

UNIVERSIDAD DE NAVARRA  
**ESCUELA SUPERIOR DE INGENIEROS**

SAN SEBASTIÁN



**Metodología de Desarrollo de Sistemas  
Interactivos Inteligentes de Ayuda al  
Aprendizaje de Tareas Procedimentales  
basados en Realidad Virtual y Mixta**

**MEMORIA**

que para optar al Grado de Doctor  
presenta

**ALBERTO LOZANO RODERO**

bajo la dirección de  
Luis Matey Muñoz  
Begoña Ferrero Martín

San Sebastián, Julio 2009

Servicio de Publicaciones de la Universidad de Navarra

ISBN 978-84-8081-348-8



*A mis padres  
A Edu y Mila*



# ***AGRADECIMIENTOS***

---

El tiempo dirá si ha merecido la pena el esfuerzo pero, en este momento, me quedo con los buenos momentos. Habéis sido muchos los que habéis estado ahí, aportando algo para que haya podido terminar este trabajo. Gracias a todos.

Gracias a Alex García-Alonso, por embarcarme y poner rumbo hacia el doctorado. He tenido tu consejo y apoyo en muchos momentos, y aunque no siempre ha sido fácil, siempre has estado ahí para ayudarme. Todo un lujo con el que espero seguir contando.

Gracias al grupo de investigación GaLan, por iniciarme en la investigación como becario en la FISS, y por su ayuda para realizar este trabajo. Especialmente, gracias a Bego Ferrero. Todas las horas de discusiones que me has dedicado y el trabajo mano a mano no tienen precio y han sido fundamentales para llegar a buen puerto. Y encima has tenido la paciencia de animarme, aguantarme y confiar en mí cuando más se torcían las cosas.

Gracias al CEIT, por darme la oportunidad de realizar este trabajo. He tenido el privilegio de trabajar en lo que me gusta con un grupo de gente excepcional. Gracias a Luis Matey, por la confianza que has depositado en mí desde el primer día, por valorarme, y por darme la libertad que necesitaba. Sin tu apoyo no habría sido posible llegar hasta aquí.

Gracias a Maite López, por tu esfuerzo, paciencia y dedicación. Ahora recoges el testigo y espero poder devolverte toda la ayuda que me has prestado.

No quería olvidarme de mis compañeros de departamento. Los de antes y los de ahora. Muchos sois muy buenos amigos, y me gustaría nombraros a todos, pero no quiero dejarme a ninguno por el camino, y los más cercanos ya sabéis las ganas que tengo de terminar este documento. Quizás debería haber hecho como Cazón y haber empezado a escribir este apartado hace meses...

Gracias a mi cuadrilla, por adoptarme en Donosti. No se cómo se podría sobrevivir en Donosti, ni mucho menos acabar una tesis, sin una cuadrilla. ¡Menos mal que tengo una de las buenas! Y gracias a los compañeros de piso que he tenido todos estos años, porque habéis conseguido hacerme más fácil el día a día.

Gracias a mi familia, por estar siempre ahí, preocupándoos por mis progresos.

Tengo muchos más amigos a los que debo dar las gracias, por su apoyo, ayuda y amistad durante estos años. Pero creo que en vez de seguir alargando esto, es mejor que os lo agradezca luego celebrándolo juntos. ¡Que ya va siendo hora de terminar!

# ÍNDICE

---

AGRADECIMIENTOS.....	I
ÍNDICE.....	III
ÍNDICE DE FIGURAS .....	VII
ÍNDICE DE TABLAS .....	XI
RESUMEN.....	XIII
ABSTRACT .....	XV
<b>0. INTRODUCCIÓN.....</b>	<b>1</b>
0.1    Objetivos de la tesis .....	3
0.2    Contexto de trabajo .....	4
0.3    Estructura de la memoria .....	6
<b>1. ANÁLISIS DE ANTECEDENTES.....</b>	<b>7</b>
1.1    Sistemas Interactivos basados en Realidad Virtual/Mixta .....	8
1.1.1    Interacción en entornos de Realidad Virtual/Mixta .....	10
1.1.2    Los componentes de un Sistema de Realidad Virtual/Mixta .....	11
1.1.2.1    Interfaz.....	12
1.1.2.2    Motor gráfico.....	13
1.1.2.3    Simulación .....	13
1.2    Sistemas Inteligentes de Ayuda al Aprendizaje .....	14
1.2.1    Enseñanza Inteligente Asistida por Ordenador (ICAI).....	15
1.2.2    Sistemas Tutores Inteligentes .....	16
1.2.3    Los Sistemas Inteligentes de Ayuda al Aprendizaje en la actualidad...18	
1.3    Sistemas Interactivos Inteligentes de Ayuda al Aprendizaje (SI <sup>2</sup> A <sup>2</sup> ) de tareas procedimentales.....	21
1.3.1    Arquitecturas de construcción de SI <sup>2</sup> A <sup>2</sup> .....	22
1.3.1.1    Arquitecturas basadas en eventos de errores .....	24
1.3.1.2    Arquitecturas basadas en el estado de la simulación .....	29
1.3.1.3    La arquitectura STEVE.....	31
1.3.1.4    Gestión de conocimiento procedimental independiente.....	37
1.3.2    Resumen de arquitecturas.....	44
1.4    Conclusiones.....	49
<b>2. ANÁLISIS DEL PROBLEMA.....</b>	<b>51</b>



2.1	Análisis de la integración de Sistemas Interactivos.....	52
2.1.1	Entornos autónomos – Entornos reactivos.....	52
2.1.2	Acciones continuas – Eventos.....	53
2.1.3	Simulación simplificada – Simulación realista.....	55
2.1.4	Ruido e incertidumbre en los datos generados por los Sistemas Interactivos.....	56
2.2	Análisis de la integración de Sistemas de Ayuda al Aprendizaje.....	58
2.2.1	Nexo de la Metodología con los Sistemas de Ayuda al Aprendizaje: Un enfoque educativo.....	59
2.2.2	Análisis de la actividad del alumno.....	63
2.3	DETECTive: El punto de partida para el diagnóstico en SI <sup>2</sup> A <sup>2</sup> .....	65
2.4	Resumen de la problemática planteada.....	70
2.5	Propuesta general para el proceso de desarrollo de SI <sup>2</sup> A <sup>2</sup> .....	72
<b>3.</b>	<b>METODOLOGÍA DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup>.....</b>	<b>75</b>
3.1	Alcance de la Metodología.....	75
3.2	Descripción general.....	78
3.3	Nivel de Observación.....	80
3.3.1	Objetivos del Nivel de Observación.....	80
3.3.2	Elementos del Nivel de Observación.....	81
3.4	Nivel de Interpretación.....	84
3.4.1	Objetivos del Nivel de Interpretación.....	85
3.4.2	Elementos Restricción.....	86
3.4.2.1	<i>Restricciones Temporales Cualitativas</i> .....	87
3.4.2.2	<i>Restricciones Temporales Cuantitativas</i> .....	92
3.4.2.3	<i>Restricciones sobre Propiedades</i> .....	94
3.4.2.4	<i>Restricciones Lógicas</i> .....	97
3.4.3	Otros elementos del Nivel de Interpretación.....	99
3.4.3.1	<i>Especificación del paso</i> .....	99
3.4.3.2	<i>Especificación de la situación</i> .....	101
3.4.4	El lenguaje del Nivel de Interpretación.....	102
3.5	Nivel de Diagnóstico.....	105
3.5.1	Objetivos del Nivel de Diagnóstico.....	105
3.5.2	Elementos del Nivel de Diagnóstico.....	107
3.5.2.1	<i>Especificación del paso</i> .....	108
3.5.2.2	<i>Especificación de la situación</i> .....	109
3.5.2.3	<i>Especificación de la solución</i> .....	110
3.5.2.4	<i>Especificación de la Tarea</i> .....	111
3.5.2.5	<i>Elementos Restricción</i> .....	112
3.6	Una visión conjunta y estructurada de los Modelos de Observación, Interpretación y Tareas.....	113
3.7	Proceso de desarrollo de SI <sup>2</sup> A <sup>2</sup> .....	116
3.7.1	Generación del Modelo de Tareas.....	119
3.7.2	Generación del Modelo de Interpretación.....	121
3.7.3	Generación del Modelo de Observación.....	123
3.8	Discusión general.....	126

---

<b>4. MARCO DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup></b> .....	<b>129</b>
4.1 Descripción general del Marco de Desarrollo.....	130
4.2 Arquitectura .....	131
4.3 Subsistema de Observación .....	134
4.3.1 Modelo Dinámico de Observación.....	134
4.3.2 Proceso de extracción de observaciones.....	136
4.4 Subsistema de Interpretación .....	137
4.4.1 Modelo Dinámico de Interpretación.....	138
4.4.2 Proceso de interpretación .....	139
4.4.2.1 Evaluación de restricciones.....	139
4.4.2.2 Interpretación de Pasos y Situaciones.....	148
4.5 Subsistema de Diagnóstico .....	150
4.5.1 Integración de DETECTive como Subsistema de Diagnóstico.....	152
4.6 Conclusiones.....	154
<b>5. APLICACIONES DE LA METODOLOGÍA Y EL MARCO DE DESARROLLO</b> .....	<b>157</b>
5.1 Criterios de validación de la Metodología y el Marco de Desarrollo.....	157
5.2 Aplicación de la Metodología en el dominio de la máquina herramienta ....	158
5.2.1 Tareas de mecanizado con máquina herramienta .....	158
5.2.2 Descripción de VIRTOOL .....	164
5.2.3 Montaje de herramienta: Extracción de Observaciones e Interpretación de Pasos.....	165
5.2.4 Ranurado de una pieza: Extracción de Observaciones e Interpretación de Pasos.....	171
5.2.5 Conclusiones .....	176
5.3 Aplicación de la Metodología en el dominio de la conducción .....	177
5.3.1 El dominio de la conducción.....	177
5.3.2 Descripción del simulador de conducción.....	178
5.3.3 Cambio de carril: Extracción de Observaciones e Interpretación de Pasos.....	180
5.3.4 Adelantamiento: Extracción de Observaciones e Interpretación de Pasos.....	185
5.3.5 Conclusiones .....	192
<b>6. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN</b> .....	<b>195</b>
6.1 Resumen .....	195
6.2 Resultados y aportaciones.....	196
6.3 Futuras líneas de investigación .....	198
<b>ANEXO A: ESTÁNDAR DE INTEGRACIÓN SCORM-SIMULACIÓN</b> .....	<b>203</b>
<b>ANEXO B: REPRESENTACIÓN DE ACTIVIDAD MEDIANTE RESTRICCIONES TEMPORALES</b> .....	<b>207</b>
<b>ANEXO C: REGLAS DE EVALUACIÓN DE RESTRICCIONES</b> .....	<b>213</b>

---

<b>ANEXO D: GRAMÁTICA DEL LENGUAJE DEL NIVEL DE INTERPRETACIÓN.....</b>	<b>223</b>
<b>ANEXO E: ACRÓNIMOS .....</b>	<b>227</b>
<b>ANEXO F: PUBLICACIONES .....</b>	<b>229</b>
<b>BIBLIOGRAFÍA.....</b>	<b>231</b>

# ÍNDICE DE FIGURAS

---

Figura 0.1: Objetivos .....	4
Figura 0.2: Proyectos en los que se enmarca este trabajo de tesis .....	5
Figura 1.1: Diagrama de Milgram: “ <i>Virtuality continuum</i> ” .....	9
Figura 1.2: Componentes de un sistema de RV/RM. Ampliada de (Burdea,1993) con los componentes software .....	11
Figura 1.3: Integración SI <sup>2</sup> A <sup>2</sup> .....	23
Figura 1.4: Arquitecturas de construcción de SI <sup>2</sup> A <sup>2</sup> .....	24
Figura 1.5: Arquitectura ICAT .....	28
Figura 1.6: El agente pedagógico STEVE .....	32
Figura 1.7: Arquitectura de integración de STEVE .....	33
Figura 1.8: Arquitectura Sistema Interactivo máquina herramienta (Lin et al., 2002) .....	38
Figura 1.9: Sistema de entrenamiento de operarios de máquina herramienta .....	38
Figura 2.1: Muestra de ruido al pisar el acelerador de un simulador .....	57
Figura 2.2: Arquitectura de DETECTive .....	66
Figura 2.3: Relación entre datos del Sistema Interactivo y el modelo de tareas .....	71
Figura 2.4: Enfoque del desarrollo de SI <sup>2</sup> A <sup>2</sup> .....	73
Figura 3.1: Abstracción de sistemas a integrar y modelado de la comunicación .....	77
Figura 3.2: Visión unificada del proceso de integración .....	78
Figura 3.3: Descripción del metamodelo de integración .....	79
Figura 3.4: Especificación de una Observación y una Propiedad .....	82
Figura 3.5: Detección de observaciones .....	83
Figura 3.6: Especificación de intervalo .....	87
Figura 3.7: Las 13 relaciones básicas del álgebra de Allen .....	88
Figura 3.8: Ejemplo de relación álgebra de intervalos — álgebra de puntos .....	90
Figura 3.9: Especificación de la Restricción Temporal Cualitativa Allen .....	91
Figura 3.10: Especificación de la Restricción Temporal Cualitativa Puntual .....	92
Figura 3.11: Especificación de elementos Duración .....	93
Figura 3.12: Especificación de la Restricción Temporal Cuantitativa .....	94
Figura 3.13: Especificación de Propiedad de Observación .....	95
Figura 3.14: Ejemplo de instancias para definir una expresión aritmética .....	96
Figura 3.15: Especificación de las Restricciones sobre Propiedades .....	97
Figura 3.16: Especificación de la Restricción Lógica .....	98
Figura 3.17: Especificación del paso en el Nivel de Interpretación .....	100
Figura 3.18: Especificación de la Situación en el Nivel de Interpretación .....	102
Figura 3.19: Ejemplo de EspIPaso .....	103
Figura 3.20: Ejemplo de EspISituacion .....	105
Figura 3.21: Especificación del paso en el Nivel de Diagnóstico .....	108
Figura 3.22: Especificación de la situación en el Nivel de Diagnóstico .....	110
Figura 3.23: Especificación de la Tarea .....	112
Figura 3.24: Ejemplo genérico de Modelos de Observación, Interpretación y Tareas .....	115

Figura 3.25: Proceso de desarrollo de SI <sup>2</sup> A <sup>2</sup> .....	116
Figura 3.26: Proceso de generación del Modelo de Tareas .....	120
Figura 3.27: Proceso de generación del Modelo de Interpretación .....	122
Figura 3.28: Proceso de generación del Modelo de Observación .....	124
Figura 4.1: Descripción general del Marco de Desarrollo de SI <sup>2</sup> A <sup>2</sup> .....	131
Figura 4.2: Arquitectura multiagente del Marco de Desarrollo.....	132
Figura 4.3: Integración del Adaptador del Sistema Interactivo en el ciclo de simulación .....	133
Figura 4.4: Definición de Observación y Propiedad .....	135
Figura 4.5: Definición de Situación y Paso .....	138
Figura 4.6: Ejemplo de evaluación de RTCAllen .....	140
Figura 4.7: Ejemplo de evaluación de RTCPuntual .....	141
Figura 4.8: Ejemplo de evaluación de RTCuantitativa.....	143
Figura 4.9: Ejemplo de árbol de Restricciones Lógicas .....	145
Figura 4.10: Ejemplo de relaciones temporales implícitas entre Observaciones.....	145
Figura 4.11: Conversión Restricción Lógica – Red de Satisfacción de Restricciones Temporales.....	147
Figura 4.12: Inferencia de precedencias entre Observaciones.....	148
Figura 4.13: Diagrama de estados de la interpretación .....	149
Figura 4.14: Plantilla de adaptación del Subsistema de Diagnóstico .....	150
Figura 4.15: Adaptación de DETECTive a la plantilla del Subsistema de Diagnóstico .....	153
Figura 5.1: Tornos y fresadoras convencionales y CNC .....	159
Figura 5.2: Fresadora vertical convencional .....	160
Figura 5.3: Ejemplo de pieza mecanizada.....	161
Figura 5.4: Ejemplo de mecanizado con fresadora.....	163
Figura 5.5: Interfaz de Virtool-Machine .....	165
Figura 5.6: Acciones para montar una herramienta.....	166
Figura 5.7: Especificación de los pasos: “Montar herramienta de corte” y “Ajustar voladizo” .....	169
Figura 5.8: Acciones para ranurar .....	172
Figura 5.9: Especificación de los pasos: “Mecanizar” y “Ranurar” .....	174
Figura 5.10: Evolución de Observaciones durante una operación de ranurado.....	175
Figura 5.11: Exterior e interior del simulador de camión y autobús .....	179
Figura 5.12: Puesto del instructor.....	179
Figura 5.13: Cambio de carril hacia la izquierda y hacia la derecha.....	180
Figura 5.14: Evolución de las distancias a los carriles izquierdo y derecho durante las maniobras de cambio de carril .....	181
Figura 5.15: Especificación de los pasos: “Cambiar de carril hacia la izquierda” y “Cambiar de carril hacia la derecha” .....	183
Figura 5.16: Evolución de la interpretación del Paso: “Cambiar de carril hacia la izquierda” .....	184
Figura 5.17: Plan de solución para la situación de “Adelantamiento” .....	187
Figura 5.18: Especificación de la situación “Adelantamiento” .....	189
Figura 5.19: Especificación del paso “Rebasar” .....	190
Figura 5.20: Evolución de la situación “Adelantamiento” .....	191

---

Figura A.6.1: Focos de interés del SCORM-Sim Study Group.....	204
Figura B.7.1: Relaciones temporales entre intervalos abiertos.....	214



# ***ÍNDICE DE TABLAS***

---

Tabla 1.1: Descripción de arquitecturas de construcción de SI <sup>2</sup> A <sup>2</sup> .....	46
Tabla 2.1: Taxonomía de objetivos de aprendizaje .....	60
Tabla 5.1: Evolución de observaciones durante una operación de montaje.....	170
Tabla B.1: Reglas de evaluación de RTCAllen (relaciones 1-9) .....	215
Tabla B.2: Reglas de evaluación de RTCAllen (relaciones 10-17) .....	215
Tabla B.3: Reglas de evaluación de RTCPuntual (relaciones 1-9).....	216
Tabla B.4: Reglas de evaluación de RTCPuntual (relaciones 10-17).....	217
Tabla B.5: Reglas de evaluación de RestriccionLogica (AND) .....	218
Tabla B.6: Reglas de evaluación de RestriccionLogica (OR) .....	218
Tabla B.7: Reglas de evaluación de RestriccionLogica (NOT).....	218
Tabla B.8: Reglas de evaluación de RTCuantitativa (operador <) .....	221
Tabla B.9: Reglas de evaluación de RTCuantitativa (operador >) .....	221
Tabla B.10: Reglas de evaluación de RTCuantitativa (operador <=).....	222
Tabla B.11: Reglas de evaluación de RTCuantitativa (operador >=).....	222
Tabla B.12: Reglas de evaluación de RTCuantitativa (operador =) .....	222





# *RESUMEN*

---

En las áreas de la Realidad Virtual y de la Inteligencia Artificial educativa se desarrollan tecnologías que resultan de gran utilidad en muchos ámbitos de formación. Este trabajo de tesis ahonda en la combinación de tecnologías de ambas áreas con el objetivo de facilitar la construcción de Sistemas Interactivos Inteligentes de Ayuda al Aprendizaje (SI<sup>2</sup>A<sup>2</sup>) de tareas procedimentales.

Se han estudiado las arquitecturas y prácticas más relevantes en el desarrollo de SI<sup>2</sup>A<sup>2</sup> en diferentes dominios de aprendizaje. A partir de las conclusiones obtenidas, se ha analizado en profundidad la problemática a la que hay que dar respuesta para integrar de forma metodológica Sistemas Interactivos basados en Realidad Virtual y Mixta con Sistemas de Ayuda al Aprendizaje. Como resultado, se ha propuesto una Metodología y un Marco de Desarrollo de propósito general para facilitar la construcción de SI<sup>2</sup>A<sup>2</sup> independientemente del dominio.

La Metodología está compuesta por un metamodelo de integración que describe el proceso de comunicación entre un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje en tres niveles: observación, interpretación y diagnóstico. Además, incluye el procedimiento de construcción de SI<sup>2</sup>A<sup>2</sup>, que se centra en particularizar cada nivel del metamodelo para el dominio de aprendizaje específico del SI<sup>2</sup>A<sup>2</sup>. Por último, el Marco de Desarrollo se encarga de hacer efectiva la comunicación entre los dos sistemas.

La Metodología y el Marco de Desarrollo han sido validados en el dominio de los procesos de fabricación con máquina herramienta y en el de la conducción de vehículos profesionales. Se han realizado experimentos que demuestran su validez para construir SI<sup>2</sup>A<sup>2</sup> independientemente del dominio de aprendizaje, mostrando características diferentes: dominios de actividad continua e instantánea; sistemas interactivos realistas y simplificados; y entornos autónomos y reactivos.



# *ABSTRACT*

---

Virtual Reality and Artificial Intelligence provide useful technologies for many different educational fields. This thesis goes deep into the combination of technologies from both areas with the aim of facilitating the development of Intelligent Interactive Learning Systems (IILS) for procedural tasks.

The most relevant architectures and common practices in the development of IILS in different learning domains have been studied. Starting from the conclusions of the study, the key problems that must be solved to methodologically integrate a Virtual or Mixed Reality-based Interactive System with an Intelligent Learning System have been analysed. As result, a Methodology and a generic development framework have been proposed to facilitate the development of IILS, regardless of the learning domain.

The Methodology is composed of an integration metamodel that describes the communication process between an Interactive System and a Learning System in three layers: observation, interpretation and diagnosis. Besides, it includes the IILS development procedure, which is focused on particularizing each layer of the metamodel for the specific learning domain of the IILS. Finally, the development framework is in charge of making the communication between the systems effective.

The Methodology and the Development Framework have been validated in two domains: the machine-tooling manufacturing and the professional vehicle driving domains. The experiments that have been performed demonstrate that both are valid for developing IILS regardless of the features of the learning domain, such as: continuous and instantaneous activity; realistic and simplified interactive systems; and, autonomous and reactive environments.



# *CAPÍTULO 0*

## *INTRODUCCIÓN*

---

El ser humano pasa gran parte de su vida aprendiendo a realizar todo tipo de actividades. Aprendemos a andar, a realizar operaciones matemáticas o a conducir. Aprendemos a utilizar cubiertos, a manejar herramientas, máquinas y ordenadores. Necesitamos adquirir destrezas de muy distinta naturaleza, tanto físicas como mentales, para desempeñar correctamente estas y otras tareas. Pero en general, la mayoría comparte algo en común: las realizamos siguiendo secuencias de pasos, lo que las define como **tareas procedimentales**.

