems, languages, and applications (págs. 63-76). Vancouver, British Columbia, Canada: ACM. doi:10.1145/141936.141943
- de Jong, T. (2006a). Computer simulations: Technological advances in inquiry learning (Vol. 312).
- de Jong, T. (2006b). Scaffolds for scientific discovery learning. Handling complexity in learning environments: Theory and research, 107–128.
- Kagal, L., Finin, T., & Joshi, A. (2003). A policy based approach to security for the semantic web. *Lecture Notes in Computer Science*, 402–418.
- KALEIDOSCOPE. (2008). Kaleidoscope Network of Excellence. Recuperado a partir de <http://www.noe-kaleidoscope.org/>
- Kalyanpur, A., Pastor, D., Battle, S., & Padget, J. (2004). Automatic mapping of OWL ontologies into Java. En SEKE (págs. 98–103).
- Kinshuk, L. T. (2004). Improving mobile learning environments by applying mobile agents technology. En Third Pan Commonwealth Forum on Open Learning, Dunedin, New Zealand, July (págs. 1–3).
- Ko, J., Hur, C., & Kim, H. (2005). A Personalized Mobile Learning System Using Multi-agent. En *Web Information Systems Engineering – WISE 2005 Workshops* (págs. 144-151). Recuperado a partir de http://dx.doi.org/10.1007/11581116_15
- Koschmann, T. (1996). Paradigm shifts and instructional technology: An introduction. CSCL, *Theory and Practice of an Emerging Paradigm: Theory and Practice of an Emerging Paradigm*.
- Krasner, G. E., & Pope, S. T. (1988). A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3), 26-49.
- Lange, D. B., & Oshima, M. (1999). Seven good reasons for mobile agents. *Commun. ACM*, 42(3), 88-89. doi:10.1145/295685.298136

- Laukkanen, M., Tarkoma, S., & Leinonen, J. (2002). FIPA-OS Agent Platform for Small-Footprint Devices. En *Intelligent Agents VIII* (págs. 447-460). Recuperado a partir de http://dx.doi.org/10.1007/3-540-45448-9_34
- Ley, D. (2007). Ubiquitous Computing. *Emerging Technologies for Learning*, 2, 64-79.
- Lonsdale, P., Baber, C., Sharples, M., & Arvanitis, T. N. (2004). A context-awareness architecture for facilitating mobile learning. En J. Attewell & C. Savill-Smith (Eds.), *Learning with mobile devices: research and development – a book of papers*. (págs. 79-85). London: Learning and Skills Development Agency.
- Martín, E., & Carro, R. M. (2009). Supporting the Development of Mobile Adaptive Learning Environments: A Case Study. *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES*, 2(1), 23-36.
- Mascardi, V., Cordì, V., & Rosso, P. (2007). A comparison of upper ontologies. En *Proc. Conf. on Agenti e industria: Applicazioni tecnologiche degli agenti software, WOA07* (págs. 55-64). Presented at the *Agenti e industria: Applicazioni tecnologiche degli agenti software, WOA07*, Genova, Italy.
- Matuszek, C., Cabral, J., Witbrock, M., & DeOliveira, J. (2006). An introduction to the syntax and content of Cyc. En *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering* (págs. 44-49).
- Mayorga, J. I., Celorrio, C., Lorenzo, E. J., Velez, J., Barros, B., & Verdejo, M. F. (2007). Comunidades Virtuales de Aprendizaje Colaborativo: de los Metadatos a la Semántica. *Inteligencia Artificial*, 11(33), 47-60.
- Mccarthy, J., & Buvac, S. (1998). Formalizing context (expanded notes). *Computing natural language*, 81, 13-50.
- Milrad M., Gottdenker, J., Hoppe, U, Verdejo, F, Baloian, N., & Pinazo A. (2002). Collaborative Learning Scenarios. COLDEX Deliverable 2.2.2.
- Mizoguchi, R., & Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal of Artificial Intelligence in Education*, 11(2), 107-121.
- Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM*, 36(6), 24-28.

- Naismith, L., & Futurelab, N. (2004). Literature review in mobile technologies and learning. Nesta Futurelab.
- Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, 18, 87-127.
- Nino, C. P., Marques, J., Barbosa, D. N. F., Barbosa, J. L. V., Geyer, C. F. R., & Augustin, I. (2007). Context-Aware Model in an Ubiquitous Learning Environment. En 3rd International Workshop on Pervasive Learning (PerEL), New York. Proceedings of PerEL (págs. 1-5).
- O'Brien, P. D., & Nicol, R. C. (1998). FIPA — Towards a Standard for Software Agents. *BT Technology Journal*, 16(3), 51-59. doi:10.1023/A:1009621729979
- Ogata, H., & Yano, Y. (2004). Context-aware support for computer-supported ubiquitous learning. En *Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on* (págs. 27-34). Presented at the *Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on*.
- OSGi. (1999). OSGi The Dynamic Module System for Java. Recuperado Junio 30, 2010, a partir de <http://www.osgi.org/Main/HomePage>
- Otero, N., Vala, A., Paiva, A., Oelinger, M., Verdejo, M. F., Barros, B., Calero, Y., et al. (2003). Local Scenarios - Description of System Prototypes. COLDEX Deliverable D.5.2.1/D.5.3.1.
- Papert, S. (1988). The conservation of Piaget: The computer as grist for the constructivist mill. *Constructivism in the computer age*, 3-13.
- Paramythis, A., & Loidl-Reisinger, S. (2004). Adaptive Learning Environments and e-Learning Standards. *Electronic Journal on e-Learning Volume*, 2(1), 181-194.
- Pavon, J., Garijo, F., & Gomez-Sanz, J. (2008). Complex Systems and Agent-Oriented Software Engineering. *Lecture Notes in Computer Science*, 5049, 3-16. doi:10.1007/978-3-540-85029-8_2
- Piaget, J., & Brown, T. (1985). The equilibration of cognitive structures: The central problem of intellectual development. University of Chicago Press Chicago.
- Poslad, S., Buckle, P., & Hadingham, R. (2000). The FIPA-OS agent platform: Open source for open standards. En *Proceedings of the 5th International*

- Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (págs. 355–368).
- Pressey, S. L. (1926). A simple apparatus which gives tests and scores—and teaches. *School and Society*, 23, 373-376.
- Rellermeyer, J. S., Alonso, G., & Roscoe, T. (2007). R-OSGi: Distributed applications through software modularization. *Lecture Notes in Computer Science*, 4834, 1.
- Rogers, Y., Price, S., Fitzpatrick, G., Fleck, R., Harris, E., Smith, H., Randell, C., et al. (2004). Ambient wood: designing new forms of digital augmentation for learning outdoors. En *Proceedings of the 2004 conference on Interaction design and children: building a community* (págs. 3-10). ACM New York, NY, USA.
- Roschelle, J. (2003). Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning*, 19(3), 260-272.
- Rosson, M. B., & Carroll, J. M. (2002a). Scenario-based design. En *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, 1032-1050. L. Erlbaum Associates Inc. <http://portal.acm.org/citation.cfm?id=772137>.
- Rosson, M. B., & Carroll, J. M. (2002). *Usability engineering: scenario-based development of human-computer interaction*. San Francisco: Morgan Kaufmann Publishers Inc.
- Sánchez, F., Pinazo, A., Martínez, R, Sebastian, J.M., Aracil, R., Baloian, N., Oelinger, M., et al. (2004). Remote Scenarios - Description of System Prototypes. COLDEX Deliverable D4.2.1 / D4.3.1.
- Scardamalia, M., & Bereiter, C. (1994). Computer Support for Knowledge-Building Communities. *The Journal of the Learning Sciences*, 3(3), 265-283. doi:0.1207/s15327809jls0303_3
- Schilit, B., & Theimer, M. M. (1994). Disseminating Active Map Information to Mobile Hosts. *IEEE Network*, 8(5), 22-32.
- Schmidt, A. (2007). *Impact of Context-Awareness on the Architecture of Learning Support Systems, Architecture Solutions for E-Learning Systems*. Idea-Group Publishing.
- Schmidt, A., Beigl, M., & Gellersen, H. W. (1999). There is more to context than location. *Computers & Graphics*, 23(6), 893-901.