Encontramos tareas procedimentales en cualquier ámbito de nuestras vidas pero, por diversas causas, el aprendizaje de algunas de ellas resulta complicado. En algunos casos la tarea puede ser arriesgada, como el manejo de maquinaria pesada o la manipulación de sustancias peligrosas. Otras tareas conviene practicarlas en un entorno real, lo que puede limitar el número de horas de aprendizaje por alumno. Por ejemplo, el entrenamiento en situaciones de emergencia implica organizar simulacros periódicos que suelen tener costes elevados. Además, hay que tener en cuenta las dificultades que pueden darse por disponer de recursos de entrenamiento limitados.

Dada la importancia de las tareas procedimentales, es fundamental disponer de mecanismos eficaces para su aprendizaje posibilitando un entrenamiento efectivo en condiciones de seguridad. En este sentido, las nuevas tecnologías han aportado al campo de la formación mejoras cualitativas en el modo de aprender tareas procedimentales. Las tecnologías

de la Realidad Virtual y Mixta (RV/RM) han abordado el tema, señalando el entrenamiento de personal como una de sus aplicaciones prácticas más evidentes y efectivas. Se ha empleado en dominios muy dispares –desde el entrenamiento de pilotos de avión (Barnhart y Ratvasky, 2006; Furness, 1986) hasta el entrenamiento de cirujanos (Liu et al., 2003; Sutherland et al., 2006)– y desde sus inicios ha demostrado su utilidad en las tareas de formación (Stone, 2001).

Asimismo, desde el área de la Inteligencia Artificial (IA) también se han venido desarrollando aportaciones para el aprendizaje de tareas procedimentales. Mientras la RV se centra en la reproducción fiel de los entornos de entrenamiento, la IA educativa lo hace en el proceso de instrucción y aprendizaje y en la asistencia al alumno. Las aproximaciones han ido evolucionando y la terminología ha cambiado a su vez, pasando por los Sistemas de Enseñanza asistida por Ordenador, los Sistemas Tutores Inteligentes (STI), los Agentes Pedagógicos, etc.

Este trabajo de tesis aborda la problemática que surge al integrar tecnología de ambas áreas para construir Sistemas Inteligentes de Ayuda al Aprendizaje de tareas procedimentales que utilicen interfaces basadas en Realidad Virtual (RV) o Realidad Mixta (RM). En adelante nos referiremos a estos sistemas como **SI<sup>2</sup>A<sup>2</sup> (Sistemas Interactivos Inteligentes de Ayuda al Aprendizaje)**. La fusión de ambos tipos de tecnología puede ser de gran ayuda en los procesos de aprendizaje, ya que los estudios realizados desde la perspectiva de la formación indican que para que los sistemas de RV sean realmente efectivos, es conveniente dotarlos de funcionalidades educativas (Mellet d'Huart, 2002). Sin embargo, aprovechar lo mejor de cada una conlleva problemas técnicos que es necesario abordar en profundidad. El trabajo que se presenta en esta memoria incide en la identificación de los nexos que existen entre ambas tecnologías. Como resultado se ha definido una Metodología y un Marco de Desarrollo de propósito general para facilitar la construcción de nuevos SI<sup>2</sup>A<sup>2</sup>. Se pretende que estas aportaciones redunden en una mayor efectividad de los sistemas de entrenamiento basados en tecnologías interactivas, ya que facilitarán la incorporación de funcionalidades educativas que le darán un valor añadido al entrenamiento.

## 0.1 OBJETIVOS DE LA TESIS

El objetivo general de este trabajo de tesis es el estudio, diseño, desarrollo y validación de una Metodología y un Marco de Desarrollo de propósito general para construir SI<sup>2</sup>A<sup>2</sup> en dominios procedimentales.

Por un lado, se busca facilitar la integración de funcionalidades educativas en todo tipo de Sistemas Interactivos diseñados para la práctica de tareas procedimentales. Se consideran tanto sistemas básicos de escritorio como sistemas altamente realistas. Por tanto, es objetivo de este trabajo que la Metodología sea independiente de la arquitectura interna, los dispositivos y las técnicas de interacción utilizados por los Sistemas Interactivos. En cualquier caso, se establece como requisito de partida que estos sistemas sean capaces de comunicar los datos necesarios para generar, aplicando la Metodología, la información requerida por el Sistema de Ayuda al Aprendizaje para evaluar el proceder del estudiante.

Por otro lado, hay que evitar predeterminar qué componentes concretos de ayuda al aprendizaje se pueden integrar, sus arquitecturas y sus procesos internos. Sin embargo, se establece como requisito de partida que los componentes de ayuda al aprendizaje se basen en un modelo de representación de tareas procedimentales y en un proceso que detecte los errores cometidos por el alumno al resolverlas. Ambos deben ser aptos para el dominio del Sistema Interactivo y se parte de la hipótesis de que existen previamente al proceso de integración. Este requisito pueden cumplirlo, por ejemplo, un sistema de guiado, un sistema tutor o un sistema de comunicación de *feedback*.

Dados los condicionantes iniciales, la Metodología estará centrada en los modelos y procesos necesarios para transformar los datos del Sistema Interactivo relacionados con la actividad del alumno en un formato adaptable al modelo de representación procedimental manejado por el Sistema de Ayuda al Aprendizaje (Figura 0.1). De esta manera, los componentes de ayuda al aprendizaje no tendrán que analizar simulaciones o variables dependientes del Sistema Interactivo para tomar sus decisiones. En su lugar, dispondrán de una visión de la actividad procedimental del alumno adecuada para tomar decisiones, independientemente del tipo de soporte que ofrezcan o de cómo hayan sido construidos.



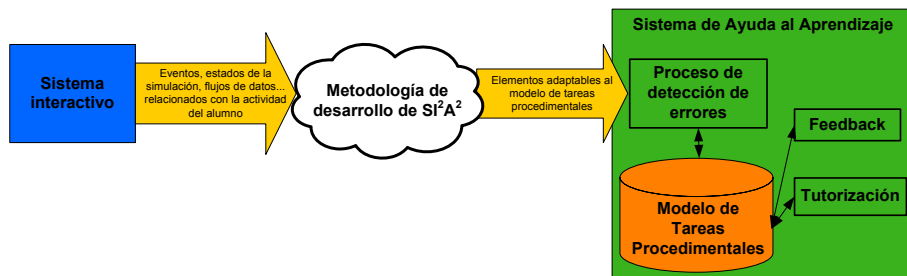


Figura 0.1: Objetivos

La Metodología tiene que indicar el procedimiento para construir un S²A² a partir de la integración de un Sistema Interactivo y de un Sistema de Ayuda al Aprendizaje. Además, se tiene que complementar con un Marco de Desarrollo que lleve a cabo todas las funcionalidades genéricas del proceso de integración establecido por la Metodología. Con el Marco de Desarrollo se simplifica este proceso, permitiendo la construcción de S²A² con un coste menor del que implican actualmente los desarrollos a medida.

## 0.2 CONTEXTO DE TRABAJO

Este trabajo de tesis se ha llevado a cabo en el contexto de varios proyectos de investigación, expuestos cronológicamente en la Figura 0.2, que han sido dirigidos por el departamento de mecánica del CEIT y por el grupo GaLan de la Universidad del País Vasco.

Con el proyecto INGALAN<sup>1</sup> se dan los primeros pasos en esta línea de trabajo. Se desarrolla un primer prototipo de Sistema Interactivo para aprendizaje de procesos de máquina herramienta y comienza el estudio de la problemática de su integración con DETECTive<sup>2</sup>, una herramienta genérica de diagnóstico cognitivo.

El proyecto MANIVIR<sup>3</sup> sirvió para profundizar en las arquitecturas de Sistemas Interactivos orientados al aprendizaje de tareas de mantenimiento de maquinaria. En VIRTOOL<sup>4</sup>, se desarrolló un nuevo

<sup>1</sup> INGALAN: Proyecto financiado por CICYT (TIC99-0252) y Gobierno Vasco

<sup>2</sup> El diagnóstico cognitivo y DETECTive se describen en la sección 2.3

<sup>3</sup> MANIVIR: Proyecto financiado por la Fundación Santa Bárbara

<sup>4</sup> VIRTOOL: Proyecto financiado por la Unión Europea (CRAFT-1999-70292)

Sistema Interactivo para el aprendizaje de procesos de máquina herramienta que ha servido de plataforma experimental y de validación de una parte de la Metodología y del Marco de Desarrollo que se expone en este trabajo de tesis.

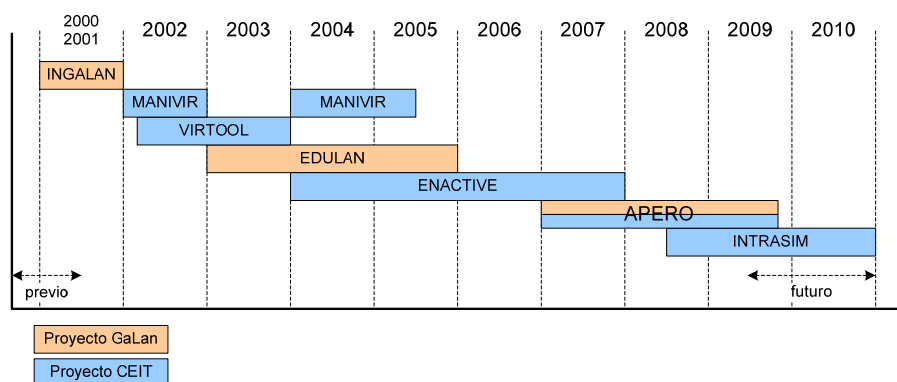


Figura 0.2: Proyectos en los que se enmarca este trabajo de tesis

EDULAN<sup>5</sup>, junto con VIRTUOL contextualizó la investigación en profundidad de la integración de DETECTive con Sistemas Interactivos, resultando en un primer prototipo integrado. También cimentó el inicio de la experimentación sobre adquisición de conocimiento mediante demostraciones en el Sistema Interactivo.

La red de excelencia ENACTIVE<sup>6</sup> enmarcó la investigación sobre interfaces interactivas avanzadas. El resultado de la investigación establece los requisitos que tiene que cumplir la Metodología de desarrollo de SP<sup>2</sup>A<sup>2</sup> para integrar interfaces avanzadas.

El proyecto APERO<sup>7</sup> ha servido para validar la Metodología y el Marco de Desarrollo de SP<sup>2</sup>A<sup>2</sup> con Sistemas Interactivos avanzados. Se ha centrado en el dominio de la conducción, empleando un simulador de conducción de camiones altamente realista. APERO ha sido la lanzadera del proyecto INTRASIM<sup>8</sup>, en el que se abordan algunas líneas futuras de este

<sup>5</sup> EDULAN: Proyecto financiado por CICYT (TIC2002-03141)

<sup>6</sup> ENACTIVE: Red de excelencia financiada por la Unión Europea

<sup>7</sup> APERO: Proyecto coordinado GaLan-CEIT, financiado por CICYT (TIN2006-14968-C02-02)

<sup>8</sup> INTRASIM: Proyecto financiado por CDTI y PROFIT

trabajo construyendo un SI<sup>2</sup>A<sup>2</sup> para la formación de conductores profesionales junto a la empresa Lander Simulation and Training Solutions ([www.landertsimulation.com](http://www.landertsimulation.com)).

### **0.3 ESTRUCTURA DE LA MEMORIA**

Esta memoria está organizada en 6 capítulos. A continuación se describe brevemente el contenido de cada uno de ellos.

En el primer capítulo se expone la revisión del estado del arte sobre el desarrollo de SI<sup>2</sup>A<sup>2</sup>. Para presentarlo, en primer lugar se describen los enfoques más significativos que se llevan a cabo en la construcción de Sistemas Interactivos y Sistemas de Ayuda al Aprendizaje. En segundo lugar se presenta una clasificación de SI<sup>2</sup>A<sup>2</sup> en función de la arquitectura de integración que emplean y se discuten dichas arquitecturas.

En el segundo capítulo se desarrolla el análisis del problema, señalando los fundamentos que apoyan los objetivos de este trabajo e indicando las restricciones y los aspectos críticos a los que se quiere dar respuesta.

El tercer capítulo presenta la Metodología propuesta para desarrollar SI<sup>2</sup>A<sup>2</sup>. Dicha Metodología contiene un metamodelo de integración que especifica los elementos necesarios para integrar un Sistema Interactivo con un Sistema de Ayuda al Aprendizaje. Asimismo, se describen los procesos de utilización del metamodelo para construir un nuevo SI<sup>2</sup>A<sup>2</sup>.

El cuarto capítulo está directamente relacionado con la Metodología presentada en el capítulo 3, y describe el Marco de Desarrollo creado para facilitar la construcción de SI<sup>2</sup>A<sup>2</sup>.

En el quinto capítulo se muestran los experimentos realizados en los dominios de la máquina herramienta y la conducción para validar la Metodología y el Marco de Desarrollo.

Finalmente, en el capítulo 6 se exponen las conclusiones y las líneas futuras de investigación, resaltando las aportaciones más relevantes de este proyecto de tesis.

# *CAPÍTULO 1*

## *ANÁLISIS DE ANTECEDENTES*

---

En cualquier proceso de desarrollo de software es necesario aplicar métodos sistemáticos para asegurar la correcta funcionalidad de la futura aplicación, su calidad y su mantenimiento (Pressman, 2005). En el caso de desarrollos complejos como pueden ser los SI<sup>2</sup>A<sup>2</sup> la necesidad es obvia; desarrollar arquitecturas complejas resolviendo los mismos problemas una y otra vez no sería en absoluto aceptable ni en coste ni en tiempo. Por esta razón, el desarrollo de software en general, tanto en el área de la RV como en el de la IA educativa, se tiende a diseñar soluciones generales y reutilizables. Este capítulo comienza introduciendo las características y arquitecturas más habituales de los Sistemas Interactivos basados en Realidad Virtual/Mixta (RV/RM) y de los Sistemas de Ayuda al Aprendizaje. Esta descripción se presenta con el propósito de ilustrar sobre la diversidad de los sistemas a integrar para la construcción de un SI<sup>2</sup>A<sup>2</sup> y el planteamiento de partida que se propone en este trabajo de tesis. Además se introducen los términos y conceptos que se emplearán en el resto de esta memoria.

Posteriormente se presenta un estudio en el que se clasifican los SI<sup>2</sup>A<sup>2</sup> en función de las arquitecturas de integración que se emplean, analizándose las ventajas y desventajas de cada una. Dicho estudio sirve para obtener una perspectiva global del alcance de los objetivos de este proyecto de tesis y para aportar una visión general de los problemas a los que se pretende dar respuesta.

## 1.1 SISTEMAS INTERACTIVOS BASADOS EN REALIDAD VIRTUAL/MIXTA

Desde una perspectiva clásica, se define la Realidad Virtual como “una interfaz persona-computador compleja que utiliza simulación en tiempo real y canales multisensoriales de interacción. Estas modalidades sensoriales son visual, auditiva, táctil, olfativa y gustativa” (Burdea y Coiffet, 2003). En esta definición ya se intuye la diversidad de factores tecnológicos y humanos que afectan a la construcción de sistemas de RV. Confluyen investigaciones en gráficos por computador, en sistemas de audio o en dispositivos hápticos, además de investigaciones psicológicas sobre los mecanismos de percepción del ser humano. Todo ello para dar lugar a sistemas que puedan engañar a nuestros sentidos haciéndonos experimentar situaciones artificiales como si fueran reales.

En este contexto, los sistemas de Realidad Virtual se caracterizan por tratar de proporcionar al usuario la sensación de inmersión completa. Esta sensación viene determinada por los siguientes factores (Perry et al., 1997):

- **Inmersión visual:** Indica la percepción de la profundidad del Entorno Virtual (EV) que tiene el usuario visualmente. Este factor es fundamental, ya que la vista es el canal por el que recibimos la mayor cantidad de información del entorno.
- **Restitución de fuerza:** Este factor se refiere a la respuesta del EV en forma de fuerzas (ej: colisiones) frente a las acciones del usuario.
- **Inmersión auditiva:** El sonido realista realza la sensación de presencia en el EV.
- **Navegación:** Indica la naturalidad de los métodos que el usuario puede utilizar para desplazarse e interactuar en el EV.

Según Zeltzer existen tres factores fundamentales que podemos considerar para valorar hasta qué punto un sistema de Realidad Virtual consigue crear con éxito experiencias realistas: Autonomía, Presencia e Interacción (Zeltzer, 1992). El autor describe estos factores como las tres dimensiones ortogonales que forman lo que llama el “cubo API”. La dimensión de Autonomía se refiere a que el estado del Entorno Virtual pueda variar con o sin la intervención del usuario. La dimensión de Presencia valora que la interfaz física proporcione al usuario una sensación convincente y natural de estar

presente en el Entorno Virtual (EV). Por último, la dimensión de Interacción indica el realismo con el que el sistema reacciona ante las acciones del usuario. Zeltzer asegura que la Realidad Virtual sólo puede existir cuando la cantidad de autonomía, presencia e interacción que muestre el sistema sea total. Sin embargo, en un contexto más amplio, las tecnologías de Realidad Mixta son empleadas para construir sistemas que no persiguen la inmersión total ni la generación de entornos completamente sintéticos, sino que entremezclan el mundo real y el virtual. Milgram clasifica los sistemas resultantes en función de la cantidad de realidad y virtualidad que combinan, representando las múltiples posibilidades que permite la tecnología como algo continuo (Figura 1.1) (Milgram y Kishino, 1994).



Figura 1.1: Diagrama de Milgram: “*Virtuality continuum*”

El lado izquierdo del diagrama define cualquier entorno en el que el usuario observe únicamente objetos reales, bien en persona, en vídeo o de cualquier otra forma. En cambio, en el otro extremo se definen entornos que contienen únicamente objetos virtuales, independientemente de que el sistema sea inmersivo o no lo sea. Todo el rango de sistemas incluidos en algún punto entre ambos extremos se conoce como Realidad Mixta. De forma más detallada, se distinguen los conceptos de Realidad Aumentada (RA) y Virtualidad Aumentada (VA). El primero de ellos se refiere a los sistemas que utilizan dispositivos de visualización transparentes que permiten la observación de objetos virtuales sin ocultar el entorno real. El segundo concepto, por su parte, define la visualización de imagen real integrada en un Entorno Virtual. En este caso el dispositivo de visualización no es determinante, pudiendo emplearse dispositivos inmersivos o no inmersivos.

### **1.1.1 Interacción en entornos de Realidad Virtual/Mixta**

Uno de los factores principales que caracterizan a los sistemas de RM es la forma de interacción del usuario con el sistema. Un principio que rige el diseño de cualquier interfaz es que el usuario sólo debe preocuparse de qué es lo que desea hacer, no de cómo tiene que controlar la interfaz para conseguir hacerlo. Cuando en el mundo real realizamos tareas aparentemente triviales como agarrar un objeto, nuestra atención no se centra en el modo de realizarlo. Simplemente lo hacemos. Si esta misma tarea hay que realizarla con un objeto virtual, el objetivo de la interfaz es lograr abstraer al usuario al mismo nivel que en el mundo real, lo que se conoce como **interacción natural**. Lograrlo depende de los dispositivos físicos de entrada y de las técnicas de interacción empleadas.

En general, se distinguen dos categorías de dispositivos de entrada: discretos y continuos (LaViola, 2001). Los dispositivos discretos son aquellos que lanzan un evento como consecuencia de una acción del usuario, por ejemplo, la pulsación de un botón. En cambio, los dispositivos continuos se caracterizan por lanzar permanentemente un flujo de eventos, independientemente de las acciones del usuario. Los dispositivos más comunes de este grupo son los *trackers* y los *datagloves*. Los *trackers* son dispositivos capaces de determinar sus coordenadas de posición y su orientación en el espacio, y de transmitirlos constantemente. Colocados en las extremidades del usuario sirven para conocer sus movimientos. Los *datagloves* son guantes sensorizados que transmiten constantemente el gesto de la mano, principalmente mediante la captura de los ángulos entre las falanges. Mediante los *datagloves* se consigue una representación virtual de las manos del usuario que sirve para interactuar manualmente con objetos virtuales.

En general, las posibilidades de los dispositivos de entrada vienen limitadas por los grados de libertad que posean. Cuantos más grados de libertad, más flexible es el dispositivo, aunque un dispositivo puede emular a otro que tenga más grados de libertad mediante botones, gestos, etc. que modifiquen su comportamiento. Por otra parte, no se debe confundir la funcionalidad de los dispositivos de entrada con las técnicas de interacción. Las técnicas de interacción son métodos o metáforas ideados para facilitar al usuario la ejecución de tareas concretas mediante los dispositivos de que dispone. Generalmente, las tareas que se pueden realizar en un EV se

pueden reducir a un conjunto básico: navegación, selección, manipulación, control del sistema e introducción de datos simbólicos. El resto de tareas posibles se consigue mediante combinaciones y/o especializaciones de las básicas (Bowman et al., 2001). Un mismo dispositivo puede servir para implementar distintas técnicas, aunque unas combinaciones ofrecen mejores resultados que otras. En cualquier caso, la elección más adecuada depende de la aplicación concreta.

### 1.1.2 Los componentes de un Sistema de Realidad Virtual/Mixta

Los Sistemas de RV/RM, independientemente de su tipo, presentan una serie de componentes comunes. En la Figura 1.2, ampliada de (Burdea, 1993) con la representación de los componentes software, se observan los elementos de los que constan los sistemas de RV. El nivel de aplicación está representado por la *tarea* que se le propone al *usuario*. Éste se comunica con el sistema mediante los *dispositivos de entrada/salida*. La base de datos contiene el mundo virtual (*geometría, modelos*) y por último se muestran los componentes software comunes a cualquier sistema de RV: la *interfaz*, el *motor gráfico* y la *simulación*.

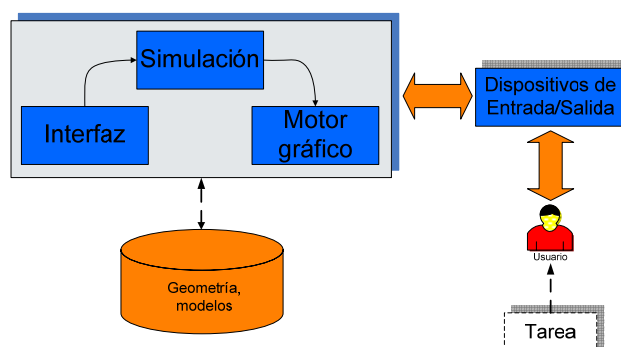


Figura 1.2: Componentes de un sistema de RV/RM. Ampliada de (Burdea, 1993) con los componentes software

El Sistema Interactivo debe ser capaz de reproducir al menos 24 fotogramas/segundo (mejor 30) para que el usuario perciba una simulación suave, lo que implica que el estado del EV se debe recalcularse cada 33 ms. Más aún, si el Sistema Interactivo utilizase además un dispositivo háptico, la frecuencia de actualización debería incrementarse hasta 1 KHz. Para cada ciclo de simulación el Sistema Interactivo realiza el siguiente proceso básico:



- 1) La *interfaz* recoge los datos de los *dispositivos de entrada* y los traduce en un formato comprensible para el sistema.
- 2) La *simulación* recalcula el estado del EV tras la acción del usuario.
- 3) El *motor gráfico* genera el fotograma representativo del nuevo estado.
- 4) La *interfaz* transmite al usuario el nuevo estado del EV mediante los *dispositivos de salida*.