- Scriven, M. (1967). The methodology of evaluation. En R. W. Tyler, R. M. Gagné, & M. Scriven (Eds.), *Perspectives of curriculum evaluation* (págs. 39-83). Chicago: Rand McNally.
- Sharples, M. (2000). The design of personal mobile technologies for lifelong learning. *Computers & Education*, 34(3-4), 177-193.
- Sharples, M., Taylor, J., & Vavoula, G. (2005). Towards a theory of mobile learning. En *Proceedings of MLearn 2005 conference*.
- Skinner, B. F. (1965). The technology of teaching. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character*. Royal Society (Great Britain), 162(989), 427.
- Sleeman, D., & Brown, J. S. (1982). *Intelligent tutoring systems*. Academic Press.
- Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R., et al. (1999). Science in the Palms of Their Hands. *Communications of the ACM*, 42(8), 21.
- Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. *Cambridge handbook of the learning sciences*, 2006.
- Stahl, G. (2000). Collaborative information environments to support knowledge construction by communities. *AI & Society*, 14(1), 71-97. doi:10.1007/BF01206129
- Strang, T., & Linnhoff-Popien, C. (2004). A context modeling survey. En *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp*.
- Tarkoma, S. (2003). Distributed event dissemination for ubiquitous agents. En *10th ISPE International Conference on Concurrent Engineering (CE-2003)* (págs. 105-110).
- Tatar, D., Roschelle, J., Vahey, P., & Penuel, W. R. (2003). Handhelds go to school: Lessons learned. *Computer*, 36(9), 30-37.
- Thiagarajan, S. (1991). *Formative Evaluation in Performance Technology*. *Performance Improvement Quarterly*, 4(2), 22-34. doi:10.1111/j.1937-8327.1991.tb00501.x
- Thordike, E. (1912). *Education: A First Book*. New York, Macmillan.

- Trifonova, A., & Ronchetti, M. (2004). A general architecture to support mobility in learning. En *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on* (págs. 26-30). Presented at the *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*. doi:10.1109/ICALT.2004.1357368
- Vavoula, G., & Sharples, M. (2002). KLeOS: a personal, mobile, knowledge and learning organisation system. En *Wireless and Mobile Technologies in Education, 2002. Proceedings. IEEE International Workshop on* (págs. 152-156). Presented at the *Wireless and Mobile Technologies in Education, 2002. Proceedings. IEEE International Workshop on*. doi:10.1109/WMTE.2002.1039239
- Vélez, J. (2009). *Pelican. Una plataforma para el diseño y desarrollo de escenarios de aprendizaje colaborativo. Soporte a los aspectos dinámicos*. (Ph. D. Dissertation). UNED.
- Verdejo, M. F., & Celorrio, C. (2007). A Multi-Agent Based System for Activity Configuration and Personalization in a Pervasive Learning Framework. En *Pervasive Computing and Communications Workshops, IEEE International Conference on* (págs. 177-181). New York, USA: IEEE Computer Society. doi:http://doi.ieeecomputersociety.org/10.1109/PERCOMW.2007.8
- Verdejo, M. F., Celorrio, C., & Lorenzo, E. J. (2006). Improving Learning Object Description Mechanisms to Support an Integrated Framework for Ubiquitous Learning Scenarios. En *Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, 2006. WMUTE'06* (págs. 93-97).
- Verdejo, M. F., Celorrio, C., Lorenzo, E. J., Millan, M., Prades, S., & Velez, J. (2009). Constructing Mobile Technology-Enabled Environments for an Integrated Learning Approach. En *Innovative Mobile Learning: Techniques and Technologies* (pág. 145). IGI Global.
- Verdejo, M. F., Celorrio, C., Lorenzo, E. J., Ruiz, A., & Sastre, T. (2007). Sustaining learning activity flow in a framework for ubiquitous learning. En *BEYOND MOBILE LEARNING WORKSHOP* (págs. 43-53).
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. *Mind in society: The development of higher psychological process*, 16(1).

- W3C. (2004a). RDF Vocabulary Description Language 1.0: RDF Schema. (D. Brickley, R. Guha, & B. McBride, Eds.)W3C recommendation, 10. Recuperado a partir de <http://www.w3.org/TR/rdf-schema/>
- W3C. (2004b). Owl web ontology language overview. (D. L. McGuinness & F. van Harmelen, Eds.)W3C recommendation, 10. Recuperado a partir de <http://www.w3.org/TR/owl-features/>
- W3C. (2009, Octubre 27). OWL 2 Web Ontology Language Document Overview. W3C Recommendation, . Recuperado Septiembre 3, 2009, a partir de <http://www.w3.org/TR/owl2-overview/>
- Wang, Y. (2004). Context awareness and adaptation in mobile learning. En *Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on* (págs. 154-158). Presented at the *Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on*.
- Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1), 91–102.
- Wasserman, A. I., & Shewmake, D. T. (1982). Rapid prototyping of interactive information systems. En *Proceedings of the workshop on Rapid prototyping* (págs. 171-180). ACM New York, NY, USA.
- Weibel, S., Kunze, J., Lagoze, C., & Wolf, M. (1998). Dublin core metadata for resource discovery. *The Internet Society*.
- Weiser, M. (1991). The Computer for the 21st Century. *Scientific American*, 265(3), 94-101.
- Weiser, M. (1993). Ubiquitous computing. *IEEE Computer*, 26(10), 71-72.
- Wenger, E. (1987). *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
- Wiley, D. A. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Wiley, D.A. (Ed.): *The instructional use of learning objects*, 571–577.
- William, U. C., Griswold, W. G., Shanahan, P., Brown, S. W., Boyer, R., & Ratto, M. (2003). ActiveCampus - Experiments in Community-Oriented. *IEEE COMPUTER*, 37, 73--81.

- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2), 115-152.
- Yau, J., & Joy, M. (2007). Architecture of a Context-aware and Adaptive Learning Schedule for Learning Java. En *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on* (págs. 252-256). Presented at the *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*. doi:10.1109/ICALT.2007.72
- Ye, J., Coyle, L., Dobson, S., & Nixon, P. (2007). Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review*, 22(04), 315-347.
- Zimmermann, A., Lorenz, A., & Oppermann, R. (2007). An Operational Definition of Context. En *Modeling and Using Context* (págs. 558-571). Recuperado a partir de http://dx.doi.org/10.1007/978-3-540-74255-5_42

PARTE IV

Apéndices

APÉNDICE A. IMPLEMENTACIÓN

En este apéndice procederemos a comentar los detalles de la implementación realizada de la arquitectura DAEDALUS. En primer lugar daremos una visión global acerca de cómo está implementada la arquitectura DAEDALUS a escala general. Después, se expondrá más de cerca un componente clave del Nivel de Infraestructura, como es el recurso de Comunicación. Acto seguido, comentaremos como se hace uso del modelo de conocimiento del Nivel de Semántica y Contexto, y cómo se produce el mapeo de la ontología dentro de la arquitectura, para que pueda ser utilizada dentro de cualquier tipo de dispositivo de la infraestructura. Finalmente, detallaremos la implementación de los componentes de la Organización Educativa que se han empleado para el escenario de ENLACE.

LA ARQUITECTURA DAEDALUS

En la Figura se puede ver un esquema descriptivo de la arquitectura DAEDALUS. Toda la implementación de DAEDALUS está realizada mediante Java. Gracias a los distintos perfiles de Java es posible utilizar el mismo lenguaje de programación para servidores con contenedores web (J2EE), ordenadores de sobremesa (J2SE) y dispositivos móviles (J2ME). En términos de implementación, una característica notable ha sido la adaptación de ICAROT para poder funcionar sobre el estándar OSGi.

OSGi (OSGi, 1999) es un modelo de despliegue dinámico de módulos (conocidos como *bundles*) para Java. Las ventajas de esto son dobles, por un lado OSGi habilita la instalación y gestión remota de componentes. Esto permite a DAEDALUS instalar y desinstalar herramientas de aprendizaje en cualquier nodo desde cualquier nodo.

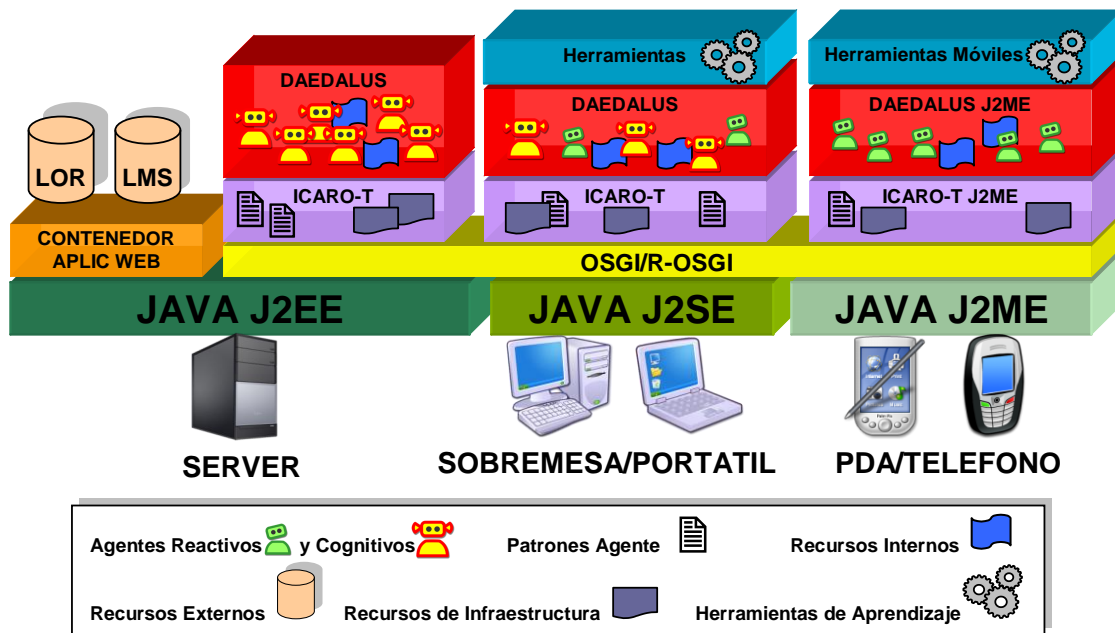


Figura A.1 Descripción de la arquitectura

Por otro lado, el uso de OSGi aporta como ventaja añadida la posibilidad de utilizar R-OSGi para implementar el recurso de comunicación, como se explica en el siguiente apartado.

IMPLEMENTACIÓN DEL RECURSO DE COMUNICACIÓN

Todos los nodos comparten la misma implementación del recurso de comunicación. Esta implementación se realiza mediante R-OSGi (Rellermeyer, Alonso, & Roscoe, 2007). R-OSGi permite la distribución de aplicaciones OSGi al conseguir que se puedan configurar módulos accesibles de forma remota transparentemente.

El recurso de comunicación hace uso de las capacidades de R-OSGi para ofrecer un sistema de descubrimiento de nodos, y un sistema de envío/recepción de mensajes, junto con un sistema de suscripción/entrega de eventos.

IMPLEMENTACIÓN DEL MODELO DE CONOCIMIENTO DE DAEDALUS

A pesar de que el modelo de contexto es totalmente trasladable a un lenguaje ontológico como OWL, se ha preferido apostar por una implementación orientada a objetos, usando técnicas de mapeo automático desde OWL a Java (Kalyanpur, Pastor, Battle, & Padget, 2004), por varios motivos. En primer lugar por la sencillez que un modelo orientado a objetos aporta y que facilita que sea fácilmente manipulable en dispositivos móviles. También, al tener codificados los modelos como clases y objetos, la distribución de éstos es directa usando las facilidades que aporta R-OSGI. Finalmente, debido a que la implementación del motor de conocimiento del agente cognitivo internamente usa DROOLS. DROOLS (DROOLS, 2009) es un motor de reglas basado en RETE-OO, una adaptación del algoritmo RETE (Forgy, 1982) para el funcionamiento orientado a objetos. Al tener una representación del conocimiento directamente manipulable por los agentes cognitivos facilita también pues la implementación.

IMPLEMENTACIÓN DE LA ORGANIZACIÓN DEL ESCENARIO DE ENLACE

Los agentes cognitivos se definen a partir de objetivos y tareas. Siguiendo el patrón cognitivo de ICARO-T, a la hora de implementar dichos elementos es necesario identificar primeramente los objetivos, diseñar las tareas, e incorporar los mecanismos de inferencia en forma de reglas que determinarán las secuencias de acciones que se ejecutarán para resolver los objetivos. Estas reglas se implementan como se ha comentado anteriormente empleando el lenguaje de reglas que provee DROOLS.

Los agentes micro-reactivos en cambio se implementan como una máquina de estados, en la que hay que definir los posibles estados y las transiciones, a modo de acciones a ejecutar por el agente. Esta máquina de estados se describe mediante un archivo XML.

Los recursos se implementan de forma variada. Por ejemplo, los recursos de recubrimiento del repositorio de objetos de aprendizaje y del gestor de usuarios y actividades se implementan mediante servicios web. La base de datos de contexto se realiza usando My-SQL y *Hibernate* como capa de conversión de los objetos Java a modelo entidad-relación.

APÉNDICE B. ONTOLOGÍA OWL/RDF

LISTADO DE LA ONTOLOGÍA DE DAEDALUS EN FORMATO RDF/XML

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY daedalus "http://enlace.uned.es/ontologies/2009/8/daedalus.owl#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2006/12/owl2-xml#"
  xml:base="http://enlace.uned.es/ontologies/2009/8/daedalus.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:daedalus="http://enlace.uned.es/ontologies/2009/8/daedalus.owl#">
  <owl:Ontology rdf:about=""/>

  <!--
  //////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////
  -->
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#confidence -->
<owl:ObjectProperty rdf:about="#confidence">
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#connectedTo -->
<owl:ObjectProperty rdf:about="#connectedTo">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:domain rdf:resource="#OrganizationNode"/>
  <rdfs:range rdf:resource="#OrganizationNode"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#contextProperty -->
<owl:ObjectProperty rdf:about="#contextProperty"/>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#hasMember -->
<owl:ObjectProperty rdf:about="#hasMember">
  <rdfs:domain rdf:resource="#Group"/>
  <rdfs:range rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#hasResource -->
<owl:ObjectProperty rdf:about="#hasResource">
  <rdfs:domain rdf:resource="#OrganizationNode"/>
  <rdfs:range rdf:resource="#Resource"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#hasType -->
<owl:ObjectProperty rdf:about="#hasType">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#LearningObject"/>
  <rdfs:range rdf:resource="#LearningObjectType"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#isMemberOf -->
<owl:ObjectProperty rdf:about="#isMemberOf">
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
  <owl:inverseOf rdf:resource="#hasMember"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#isRequiredBy -->
<owl:ObjectProperty rdf:about="#isRequiredBy">
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
  <owl:inverseOf rdf:resource="#requires"/>
```