Dado que existen soluciones reutilizables para el desarrollo de cada uno de los componentes software, los esfuerzos de desarrollo se pueden invertir en implementar funcionalidades específicas del dominio de aplicación, en lugar de hacerlo en la tecnología básica.

### 1.1.2.1 Interfaz

La interfaz se encarga de un proceso de traducción entre el usuario y el sistema de RV que consta de varios pasos (Bowman et al., 2004). En primer lugar el usuario determina su objetivo en el EV y lo traduce en acciones físicas mediante los dispositivos de entrada. Posteriormente, las acciones son traducidas a un formato electrónico para que el sistema de RV las comprenda, y finalmente se determina en qué afectan las acciones al EV. El sistema de RV debe dar una respuesta a las acciones del usuario, por lo que la interfaz traduce dicha respuesta utilizando los dispositivos de salida de forma que el usuario lo pueda percibir (sonido, imagen en movimiento, fuerzas...). Así el usuario interpreta esas percepciones como una representación semántica significativa en el contexto de la tarea que realiza.

El proceso de traducción que realiza la interfaz consiste en asociar cada evento producido por los dispositivos de entrada, independientemente de que provenga de un dispositivo discreto o uno continuo, con alguna función que modificará el estado de la aplicación o del EV, y que desencadenará la respuesta del sistema. Este proceso que inicialmente se implementaba en Sistemas de Gestión de Interfaces de Usuario (Green, 1985; Lowgren, 1988), actualmente se integra en el sistema operativo y en librerías externas como DirectInput ([www.microsoft.com](http://www.microsoft.com)). Sobre ellas se implementan las técnicas de interacción que determinan el efecto sobre el EV de los eventos transmitidos por la librería.

### 1.1.2.2 Motor gráfico

El motor gráfico gestiona la representación gráfica del entorno 3D y se encarga de dibujarlo. La aproximación más extendida para representar una escena 3D consiste en una jerarquía de nodos. Cada nodo contiene datos específicos necesarios para el dibujo: transformaciones, materiales, luces, geometría, etc. Los datos de la jerarquía sirven como entrada al *rendering pipeline*<sup>9</sup> que se encarga de generar la imagen 2D final. Este proceso lo implementa la API gráfica, habitualmente OpenGL ([www.opengl.org](http://www.opengl.org)) o Direct3D ([www.microsoft.com](http://www.microsoft.com)) que son capaces de aprovechar las capacidades actuales del hardware de aceleración gráfica ([www.nvidia.com](http://www.nvidia.com), <http://ati.amd.com>).

Existen múltiples motores gráficos en el mercado como Unreal Engine (Epic Games, [www.unrealtechnology.com](http://www.unrealtechnology.com)) o de libre distribución como OpenSceneGraph ([www.openscenegraph.com](http://www.openscenegraph.com)), Ogre 3D ([www.ogre3d.org](http://www.ogre3d.org)) o Java 3D (Sun Microsystems, <https://java3d.dev.java.net/>). El uso de uno de estos motores gráficos reduce en gran medida el tiempo de desarrollo de un sistema 3D interactivo. Gracias a este componente, la gestión de la escena 3D y el proceso de dibujo se pueden abstraer tanto del hardware como del resto de la lógica de la aplicación.

### 1.1.2.3 Simulación

La simulación determina el comportamiento de los elementos del Entorno Virtual, es decir, es la responsable de que cada elemento sea autónomo y/o reaccione de forma adecuada a la interacción con el usuario o a las reglas del EV. Dado que la simulación es dependiente del dominio de aplicación, este componente habitualmente se realiza a medida. Sin embargo, existen componentes reutilizables que simplifican el desarrollo.

El concepto de “comportamiento” es muy genérico. Encaja desde el comportamiento físico de los elementos, para lo cual se dispone de motores como Bullet ([www.continuousphysics.com](http://www.continuousphysics.com)) y Open Dynamics Engine ([www.ode.org](http://www.ode.org)) de libre distribución, o Havok ([www.havok.com](http://www.havok.com)) y PhysX<sup>10</sup>

---

<sup>9</sup> *Rendering pipeline*: Secuencia de procesos aplicados a las primitivas de una escena 3D para rasterizar una imagen 2D

<sup>10</sup> Physx depende de un hardware del mismo nombre especializado en cálculos para simulación física o de una tarjeta gráfica NVIDIA

([www.nvidia.com](http://www.nvidia.com)) como productos comerciales. Asimismo, el comportamiento inteligente de los elementos también estaría contemplado como parte de la simulación, con funcionalidades como navegación autónoma o percepción del entorno. Para simplificar su desarrollo se pueden reutilizar componentes comerciales como Kynapse ([www.kynogon.com](http://www.kynogon.com)) o SimBionic ([www.simbionic.com](http://www.simbionic.com)) y de libre distribución como Fear (<http://fear.sourceforge.net>) y OpenSteer (<http://opensteer.sourceforge.net>).

Existen muchos más motores de simulación especializados en diferentes tipos de sistemas (eléctricos, mecánicos, etc.) o diseñados para simulación basada en eventos. Gracias a ellos se facilita la integración de componentes complejos en el Sistema Interactivo. En este grupo destacan ProModel ([www.promodel.com](http://www.promodel.com)) y ExtendSim (<http://imaginethatinc.com>), así como C++SIM (<http://cxsim.ncl.ac.uk>) y DSOL (<http://sk-3.tbm.tudelft.nl>) de libre distribución.

También merece la pena destacar la existencia de arquitecturas estándar para el desarrollo de simulaciones complejas que pueden ser incluso distribuidas, como HLA (2000) o DIS (1995).

## 1.2 SISTEMAS INTELIGENTES DE AYUDA AL APRENDIZAJE

El área de la Inteligencia Artificial aplicada a la educación ha seguido distintos paradigmas de desarrollo a lo largo de su historia. Durante las décadas de los 80 y los 90, el de los Sistemas Tutores Inteligentes (STIs) dominaba el área ofreciendo una visión unificada de los Sistemas de Ayuda al Aprendizaje y de sus objetivos. Sin embargo, en la actualidad, el planteamiento y los objetivos se han diversificado. Se abordan contextos de aprendizaje de distinta naturaleza, y las investigaciones se centran en resolver problemas específicos de forma general para poder reutilizar soluciones.

Aunque el enfoque actual de desarrollo de Sistemas de Ayuda al Aprendizaje es sensiblemente diferente al de los STIs, hay muchos aspectos que siguen vigentes. Por esta razón, en esta sección se ofrece una visión general del área desde una perspectiva histórica. Brusilovsky y Peylo (2003) hacen un análisis de su evolución desde sus orígenes e identifican distintos

grupos de tecnologías relevantes en cada etapa que hoy en día también se utilizan. Señalan dos paradigmas importantes que han marcado el área: los Sistemas Inteligentes de Enseñanza asistida por Ordenador (*Intelligent Computer-Assisted Instruction, ICAI o AI-CAI*) y los Sistemas Tutores Inteligentes (*Intelligent Tutoring Systems*). Uno de los focos de investigación que tienen ambos en común, que además sigue estando vigente, es el problema del modelado del estudiante (Self, 1999). Sin embargo, la diferencia entre ambos radica principalmente en la evolución de las técnicas empleadas y en que los STIs facilitan una mayor interacción entre el alumno y el sistema.

Además de estos paradigmas, Brusilovsky y Peylo proponían como nuevo paradigma los sistemas educativos inteligentes y adaptativos para la web (*Adaptive and Intelligent Web-Based Educational Systems, AIWBES*). Es cierto que el panorama actual de la IA educativa está protagonizado por las tecnologías web. Sin embargo, las técnicas empleadas son válidas para otros contextos. El aprendizaje colaborativo y el rol del instructor han cobrado importancia, y se investiga el uso de las tecnologías de ayuda al aprendizaje tanto en escenarios de formación a distancia como presenciales. En las siguientes secciones se hace un breve repaso de los paradigmas adoptados en el área, haciendo mención especial a los objetivos planteados en cada época, arquitecturas relevantes y técnicas de interés. Finalmente, se presenta un resumen de las tendencias actuales con la intención de ilustrar sobre las múltiples posibilidades existentes para abordar distintos contextos y dominios de aprendizaje.

### **1.2.1 Enseñanza Inteligente Asistida por Ordenador (ICAI)**

Los ICAI surgen en los años 70 con la intención de paliar la carencia de inteligencia que se achacaba a los antiguos CAI (*Computer-Assisted Instruction*). Era necesario un salto cualitativo que superase la secuenciación prefijada, fuera lineal o ramificada, de contenidos educativos y ejercicios simples que caracterizaban a los CAI (Carbonell, 1970). Mediante la aplicación de técnicas de Inteligencia Artificial se pretendía conseguir sistemas que se adaptasen al proceso de aprendizaje de cada alumno.

Durante esta etapa, las tecnologías predominantes que caracterizaban a los sistemas ICAI fueron principalmente dos: la planificación instruccional (Brown et al., 1973; Wescourt et al., 1977) y el análisis inteligente de

soluciones (Self, 1974). La primera de ellas se encarga de decidir la secuenciación más adecuada de unidades de aprendizaje y tareas que se le han de proponer al alumno. En (Brusilovsky, 1999) se distinguen dos tipos de planificación instruccional: la activa y la pasiva. La activa es aquella que considera los objetivos de aprendizaje para seleccionar la siguiente unidad de aprendizaje o tarea a proponer. En cambio, la pasiva no requiere de la existencia de un objetivo de aprendizaje. Este tipo de planificación aplica medidas correctoras cuando el alumno no ha podido realizar correctamente una tarea, es decir, propone un conjunto de unidades de aprendizaje que puedan ayudar al alumno a subsanar la deficiencia de conocimiento que ocasionó su error.

La segunda tecnología relevante en esta etapa, el análisis inteligente de soluciones, verifica las respuestas del alumno cuando finaliza una tarea. El término “inteligente” se refiere a la capacidad del software para determinar no sólo si una respuesta es correcta o incorrecta, sino también para señalar qué parte está incompleta o es inadecuada y cuáles son las carencias de conocimiento que motivaron el error.

### **1.2.2 Sistemas Tutores Inteligentes**

A finales de los años 70, la evolución de la IA educativa condujo hacia un cambio de paradigma: los Sistemas Tutores Inteligentes (STIs). La idea de sustituir total o parcialmente la enseñanza clásica en aula por sistemas CAI o ICAI fue abandonada. En su lugar, la nueva propuesta sugería la construcción de sistemas más flexibles, dotados de iniciativa y que favorecieran la interacción con el alumno para ofrecerle tutorización individualizada en su proceso de aprendizaje (Burton y Brown, 1979; Clancey, 1979). El cambio de paradigma se afianza, en parte, favorecido por el salto en la plataforma de ejecución, de los *mainframes* a los ordenadores personales, por las nuevas posibilidades de interacción que ofrecían (Brusilovsky y Peylo, 2003). Sin embargo, un Sistema Tutor Inteligente se puede considerar en general como un sistema ICAI más evolucionado. De hecho, hay autores que utilizan los términos STI y sistema ICAI como sinónimos (Nwana, 1990; Rickel, 1989), aunque el primero surgió después que el segundo (Sleeman y Brown, 1982).

El paradigma de los Sistemas Tutores Inteligentes dominó las décadas de los 80 y los 90, y probablemente es el que más ha marcado el área de la

IA educativa. Se caracteriza por una arquitectura general en la que tradicionalmente se identifican cuatro componentes: el módulo del dominio, el módulo pedagógico, el modelo del estudiante y el módulo de diálogo (Wenger, 1987):

- **Módulo del dominio o módulo experto:** Contiene la representación del conocimiento de la materia objeto de aprendizaje. Dicho conocimiento se organiza en función de las características de la materia y de forma que facilite su presentación al estudiante. Además, sirve de referencia para evaluar el conocimiento adquirido por el alumno.
- **Módulo pedagógico:** Este módulo incluye la estrategia o estrategias de enseñanza que va a aplicar el STI. Determina qué contenidos y en qué forma se le han de presentar al alumno en cada momento, corrige las desviaciones de conocimiento del alumno y se adapta a sus necesidades. Para conseguirlo emplea la información contenida en el módulo del dominio y en el modelo del estudiante.
- **Modelo del estudiante:** Representa la visión de cada estudiante que tiene el STI sobre todo aquello que pueda afectar al proceso de instrucción, como por ejemplo: conocimiento adquirido del dominio, ritmo de progreso, nivel de conocimientos previos, motivación, formas de aprendizaje preferidas, etc. Además, habitualmente incorpora algún proceso de diagnóstico que se encarga de valorar las actuaciones del alumno ante las tareas que se le proponen. Mediante este modelo, el tutor consigue uno de sus fines, que es la instrucción adaptada a las características particulares de cada alumno. La forma y contenido que debe tener el modelo de estudiante es un problema de gran dificultad, por lo que se han seguido aproximaciones distintas a lo largo de la historia (Self, 1990), y sigue siendo objeto de debate hoy en día.
- **Módulo de diálogo:** Este módulo se encarga de controlar la comunicación con el alumno. Dada la importancia que se atribuye a la interacción y a la participación del alumno, el módulo de diálogo es responsable de obtener las entradas del estudiante y de presentar las intervenciones del STI de forma adecuada para que la comunicación sea efectiva.

Entre las innumerables contribuciones que ha dado la investigación sobre Sistemas Tutores Inteligentes cabe destacar algunas técnicas asociadas a su

construcción. Por ejemplo, el seguimiento de modelos o *model-tracing* (Anderson et al., 1990; Anderson y Reiser, 1985) es destacable por haber sido empleado de forma generalizada. Esta técnica está basada en las teorías cognitivas ACT\* (Anderson, 1983) y ACT-R (Anderson y Lebiere, 1998) y consiste principalmente en hacer un seguimiento de la actividad del alumno y comparar cada paso que realiza con la planificación prevista por el tutor. Otras técnicas empleadas para la construcción de STIs son la tutorización basada en restricciones (Mitrovich, 2008; Ohlsson, 1994), y la representación mediante redes bayesianas, que se ha utilizado con asiduidad para desarrollar el modelo del estudiante (Pearl, 1988).

Generar dominios completos, modelar a los estudiantes con la precisión necesaria y desarrollar las estrategias de enseñanza componen una labor de gran complejidad, lo que hace que el coste de construcción de un STI sea elevado. Para minimizarlo surgen las herramientas de construcción de STIs, con las que los expertos en las materias pueden crear sus propios tutores inteligentes sin necesidad de contar con un experto en informática.

Existe una gran variedad de herramientas de autor de distintas características (Murray, 1999; Murray et al., 2003), algunas de las cuales incorporan capacidades limitadas de simulación para crear tutores con una marcada orientación práctica. En este grupo se encuentran SIMQUEST (Joolingen y Jong, 1996), XAIDA (Wenzel et al., 1998), Demonstr8 (Blessing, 1997), RIDES y VIVIDS (Munro et al., 1997; Pizzini et al., 1996).

### **1.2.3 Los Sistemas Inteligentes de Ayuda al Aprendizaje en la actualidad**

Muchas soluciones desarrolladas en el campo de la IA educativa se limitaban a contextos y dominios restringidos. Sin embargo, en la actualidad los esfuerzos se centran en resolver problemas genéricos, dando lugar a productos reutilizables en distintos contextos (Andriessen y Sandberg, 1999). El paradigma de los STIs basado en la tutorización individualizada ha dado paso a la exploración de otras alternativas.

Actualmente los Sistemas de Ayuda al Aprendizaje no se consideran únicamente herramientas de autoformación y el instructor ha cobrado mayor protagonismo del que tenía. En lugar de presuponerle un rol pasivo, se estudia su interacción con los alumnos en distintos contextos y el papel

más adecuado de las tecnologías informáticas para facilitar el aprendizaje. En este sentido, es de gran interés proveer al instructor de herramientas que le faciliten obtener información sobre la actividad de los alumnos que usan los sistemas de aprendizaje. De hecho, esta línea de investigación ha adquirido notoriedad, tal y como se puede observar por los *workshops* en minería de datos para la educación celebrados en las conferencias ITS'04 (Beck, 2004), AIED'05 (Choquet et al., 2005), AAAI'05 (Beck, 2005), AAAI'06 (Beck et al., 2006), ITS'06 (Heiner et al., 2006), AIED'07 (Heiner et al., 2007) y la primera conferencia internacional centrada en el tema: EDM'08 (Baker et al., 2008).

La obtención de información sobre los alumnos no se refiere exclusivamente a información individualizada. La investigación sobre el modelado del estudiante sigue siendo de gran importancia, pero también es relevante la información sobre grupos de alumnos. Esta información puede obtenerse en distintos contextos, por ejemplo, grupos de alumnos aprendiendo las mismas materias de forma individual, en subgrupos, o colaborando para resolver una misma tarea. Este último contexto, el aprendizaje colaborativo (Dillenbourg, 1999; Jermann et al., 2001; Koschmann, 1996), trata de favorecer el desarrollo de habilidades en grupo y sociales que de forma individual son difíciles o imposibles de adquirir, como la cooperación o la coordinación (Vigotsky, 1978).

En cuanto a las arquitecturas de los Sistemas de Ayuda al Aprendizaje, la novedad más significativa ha venido de la mano de las plataformas multi-agente. En este tipo de arquitectura, las funciones del sistema de aprendizaje se distribuyen entre varios agentes inteligentes y autónomos que colaboran y se coordinan (Ritter, 1997). Entre ellos hay agentes que observan y controlan la actividad del alumno (Chan, 1996), agentes tutores encargados de tomar decisiones didácticas (Ritter y Koedinger, 1996) y mediadores que escogen la forma de diagnosticar las soluciones del alumno (Conati y Klawe, 2000). Asimismo, otros agentes se dedican a proveer información al instructor o a gestionar aspectos del aprendizaje relativos a grupos de alumnos (Ayala y Yano, 1998). La naturaleza distribuida de las arquitecturas multi-agente facilita la investigación en funcionalidades específicas de los sistemas de aprendizaje, lo cual es una ventaja sobre las arquitecturas clásicas de STIs. Además, hace que su adaptación a la web sea idónea y, teniendo en cuenta que en la actualidad la web es el escenario de ejecución dominante de los Sistemas de



Ayuda al Aprendizaje, se justifica el gran número de investigaciones que emplean esta arquitectura.

En el ámbito de la web, cabe destacar otra línea de investigación de importancia: los objetos de aprendizaje. Tras los objetos de aprendizaje se esconden los esfuerzos de decenas de grupos de investigación por encontrar la forma de reutilizar el conocimiento de cualquier materia en forma de objetos. Además, se pretende conseguir que éstos sean interoperables para construir Sistemas de Ayuda al Aprendizaje de forma sencilla y eficaz. Existen múltiples definiciones de objeto de aprendizaje, y no parece que alcanzar una definición consensuada sea sencillo. Por ejemplo, el *IEEE Learning Technology Standards Committee* considera un objeto de aprendizaje a “cualquier entidad, digital o no digital, que pueda ser utilizada, reutilizada o referenciada durante un proceso de aprendizaje basado en algún soporte tecnológico”. En esta definición encajaría desde material multimedia, contenidos educativos y programas informáticos hasta personas, organizaciones y eventos que puedan ser referenciados en un proceso de aprendizaje. Esta definición ha recibido críticas por considerar entidades no digitales como objetos de aprendizaje (Sosteric y Hesemeier, 2002), pero no es menos cierto que sería difícil encontrar una definición que no haya sido criticada (Friesen y McGreal, 2004). Otros grupos definen conceptos similares a los objetos de aprendizaje, como los objetos de conocimiento (Merrill, 1998) o los objetos de enseñanza (Gibbons et al., 2002), cada uno con distintas funcionalidades que creen necesarias.

La polémica por la definición no está resuelta, pero las investigaciones progresan y los esfuerzos por estandarizar la tecnología de los objetos de aprendizaje han dado sus frutos. Aunque han surgido múltiples iniciativas de estandarización – IEEE Learning Object Metadata standard (IEEE, 2005), IMS Learning Resource Meta-data specification (IMS, 2006), ARIADNE metadata specification (ARIADNE, 1998), Dublin Core metadata standar (Dublin-Core, 2006) – probablemente la que recoge lo mejor de cada una es SCORM (Dodds, 2004), una iniciativa promovida por ADL (*Advanced Distributed Learning Initiative*) bajo los auspicios del Departamento de Defensa de EEUU. SCORM define un formato para empaquetar contenidos en forma de objetos de aprendizaje, un modelo de metadatos para describirlos y un entorno de ejecución con el que establecer la secuenciación de los objetos durante sesiones de aprendizaje en entornos web. Además de esta iniciativa, entre los objetivos de ADL se encuentra

aplicar la misma filosofía de SCORM para otras tecnologías, como *Serious Games*, simulaciones y STIs. En estos casos el objetivo es hacer de la simulación una parte integral del proceso de entrenamiento.

### **1.3 SISTEMAS INTERACTIVOS INTELIGENTES DE AYUDA AL APRENDIZAJE (SI<sup>2</sup>A<sup>2</sup>) DE TAREAS PROCEDIMENTALES**

El entrenamiento de personal es una de las aplicaciones prácticas más claras de los Sistemas Interactivos, por lo que la lista de sistemas que incorporan algún tipo de funcionalidad educativa puede ser interminable. La funcionalidad más básica que se puede encontrar, y en muchos casos la única, es la de permitir a un instructor humano seguir la actividad de sus alumnos y comunicarse con ellos. También existe algún ejemplo más sofisticado en el que el instructor puede colaborar activamente en el Entorno Virtual (Hutchins et al., 2006). Sin embargo, la lista se reduce cuando las funcionalidades buscadas son más avanzadas. No se trata de limitar la lista únicamente a Sistemas Tutores Inteligentes que dejan al instructor un rol pasivo. También se considera la automatización de algunas actividades específicas que habitualmente lleva a cabo el instructor, como asesorar en el desempeño de una tarea, colaborar en la resolución, adaptar la tarea a las características de cada alumno, etc.

La integración de funcionalidades educativas automatizadas e inteligentes es un trabajo que habitualmente se realiza a medida. Además, la arquitectura de integración no suele ser parte central de la descripción de los sistemas en la bibliografía existente. Sin embargo, analizando dichas descripciones se pueden extraer características comunes representativas del tipo de arquitectura utilizada. En la siguiente sección se presentan las cuatro clases principales de arquitecturas de integración identificadas y se analizan las ventajas y desventajas de cada una.