```
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#isTypeOf -->
<owl:ObjectProperty rdf:about="#isTypeOf">
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
  <owl:inverseOf rdf:resource="#hasType"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#object -->
<owl:ObjectProperty rdf:about="#object">
  <rdfs:range rdf:resource="#ContextEntity"/>
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#predicate -->
<owl:ObjectProperty rdf:about="#predicate">
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#privacy -->
<owl:ObjectProperty rdf:about="#privacy">
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#provenance -->
<owl:ObjectProperty rdf:about="#provenance">
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#requires -->
<owl:ObjectProperty rdf:about="#requires">
  <rdfs:domain rdf:resource="#LearningActivity"/>
  <rdfs:range rdf:resource="#LearningResource"/>
  <rdfs:subPropertyOf rdf:resource="#contextProperty"/>
</owl:ObjectProperty>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#statementProperty -->
<owl:ObjectProperty rdf:about="#statementProperty"/>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#subject -->
<owl:ObjectProperty rdf:about="#subject">
  <rdfs:range rdf:resource="#ContextEntity"/>
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#temporalValidity -->
<owl:ObjectProperty rdf:about="#temporalValidity">
  <rdfs:domain rdf:resource="#ContextStatement"/>
  <rdfs:subPropertyOf rdf:resource="#statementProperty"/>
</owl:ObjectProperty>

<!--//////////////////////////////////////
//
// Classes
//
//////////////////////////////////////
-->

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Activity -->
<owl:Class rdf:about="#Activity">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Administrator -->
<owl:Class rdf:about="#Administrator">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Agent -->
<owl:Class rdf:about="#Agent">
  <rdfs:subClassOf rdf:resource="#OrganizationComponent"/>
  <rdfs:subClassOf rdf:resource="#SocialUnit"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#AgentGroup -->
<owl:Class rdf:about="#AgentGroup">
  <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Chat -->
<owl:Class rdf:about="#Chat">
  <rdfs:subClassOf rdf:resource="#CommunicationTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ClassroomGroup -->
<owl:Class rdf:about="#ClassroomGroup">
  <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ClassroomLocation -->
<owl:Class rdf:about="#ClassroomLocation">
  <rdfs:subClassOf rdf:resource="#SchoolLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#CommunicationTool -->
<owl:Class rdf:about="#CommunicationTool">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#CommunityEvent -->
<owl:Class rdf:about="#CommunityEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#CommunityGroup -->
<owl:Class rdf:about="#CommunityGroup">
  <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ComputerLabLocation -->
<owl:Class rdf:about="#ComputerLabLocation">
  <rdfs:subClassOf rdf:resource="#LabLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ContextEntity -->
<owl:Class rdf:about="#ContextEntity"/>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ContextEvent -->
<owl:Class rdf:about="#ContextEvent"/>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ContextStatement -->
<owl:Class rdf:about="#ContextStatement"/>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#CoordinatesLocation -->
<owl:Class rdf:about="#CoordinatesLocation">
  <rdfs:subClassOf rdf:resource="#Location"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#DataCapturingTool -->
<owl:Class rdf:about="#DataCapturingTool">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#DesktopComputer -->
<owl:Class rdf:about="#DesktopComputer">
  <rdfs:subClassOf rdf:resource="#StaticDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Device -->
<owl:Class rdf:about="#Device">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
  <owl:disjointWith rdf:resource="#OrganizationNode"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#EReader -->
<owl:Class rdf:about="#EReader">
  <rdfs:subClassOf rdf:resource="#MobileDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#EWhiteBoard -->
<owl:Class rdf:about="#EWhiteBoard">
  <rdfs:subClassOf rdf:resource="#StaticDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#EducationalArtefactEvent --
>
<owl:Class rdf:about="#EducationalArtefactEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#FieldTripLocation -->
<owl:Class rdf:about="#FieldTripLocation">
  <rdfs:subClassOf rdf:resource="#LogicalLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#FieldTripMonitor -->
<owl:Class rdf:about="#FieldTripMonitor">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>
```



```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Forum -->
<owl:Class rdf:about="#Forum">
  <rdfs:subClassOf rdf:resource="#CommunicationTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Group -->
<owl:Class rdf:about="#Group">
  <rdfs:subClassOf rdf:resource="#SocialUnit"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#IndividualEvent -->
<owl:Class rdf:about="#IndividualEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#InfrastructureEvent -->
<owl:Class rdf:about="#InfrastructureEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#InstructionalDesigner -->
<owl:Class rdf:about="#InstructionalDesigner">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LMS -->
<owl:Class rdf:about="#LMS">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LOR -->

<owl:Class rdf:about="#LOR">
  <rdfs:subClassOf rdf:resource="#Repository"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LOVisualizationTool -->
<owl:Class rdf:about="#LOVisualizationTool">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LabLocation -->
<owl:Class rdf:about="#LabLocation">
  <rdfs:subClassOf rdf:resource="#LogicalLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Laptop -->
<owl:Class rdf:about="#Laptop">
  <rdfs:subClassOf rdf:resource="#MobileDevice"/>
  <rdfs:subClassOf rdf:resource="#StaticDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Learner -->
<owl:Class rdf:about="#Learner">
  <rdfs:subClassOf rdf:resource="#Role"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LearningActivity -->
<owl:Class rdf:about="#LearningActivity">
  <rdfs:subClassOf rdf:resource="#Activity"/>
  <owl:disjointWith rdf:resource="#SupportActivity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LearningObject -->
<owl:Class rdf:about="#LearningObject">
  <rdfs:subClassOf rdf:resource="#LearningResource"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LearningObjectType -->
<owl:Class rdf:about="#LearningObjectType">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LearningResource -->
<owl:Class rdf:about="#LearningResource">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LearningTool -->
<owl:Class rdf:about="#LearningTool">
  <rdfs:subClassOf rdf:resource="#LearningResource"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Location -->
<owl:Class rdf:about="#Location">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#LogicalLocation -->
<owl:Class rdf:about="#LogicalLocation">
  <rdfs:subClassOf rdf:resource="#Location"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#MobileDevice -->
<owl:Class rdf:about="#MobileDevice">
  <rdfs:subClassOf rdf:resource="#Device"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ModelingTool -->
<owl:Class rdf:about="#ModelingTool">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#MuseumLocation -->
<owl:Class rdf:about="#MuseumLocation">
  <rdfs:subClassOf rdf:resource="#FieldTripLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#NatureParkLocation -->
<owl:Class rdf:about="#NatureParkLocation">
  <rdfs:subClassOf rdf:resource="#FieldTripLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#OrganizationComponent -->
<owl:Class rdf:about="#OrganizationComponent">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#OrganizationNode -->
<owl:Class rdf:about="#OrganizationNode">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#PDA -->
<owl:Class rdf:about="#PDA">
  <rdfs:subClassOf rdf:resource="#MobileDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#PedagogicProcessEvent -->
<owl:Class rdf:about="#PedagogicProcessEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Person -->
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf rdf:resource="#SocialUnit"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Project -->
<owl:Class rdf:about="#Project">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Projector -->
<owl:Class rdf:about="#Projector">
  <rdfs:subClassOf rdf:resource="#StaticDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#RangeLocation -->
<owl:Class rdf:about="#RangeLocation">
  <rdfs:subClassOf rdf:resource="#Location"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Repository -->
<owl:Class rdf:about="#Repository">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Resource -->
<owl:Class rdf:about="#Resource">
  <rdfs:subClassOf rdf:resource="#OrganizationComponent"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Role -->
<owl:Class rdf:about="#Role">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#ScheduledActivity -->
<owl:Class rdf:about="#ScheduledActivity">
  <rdfs:subClassOf rdf:resource="#Activity"/>
  <owl:disjointWith rdf:resource="#SpontaneousActivity"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SchoolGroup -->
<owl:Class rdf:about="#SchoolGroup">
  <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SchoolLocation -->
<owl:Class rdf:about="#SchoolLocation">
  <rdfs:subClassOf rdf:resource="#LogicalLocation"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Server -->
<owl:Class rdf:about="#Server">
  <rdfs:subClassOf rdf:resource="#StaticDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SimulationTool -->
<owl:Class rdf:about="#SimulationTool">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SmartPhone -->
<owl:Class rdf:about="#SmartPhone">
  <rdfs:subClassOf rdf:resource="#MobileDevice"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SocialUnit -->
<owl:Class rdf:about="#SocialUnit">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SpatioTemporalEvent -->
<owl:Class rdf:about="#SpatioTemporalEvent">
  <rdfs:subClassOf rdf:resource="#ContextEvent"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SpontaneousActivity -->
<owl:Class rdf:about="#SpontaneousActivity">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Staff -->
<owl:Class rdf:about="#Staff">
  <rdfs:subClassOf rdf:resource="#Role"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#StaticDevice -->
<owl:Class rdf:about="#StaticDevice">
  <rdfs:subClassOf rdf:resource="#Device"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#SupportActivity -->
<owl:Class rdf:about="#SupportActivity">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Task -->
<owl:Class rdf:about="#Task">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Teacher -->
<owl:Class rdf:about="#Teacher">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#WebBrowser -->
<owl:Class rdf:about="#WebBrowser">
  <rdfs:subClassOf rdf:resource="#LearningTool"/>
</owl:Class>
```

```
<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#Wiki -->
<owl:Class rdf:about="#Wiki">
  <rdfs:subClassOf rdf:resource="#CommunicationTool"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#WorkGroup -->
<owl:Class rdf:about="#WorkGroup">
  <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>

<!-- http://enlace.uned.es/ontologies/2009/8/daedalus.owl#WorkLocation -->
<owl:Class rdf:about="#WorkLocation">
  <rdfs:subClassOf rdf:resource="#LogicalLocation"/>
</owl:Class>

</rdf:RDF>
```


APÉNDICE C. DIAGRAMAS AGENTES PEDAGÓGICOS

A continuación se muestran algunos de los distintos diagramas correspondientes a los agentes pedagógicos diseñados en los escenarios de aplicación presentados en los capítulos 9 y 10.

DIAGRAMAS AGENTE MICRO-REACTIVO PERSONALAGENT

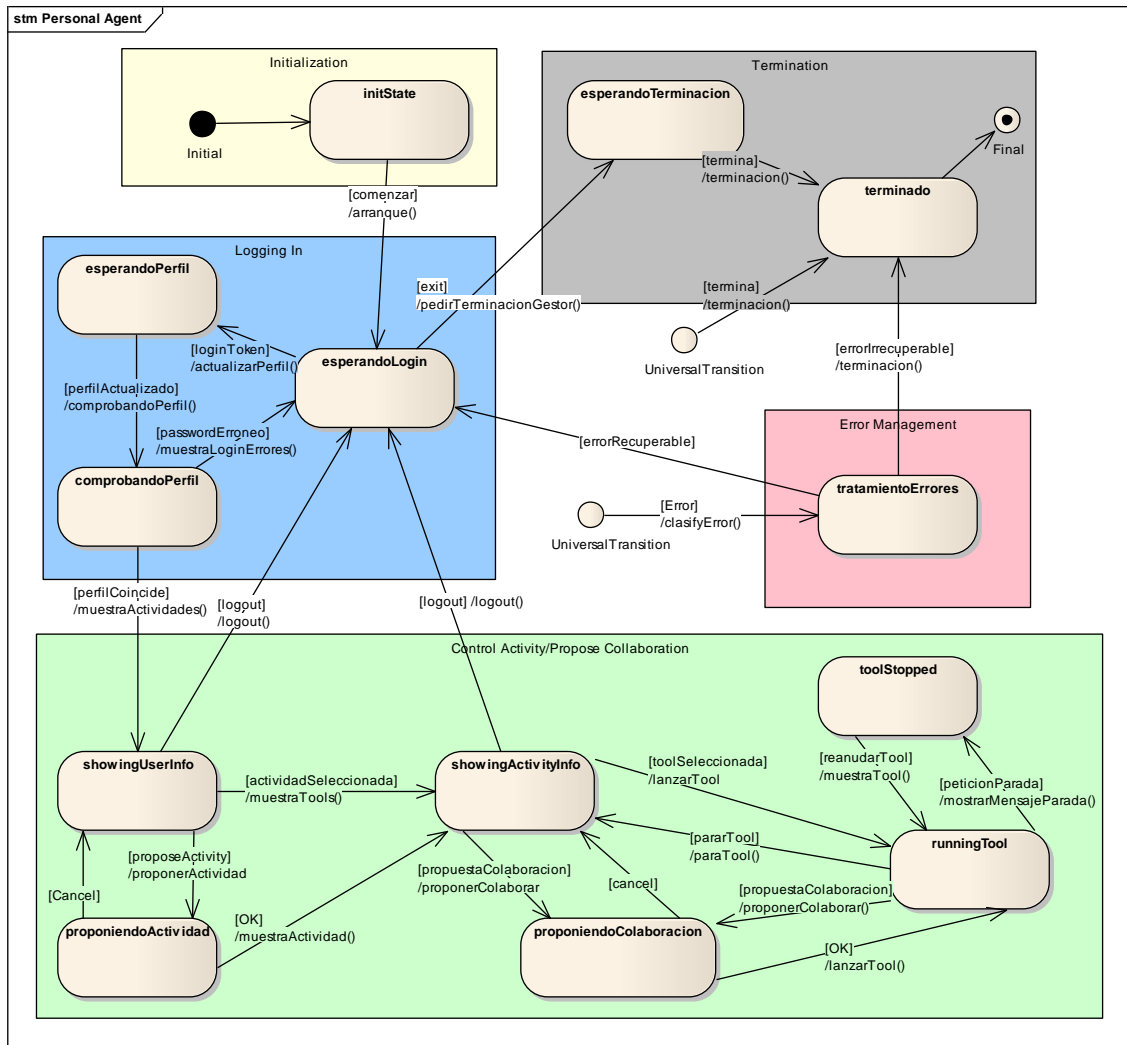


Figura C.1 Máquina de estados del agente PersonalAgent

DIAGRAMAS AGENTE MICRO-REACTIVO DEVICEAGENT

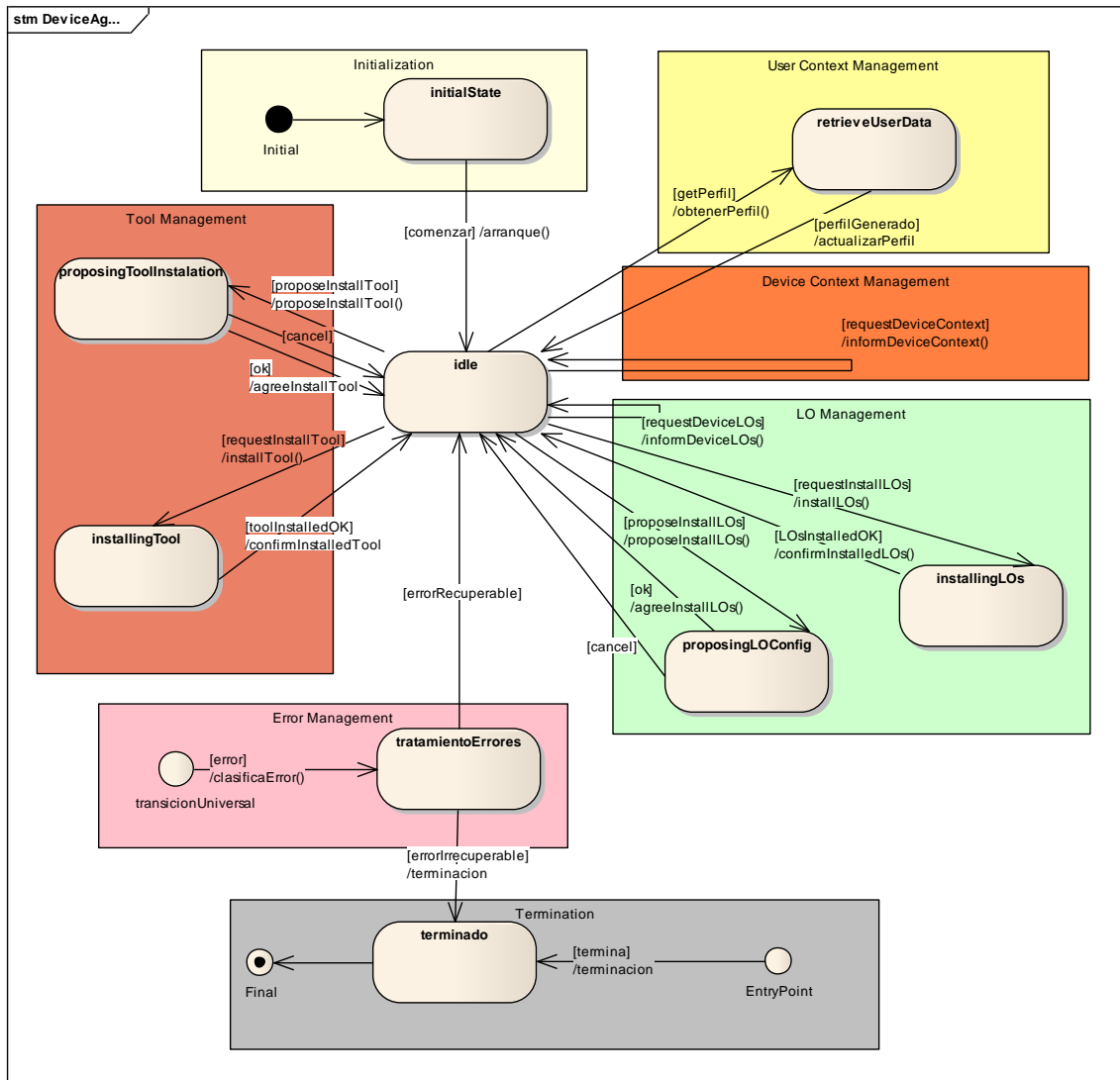


Figura C.2 Máquina de estados del agente DeviceAgent

DIAGRAMAS AGENTE MICRO-REACTIVO COLLABORATIONAGENT

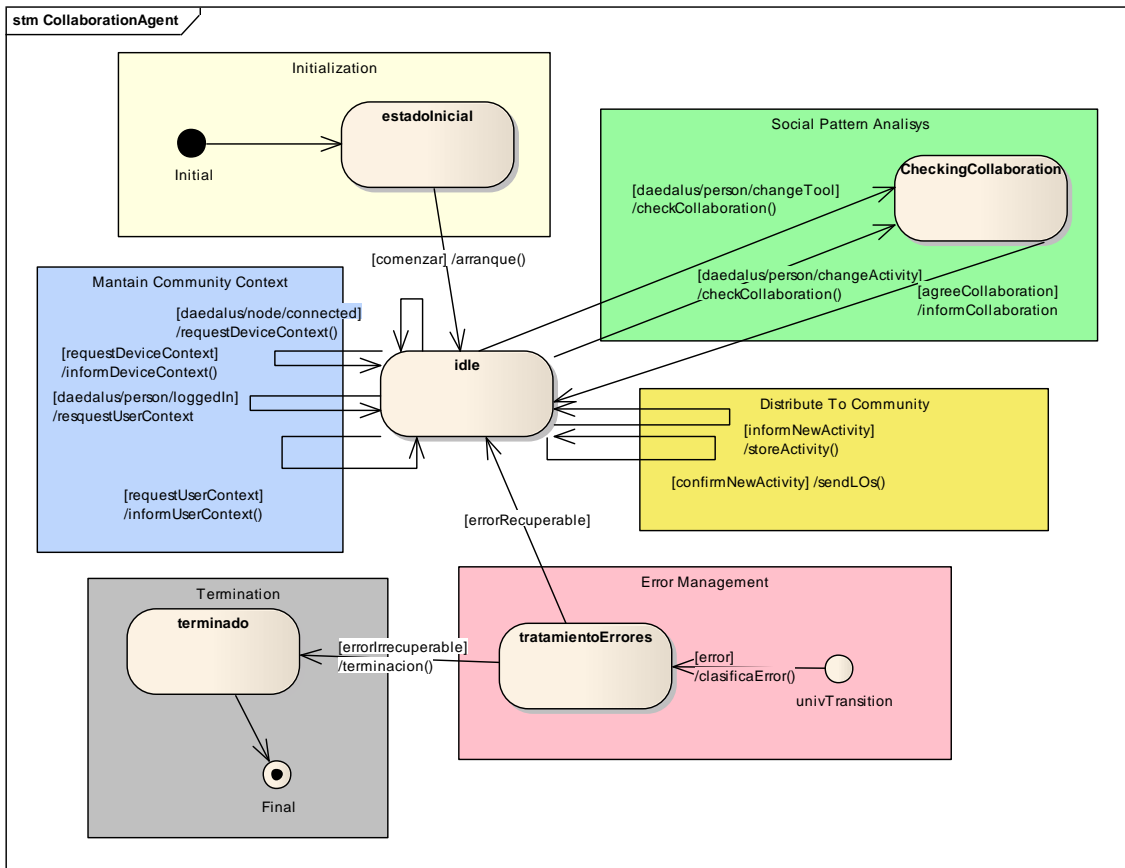


Figura C.3 Máquina de estados del agente CollaborationAgent

DIAGRAMAS AGENTE COGNITIVO DEVICEMANAGERAGENT

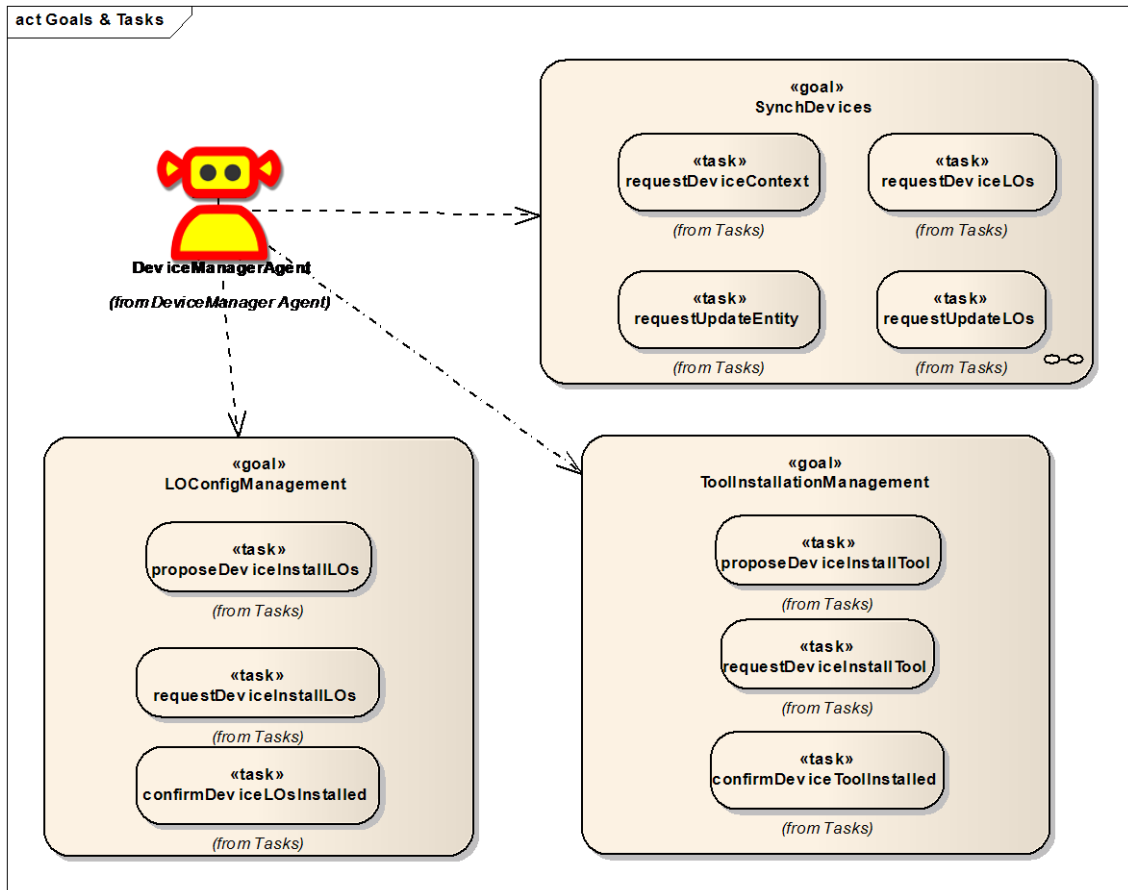


Figura C.4 Objetivos y Tareas del agente DeviceManagerAgent