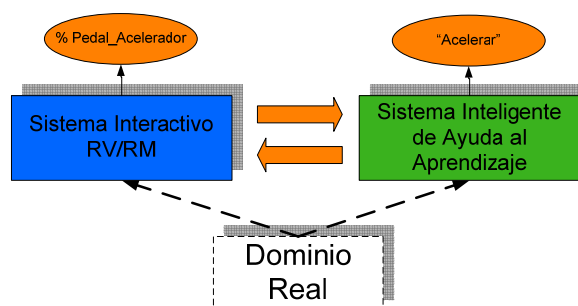
Para finalizar el capítulo, se resumen las ventajas y desventajas de cada arquitectura y se relacionan con las restricciones que impone cada tipo de dominio, simulación y funcionalidades educativas.

### 1.3.1 Arquitecturas de construcción de SI<sup>2</sup>A<sup>2</sup>

La bibliografía refleja que, de todas las tareas que implica la construcción de un SI<sup>2</sup>A<sup>2</sup>, el proceso de integración entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje no es de las que reciben más atención. Habitualmente no se observa una estrategia de integración en la que se señalen claramente los nexos de unión. Probablemente porque su identificación no es obvia y en muchos casos, los nexos establecidos para la construcción de un SI<sup>2</sup>A<sup>2</sup> no son válidos para otros, lo que dificulta la reutilización de estrategias. Sin embargo, el proceso de integración afecta directamente a las capacidades finales del SI<sup>2</sup>A<sup>2</sup> y, por tanto, es necesario considerar a priori el alcance del proceso.

Al contrario que otros procesos de integración de software, éste presenta una peculiaridad que proviene de la propia naturaleza individual de los sistemas a integrar. Mientras un Sistema Interactivo basado en Realidad Virtual/Mixta representa un entorno real para poder **interactuar** en él, un Sistema Inteligente de Ayuda al Aprendizaje representa el **conocimiento** subyacente a las interacciones en el mismo entorno real. En consecuencia, el proceso de integración no se limita a la definición de una interfaz de acceso a cada sistema, sino a la definición de mecanismos de interoperabilidad entre dos sistemas que representan el mismo entorno, aunque de forma distinta y con objetivos diferentes.

En general, conseguir la interoperabilidad entre sistemas comienza por definir la forma de representar e intercambiar datos. En el caso de los SI<sup>2</sup>A<sup>2</sup>, el problema que se aborda es el de transformar datos en conocimiento. Por ejemplo, en el dominio de la conducción el modelo de simulación contendría una variable para indicar mediante un porcentaje cuánto está pisado el pedal de aceleración. El Sistema de Ayuda al Aprendizaje por su parte, representaría el término “acelerar” junto a una serie de atributos y otros conceptos relacionados. Un ser humano es capaz de reconocer de forma aparentemente sencilla la relación entre el porcentaje que está pisado el acelerador y el término “acelerar”. Sin embargo, conseguir que el software represente la misma relación requiere un proceso de transformación que no es inmediato (Figura 1.3).

Figura 1.3: Integración SI<sup>2</sup>A<sup>2</sup>

Tal vez implementar este tipo de relaciones para un sólo caso pueda ser aceptable, pero cuando hay que definir un dominio completo en el que intervienen decenas de variables del modelo de simulación y de conceptos del Sistema de Aprendizaje, es recomendable abordar el proceso de forma metodológica. No hacerlo así puede llevar a integraciones incompletas, y si la interoperabilidad es insuficiente se pueden producir pérdidas de información sobre la actividad del alumno en el Entorno Virtual. En estos casos, el Sistema de Ayuda al Aprendizaje se basaría en hechos incompletos o incorrectos y podría tomar decisiones didácticas incoherentes con la situación real. Por tanto, es necesario evitar este tipo de situaciones atendiendo a las conexiones entre los dos sistemas que intervienen en el proceso de integración.

Aunque la arquitectura de integración suele desarrollarse a medida para cada aplicación, tras una exhaustiva revisión bibliográfica, se han encontrado características comunes entre arquitecturas de SI<sup>2</sup>A<sup>2</sup>. Teniendo en cuenta la peculiaridad del proceso de integración comentada en los párrafos anteriores, se ha realizado un análisis centrado en la forma de gestionar, distribuir e intercambiar datos y conocimiento procedimental. Como resultado, se han identificado cuatro aproximaciones relevantes de diseño de arquitecturas de integración para construir SI<sup>2</sup>A<sup>2</sup> (Figura 1.4):

- 1) Arquitecturas basadas en eventos de errores.
- 2) Arquitecturas basadas en el estado de la simulación.
- 3) Arquitectura Soar Training Expert for Virtual Environments (STEVE).
- 4) Arquitecturas con gestión de conocimiento procedimental independiente.

En las secciones siguientes se presentan estas arquitecturas y se analizan sus ventajas y desventajas.

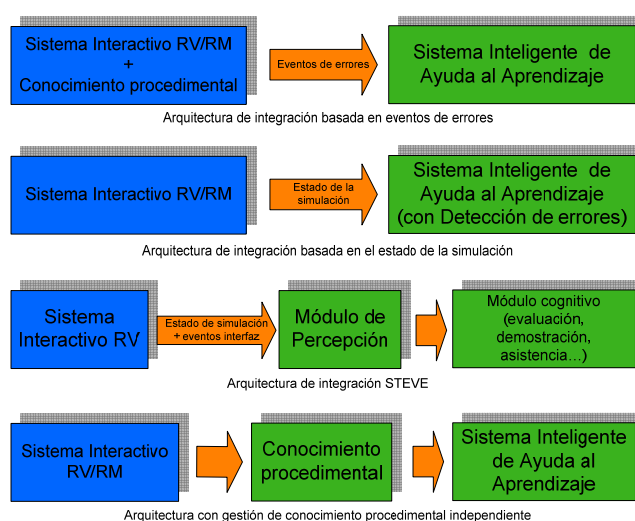


Figura 1.4: Arquitecturas de construcción de SI<sup>2</sup>A<sup>2</sup>

La descripción de los SI<sup>2</sup>A<sup>2</sup> emplea términos como caso, situación, procedimiento, paso o acción que se relacionan con alguno de los niveles de abstracción necesarios para representar las tareas que realiza el alumno. Su significado no está unificado, por lo que para la presentación de cada ejemplo en este capítulo se ha empleado la misma terminología utilizada por los autores de los sistemas en sus publicaciones. Así, algunos autores hablan de resolver un “caso” siguiendo una serie de “pasos”, mientras que otros plantean “situaciones” en las que el alumno debe realizar una serie de “acciones”, refiriéndose a abstracciones de la actividad similares. Sin embargo, cada ejemplo tiene sus particularidades y cada sistema plantea sus propios niveles de abstracción de la actividad, por lo que no es posible determinar que términos empleados por un autor son sinónimos de los empleados por otro. En su lugar, cada término debe ser entendido en el contexto de la aplicación descrita.

### 1.3.1.1 Arquitecturas basadas en eventos de errores

Los sistemas que emplean este tipo de arquitectura se caracterizan por delegar en el Sistema Interactivo la detección de los errores que cometen los alumnos durante las sesiones de aprendizaje. Cuando el sistema detecta un

error envía un evento con su identificación al Sistema de Ayuda al Aprendizaje para que éste lo procese. La detección de errores se realiza a medida del dominio de aplicación, y el módulo encargado presenta siempre una dependencia total de la simulación. El conocimiento procedimental lo representa mediante una biblioteca de errores y se emplean mecanismos específicos para detectar cada error individualmente.

Esta aproximación se emplea habitualmente en dominios en los que la forma de medir el rendimiento del alumno no está claramente especificada o es difícil de estructurar. Por ejemplo, Bernold et al. (2002) presentan un sistema de entrenamiento para operarios de excavadora que integra un tutor capaz de asesorar y dar instrucciones en tiempo real. Según su estudio, la destreza del operario se puede medir por la suavidad del movimiento de la excavadora durante la tarea, y para ello emplean principalmente dos indicadores: el patrón de fuerzas ejercidas por la máquina y el patrón de movimiento. Mediante su análisis, se detectan irregularidades respecto a los patrones esperados que indiquen posibles situaciones en las que el alumno tiene problemas. El sistema no considera ningún tipo de estructuración de las tareas, por lo que no puede detectar errores originados por realizar algún paso en un orden inadecuado. Tampoco puede especificar alternativas para realizar la tarea ni mucho menos valorarlas individualmente. Por otra parte, al realizar una detección de errores completamente ligada al dominio de la excavadora, no es posible reutilizar el software para otros dominios.

### ➤ **El dominio de la cirugía**

Los mismos problemas originados por la arquitectura del sistema de entrenamiento para operarios de excavadora aparecen en el dominio de la cirugía. Los parámetros para valorar la destreza de un futuro cirujano no son evidentes, y se tiende a desarrollar soluciones específicas para cada tipo de operación quirúrgica. Lo más habitual es que ni siquiera sea posible reutilizar el mismo sistema para el entrenamiento de operaciones similares (Cosman et al., 2002). Aunque existen numerosas investigaciones sobre RV aplicada a la cirugía, los sistemas que han pasado validaciones rigurosas escasean (Schijven y Jakimowicz, 2003). A continuación se describen las características de algunos sistemas de compañías de prestigio validados por estudios independientes (Carter et al., 2005; Carter et al., 2006) que son referencia en el sector de los simuladores de entrenamiento de cirujanos:

- *Mentice*: Compañía sueca que cuenta con tres líneas principales de productos: (1) PROCEDICUS VIST es la línea de simuladores de entrenamiento de intervenciones cardiovasculares, (2) PROCEDICUS MIST es la de entrenamiento de intervenciones quirúrgicas mínimamente invasivas y (3) PROCEDICUS VA la de artroscopias de espalda y rodilla. Los simuladores de artroscopias no integran funciones de evaluación. En cambio, los otros dos tipos de simuladores permiten valorar la destreza del alumno. Ambos emplean el mismo tipo de arquitectura, consistente en un módulo de detección de errores integrado en la simulación. El simulador contabiliza el número de colisiones no deseadas, incisiones inapropiadas, etc. Junto a la longitud del camino seguido con el instrumental y el número de movimientos realizado en la intervención, obtiene un indicador del rendimiento del alumno. La validez de estas medidas y de los procedimientos adecuados para evaluar a los alumnos son algunos de los temas centrales investigados por la compañía junto a hospitales y universidades, aunque los criterios óptimos de evaluación siguen sin determinarse (Aggarwal et al., 2006; Patel et al., 2006).
- *Select-IT VEST Systems*: Los sistemas de entrenamiento de esta compañía alemana se centran en la práctica de los procedimientos de cirugía mínimamente invasiva más habituales, tales como suturas, incisiones o extirpación de tejido. La arquitectura es común en todos sus sistemas, aunque su implementación es dependiente de cada procedimiento. Consta de un módulo de evaluación integrado en la simulación que obtiene un indicador basado en el tiempo empleado para finalizar la tarea, el número de contactos en tejidos no permitidos y el uso de fuerzas mayores que unos umbrales predeterminados. Asimismo, incorpora un tutor paso a paso para el entrenamiento de colecistectomías, aunque la tutorización no es individualizada para cada alumno y además es dependiente de la simulación (Çakmak et al., 2005).
- *Simbionix*: Al igual que Select-IT VEST Systems, la compañía estadounidense Simbionix ([www.simbionix.com](http://www.simbionix.com)) centra sus líneas de simuladores en el entrenamiento de procedimientos de cirugía mínimamente invasiva. Actualmente, los métodos que integran para evaluar a los alumnos son también similares a los de su competencia, es decir, integran un módulo que detecta errores específicos para cada tipo de procedimiento. Recoge el número de errores cometidos, y número de

movimientos realizados y los analiza al final de la tarea para presentarle al instructor los indicadores de rendimiento del alumno. Sin embargo, Symbionix destaca entre el resto por su línea de investigación sobre tecnologías de aprendizaje. La compañía ha creado una división especializada en e-learning llamada eTrinsic ([www.etrinsic.com](http://www.etrinsic.com)), que se centra en el desarrollo de contenidos educativos en el ámbito de la medicina. La tecnología que emplean se basa en el uso de objetos de aprendizaje compatibles con SCORM e IMS capaces de integrar simulaciones interactivas. Aún no constan progresos en la integración de sus sistemas educativos con los simuladores de Symbionix, aunque aparece entre sus objetivos.

- *Xitact*: El caso de los suizos Xitact ([www.xitact.com](http://www.xitact.com)) es algo diferente al resto de compañías descritas. Su actividad principal se centra en la integración de los simuladores PROCEDICUS con los dispositivos hápticos que fabrican y su propia plataforma de evaluación. La arquitectura de integración no varía del resto y se basa en un módulo de detección de errores dependiente de la simulación. La plataforma de evaluación se encarga de recoger la información de este módulo y facilitar la organización y análisis de la actividad de los alumnos, tanto individual como colectivamente (Rosenthal et al., 2006).

### ➤ **Arquitectura ICAT**

En un intento por facilitar la construcción de SI<sup>2</sup>A<sup>2</sup>, la NASA diseñó una arquitectura de propósito general denominada ICAT (Loftin et al., 1994). La arquitectura consta de un módulo con el dominio experto, un modelo de estudiante y un gestor de sesiones de entrenamiento encargado de detectar los errores del alumno y de guiarle. Todos estos módulos se comunican entre sí y con la interfaz mediante una arquitectura de pizarra<sup>11</sup> (Figura 1.5).

---

<sup>11</sup> Arquitectura de pizarra: aquella en la que varios módulos utilizan una zona de memoria compartida, denominada pizarra, para comunicarse.



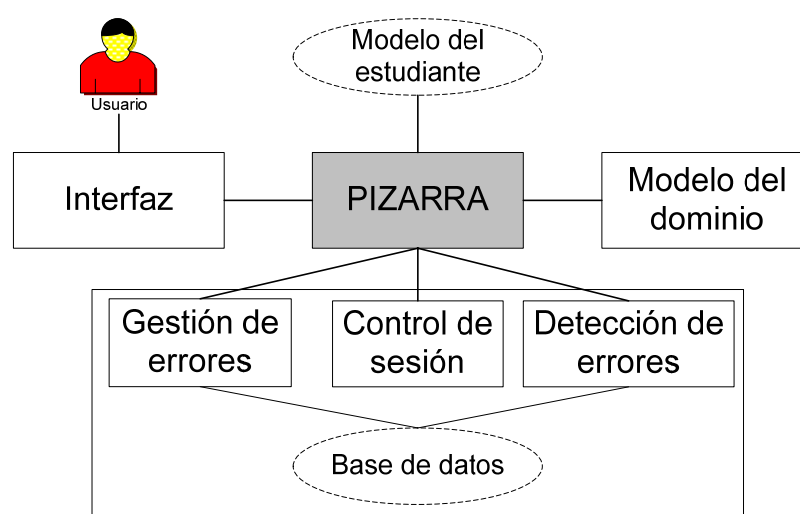


Figura 1.5: Arquitectura ICAT

El modelo de detección de errores se basa en una biblioteca de errores que intenta describir el conocimiento erróneo del estudiante (Brown y Burton, 1978; Burton, 1982). La técnica de las bibliotecas de errores requiere un proceso exhaustivo de identificación de los errores que se pueden cometer en el dominio, y que en ocasiones resultan de la combinación de varios de ellos. En dominios complejos, el análisis da lugar a una explosión combinatoria de posibles errores y sus causas, y a menudo es complicado controlarla (Baffes y Mooney, 1996; Ohlsson, 1994). La arquitectura ICAT plantea la representación de errores mediante reglas IF-THEN creadas específicamente para cada tarea y cada interfaz, lo que conlleva que este módulo no se pueda reutilizar porque la biblioteca depende completamente de la simulación. Además, es necesario que la simulación genere datos de forma compatible con la arquitectura (implementada en CLIPS), lo cual no siempre es posible.

Para minimizar el coste de desarrollo, la NASA creó una herramienta de autor con la que facilitar el proceso de creación de reglas (Saito et al., 1992), aunque el coste asociado a la identificación de errores y diseño de la biblioteca sigue siendo elevado. Utilizando la arquitectura ICAT, la NASA desarrolló varios sistemas de entrenamiento, la mayoría de ellos para entrenar a sus operarios en los procedimientos de mantenimiento de sus sistemas (Loftin y Savely, 1991). Al margen, se llegaron a idear otros entornos de entrenamiento para dominios como la cirugía que implicaban

extensiones de la arquitectura ICAT para poder integrarse con la simulación quirúrgica (Ota et al., 1995).

### 1.3.1.2 Arquitecturas basadas en el estado de la simulación

Este tipo de arquitectura se caracteriza por emplear el estado de la simulación como eje de comunicaciones entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje. Para que esta comunicación sea posible, se parte de la premisa de que el estado de la simulación se puede integrar fácilmente en el modelo de representación de tareas que controla el Sistema de Ayuda al Aprendizaje. Una vez que dispone de esta representación, puede analizarla desde el punto de vista educativo para reaccionar de acuerdo al contexto de aprendizaje de cada momento.

El uso de esta arquitectura es muy poco común. Los SI<sup>2</sup>A<sup>2</sup> que la emplean han sido desarrollados por la compañía estadounidense Stottler Henke Associates ([www.stottlerhenke.com](http://www.stottlerhenke.com)). Esta empresa está especializada en la aplicación de técnicas de Inteligencia Artificial al desarrollo de sistemas de entrenamiento, y cuentan con premios y publicaciones que avalan sus creaciones. Han desarrollado SI<sup>2</sup>A<sup>2</sup> para el entrenamiento de tareas procedimentales de los operarios de la NASA (Ong y Noneman, 2000), la toma de decisiones tácticas para oficiales de la Marina y la Armada de EEUU (Jensen et al., 2003; Stottler y Jensen, 2002; Stottler y Panichas, 2006; Stottler y Pike, 2002; Stottler y Vinkavich, 2000) y el pilotaje de helicópteros (Remolina et al., 2004; Richards, 2002a; Richards, 2002b) entre otros.

La arquitectura general de todos los SI<sup>2</sup>A<sup>2</sup> de Stottler Henke es muy similar. Además de basarse en la monitorización del estado de la simulación, emplean habitualmente el mismo tipo de representación de las tareas. Describen mediante máquinas de estados finitos la forma en que ha de proceder el alumno para cada caso o situación que se va a encontrar. Además, las máquinas de estados llevan asociadas las acciones que realizará el sistema de aprendizaje al detectar alguna actividad del alumno que considere importante, como errores, objetivos logrados, acciones que desencadenan cambios en la configuración de la tarea, etc.

Analizando esta aproximación se puede observar una serie de características que es necesario tener en cuenta:

- 1) *La relación entre las variables del estado de la simulación y las variables de las máquinas de estados ha de ser muy estrecha.* Esto es debido a la comunicación directa entre la simulación y el sistema de ayuda al aprendizaje. Esta relación no siempre es posible, como por ejemplo en (Stottler et al., 2001) donde para integrar un STI con un simulador interactivo 3D de toma de decisiones tácticas (FBCB2), se vieron forzados a desarrollar un módulo específico de traducción entre las variables de la simulación y las del STI.

Para conseguir esta relación tan estrecha, o bien la simulación integra muchas variables que representan conceptos, o bien aparecen muchas variables de la simulación en las máquinas de estados que por sí mismas no tienen significado en el contexto educativo. El primer caso se da cuando la interfaz simplifica la interacción con el alumno mediante comandos que representan los mismos conceptos que el sistema de aprendizaje tiene que valorar. Por ejemplo, en el sistema de entrenamiento de control táctico de vehículos de combate presentado en (Jensen et al., 2003), el alumno envía ordenes a los vehículos para que se posicionen. Cada orden genera un evento que lo representa y se lo envía al Sistema de Aprendizaje. Éste incorpora una máquina de estados con las órdenes posibles en cada contexto y valora inmediatamente el evento recibido. Hay que resaltar la diferencia que supondría que el Sistema de Aprendizaje, en lugar de recibir un evento representativo de la orden para posicionar el vehículo, recibiera permanentemente sus coordenadas de posición en el espacio. En este caso, las coordenadas no aportarían por sí solas la información conceptual de lo que desea hacer el alumno, que es en realidad lo que el sistema de aprendizaje tiene que evaluar. Esta situación implicaría que el sistema de ayuda al aprendizaje tendría que representar las tareas a partir de variables de la simulación sin significado en el contexto educativo, lo que sería más complejo. Por ejemplo, el sistema de entrenamiento de pilotos de helicóptero presentado en (Remolina et al., 2004) emplea variables como altitud, revoluciones del motor o fuerza del viento para representar los procedimientos de vuelo.

- 2) *Si la representación de tareas es complicada se hace necesario el uso de herramientas de autor que simplifiquen la labor.* Extraer los conceptos del dominio de aprendizaje y sus relaciones a partir de variables de la simulación sin significado educativo es una tarea costosa. Implica ser

capaces de relacionar las variables, identificar condiciones sobre sus valores que representen conceptos conocidos, organizarlas, etc. Además, la representación de las tareas hay que completarla con las acciones educativas a ejecutar en cada situación (dar pistas, consejos, ofrecer recursos didácticos, etc.) Si el dominio es complejo, la probabilidad de cometer errores en su representación, o de generar dominios incompletos, es muy alta. Por esta razón, se hace imprescindible el uso de herramientas de autor que ayuden a construir la representación del conocimiento que necesita el Sistema de Aprendizaje.

- 3) *Cuanto más estrecha es la relación entre la representación de conocimiento y las variables del estado de la simulación, mayor es la dependencia entre la herramienta de autor y el Sistema Interactivo.* Stottler Henke desarrolla herramientas de autor específicas para cada uno de sus SI<sup>2</sup>A<sup>2</sup>. El hecho de que no sean reutilizables para distintos sistemas no se puede considerar un factor negativo, ya que se debe facilitar que los instructores y expertos de cada dominio usen estas herramientas. Sin duda, emplear interfaces adaptadas al dominio para que sus usuarios se sientan cómodos al usarlas, facilita la labor de adquisición de conocimiento. Sin embargo, al margen de la interfaz empleada, representar un dominio en el que se mezclan conceptos y variables de la simulación sin significado en el contexto educativo, genera otros problemas. La representación de tareas, casos, conceptos y demás información asociada al aprendizaje, es tan dependiente del Sistema Interactivo y parte de una información de tan bajo nivel que la gestión global del dominio se vuelve muy complicada, aun disponiendo de una herramienta de autor. Como resultado de esta estrategia de integración, aumenta la probabilidad de generar dominios incompletos o incluso erróneos, lo que implica alargar el proceso de refinamiento y testeo del SI<sup>2</sup>A<sup>2</sup>. Además, la experiencia adquirida en la construcción del sistema, el software desarrollado y el dominio educativo generado son muy difíciles de emplear para construir otros SI<sup>2</sup>A<sup>2</sup>.