DIAGRAMAS AGENTE COGNITIVO GLOBALCONTEXTAGENT

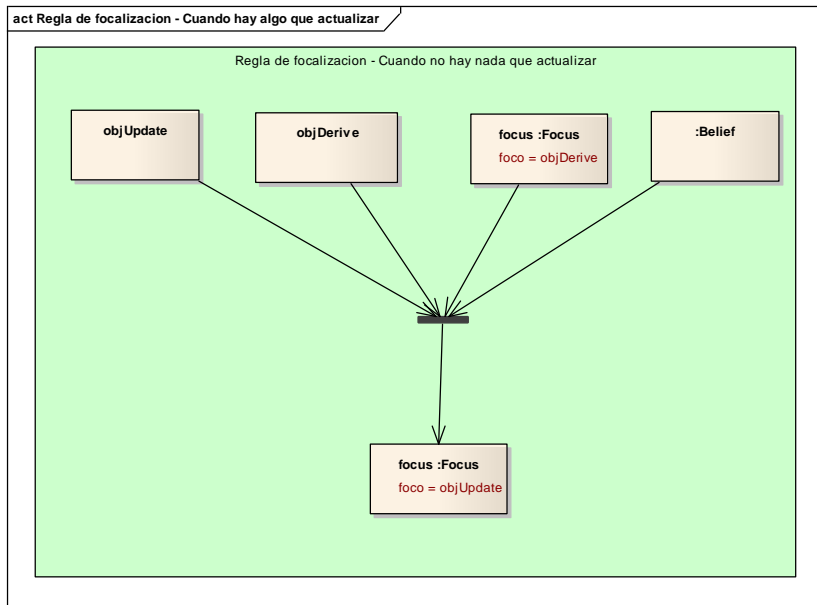


Figura C.5 Focalización de objetivo en agente GlobalContextAgent

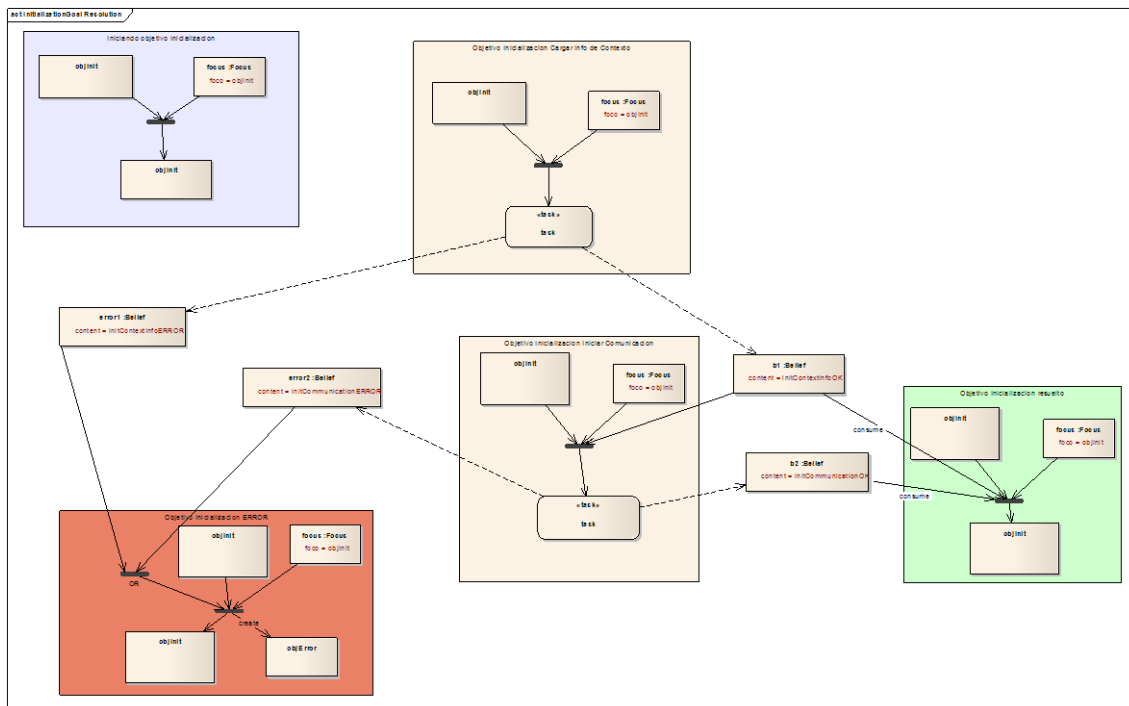


Figura C.6 Resolución de objetivo del agente GolbalContextAgent

DIAGRAMAS AGENTE COGNITIVO MATHACTIVITY

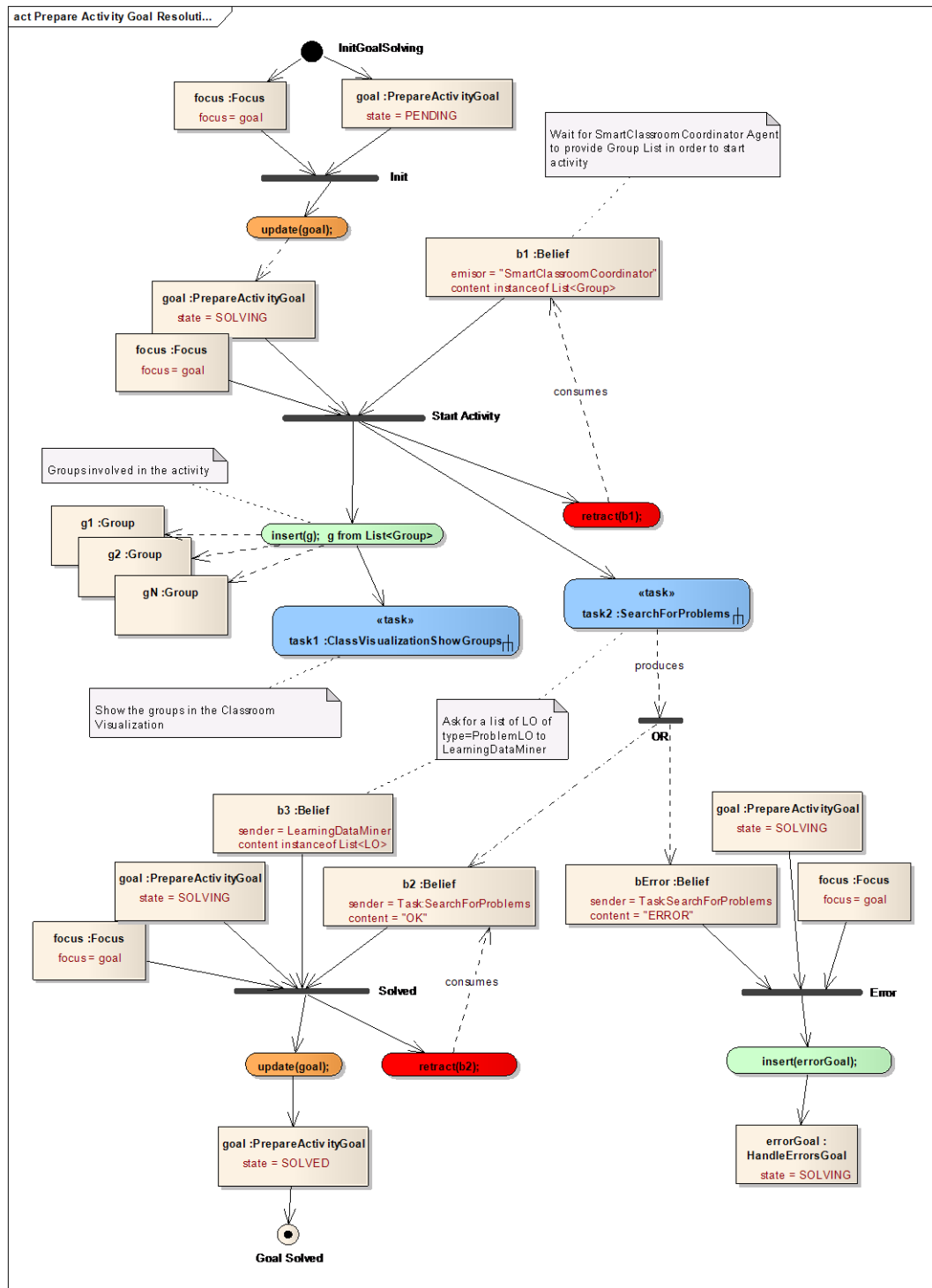


Figura C.7 Desarrollo del objetivo PrepareActivity del agente MathActivity

APÉNDICE D. PUBLICACIONES

Las publicaciones en las que ha participado el autor de esta tesis y que muestran relación con los temas aquí expuestos son:

- Verdejo, M. F., Celorrio, C., Lorenzo, E. J., Millan, M., Prades, S., & Velez, J. (2009). Constructing Mobile Technology-Enabled Environments for an Integrated Learning Approach. En *Innovative Mobile Learning: Techniques and Technologies* (pág. 145). IGI Global.
- Celorrio, C., & Verdejo, M. F. (2008). An Interoperable, Extensible and Configurable Service Architecture for an Integrated Educational Networking Infrastructure. En *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on* (págs. 207-211). Presented at the Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on. doi:10.1109/ICALT.2008.248
- Celorrio, C., & Verdejo, M. F. (2007). Adapted Activity Deployment and Configuration in a Pervasive Learning Framework. *Pervasive Learning 2007: Design challenges and Requirements*, 51-58.