### **1.3.1.3 La arquitectura STEVE**

Soar Training Expert for Virtual Environments, STEVE, es el nombre del agente pedagógico animado 3D (Figura 1.6) desarrollado por la University of Southern California (Rickel y Johnson, 1999). Este agente está enfocado al

entrenamiento de tareas procedimentales, sobre todo de control de maquinaria. Más concretamente, STEVE ha sido empleado para entrenamiento de operarios en procedimientos de navíos, como utilizar la consola de control de turbinas o manejar las válvulas y compresores de aire de los motores (Stiles, 1999). Existen varias evoluciones de esta arquitectura basadas en los mismos principios que la original. Por ejemplo, se ha empleado para entrenamiento de tareas de equipo (McCarthy et al., 1999), o para la toma de decisiones en escenarios militares, añadiendo expresividad a la representación gráfica del agente y reconocimiento de lenguaje natural a su arquitectura (Rickel et al., 2001).

Entre las capacidades de STEVE se encuentran: demostrar como se realiza un procedimiento paso a paso en el Entorno Virtual, comunicarse con el alumno mediante síntesis y reconocimiento de voz, utilizar mirada y gestos cuando se dirige al alumno o realiza una tarea, contestar a preguntas, adaptar los procedimientos del dominio a eventos inesperados y recordar acciones pasadas. Para mostrar estas capacidades STEVE presenta una arquitectura de características únicas en el ámbito de los SPA<sup>2</sup>.

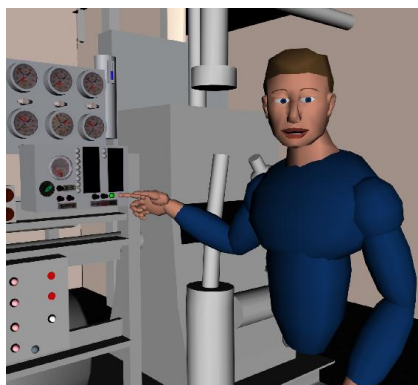


Figura 1.6: El agente pedagógico STEVE

Los Entornos Virtuales en los que se pueden integrar los agentes deben construirse ciñéndose a una arquitectura concreta. Está formada por los siguientes componentes independientes que se comunican mediante paso de mensajes (Figura 1.7):

- Simulador: Controla el comportamiento del Entorno Virtual. Emplean el simulador VIVIDS (Munro y Surmon, 1997) desarrollado por la propia University of Southern California. El modelo de simulación se basa en

eventos y comportamientos, y fue ideado para constituir el núcleo de STIs centrados en simulaciones. Han abordado la creación de tutores para el aprendizaje del control de dispositivos, monitorización de controles, y dominios similares.

- Interfaz visual: Se encarga de la visualización e interacción en el EV. Cuando algún componente cambia algún atributo visual de los objetos, se le notifica para que lo actualice, y en caso de que el usuario interactúe con algún objeto, la interfaz envía la notificación al resto de componentes. Las posibilidades de interacción están limitadas a tocar y agarrar objetos mediante un *dataglove*.
- Audio: Controla la generación de audio en función de los mensajes del simulador que describen las fuentes de sonido en el EV y sus radios de alcance.
- Síntesis de voz: Conversor de texto a voz para que los usuarios escuchen los mensajes de los agentes.
- Reconocimiento de voz: Traduce las preguntas del usuario a mensajes de texto para transmitírselos al agente.

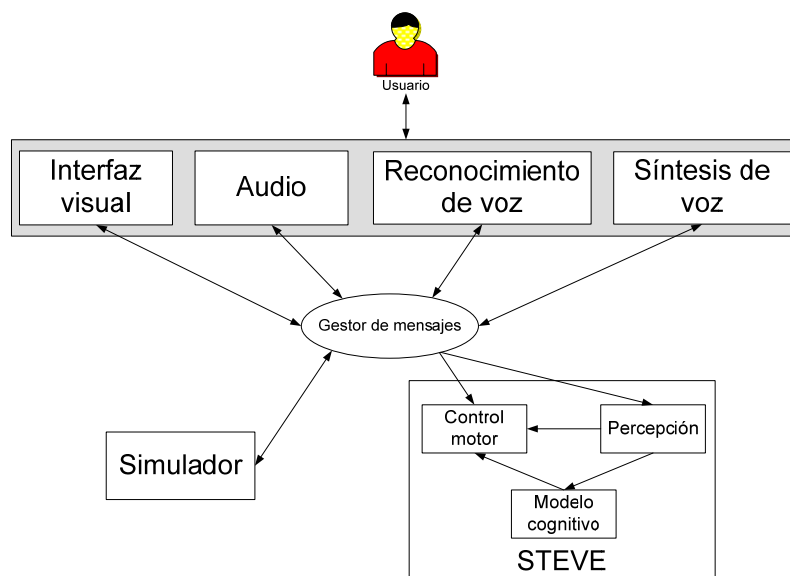


Figura 1.7: Arquitectura de integración de STEVE

El último componente de la arquitectura del Entorno Virtual es el agente STEVE. Su arquitectura interna está compuesta por los siguientes módulos:

- **Módulo de control motor:** Este módulo recibe órdenes de alto nivel para que el agente STEVE ejecute alguna acción en el EV. Cada una de estas órdenes es traducida como una secuencia de comandos de bajo nivel que es enviada a los demás componentes. No sólo controla la posición y gestos del agente, sino también su voz y la forma de ejecutar acciones en el EV. Las acciones que puede ejecutar STEVE dependen de los comandos aceptados por la interfaz visual (tocar, agarrar objetos y pulsar botones) y por el simulador (ej: asignar un valor a un atributo de un objeto del Entorno Virtual).
- **Módulo de percepción:** Mantiene una “instantánea” del estado del EV a partir de los mensajes del simulador y de los eventos recibidos de la interfaz. Lo representa mediante un conjunto de pares atributo-valor, cada uno de los cuales se asocia a una variable del simulador, actualizándose periódicamente. Además, gestiona una lista de eventos producidos por la interfaz que identifican unívocamente las acciones del usuario. Por otra parte, también mantiene una representación de las propiedades espaciales de los objetos del EV paralela a la que lleva la interfaz, pero extendida con la información que STEVE necesita para interactuar (posiciones en las que ubicarse, forma de agarrar o presionar objetos, etc.). Además, controla en todo momento el campo de visión del usuario y un grafo espacial del EV para desplazarse de objeto a objeto evitando colisiones.
- **Módulo cognitivo:** El modelo cognitivo de STEVE está estructurado en tres capas principales: conocimiento específico de la tarea, capacidades pedagógicas independientes del dominio y Soar.
  - El conocimiento de la tarea está representado mediante planes jerárquicos compuestos por los pasos a ejecutar para resolverla. Cada paso incluye relaciones de precedencia con otros pasos y un enlace que especifica que al realizarlo se consigue un objetivo que es precondition para realizar otro. Cuando el alumno ejecuta un paso, el modelo de percepción de STEVE se actualiza para reflejar el objetivo logrado y dar pie a seguir con el siguiente paso de la tarea.

- Las capacidades pedagógicas independientes del dominio se apoyan en el modelo de conocimiento de la tarea. Así, STEVE puede evaluar si los pasos que realiza el alumno le llevan a conseguir los objetivos planteados. También puede construir un plan para realizar una demostración sobre como conseguir un objetivo y responder a preguntas del tipo: ¿cuál es el siguiente paso? ¿por qué este paso es importante?
- Las capas anteriores están implementadas en Soar (Laird et al., 1987), una arquitectura cognitiva de propósito general para el desarrollo de sistemas que tengan que exhibir comportamientos inteligentes. El módulo cognitivo aprovecha la capacidad de Soar de modelado mediante reglas de producción, y su motor de inferencia rige el funcionamiento del sistema.

La arquitectura STEVE presenta ventajas que no se encuentran en otras arquitecturas. Su organización en componentes independientes resulta muy apropiada para la integración y mantenimiento del sistema completo. Más aún teniendo en cuenta que puede ser conveniente el uso de distintos lenguajes de programación. En el caso de STEVE, se emplearon C, Soar y Tcl/Tk. Además, el módulo cognitivo diferencia explícitamente las funcionalidades que son independientes del dominio de aquellas que no lo son, lo cual supone una ventaja a la hora de emplear el mismo sistema en distintos dominios.

Esta arquitectura también facilita la tarea de adquisición de conocimiento. Angros (2000) propone una herramienta llamada *Diligent* para generar el conocimiento que necesita STEVE en un proceso que consta de tres fases. La primera consiste en capturar los pasos que sigue el instructor a modo de demostración para resolver cada tarea. En la segunda fase, *Diligent* experimenta de forma automática con el simulador la ejecución de caminos alternativos al seguido por el instructor para llegar a la solución. Así obtiene nuevas soluciones que lleven al mismo resultado del instructor. Finalmente, el instructor emplea un editor gráfico para refinar y completar los modelos obtenidos por *Diligent* en la fase anterior. Este proceso reduce en gran medida el esfuerzo de adquisición de conocimiento. Sin embargo, impone restricciones que impiden su uso con otros SI<sup>2</sup>A<sup>2</sup>. La principal es que resulta imprescindible poder controlar el simulador mediante comandos para simular ejecuciones de tareas como si



un alumno estuviera utilizando la interfaz. VIVIDS lo permite, pero en muchos casos este requisito puede ser excesivo.

La herramienta de adquisición de conocimiento aprovecha otra característica de la arquitectura STEVE que también implica ventajas y desventajas: el modelo de definición de las tareas considera cada paso del alumno como una acción de bajo nivel que se corresponde unívocamente con un comando para la interfaz o el simulador. De esta forma, cuando *Diligent* quiere experimentar la ejecución de un paso del modelo de tareas para comprobar si lleva a la misma solución del instructor, puede enviar directamente el comando representativo de ese paso al simulador sabiendo que éste lo va a reconocer. En este sentido, la correspondencia entre acciones de la interfaz y pasos del modelo cognitivo es una ventaja. Además, el modelo cognitivo se asegura de tener un conocimiento muy detallado de todo lo que puede hacer el alumno. Sin embargo, esta aproximación no es aplicable a muchos dominios. Por ejemplo, en los SI<sup>2</sup>A<sup>2</sup> para entrenamiento de pilotos de helicóptero de Stottler Henke (Remolina et al., 2004; Richards, 2002a; Richards, 2002b) no es posible representar los pasos del alumno de una forma tan simple.

Por tanto, aunque es posible utilizar la arquitectura STEVE en distintos dominios, está limitada a aquellos con una representación procedimental muy bien estructurada y en los que la interacción del alumno en el EV sea simple.

Por otra parte, las capacidades pedagógicas de STEVE están muy acotadas por el modelo de tareas propuesto. Esto no debe entenderse como una crítica, pero pensando en una metodología general para construcción de SI<sup>2</sup>A<sup>2</sup>, no resulta evidente cómo aprovechar la experiencia de integración de STEVE en sistemas de RV para construir otros SI<sup>2</sup>A<sup>2</sup> con capacidades de ayuda al aprendizaje distintas. Además, tampoco queda claro hasta qué punto la arquitectura es independiente de la simulación de VIVIDS. En algunos casos podría ser necesario emplear arquitecturas que empleen algo más que una lista de pares atributo-valor para la simulación. Por ejemplo, podría ser necesario el uso de una arquitectura compleja que aprovechara al máximo un desarrollo orientado a objetos.

### **1.3.1.4 Gestión de conocimiento procedimental independiente**

La mayoría de las arquitecturas que utilizan un módulo de gestión de conocimiento procedimental independiente son muy similares a las basadas en el estado de la simulación. La comunicación entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje tiene las mismas características generales. En cambio, la estructura del Sistema de Ayuda al Aprendizaje presenta una clara división que no se encuentra en las demás arquitecturas. Independientemente del tipo de ayuda al aprendizaje que provea el sistema, la arquitectura separa del resto del Sistema de Aprendizaje: la gestión de conocimiento procedimental y los procesos de detección de errores. La comunicación entre ambas partes del Sistema de Aprendizaje normalmente se basa en informar de si los pasos que ejecuta el alumno sirven para llegar a la solución de la tarea, aunque el esquema de comunicación puede diferir entre unos sistemas y otros.

Los SI<sup>2</sup>A<sup>2</sup> analizados que siguen este tipo de arquitectura son desarrollos hechos a medida para dominios de aprendizaje concretos, por lo que resulta complicado identificar sin lugar a duda qué módulos son reutilizables para construir otros sistemas. Sin embargo, el hecho de encontrar características similares entre SI<sup>2</sup>A<sup>2</sup> permite detectar algunas pautas de integración exitosas que podrían ser aplicables en otros casos. A continuación se describen varios sistemas que emplean esta arquitectura:

#### **➤ Sistema de entrenamiento de operarios de máquina herramienta**

Lin et al. (2002) presentan un sistema basado en RV para el entrenamiento de operarios de máquina herramienta. Concretamente el sistema contempla el aprendizaje de procedimientos de fresado y taladrado con máquinas de control numérico (CNC). Su arquitectura está formada por los siguientes componentes (Figura 1.8):

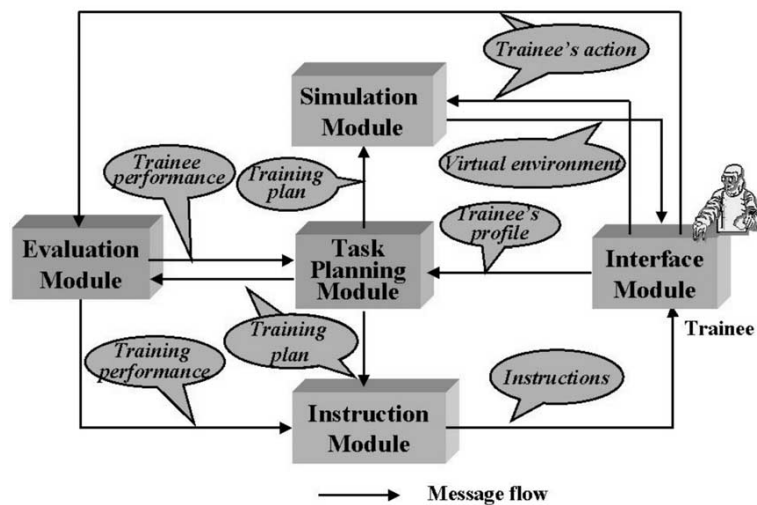


Figura 1.8: Arquitectura Sistema Interactivo máquina herramienta (Lin et al., 2002)

- Interfaz: Controla la visualización e interacción en el Entorno Virtual (Figura 1.9). El sistema es inmersivo y se emplean *Head-Mounted Displays* para la visualización y *datagloves* como dispositivos de interacción. El alumno puede navegar por el Entorno Virtual e interactuar con la máquina. Cada vez que el alumno acciona algún control de la máquina se genera un evento que es enviado al módulo de evaluación para que determine su corrección y al módulo de simulación para que modifique el estado del EV.



Figura 1.9: Sistema de entrenamiento de operarios de máquina herramienta

- Simulación: El módulo de simulación controla los objetos virtuales del entorno mediante la modificación de los atributos que lleva asociados cada objeto. Gestiona tanto su apariencia visual como su comportamiento.
- Planificación de tareas: Este módulo representa el plan de tareas que se le van a proponer al alumno. Para ello utiliza una red de Petri, en la que los nodos se corresponden con los objetivos de aprendizaje planteados en el dominio, y las transiciones con las tareas que debe superar para ir progresando. Así, cuando se considera que el alumno ha llegado a dominar un objetivo, se le permite practicar un conjunto de tareas que le van a llevar a alcanzar otros objetivos más avanzados.
- Evaluación: Es el módulo que se encarga de valorar el rendimiento del alumno en cada tarea. Es capaz de detectar si cada acción del alumno cumple las restricciones del contexto (condiciones sobre parámetros y dependencias de orden entre acciones) y si cumple los objetivos. También valora el camino seguido por el alumno para llegar al objetivo, identificando soluciones eficientes, óptimas y no-óptimas. Asimismo, el módulo de evaluación calcula el nivel de competencia del alumno para cada tarea a partir del número de errores cometidos y el número de intentos efectuados.
- Instrucción: El módulo de instrucción recibe mensajes de los demás módulos y proporciona ayuda al alumno durante la sesión de aprendizaje. En función de los resultados del módulo de evaluación y de la planificación de tareas preparada para el alumno, es capaz de darle pistas y hacerle recomendaciones adaptadas al contexto para que llegue a la solución de la tarea.

La distribución de módulos diseñada en esta arquitectura destaca fundamentalmente por la independencia de los módulos del Sistema de Aprendizaje. Por una parte el módulo de evaluación mantiene un modelo propio de representación de la actividad del alumno para extraer toda la información que considera oportuna sobre la corrección de cada acción. Cuando finaliza el proceso de detección de errores, el módulo es capaz de servir sus resultados al resto de módulos para que los empleen como les resulte adecuado. De esta forma, un cambio de estrategia en el módulo de evaluación no afectaría al resto de módulos. De igual manera, el módulo de planificación de tareas y el de instrucción, siguen la misma filosofía.

No obstante, es necesario remarcar que aunque la estrategia de distribución de módulos es general, los procesos implementados en cada uno son dependientes del dominio. Además, el dominio de la máquina herramienta es un dominio claramente estructurado, en el que los procedimientos a realizar se pueden formalizar fácilmente. Así, en este sistema se pueden observar prácticas de diseño convenientes para muchos dominios. Por ejemplo, se puede observar esta misma arquitectura en (Bin et al., 2006), donde presentan un sistema de entrenamiento de moldeado por inyección con las mismas características que éste. Sin embargo no se pueden extraer conclusiones sobre su generalidad, especialmente para la construcción de SI<sup>2</sup>A<sup>2</sup> en dominios menos estructurados. Tampoco se contemplan interacciones complejas, en las que los pasos del alumno no estén unívocamente identificados por eventos de la interfaz y de la simulación.

➤ **LAHYSTOTRAIN: Un sistema inteligente para entrenamiento de laparoscopias e histeroscopias**

En el marco del proyecto europeo LAHYSTOTRAIN se desarrolló un SI<sup>2</sup>A<sup>2</sup> para la práctica de procedimientos médicos, concretamente laparoscopias e histeroscopias (Los Arcos et al., 2000). El sistema sigue una arquitectura de agentes conectada a un entorno de Realidad Virtual en el que el alumno realiza las tareas. Los agentes que forman el sistema son los siguientes:

- Enfermera, anestesista y cirujano ayudante: Estos tres agentes están representados mediante un avatar en el EV y se encargan de emular el comportamiento del personal que interviene en la operación. Su arquitectura interna está compuesta por un módulo de percepción, un motor de razonamiento y un módulo de control de acciones. El primero recoge los eventos producidos en el sistema de RV y representa el estado del paciente. El segundo se encarga de procesar la información recibida y decidir qué hacer en cada momento. Para ello contiene un modelo de tareas que representa qué pasos dar. Finalmente, el módulo de control de acciones se encarga de hacer efectivas las decisiones del módulo de razonamiento enviando los comandos de bajo nivel apropiados a cada módulo del sistema para presentar sus decisiones al usuario.

- Asistente: Es un agente pedagógico que se encarga de ayudar al alumno durante la sesión de aprendizaje. Su funcionamiento está distribuido en varios componentes, entre los que se encuentran:
  - Monitor: Detecta los errores cometidos por el alumno. Emplea un modelo de tareas representado mediante un grafo de eventos que contiene las secuencias óptimas de las acciones, fases y tareas por las que puede pasar el alumno. También representa grafos alternativos a las secuencias óptimas.
  - Diagnóstico: Recoge los errores detectados por el Monitor y trata de analizar sus causas. La implementación de este componente está basada en reglas.
  - Generador de explicaciones: Este componente se encarga de generar explicaciones adaptadas al tipo de alumno y al tipo de error detectado por el módulo de diagnóstico. Las explicaciones que puede presentar van desde recursos multimedia (texto, gráficos, sonido, video...) hasta demostración de acciones mediante un avatar animado.
- Tutor: El agente Tutor es el responsable de supervisar las sesiones de entrenamiento. Está formado por los siguientes componentes:
  - Supervisor de entrenamiento: Valora el rendimiento del alumno en función de las unidades de aprendizaje planteadas en la sesión. En caso necesario puede replanificarla.
  - Planificador pedagógico: Escoge el plan pedagógico que se le ha de proponer a cada alumno en función de la información contenida en el modelo del estudiante. El plan pedagógico contiene los objetivos de aprendizaje (destrezas que el tutor quiere transmitir al alumno), las estrategias de enseñanza (tipos de actividades previstas para alcanzar un objetivo de aprendizaje) y los objetivos del tutor (tareas específicas que tiene que llevar a cabo el alumno durante una sesión de entrenamiento).

La arquitectura presentada es un buen ejemplo de distribución de funcionalidades de ayuda al aprendizaje entre varios agentes vinculados a un Entorno Virtual. Esta distribución se debe en buena parte a la asignación de roles de los avatares animados que representan a cada agente. Sin embargo, merece la pena resaltar que aplicar esta misma estrategia para construir otros SI<sup>2</sup>A<sup>2</sup> tiene efectos positivos aunque no cuenten con una representación de avatares. Una ventaja inmediata es la claridad del diseño

de la arquitectura respecto a otras en las que varias funcionalidades están compactadas en un solo módulo. Esto redundaría en una mayor facilidad a la hora de mantener el sistema, ampliar funcionalidades o introducir modificaciones. Más aún cuando los Sistemas de Ayuda al Aprendizaje suelen contener módulos basados en reglas que en ocasiones son muy difíciles de mantener.

También se observan efectos positivos sobre el proceso de integración. No todos los componentes necesitan estar relacionados directamente con el sistema de RV, con lo que se limita la complejidad de la comunicación de datos de la simulación hacia los agentes. En LAHYSTOTRAIN, el componente más cercano al sistema de RV es el componente Monitor del agente Asistente. La comunicación se basa en el envío de eventos de la interfaz junto a la representación del estado del paciente virtual. Al igual que sucede con el resto de sistemas presentados en esta sección, los eventos que le envía el sistema de RV identifican las acciones del alumno. En este sentido, LAHYSTOTRAIN no aporta características diferentes a las ya presentadas en otros sistemas.

➤ **Un sistema de entrenamiento inteligente multimodal para cirujanos**

Billinghurst et al. (1995) presentan un sistema de entrenamiento de cirujanos que integra un sistema experto para proporcionar capacidades educativas inteligentes. El prototipo desarrollado está centrado en el aprendizaje de intervenciones quirúrgicas en la cavidad nasal, aunque los autores afirman que puede reutilizarse para otro tipo de operaciones. Su arquitectura distingue varios componentes especializados con una distribución muy similar a las del resto de esta sección: un sistema de Realidad Virtual, un modelo de tareas que representa mediante un grafo los pasos que debe realizar el alumno en cada ejercicio y un sistema experto basado en reglas que se puede configurar para exhibir distintos comportamientos que ayuden al estudiante.

El comportamiento del sistema experto más restrictivo para el estudiante es en el que no se permite realizar ninguna acción que no esté considerada en el modelo de tareas. En el siguiente nivel de configuración, el estudiante tiene libertad para realizar las acciones que crea necesarias en cada paso de la tarea, pero no puede pasar al siguiente hasta que no lo haya completado. Finalmente, el sistema experto también puede ser configurado

en un modo de exploración libre, en el que el alumno puede hacer lo que quiera durante todo el ejercicio y sólo se le notifican sus errores cuando ha finalizado.

La diferencia que este sistema presenta respecto al resto reside en la transformación de datos que se realiza para poder comunicar el sistema de RV con el sistema experto. La comunicación en los sistemas presentados anteriormente se basa en el envío de eventos representativos de la actividad del estudiante. En cambio, en este SI<sup>2</sup>A<sup>2</sup> cambia el fundamento, motivado principalmente porque la interfaz del sistema de RV es multimodal. El usuario utiliza sus gestos y la voz para actuar en el Entorno Virtual. Cada vez que utiliza el instrumental, cuando se desplaza o cuando envía un comando de voz, combina varios canales de interacción para realizar una acción. La fusión de todos los canales da lugar a un contexto de acción que es necesario interpretar para poder comunicarse con el sistema experto. Es necesario conceptualizar el contexto para determinar exactamente la relación entre lo sucedido en el EV y los pasos representados en el modelo de tareas. Por ejemplo, uno de los pasos que se realizan en este SI<sup>2</sup>A<sup>2</sup> es "extirpar glándula". Este paso requiere de varias acciones entre las que se encuentran la selección del instrumental, acercarse a la glándula y realizar varias incisiones. Cuando el alumno las realiza, el sistema de RV detecta eventos asociados a los comandos ejecutados, el movimiento del instrumental y las colisiones con el modelo 3D, lo que da lugar a un conjunto de "acciones primitivas". Estas primitivas por sí solas no identifican el paso realizado por el alumno, ya que pueden formar parte de varios pasos diferentes. Es necesario analizarlas en conjunto para interpretar el contexto global y poder tenerlo en cuenta a la hora de determinar la corrección de las acciones. En el caso de "extirpar glándula", cuando se detectan colisiones sobre los tejidos que recubren la glándula se infiere que las acciones que va a realizar el alumno están ligadas a dicho paso. De esta forma la respuesta del sistema experto cuando se le pide que detecte si se ha cometido algún error será ajustada al contexto.

El proceso de interpretación que se realiza está basado en la representación de la actividad multimodal mediante un modelo de Dependencias Conceptuales (DC). Esta técnica de representación de conocimiento surge del área del procesamiento de lenguaje natural. Se basa en la representación de cualquier acción a partir de un conjunto de primitivas básicas, estados intermedios y relaciones causales (Schank, 1975).



Utiliza una estructura sencilla compuesta por los campos "Actor", "Acción", "Objeto" y "Dirección". Completando esta estructura y ligando varias entre sí se puede representar el significado de cualquier evento, sea cual sea su origen. De esta forma se pueden integrar en un formato común tanto eventos producidos en el EV como comandos de voz y cualquier otra información que represente acción. Cada paso del modelo de tareas incluye su representación en forma de DCs, de tal forma que cuando un conjunto de DCs generadas desde la interfaz multimodal coinciden con las DCs de algún paso, se dispara un conjunto de reglas en el sistema experto para activar el contexto de dicho paso en el proceso de detección de errores.

Las Dependencias Conceptuales facilitan la combinación de canales de interacción. Sin embargo, no carecen de inconvenientes. Este tipo de estructura es muy fuerte, lo que conlleva inflexibilidad para representar situaciones muy dinámicas en las que las relaciones entre DCs puedan romperse para manejar excepciones a las situaciones habituales. En el prototipo de entrenamiento de cirujanos presentado no se da el problema, ya que en dicho dominio no se dan cambios de contexto excepcionales de forma habitual y por tanto puede ser controlable. En cambio, aplicar esta técnica a dominios como el pilotaje de aviones limitaría los escenarios de aprendizaje a simplificaciones fácilmente formalizables. En ese caso, se perdería el potencial que las simulaciones realistas pueden aportar al aprendizaje de tareas complejas y por tanto esta estrategia no sería aconsejable.

### **1.3.2 Resumen de arquitecturas**

En la sección anterior se ha presentado un conjunto de sistemas que muestran las estrategias de construcción de SI<sup>2</sup>A<sup>2</sup> más representativas. Aunque sus arquitecturas son completamente distintas, se han presentado agrupados en función de algunas características comunes que resaltan sobre el resto. Todas ellas indican la forma en que los diseñadores han decidido distribuir e independizar funcionalidades, así como intercambiar datos y conocimiento. Cada decisión es un reflejo de las necesidades surgidas por los requerimientos de cada aplicación, de cada dominio y de las expectativas de los usuarios. Por esta razón ninguna de las arquitecturas se puede reutilizar ni desechar íntegramente a priori para la creación de un nuevo sistema. Sin embargo, observando en conjunto las descripciones de las arquitecturas presentadas se revelan algunos indicios de por qué se ha

utilizado cada una de ellas y del tipo de ventajas o desventajas que ofrecen. En esta sección se expone el análisis realizado, comenzando por la presentación conjunta de las características de cada SI<sup>2</sup>A<sup>2</sup> en la Tabla 1.1.

Las filas muestran las características de los SI<sup>2</sup>A<sup>2</sup> agrupados por tipo de arquitectura. Las columnas señalan las particularidades que se han considerado. De izquierda a derecha se presentan, en primer lugar, las asociadas al Sistema Interactivo. Las dos siguientes columnas están relacionadas con el proceso de integración: la primera es el eje de integración, que se refiere a la entrada de datos que el sistema de aprendizaje espera obtener del Sistema Interactivo; y la segunda es la forma de comunicación, que indica el método empleado para obtener los datos. Las últimas columnas indican características relativas al Sistema de Ayuda al Aprendizaje.

Las observaciones realizadas se dividen en dos líneas generales. La primera gira en torno a la representación de las acciones del alumno en el sistema, y la segunda en torno a la distribución de funcionalidades de ayuda al aprendizaje que se hace en cada caso.

Características SI2A2 Arquitecturas		Sistema Interactivo		Integración		Sistema de Ayuda al Aprendizaje		
		Disp. Interacción	Acciones	Eje de integración	Comunicación	Detección de errores	Modelo de tareas	Ayuda al aprendizaje
Eventos de errores	Excavadora	Simulador excavadora	Continuas	Eventos de errores		Específica del dominio		Evaluar Asesorar
	Cirugía	Hapticos	Eventos y continuas	Eventos de errores	Mensajes	Específica del dominio		Evaluar
	ICAT		Eventos	Eventos de errores	Pizarra	Biblioteca de errores	Grafo de pasos + Reglas	Asesorar, Guiar
Estado de simulación	Stottler Inc.	Teclado y ratón	Eventos	Estado de la simulación	HLA/DIS	Restric. sobre Maq. Est. Fin.	Máquinas de Estados Finitos	Tutorizar, Evaluar, Asesorar
	Stottler Inc. Helicópteros	Simulador de vuelo	Eventos	Estado de la simulación	Mensajes	Restric. sobre Maq. Est. Fin.	Máquinas de Estados Finitos	Tutorizar, Evaluar, Asesorar
STEVE	STEVE	Dataglove	Eventos	Estado de la simulación	Mensajes	Situated Plan Attribution	Red causal	Diagnostic, Evaluar, Contestar preguntas, Demostrar
Gestión proc. independiente	Máquina herramienta	Dataglove	Eventos	Estado de la simulación	Mensajes	Modelo de restricciones y Modelo de Errores	Red de Petri	Evaluar, Tutorizar
	LAHYSTOTRAIN	Haptico	Eventos	Estado de la simulación	Mensajes	Situated Plan Attribution	Grafo de eventos	Evaluar, Explicar, Responder preguntas, Tutorizar
	Cirugía Billingham	Voz, trackers	Eventos multimodales	Estado de la simulación + DCs	Mensajes	Reconocimiento de Planes y Restricciones	Grafo de pasos + reglas	Asesorar, Demostrar

Tabla 1.1: Descripción de arquitecturas de construcción de SI<sup>2</sup>A<sup>2</sup>

Las acciones de los alumnos se pueden considerar de dos maneras: como eventos instantáneos o como actividades continuas que se dan durante un intervalo de tiempo, es decir, con un instante de inicio y un instante de fin. En el grupo de los sistemas que consideran acciones continuas se encuentran el sistema de entrenamiento de operarios de excavadora y los sistemas de entrenamiento de cirujanos. El primero considera los movimientos del brazo articulado de la excavadora como una acción con principio y fin que debe ser valorada en toda su duración, y en el caso de la cirugía sucede lo propio con los movimientos que hace el cirujano con el instrumental. En cambio, los sistemas que consideran las acciones del alumno como eventos se emplean en dominios en los que las acciones son

del tipo pulsar botones y activar controles (STEVE, máquina herramienta, Stottler). En otros casos se generan eventos para identificar actividades continuas, es decir, una acción que en el Sistema Interactivo es ejecutada durante un intervalo de tiempo es transformada en un evento instantáneo (ICAT, Stottler helicópteros, LAHYSTOTRAIN, cirugía Billingham). No queda claro en que momento se lanza el evento (al principio de la acción, al final, en algún momento intermedio). En cualquier caso, esta transformación implica una pérdida de información sobre la forma en que el alumno realiza la acción a lo largo del intervalo.

Teniendo en cuenta esta diferencia, se observa que:

- Los sistemas que consideran las **acciones** del alumno **como eventos**:
  - Son los que cuentan con las capacidades de ayuda al aprendizaje más avanzadas, como tutorizar, hacer demostraciones, responder preguntas, además de formas de evaluación sofisticadas. En este sentido, es necesario analizar cuales son las dificultades que implica tratar con Sistemas Interactivos en los que la actividad es continua.
  - En la mayoría de los casos, los sistemas que transforman las acciones del alumno en eventos emplean alguna forma de grafo para representar la actividad procedimental.
  - Implementan un proceso de detección de errores basado en la verificación de restricciones sobre las acciones que contiene el modelo procedimental (en los casos en que existe).
  - Su eje de integración es el estado de la simulación, es decir, analizan la información disponible en el Sistema Interactivo en lugar de delegar la detección de errores en algún módulo de la simulación como hacen los sistemas que representan acciones continuas.
- Los sistemas que representan **acciones continuas**:
  - Aplican mecanismos específicos de detección de errores que solo son aplicables en cada aplicación particular.
  - Lanzan eventos para informar sobre los errores cometidos, aunque sus capacidades son más limitadas e inflexibles y no profundizan en el alcance educativo de cada error.

El segundo grupo de observaciones sobre las arquitecturas de construcción de SI<sup>2</sup>A<sup>2</sup> se centra en la forma de distribuir las funcionalidades de ayuda al aprendizaje:

- Las arquitecturas basadas en eventos de errores, prácticamente se limitan a informar sobre los errores cometidos, por lo que no existe una distribución clara.
- Los sistemas de Stottler Inc. (Sección 1.3.1.2) integran varias funcionalidades sobre un mismo modelo basado en máquinas de estados finitos.
- Destacan aquellos sistemas que como mínimo independizan la gestión del conocimiento procedimental y el componente de detección de errores. Esta aproximación es interesante, ya que cualquiera que sea el tipo de ayuda al aprendizaje que ofrezca el SI<sup>2</sup>A<sup>2</sup>, las decisiones didácticas comienzan por el análisis de los errores cometidos por el alumno. Esto se traduce en la obligatoriedad de implementar el proceso de detección de errores en cualquier SI<sup>2</sup>A<sup>2</sup>.
- Se puede constatar que distribuir varias funcionalidades educativas en módulos independientes no supone un problema, tal y como se desprende del sistema LAHYSTOTRAIN que divide sus funciones en varios agentes especializados.

Para finalizar, es necesario comentar la existencia de una iniciativa promovida conjuntamente por la Simulation Interoperability Standards Organization (SISO) y el IEEE Learning Technology Standards Committee (LTSC). El grupo de trabajo que han formado está estudiando la creación de un estándar para la unión de entornos de simulación con entornos de aprendizaje basados en SCORM. Actualmente, su trabajo se limita a algunas propuestas generales que están en proceso de discusión y que no se puede garantizar que formen parte del estándar final. Sin embargo, por la relevancia de sus objetivos para este proyecto de tesis, se incluyen y analizan algunas de ellas en el Anexo A. Siguiendo la misma línea, existen otras iniciativas para unir SCORM u otras arquitecturas de aprendizaje desarrolladas a medida y algún tipo de entorno interactivo (de Antonio et al., 2005; Moreno-Ger et al., 2008). Sin embargo, están dirigidas a entornos de interacción simple (*point and click*), que pueden ser aptas para entornos web, pero que son comparables a la interacción con aplicaciones 2D, por lo que no se han incluido en este análisis.

## 1.4 CONCLUSIONES

Este capítulo comenzó describiendo las características generales de los Sistemas Interactivos basados en Realidad Virtual/Mixta y de los Sistemas de Ayuda al Aprendizaje. Dicha descripción fijó el punto de partida para incidir en el problema principal de integración de un SI<sup>2</sup>A<sup>2</sup>: el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje modelan el mismo mundo con representaciones y objetivos distintos. Esta particularidad implica que los datos del Sistema Interactivo que sirven para simular el Entorno Virtual intersectan con los datos que necesita el Sistema de Ayuda al Aprendizaje. Por tanto, el SI<sup>2</sup>A<sup>2</sup> tiene que utilizar mecanismos de comunicación que le permitan mantener la coherencia de los datos, además de asignar a cada sistema la responsabilidad de gestionar su parte. El estudio realizado en el resto del capítulo se ha centrado en identificar las arquitecturas más relevantes en la construcción de SI<sup>2</sup>A<sup>2</sup> para determinar cómo se organizan, qué mecanismos de gestión emplean y cuáles son sus carencias. Hay varias conclusiones obtenidas del análisis realizado que merece la pena resaltar:

- Las arquitecturas descritas muestran características comunes entre SI<sup>2</sup>A<sup>2</sup> diferentes que sugieren pautas de desarrollo útiles en función de cada tipo de dominio. Sin embargo, ninguna de las aproximaciones es suficientemente general como para ser aplicada a una variedad relevante de sistemas.
- Los SI<sup>2</sup>A<sup>2</sup> más avanzados por sus funcionalidades se limitan a dominios fácilmente formalizables o a simplificaciones de los dominios. Por ejemplo, la actividad del alumno en los sistemas de entrenamiento de cirujanos, o bien se simplifica transformando acciones continuas en eventos, o bien se suprimen del diseño las funcionalidades educativas. Cualquiera de las dos alternativas reduce la capacidad del SI<sup>2</sup>A<sup>2</sup> y, por tanto, no son deseables. Esta situación pone de manifiesto un problema que es necesario analizar.
- Ahondando en la conclusión anterior, resulta llamativa la ausencia de sistemas de entrenamiento que incorporen capacidades de ayuda al aprendizaje en algunos dominios. Por ejemplo, hay grandes progresos en simuladores de conducción de automóviles, pero sólo pequeños avances para construir SI<sup>2</sup>A<sup>2</sup> (Weevers et al., 2003). Lo mismo sucede

con otro tipo de vehículos o maquinaria, la enseñanza de destrezas físicas, etc.

- En general, los mayores problemas para la construcción de SI<sup>2</sup>A<sup>2</sup> aparecen en dominios procedimentales poco estructurados, desarrollados en entornos muy dinámicos, y en los que la destreza física juega un papel importante.

A la vista de estas conclusiones, es necesario estudiar en profundidad y acotar cuáles son los problemas principales a los que una Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> tiene que dar respuesta para progresar en la investigación sobre SI<sup>2</sup>A<sup>2</sup> en todo tipo de dominios procedimentales. El capítulo siguiente presenta dicho estudio.

## CAPÍTULO 2

# ANÁLISIS DEL PROBLEMA

A partir del análisis de antecedentes descrito en el capítulo anterior, resulta claro que definir una Metodología que homogeneice los pasos para construir un SI<sup>2</sup>A<sup>2</sup> es complicado:

- 1) Cada dominio tiene sus propias características individuales;
- 2) Los intereses del instructor y el tipo de alumnos particularizan aún más el desarrollo de los SI<sup>2</sup>A<sup>2</sup>;
- 3) En algunos casos, las limitaciones técnicas de los Sistemas Interactivos también restringen las decisiones de diseño.

De ahí la variedad de enfoques diferentes descritos en el capítulo anterior y la cantidad de particularidades que se observan. Por esta razón, diseñar una Metodología de propósito general para el desarrollo de SI<sup>2</sup>A<sup>2</sup> requiere un trabajo de análisis en el que se especifique la problemática a la que se ha de dar respuesta. En este capítulo se expone dicho análisis.

El objetivo planteado es diseñar una Metodología para simplificar y generalizar el proceso de construcción de SI<sup>2</sup>A<sup>2</sup>, destacando que dicha **Metodología** tiene que ser válida **para dominios diferentes**. Por este motivo, el estudio expuesto en este capítulo incide en los requerimientos que surgen de esta decisión. Para abordar distintos dominios la Metodología debe ser independiente tanto del Sistema Interactivo como de las funcionalidades de ayuda al aprendizaje que se quieran integrar. A continuación se presenta dicho estudio que comienza revisando las características de los Sistemas Interactivos y continúa acotando el nexo entre



la Metodología y los Sistemas de Ayuda al Aprendizaje. La última sección del capítulo presenta un análisis general del proceso de integración y una síntesis del diseño de la Metodología y el Marco de Desarrollo que se presentarán en los capítulos 3 y 4 de esta memoria.

## **2.1 ANÁLISIS DE LA INTEGRACIÓN DE SISTEMAS INTERACTIVOS**

En la sección 1.1 de esta memoria se describían las posibilidades que ofrecen, en general, las tecnologías de Realidad Virtual para recrear entornos y formas de interacción. En esta sección también se describen algunas características de los Sistemas Interactivos aunque, en este caso, se centra en las más relevantes atendiendo a la conexión con un Sistema de Ayuda al Aprendizaje. En este sentido, resultan de interés las siguientes características, ya que influyen en el tipo de información que se podrá obtener sobre la actividad del alumno y el entorno virtual: diferencias entre entornos autónomos y entornos reactivos, representación de acciones continuas y eventos, realismo del Sistema Interactivo. Además, se describirán dos problemas añadidos que hay que considerar: el ruido en los datos y la incertidumbre.

### **2.1.1 Entornos autónomos – Entornos reactivos**

La diferencia entre los entornos autónomos y los reactivos reside en el comportamiento de los elementos del entorno simulado. En el primer caso, el entorno puede cambiar por sí mismo, sin la intervención del usuario. Por ejemplo, en un simulador de conducción aparecen vehículos que circulan por sí solos interactuando con el que controla el alumno. En cambio, en los entornos reactivos los elementos sólo cambian como reacción a las intervenciones del usuario. Es lo que ocurre en un sistema para aprender a desmontar componentes mecánicos: ninguna pieza cambiará de posición salvo que el usuario la mueva deliberadamente. Esta diferencia en el comportamiento del entorno influye en la arquitectura del SI<sup>2</sup>A<sup>2</sup> en dos aspectos principales: la coherencia de datos y la representación de la actividad procedimental.

Partimos de la base de que la arquitectura debe garantizar la coherencia de datos entre el Sistema Interactivo y el Sistema de Ayuda al

Aprendizaje. Si no lo hiciera, cuando el Sistema de Aprendizaje tomara una decisión educativa, cualquiera que fuera, ésta podría resultar inconsistente con la situación real. Para mantener la coherencia hay que tener en cuenta que al Sistema de Aprendizaje puede no interesarle todo lo representado en el Sistema Interactivo. El foco se encuentra en el estado de los elementos que intervienen en la actividad que desarrolla el alumno en cada instante. En los entornos autónomos hay que considerar que los elementos pueden crearse y destruirse, aparecer y desaparecer, alterar el contexto en el que está el alumno o ser intrascendentes. Por tanto, los entornos autónomos implican estrategias de gestión de datos dinámicas tanto en el lado del Sistema Interactivo como en el del Sistema de Aprendizaje. En cambio, en los entornos reactivos, los datos que el Sistema de Aprendizaje debe mantener actualizados son más previsibles, ya que no se modifican, generan o destruyen espontáneamente. Por consiguiente, las estrategias para gestionar y mantener la coherencia de datos no requieren el mismo grado de control dinámico del entorno. Ante estas diferencias, consecuentemente, la Metodología de construcción de SI<sup>2</sup>A<sup>2</sup> debe contemplar ambas posibilidades.

La distinción entre entornos autónomos y reactivos también afecta a otro aspecto del diseño del SI<sup>2</sup>A<sup>2</sup>. En un entorno autónomo, las acciones del alumno no son únicamente producto de su iniciativa sino que pueden producirse como respuesta a las intervenciones de otros elementos autónomos. Como no es posible saber en qué momentos van a producirse las interacciones, los pasos del alumno para solucionar una tarea no pueden establecerse completamente a priori, aunque sí de forma parcial o con la posibilidad de modificarlos dinámicamente. En el caso de los entornos reactivos no existe este problema, ya que las acciones del alumno producen siempre resultados previsibles, es decir, la tarea realizada en las mismas condiciones produce siempre idénticos resultados. Esta diferencia condiciona el tipo de información que se le va a comunicar al Sistema de Ayuda al Aprendizaje, que debe ser adecuada tanto para entornos autónomos como reactivos.

### **2.1.2 Acciones continuas – Eventos**

En el capítulo anterior se indicó que algunos SI<sup>2</sup>A<sup>2</sup> representan las acciones del alumno en forma de eventos instantáneos mientras que otros consideran el tiempo empleado para realizar la acción. La decisión de escoger una

forma de representación u otra viene determinada por el Sistema Interactivo y, en última instancia, por la naturaleza de cada acción. Así, para representar la pulsación de un botón virtual lo más habitual es definir un evento que indique el instante en el que el botón se activa, independientemente de que la interfaz permita simular la duración completa de la acción desde que existe un contacto con el botón hasta que éste ha sido completamente pulsado. Sin embargo, algunos SI<sup>2</sup>A<sup>2</sup> también representan en forma de evento instantáneo acciones como “hacer una incisión en un tejido”, aunque la interfaz permita realizarla como una acción continua durante un periodo de tiempo (Billinghurst et al., 1995). Esta representación se limita a detectar el instante en que se produce la colisión entre el instrumental y el tejido para informar al Sistema de Aprendizaje de que el alumno está realizando una incisión, obviando todo lo que sucede desde que comienza hasta que termina. Sin embargo, podría ser importante detectar si la fuerza ejercida por el alumno es constante, si le tiembla el pulso o si la orientación del instrumental es la adecuada de principio a fin.