- Verdejo, M. F., & Celorrio, C. (2007). A Multi-Agent Based System for Activity Configuration and Personalization in a Pervasive Learning Framework. En *Pervasive Computing and Communications Workshops, IEEE International Conference on* (págs. 177-181). New York, USA: IEEE Comput-

- er Society. doi:
<http://doi.ieeecomputersociety.org/10.1109/PERCOMW.2007.8>
- Verdejo, M. F., Celorrio, C., Lorenzo, E. J., Ruiz, A., & Sastre, T. (2007). Sustaining learning activity flow in a framework for ubiquitous learning. En *BEYOND MOBILE LEARNING WORKSHOP* (págs. 43-53).
- Mayorga, J. I., Celorrio, C., Lorenzo, E. J., Velez, J., Barros, B., & Verdejo, M. F. (2007). Comunidades Virtuales de Aprendizaje Colaborativo: de los Metadatos a la Semántica. *Inteligencia Artificial*, 11(33), 47-60.
- Celorrio, C., Verdejo, M. F., & Barros, B. (2006). Un Modelo de Distribución de Repositorios para un Portal de Comunidades de Aprendizaje Colaborativo. *Informatica Educativa Comunicaciones*, 2(3).
- Verdejo, M. F., Celorrio, C., & Lorenzo, E. J. (2006). Improving Learning Object Description Mechanisms to Support an Integrated Framework for Ubiquitous Learning Scenarios. En *Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, 2006. WMUTE'06* (págs. 93-97).