Cuando se modela una acción continua como si fuera un evento instantáneo surgen dos posibilidades: (i) ignorar parte de lo que está haciendo el alumno o (ii) dejar que el Sistema Interactivo automatice parte de la acción para que su resultado sea el preestablecido por la simulación. En ambos casos, el Sistema de Aprendizaje puede concluir que el alumno sabe **qué** es lo que tiene que hacer, pero no puede determinar si sabe **cómo** hacerlo.

Los Sistemas de Ayuda al Aprendizaje que ofrecen las funcionalidades más avanzadas están centrados en dominios en los que las acciones son claramente representables mediante eventos. Sin embargo, no podemos desdeñar los dominios en los que las acciones son continuas. En este conjunto entraría el aprendizaje de destrezas físicas, y hay que tener en cuenta que la evolución de las tecnologías de Realidad Virtual va a permitir que la interacción física en un Entorno Virtual sea cada vez más realista. En esta línea, los avances en dispositivos hápticos nos permiten practicar tareas en las que el tacto juega un papel principal. Los sistemas de captura de movimiento también sirven para adquirir destrezas físicas, por ejemplo, para mejorar el rendimiento deportivo. Tampoco podemos olvidar el desarrollo de simuladores de vehículos como el de entrenamiento de operarios de excavadora presentado en la sección 1.3.1.1 (Bernold et al., 2002). Este sistema considera acciones continuas, y aunque sus capacidades

educativas son bastante limitadas y difícilmente exportables a otros dominios demuestra el interés y la utilidad de la tecnología. Por estas razones, ser capaces de abordar dominios con acciones continuas es parte prioritaria de este trabajo de tesis.

### **2.1.3 Simulación simplificada – Simulación realista**

El grado de realismo de la simulación del dominio también afecta a la construcción del SI<sup>2</sup>A<sup>2</sup>. Este factor está relacionado con el tipo de información que se puede obtener del Sistema Interactivo para transmitírsela al Sistema de Aprendizaje. Las simplificaciones de una simulación pueden afectar tanto al comportamiento de los objetos del Entorno Virtual como a la forma de interactuar del usuario, eliminando parte de la interacción que existiría en el mundo real. En esta descripción suelen encajar los sistemas de RV de escritorio, aplicaciones 3D interactivas que emplean un monitor convencional como dispositivo de visualización, además de ratón y teclado, o un dispositivo de videojuegos para interactuar. Este tipo de plataforma sirve como ejemplo para diferenciar los tipos de información que se pueden obtener de los Sistemas Interactivos mientras el alumno realiza una tarea.

En un Entorno Virtual se llevan a cabo acciones básicas como seleccionar y manipular objetos o navegar por el entorno. Sin embargo, en un contexto de aprendizaje se abstraen estas acciones básicas y se manejan conceptos de más alto nivel propios del dominio de aprendizaje. Por ejemplo, desmontar un motor es una tarea que conlleva acciones primitivas de manipulación de las partes del motor. No obstante, no nos referimos a cada paso realizado como a una manipulación. Simplemente diremos que hay que desmontar la pieza en cuestión. Ahora bien, la forma en que se realiza cada paso de desmontaje puede ser muy variable en función del Sistema Interactivo. En un sistema de escritorio las opciones de manipulación son limitadas y es probable que se emplee un comando que automatice el desmontaje. En este caso, informar al Sistema de Aprendizaje sobre lo que está haciendo el alumno es inmediato, ya que la activación del comando “Desmontar” proporciona la semántica de la acción del alumno.

En el caso de los Sistemas Interactivos realistas, siguiendo con el ejemplo anterior, el desmontaje de una pieza sería un proceso más próximo a la realidad: *el alumno busca la zona de accesibilidad a la pieza, la selecciona, y la*

*desplaza a lo largo de una trayectoria que permita su extracción evitando colisiones con otras piezas* (Borro et al., 2004). En la simulación realista no existe ningún comando que ofrezca la semántica completa de la actividad del alumno. Para un observador sería obvio que el alumno está realizando un desmontaje, pero el Sistema Interactivo no tiene por qué manejar el concepto "Desmontar". Simplemente conoce el movimiento de una herramienta, el cambio de coordenadas de una pieza, posibles colisiones, etc.

En resumen, el ejemplo anterior es una muestra sencilla de cómo el tipo de simulación se relaciona con los conceptos de aprendizaje. En general, cuanto más realista es el Sistema Interactivo, menos conceptos del dominio de aprendizaje maneja, es decir, los datos de la simulación que contiene el Sistema Interactivo no suelen ser útiles directamente para el Sistema de Aprendizaje, por lo que es necesario interpretarlos antes de transmitirlos. La elección de un sistema u otro vendrá determinada por los requerimientos educativos que se planteen para el SI<sup>2</sup>A<sup>2</sup>. Mientras que con la simulación simplificada únicamente se puede determinar que el alumno conoce qué piezas debe desmontar en cada situación, con la simulación realista se puede verificar que además sabe cómo hacerlo.

#### **2.1.4 Ruido e incertidumbre en los datos generados por los Sistemas Interactivos**

En líneas generales, buena parte de las decisiones educativas que toma un SI<sup>2</sup>A<sup>2</sup> se basan en el análisis de la actividad del alumno cuando utiliza el Sistema Interactivo (Self, 1974). Por tanto, es necesario valorar la fiabilidad de los datos que éste genera. En este sentido, se han identificado dos problemas principales que pueden comprometer la funcionalidad del Sistema de Ayuda al Aprendizaje. Se trata del ruido y de la incertidumbre.

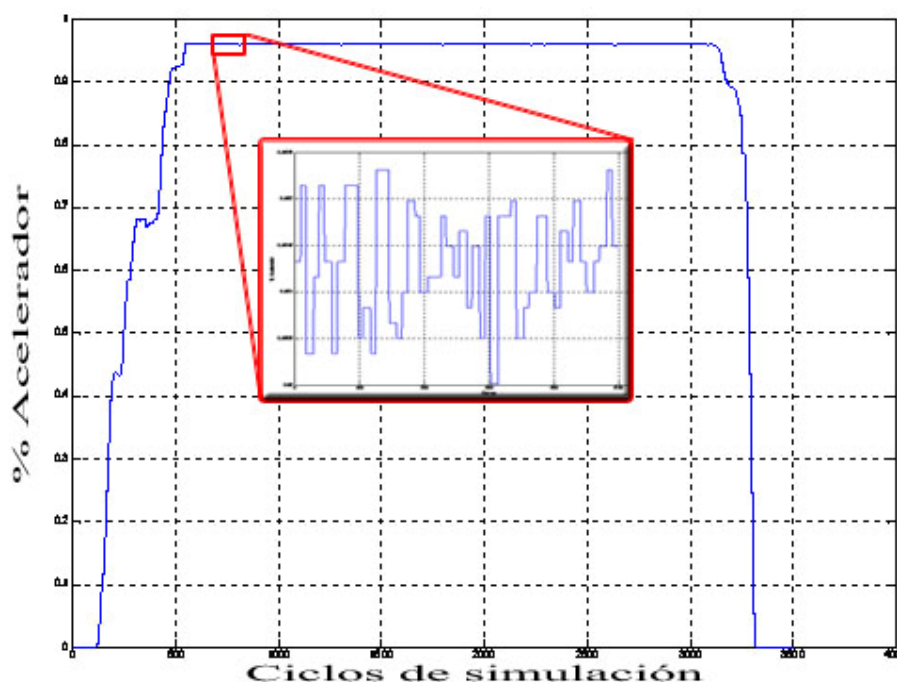


Figura 2.1: Muestra de ruido al pisar el acelerador de un simulador

Hablamos de ruido cuando un flujo de datos incorpora una distorsión que altera el significado del mensaje que se está comunicando. En la Figura 2.1 se muestra un ejemplo de ruido en los datos de posición de un pedal de aceleración de un simulador al realizar las siguientes acciones: “Acelerar a fondo”, “Mantener pisado el acelerador” y “Soltar el acelerador”. Como se puede observar, los valores de posición del acelerador fluctúan en cada ciclo de simulación. Incluso cuando el pedal está pisado a fondo (detalle de la gráfica) existen ligeras variaciones. Si considerásemos que el estado del pedal en cada ciclo representa la actividad del estudiante, comunicárselo al Sistema de Aprendizaje le estaríamos induciendo a error. Solamente en el detalle de la gráfica, que corresponde con menos de 4 segundos de ejecución, se observan dos docenas de variaciones en la posición del pedal, que obviamente no son representativas de la actividad. Esta fluctuación es ruido que altera el mensaje real de las acciones del estudiante. Por consiguiente, la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> tiene que facilitar la incorporación de medidas para evitar este problema.

El problema de la incertidumbre también afecta a la actividad del Sistema de Ayuda al Aprendizaje. Hablamos de incertidumbre cuando no se dispone de datos suficientes en el Sistema Interactivo para especificar exactamente la actividad realizada por el alumno. Esta situación se puede producir por varios motivos. El más evidente es que realmente el Sistema Interactivo no genere los datos necesarios. Puede deberse a un error de diseño, a que no sea posible que el software obtenga esos datos o a que el coste computacional para obtenerlos sea excesivo. La gravedad de ignorar parte de las acciones del alumno habría que valorarla en cada caso. Otra posibilidad es que aparezca una incertidumbre momentánea. La incertidumbre es momentánea si en un ciclo de simulación se detectan indicios de que el alumno está realizando una acción, pero éstos son insuficientes para especificarla en ese instante; no obstante en ciclos posteriores se dispondrá de los datos necesarios para hacerlo. Esta posibilidad añade complejidad al SI<sup>2</sup>A<sup>2</sup> ya que surgen las siguientes cuestiones: en un momento de incertidumbre, ¿se puede estar seguro de que llegarán todos los datos en ciclos posteriores? ¿es necesario esperar a tener la especificación completa de la acción del alumno para analizarla? Si se espera a tener los datos completos, el Sistema de Aprendizaje analizará acciones pasadas, no presentes. ¿Cómo se gestiona el tiempo en esos casos? Lo que en un primer momento parece un indicio de una acción, ¿puede finalmente no serlo? ¿Pueden estar los indicios asociados a distintas acciones? ¿Hay que resolver ambigüedades? Estos problemas asociados a la incertidumbre deben ser controlados por el SI<sup>2</sup>A<sup>2</sup>, por lo que se abordarán en el Marco de Desarrollo de SI<sup>2</sup>A<sup>2</sup>.

## **2.2 ANÁLISIS DE LA INTEGRACIÓN DE SISTEMAS DE AYUDA AL APRENDIZAJE**

Un Sistema Inteligente de Ayuda al Aprendizaje está compuesto por una variedad significativa de componentes que proveen diversas funcionalidades educativas. Por tanto, a la hora de plantear de forma general la construcción de un SI<sup>2</sup>A<sup>2</sup> no se puede considerar la parte de aprendizaje como un único elemento que presente las mismas características para cualquier dominio. Uno de los objetivos planteados en este trabajo es que la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> sea independiente de los componentes específicos de ayuda al aprendizaje que se pretendan

integrar y de sus características internas. En este sentido, se indicó como requisito de partida que los componentes de ayuda al aprendizaje relevantes para la Metodología son un modelo de representación de tareas procedimentales y un proceso de detección de errores de los que ya se debe disponer. De esta forma, en este trabajo no se abordará la integración de ningún componente específico de Ayuda al Aprendizaje aparte de los previamente mencionados, y la Metodología se diseñará teniendo en cuenta esta acotación.

Puesto que los motivos de esta decisión no son obvios, a continuación se presenta el análisis realizado que la justifica. En este análisis se ha recurrido a los estudios de la psicología y de la informática educativa que inciden en los distintos niveles del proceso educativo. A partir de ellos, se pueden determinar las necesidades comunes de representación de conocimiento en los Sistemas de Ayuda al Aprendizaje de tareas procedimentales, y establecer las bases para su integración con los Sistemas Interactivos. La sección posterior analiza las dificultades que implica la decisión tomada.

### ***2.2.1 Nexo de la Metodología con los Sistemas de Ayuda al Aprendizaje: Un enfoque educativo***

La finalidad de cualquier proceso educativo es conseguir que el alumno adquiera nuevos conocimientos, habilidades o destrezas, lo que se conoce como **objetivos de aprendizaje**. Adquirirlos no requiere de las mismas capacidades intelectuales o físicas por parte del alumno. Por esta razón, varios investigadores han propuesto taxonomías para diferenciarlos (Marzano y Kendall, 2007; Pitrat, 1991; Romizowski, 1981). La más extendida es la de Benjamin S. Bloom, que separa los objetivos en tres categorías principales: Afectivos, Cognitivos y Psicomotores. Bloom especifica las subcategorías de los dos primeros en (Bloom et al., 1956; Krathwohl et al., 1964), siendo la taxonomía de objetivos cognitivos la más relevante. Los objetivos psicomotores no llega a desarrollarlos, aunque otros autores lo han hecho posteriormente (Dave, 1975; Harrow, 1972; Simpson, 1972).

En la Tabla 2.1 se muestra un resumen de las taxonomías en cada categoría.



Cognitivos	Afectivos	Psicomotores		
		(Dave, 1975)	(Harrow, 1972)	(Simpson, 1972)
Conocimiento	Recibir	Imitación	Movimientos reflejo	Percepción
Comprensión	Responder	Manipulación	Movimientos fundamentales	Preparación
Aplicación	Valorar	Precisión	Percepción, respuesta a estímulos	Respuesta guiada
Análisis	Organización, jerarquía de valores	Coordinación	Cualidades físicas	Mecanización
Síntesis	Caracterización, opción, elección	Naturalización	Destrezas	Respuesta compleja evidente
Capacidad crítica			Comunicación corporal	Adaptación
				Creación

Tabla 2.1: Taxonomía de objetivos de aprendizaje

La especificación de los objetivos de aprendizaje es fundamental, ya que constituye la base del proceso de enseñanza. En función del tipo de objetivos definidos, el instructor selecciona la estrategia de enseñanza que cree apropiada para que los alumnos aprendan de forma efectiva (Cox y Ram, 1994). *La estrategia define la forma en que se le presenta al alumno el conocimiento que se desea transmitir, las actividades que se le proponen y el proceso de instrucción que seguirá el sistema.* Es importante una buena elección, ya que una estrategia inadecuada puede causar problemas de aprendizaje (Clark, 1988). Por esta razón muchas herramientas educativas fundamentan sus estrategias en teorías cognitivas sobre el aprendizaje (Kearsly, 2001). Dichas teorías establecen hipótesis sobre los mecanismos que emplea el ser humano para aprender. De esta manera, el sistema informático se rige por criterios pensados para facilitar el proceso de aprendizaje de los alumnos.

Contextualizando en el marco de los dominios procedimentales, una de las teorías cognitivas más utilizadas es la ACT\* propuesta por Anderson (1983; 1993). Según esta teoría el conocimiento de un individuo es inicialmente declarativo y el conocimiento procedimental lo adquiere mediante la práctica de las tareas realizando inferencias a partir del conocimiento que ya tiene. Gracias a la práctica de tareas en el contexto adecuado el individuo refina, refuerza y adquiere nuevo conocimiento. Esta teoría da validez a los sistemas de aprendizaje que aplican la técnica de “aprender haciendo” (*learning by doing*) (Anzai y Simon, 1979). Mediante

esta técnica el alumno progresa en su aprendizaje al aplicar sus conocimientos para realizar tareas y al comprobar la utilidad del conocimiento adquirido en situaciones reales. Así, independientemente de cuáles sean las estrategias y las técnicas escogidas, es seguro que la construcción del SI<sup>2</sup>A<sup>2</sup> implica la definición del conjunto de tareas que se le van a proponer al alumno.

Puesto que las tareas se realizan en un Sistema Interactivo, el nexo desde el punto de vista educativo entre éste y el Sistema de Aprendizaje tiene que quedar acotado en el contexto de la tarea. El alcance y la problemática de ese contexto son importantes para definir la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup>. Su estudio parte de las investigaciones que indican en términos generales qué factores hay que considerar para definir cada tarea (Bauer, 2002; Ferrero, 2004; Shute, 2004). En los puntos siguientes se analiza la relación de cada factor con los SI<sup>2</sup>A<sup>2</sup>:

- **La definición de los objetivos de aprendizaje:** El primer factor a considerar para definir una tarea son los objetivos de aprendizaje que se quiere que trabaje el alumno. Según la taxonomía de Bloom, existen objetivos de diferente naturaleza. Sin embargo, la definición de tareas no se puede realizar aislando los objetivos de distintos tipos. En un proceso de aprendizaje complejo se entremezclan los objetivos de distintas categorías y subcategorías, con lo cual, una estrategia efectiva tiene que proponer tareas que los integren y coordinen (van Merriënboer, 1997). En el contexto de los SI<sup>2</sup>A<sup>2</sup>, este planteamiento es de gran relevancia, ya que una tarea procedimental puede requerir la puesta en práctica de destrezas psicomotoras mediante un sistema de Realidad Mixta. Pero además de comprobar la destreza del alumno, hay que evaluarlo en un contexto en el que demuestre su conocimiento sobre el dominio. Por ejemplo, si un SI<sup>2</sup>A<sup>2</sup> en los dominios de la cirugía o la conducción pretende facilitar la transferencia de habilidades del Entorno Virtual al real, es necesario que incluya tareas que integren habilidades psicomotoras y cognitivas.
- **La identificación de los datos que evidencien el nivel de aprendizaje del alumno por cada objetivo:** Este factor es la respuesta a la pregunta: ¿Cómo sabemos si un alumno ha alcanzado un objetivo de aprendizaje? De hecho, además del diseñador de la tarea, el Sistema de Aprendizaje tiene que dar respuesta a esta misma pregunta. Para que éste pueda

tomar decisiones educativas tiene que saber cuál es el estado de aprendizaje del alumno para cada objetivo. Además de proponer tareas ideadas para poner en práctica una serie de objetivos de aprendizaje, hay que determinar si el alumno los ha adquirido, o en los SI<sup>2</sup>A<sup>2</sup> más sencillos, al menos proveer de información para que el instructor lo haga. Por tanto, cuando se plantea un objetivo de aprendizaje, hay que saber si se pueden obtener los datos necesarios. En algunos dominios, la mera identificación es realmente complicada. Por ejemplo, en el caso de la cirugía la identificación de datos objetivos que valoren las destrezas del alumno sigue siendo objeto de discusión (Rosenthal et al., 2006).

- **La identificación de las tareas que faciliten obtener los datos considerados:** Proponer una tarea de la que obtener los datos que permitan determinar si un alumno ha alcanzado los objetivos de aprendizaje repercute directamente en la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup>. Hay que recordar que los datos se obtienen a partir de la actividad del alumno en el Sistema Interactivo. Por tanto, su representación en el Sistema de Aprendizaje es un foco de interés. La representación procedimental ha de ser apta para razonar sobre el estado de conocimiento del alumno, y lo normal es que se empleen los términos utilizados habitualmente en el dominio de aprendizaje. Por tanto, hay que ser cuidadosos a la hora de especificar las tareas y preguntarse cómo de directa es la relación entre dichos términos y los datos gestionados por el Sistema Interactivo. Más aún, al diseñar la tarea es necesario asegurarse de que el Sistema Interactivo permite practicarla con la fidelidad suficiente como para obtener datos válidos. Por ejemplo, en un sistema para evaluar la destreza de un soldador, si se pretende comprobar que al alumno no se le queda pegado el electrodo al metal, o su reacción si le ocurriera, es necesario que el sistema permita una interacción y simulación realistas. Este caso incluso impondría el uso de un dispositivo háptico para manejar el soldador virtual, delatando así la relación entre los objetivos de aprendizaje y el diseño del Sistema Interactivo.

De los puntos anteriores se extrae una serie de conclusiones que sirven para delimitar los problemas a los que la Metodología tiene que dar respuesta desde el punto de vista del Sistema de Ayuda al Aprendizaje. Estas conclusiones se sintetizan en los siguientes puntos:

- La funcionalidad de un Sistema de Ayuda al Aprendizaje puede ser muy variable en función de los requerimientos impuestos por los expertos de las materias. Por tanto, la estrategia de enseñanza, teorías de aprendizaje o el tipo de ayuda que se le de al alumno (pistas, demostraciones, asistencia interactiva...) no pueden ser predeterminados por la Metodología.
- Nos centraremos en la técnica de “aprender haciendo” como base para el aprendizaje de tareas procedimentales mediante SI<sup>2</sup>A<sup>2</sup>.
- La relación entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje comienza en la definición de tareas, que se supone que cubren de manera correcta y completa los objetivos de aprendizaje para los que hayan sido diseñados.
- Las tareas se tienen que diseñar para obtener datos que evidencien el conocimiento del estudiante. Es responsabilidad del experto en la materia diseñar las tareas adecuadas para que el aprendizaje sea efectivo, pero es responsabilidad de la Metodología especificar la representación de los datos asociados a las tareas.
- No se puede dar por supuesto que la representación de datos en el Sistema Interactivo tenga una forma similar a la del Sistema de Aprendizaje de forma que la comunicación sea inmediata. Por tanto, la Metodología tiene que ofrecer soluciones apropiadas de interpretación de datos para hacer posible la comunicación.

La obtención de datos que evidencien el estado de conocimiento del estudiante implica el análisis de la actividad del estudiante resolviendo las tareas. Los SI<sup>2</sup>A<sup>2</sup> presentados en el capítulo de antecedentes realizan esta función de forma específica para los dominios de aprendizaje que pueden modelar. En el caso de este trabajo, se necesitan métodos de propósito general que permitan abordar el análisis en un amplio espectro de dominios procedimentales. Puesto que esta funcionalidad puede conllevar una gran complejidad, en la siguiente sección se estudia la problemática asociada y se define el enfoque a seguir para diseñar la Metodología.

### **2.2.2 Análisis de la actividad del alumno**

Determinar el estado de conocimiento del alumno a partir del análisis de su actividad es el objetivo del **diagnóstico cognitivo** (Ohlsson, 1994). Una

parte fundamental de este proceso consiste en la detección de los errores cometidos por el alumno mientras resuelve tareas. Esta labor es compleja, más aún en dominios procedimentales donde cada tarea puede tener múltiples soluciones. Esta peculiaridad requiere reconocer los errores y analizarlos desde distintos puntos de vista: cómo afectan a la solución global de la tarea, en qué modifican la resolución de la tarea desde el momento en que se producen, cuál es su gravedad individual en el contexto en el que se han producido, y qué denotan respecto al estado de conocimiento del alumno.