- Mayorga, J. I., Barros, B., Celorrio, C., & Verdejo, M. F. (2006). Accessing a learning object repository through a semantic layer. En *Workshop on Learning Object Repositories as Digital Libraries: Current challenges*. Presented at the 10th European Conference on Digital Libraries, Alicante, Spain.
- Celorrio, C., Verdejo, M. F., & Barros, B. (2005). Una Aproximación a la Distribución de Repositorios de Objetos de Aprendizaje basada en Servicios Web. En *Actas VI Congreso Nacional de Informática Educativa Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación*. Presented at the SINTICE, Granada, Spain.
- Mayorga, J., Celorrio, C., Lorenzo, E., Velez, J., Barros, B., & Verdejo, M. F. (2005). Una aplicación semántica de apoyo a Comunidades Virtuales de Aprendizaje Colaborativo. En *Actas del taller "Técnicas de la Inteligencia Artificial Aplicadas a la Educación"*. Presented at the XI Conferencia de la Asociación Española para la Inteligencia Artificial, Santiago de Compostela, Spain.
- Mayorga, J. I., Barros, B., Celorrio, C., & Verdejo, M. F. (2005). An Ontology-driven portal for a collaborative learning community. En *12th International Conference on Artificial Intelligence in Education, Frontiers in*

Artificial Intelligence and Applications (Vol. 125, págs. 872-874). IOS Press.