Al margen de sus capacidades, la localización del proceso de diagnóstico en la arquitectura de un SI<sup>2</sup>A<sup>2</sup> tampoco es trivial. En el capítulo de antecedentes se prestó especial atención a la gestión y dependencias del proceso de detección de errores que mostraban las aplicaciones. Se realizan aproximaciones de todo tipo y generalmente ligadas al dominio: Se observa dependencia del Sistema Interactivo, de otras funcionalidades del Sistema de Aprendizaje, o excepcionalmente, aproximaciones generales limitadas a dominios muy estructurados y de actividad basada en eventos.

La localización del proceso de diagnóstico en el Sistema de Ayuda al Aprendizaje tampoco es evidente. Algunas aproximaciones lo integran en el módulo del estudiante (Self, 1992), ya que el objetivo final es actualizar el modelo con el estado de conocimiento deducido por el proceso de diagnóstico. Otras, en cambio, definen el módulo del estudiante como una estructura de datos separada del proceso de diagnóstico encargado de manipularla (Ragnemalm, 1996; VanLehn, 1988). Ambas aproximaciones pueden estar justificadas. Sin embargo, cualquiera de ellas resulta poco prometedora en el contexto de este trabajo. Parece claro que el módulo del estudiante es necesario para adaptar el proceso de enseñanza a cada alumno, por lo que carece de sentido eliminarlo. Ahora bien, su contenido y su forma para lograr la conducta inteligente deseada ha sido objeto de intenso debate desde el comienzo de la investigación en Sistemas Tutores Inteligentes y lo seguirá siendo.

Desde un punto de vista pragmático, el modelo de estudiante debe contener los datos que requieran los procesos que lo manipulen y representarlos de la forma más adecuada para ello (Self, 1990). Por tanto, si consideramos el proceso de diagnóstico como parte del modelo de estudiante o como un proceso que lo manipula, estamos estableciendo una

dependencia directa entre ambos que supone un problema: abordar el diagnóstico cognitivo desde una perspectiva general, ¿implica que estamos obligados a tener un modelo de estudiante genérico e independiente del dominio? La discusión de si es posible o no, o de lo completo que tendría que ser dicho modelo, podría ser interminable.

Sin embargo, este problema se puede resolver mediante otra alternativa. Ferrero (2004; 2005) propone en su tesis doctoral un componente de diagnóstico genérico e integrable capaz de diagnosticar en dominios procedimentales denominado DETECTive. Cuenta en su arquitectura con un modelo de estudiante específico para un componente de diagnóstico independiente. El modelo no está planteado para realizar labores de planificación instruccional, sino que está orientado a la tarea. El sistema registra en este modelo los resultados de diagnóstico de las tareas realizadas por cada alumno. Así el instructor obtiene información útil para seguir el progreso de sus alumnos (Gugerty, 1997). Además, DETECTive se ha integrado en STIs generados por la herramienta de autor IRIS (Arruarte et al., 2003), que se encargarían de la planificación instruccional. Desde una perspectiva más general, sin contar con la herramienta IRIS, el modelo de estudiante de DETECTive podría formar parte de una arquitectura más compleja basada en componentes interoperables (Brusilovsky et al., 2005).

Desde este punto de vista, esta aproximación de DETECTive constituye un punto de partida adecuado para tratar el diagnóstico cognitivo en SI<sup>2</sup>A<sup>2</sup> ejerciendo de nexo con otras posibles funcionalidades del Sistema de Ayuda al Aprendizaje. En la siguiente sección se describe el sistema en profundidad y se analiza cómo encajan sus características en el contexto de los SI<sup>2</sup>A<sup>2</sup>.

### **2.3 DETECTIVE: EL PUNTO DE PARTIDA PARA EL DIAGNÓSTICO EN SI<sup>2</sup>A<sup>2</sup>**

DETECTive es un componente de diagnóstico genérico capaz de diagnosticar en distintos tipos de dominio, incluyendo los procedimentales (Ferrero, 2004; Ferrero et al., 2005). Su diseño está concebido de acuerdo a los siguientes propósitos: **Genericidad** para poder ser adaptado a distintos dominios, capacidad de **diagnóstico apropiado y suficiente** para ser útil en procesos de aprendizaje, y **usabilidad** para facilitar su utilización. Los dos

primeros propósitos los consigue formalizando mecanismos generales de diagnóstico. Como resultado, DETECTive está formado por un *Modelo de Tareas* y por un *Modelo de Diagnóstico múltiple* que se presentan a continuación (Figura 2.2). Por otra parte, el propósito de usabilidad se alcanza gracias a una herramienta de autor, KADI (Martín et al., 2004), que simplifica la definición de las características del dominio mediante una interfaz gráfica.

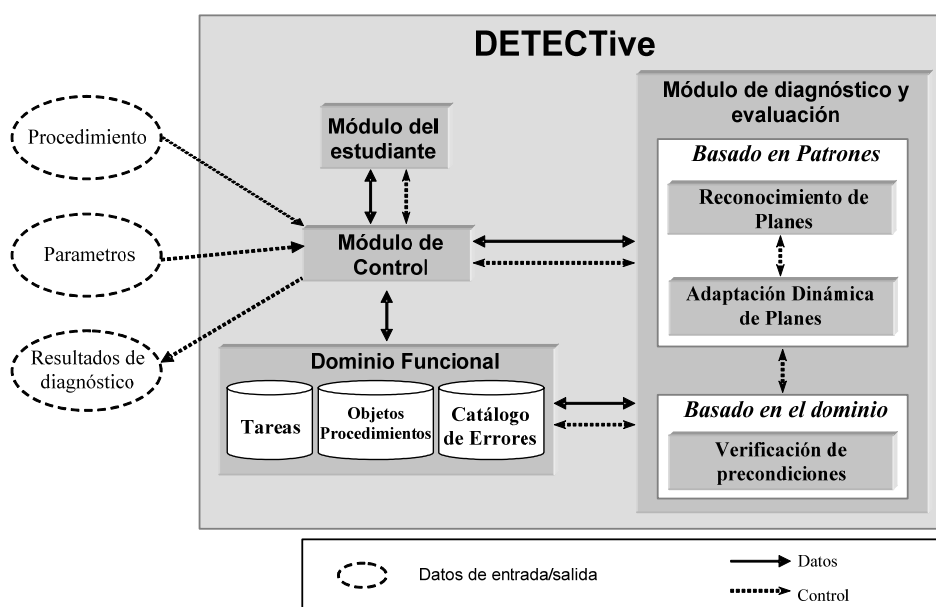


Figura 2.2: Arquitectura de DETECTive

### ➤ El modelo de tareas de DETECTive

El modelo de tareas establece una ontología para describir los elementos del dominio procedimental que deben definirse para formalizar el proceso de diagnóstico. Estos elementos son los siguientes:

- **Los objetos y procedimientos del dominio:** Un *objeto* es un elemento del dominio que tiene su propia representación en el Sistema Interactivo. El Modelo de Tareas establece su caracterización mínima, permitiendo introducir nuevas relaciones y propiedades para definir sus peculiaridades. Los *procedimientos* representan las actividades que el alumno puede llevar a cabo para resolver tareas. La ejecución de un procedimiento se refleja en las propiedades de los objetos, que cambian

de valor modificando su *estado*. La especificación de los *procedimientos* contiene las precondiciones que deben cumplirse para poder ser ejecutados y las postcondiciones que se darán tras su realización. Asimismo, DETECTive puede determinar los cambios que la ejecución del procedimiento causa en los objetos del dominio. Mediante este mecanismo, DETECTive tiene el potencial para hacer simulaciones sencillas, aunque sólo en entornos reactivos en los que los cambios se producen expresamente por la acción del alumno.

- **Las tareas:** La *tarea* se emplea para inferir el estado de conocimiento del alumno (Almond et al., 2002; Mislevy et al., 1998). Incluye el criterio para evaluarla y está asociada a un conjunto de objetivos de aprendizaje definidos por el instructor. También contiene los patrones de solución más habituales y un conjunto de errores típicos y desviaciones que identifican lagunas de conocimiento.
- **Resultados de diagnóstico:** Contiene la solución propuesta por el alumno a la tarea junto con el diagnóstico de cada paso realizado y una valoración global en forma de puntuación. Esta información identifica los aspectos de la resolución de la tarea que se consideran relevantes para el instructor, el alumno, o para otros módulos de un Sistema de Ayuda al Aprendizaje.

La visión computacional del modelo de tareas de DETECTive identifica tres niveles de abstracción o *Planos de Conocimiento*. El nivel superior o *Plano Abstracto* contiene la metaontología para definir el dominio, constituyendo la base sobre la que se sustenta la independencia del dominio. Las especificaciones contenidas en él, permiten definir los tipos de ejercicios más adecuados a las características de la materia y a los objetivos de aprendizaje que se desean diagnosticar. Este plano se refina en los *Planos Concreto y de Resolución* que contienen los elementos que describen el dominio de la materia en cuestión.

### ➤ El modelo de diagnóstico múltiple de DETECTive

El modelo de diagnóstico aplica un proceso híbrido que integra distintas técnicas para identificar los errores cometidos y evaluar al alumno. Las técnicas empleadas son: Reconocimiento de planes (Genesereth, 1982; Kautz y Allen, 1986), adaptación dinámica de planes y verificación de precondiciones. DETECTive aplica en cada acción del alumno la técnica más



efectiva que puede siguiendo una estrategia estática. En primer lugar, utilizando la técnica del Reconocimiento de Planes, intenta reconocer la acción del alumno como parte de alguno de los patrones de solución definidos por el instructor. En caso de que la acción no se ajuste a ninguno de ellos, intenta adaptarlos mediante alguna de las desviaciones que representan lagunas de conocimiento del alumno. Si ninguna de estas técnicas produce resultados, abandona el diagnóstico basado en planes pasando a aplicar un diagnóstico específico que utiliza la articulación de los procedimientos en pre- y post-condiciones para determinar la adecuación de cada paso del alumno. Esta última técnica no puede garantizar unos resultados de diagnóstico exactos: puede ocurrir que el alumno llegue a una solución errónea ejecutando procedimientos para los que se verifican todas sus precondiciones. No obstante, estas situaciones suelen ser debidas a una incorrecta o incompleta definición del dominio funcional y pueden solventarse incorporándole nuevo conocimiento. Por otro lado se ofrece al instructor la posibilidad de revisar la respuesta del alumno e incorporarla, si lo considera adecuado, como un nuevo patrón de solución para el ejercicio.

Cuando DETECTive detecta un error del alumno, puede ofrecerle *feedback* inmediatamente o bien esperar al final de la tarea para hacerlo (Mathan y Koedinger, 2003). El *feedback* lo define el instructor junto al resto del dominio, aunque la estrategia para dar *feedback* podría quedar en manos de otros componentes (López-Garate et al., 2008a) en el caso de que DETECTive forme parte de algún Sistema de Ayuda al Aprendizaje.

#### ➤ **Idoneidad de DETECTive para construir SI<sup>2</sup>A<sup>2</sup>**

En este apartado se identifican los puntos a favor (✓) y en contra (✗) de DETECTive para formar parte de una arquitectura de propósito general para construir SI<sup>2</sup>A<sup>2</sup>. Estos puntos se exponen a continuación:

- ✓ La aproximación de DETECTive consigue que el modelo de tareas, el módulo del estudiante y las técnicas de diagnóstico sean independientes de los sistemas en los que se integre. Son independientes tanto de otras funcionalidades del Sistema de Ayuda al Aprendizaje como de los Sistemas Interactivos.
- ✓ El modelo de tareas trata los dominios procedimentales de forma genérica, y es apto para abordar el diagnóstico en dominios diferentes.

- ✓ La representación de los procedimientos del dominio es apta para definir tareas que integren objetivos de aprendizaje de distinta categoría, incluidos objetivos psicomotores. No obstante, el diagnóstico cognitivo de destrezas psicomotoras es de gran complejidad, ya que para conseguir representarlas es necesario identificar y capturar datos adecuados, lo cual implica una gran dificultad (Gopher, 1993).
- ✓ Combinar técnicas distintas para el diagnóstico de una tarea es una aproximación totalmente ajustada a las necesidades de los SI<sup>2</sup>A<sup>2</sup>, especialmente de los que conceden gran libertad de acción al estudiante. En dichos casos, la probabilidad de una acción inesperada del alumno aumenta, con lo que el diagnóstico se complica. Combinando varias técnicas se ofrece la posibilidad de llegar a resultados de diagnóstico adecuados cuando una sola técnica no podría hacerlo.
- ✗ La representación de patrones de solución en el modelo de tareas implica conocer todos los pasos que realizará el alumno del principio al fin de la tarea. En entornos autónomos, en los que la actividad del alumno puede ser una respuesta a la interacción con un elemento autónomo, la solución no se puede predefinir. Por tanto, el planteamiento de los patrones de solución no se ajusta a todos los tipos de Sistema Interactivo que se abordan en este trabajo.
- ✗ El modelo de tareas de DETECTive no considera la actividad continua. En este sentido no se diferencia de las aproximaciones de otros SI<sup>2</sup>A<sup>2</sup> para detectar errores. Sin embargo, como ya se ha comentado en secciones anteriores, la representación de la actividad continua es de gran importancia en algunos dominios.
- ✗ La estrategia de secuenciación de técnicas de diagnóstico es estática. Sin embargo, en dominios altamente dinámicos en los que se podrían llevar a cabo tareas concurrentemente, podría ser necesario el uso de estrategias distintas.
- ✗ Las técnicas de diagnóstico empleadas por DETECTive, en general, son adecuadas para cualquier dominio procedimental. Sin embargo, podría ser interesante considerar formas de facilitar la escalabilidad del sistema. Por ejemplo, para incorporar nuevos tipos de desviaciones. En general, sería muy positivo contar con mecanismos que ayuden a

controlar las particularidades de cada dominio que no se ajusten a los procesos generales.

Concluyendo, los principios generales que rigen el diseño de DETECTive encajan perfectamente con los requisitos de los SI<sup>2</sup>A<sup>2</sup>. Asimismo, la forma de abordar el diagnóstico en dominios procedimentales constituye un punto de referencia que no se puede ignorar. Sin embargo, sus capacidades no dan respuesta a toda la problemática expuesta en este capítulo. La Metodología tiene que contemplar una mayor flexibilidad en la elección de la estrategia de diagnóstico, posibilitando la incorporación de más técnicas y permitiendo la representación de acciones continuas. Por estas razones, la Metodología establecerá el diagnóstico de tareas como nexo con los Sistemas de Ayuda al Aprendizaje siguiendo los principios de diseño de DETECTive. En cambio, dadas las carencias identificadas en DETECTive, la Metodología planteará el diagnóstico dando prioridad en su diseño a un nuevo propósito estratégico: la **escalabilidad**. Del componente de diagnóstico. El objetivo es aprovechar las capacidades de diagnóstico de DETECTive en los dominios en los que pueda diagnosticar, y mantener la puerta abierta a integrar nuevos elementos para cubrir sus carencias en los dominios en los que no resulte suficiente.

## 2.4 RESUMEN DE LA PROBLEMÁTICA PLANTEADA

Tras analizar por separado la problemática inherente a los Sistemas Interactivos y a los Sistemas de Ayuda al Aprendizaje relacionada con la construcción de SI<sup>2</sup>A<sup>2</sup>, la integración de ambos tipos de sistemas queda acotada. El objetivo general de la integración es capturar datos del Sistema Interactivo necesarios para representar la actividad del estudiante posibilitando su diagnóstico. Conseguir este objetivo implica establecer un proceso de **interpretación** que, a partir de los datos recogidos del Sistema Interactivo, identifique si el alumno ha realizado alguna acción de la que el componente de diagnóstico deba ser informado.

El diseño del proceso de interpretación tiene que considerar la variedad de datos de entrada que pueden provenir del Sistema Interactivo. En el caso más simple la recepción de un evento en un único instante podría ser suficiente para identificar una acción. Sin embargo, cuando se ejecutan

acciones continuas se pueden recibir flujos continuos de datos que, independientemente, pueden no tener significado, pero tal vez sí lo tengan analizándolos a lo largo del tiempo. Es posible, incluso, que ese flujo de datos contenga ruido o que carezca de significado si no viene acompañado de otros eventos o flujos de datos.

La interpretación se puede complicar más si tenemos en cuenta el realismo del Sistema Interactivo. Cabe la posibilidad de que los datos recibidos, aun siendo suficientes para simular una acción, estén muy lejos de representar la semántica que un observador captaría fácilmente en la interfaz y que coincide con la establecida en el modelo de tareas (Figura 2.3). Si además existe incertidumbre en los datos, el problema no sólo radica en extraer la semántica sino también en determinar en cada instante si es posible extraerla o si hay que esperar a la recepción de más datos.

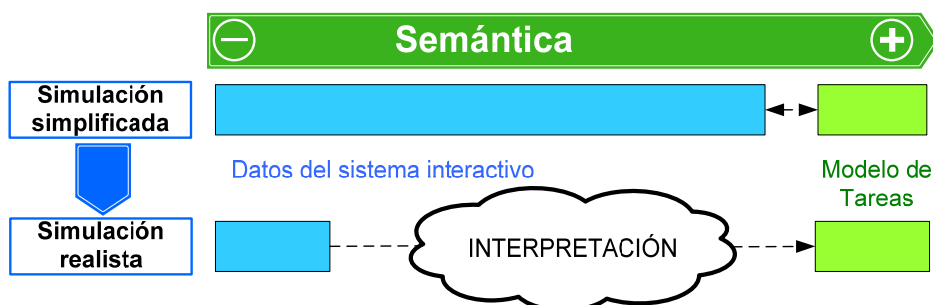


Figura 2.3: Relación entre datos del Sistema Interactivo y el modelo de tareas

Bajo este enfoque, el proceso de interpretación consiste en la fusión en tiempo real de un conjunto heterogéneo de datos y su análisis a lo largo del tiempo para extraer el conocimiento procedimental necesario para el diagnóstico. Obtener esta funcionalidad implica disponer de un formato de representación de conocimiento para definir las acciones del modelo de tareas a partir de sus componentes básicas, es decir, a partir de los datos disponibles en el Sistema Interactivo. Además, debe permitir integrar el factor tiempo tanto en las componentes básicas como en las relaciones entre ellas.

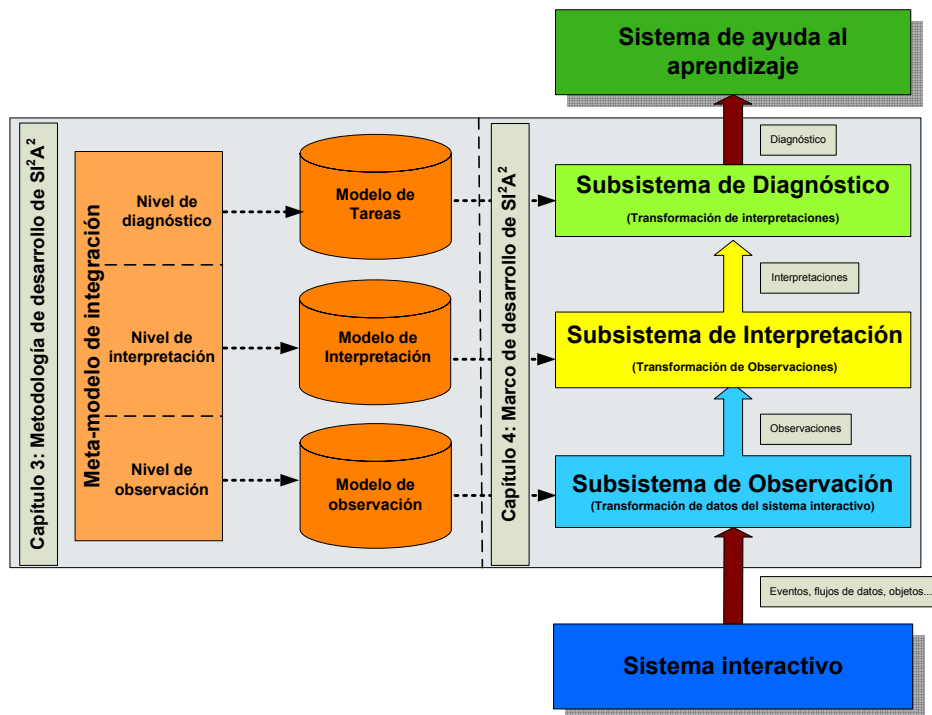
El proceso de definición puede llegar a ser muy complejo. Hay que tener en cuenta que la interpretación busca la conexión entre los términos en lenguaje natural que representan el conocimiento del instructor y un conjunto de datos a priori indeterminados suministrados por el Sistema

Interactivo. Definir qué representa un término en lenguaje natural no siempre es obvio, y encontrar todas las componentes que lo definen puede ser complicado. Por este motivo, el formato de representación debe facilitar un proceso de definición incremental que permita definir las acciones en varias iteraciones hasta encontrar la representación adecuada (Carbonell y Hood, 1986; Fisher, 1987).

## **2.5 PROPUESTA GENERAL PARA EL PROCESO DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup>**

Vista la problemática presentada en este capítulo y las experiencias de desarrollo de otros SI<sup>2</sup>A<sup>2</sup> en el de antecedentes, en los siguientes capítulos se propone una Metodología y un Marco de Desarrollo de propósito general para facilitar la construcción de SI<sup>2</sup>A<sup>2</sup> (Figura 2.4). El enfoque propuesto se inspira en la actividad de un instructor mientras trabaja con sus alumnos. El instructor observa lo que hacen los aprendices, interpreta sus observaciones, y, una vez identificada la actividad en curso, determina su corrección o adecuación. A partir de este punto, su estrategia educativa establecerá qué hacer con esa información: ofrecer asistencia, una demostración, no hacer nada... En definitiva, lo que considere más conveniente para lograr que los alumnos transfieran a situaciones reales las destrezas adquiridas durante su formación.

Esta aproximación se traduce en un Marco de Desarrollo formado por tres subsistemas que representan explícitamente el mismo proceso (Figura 2.4 parte derecha). Cada subsistema recibe en tiempo real los datos generados por el anterior, los analiza y los transforma en una nueva representación hasta conseguir la información que necesita el Sistema de Ayuda al Aprendizaje.

Figura 2.4: Enfoque del desarrollo de SI<sup>2</sup>A<sup>2</sup>

La base de este enfoque es la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> (Figura 2.4 parte izquierda). Consiste en un metamodelo de integración que especifica un conjunto de elementos que abstraen la información que necesita cada subsistema. Estos elementos describen cómo son los modelos que hay que generar para cada subsistema al construir un SI<sup>2</sup>A<sup>2</sup>, es decir, indican:

- cómo describir lo que hay que observar en el Sistema Interactivo;
- cómo describir lo que hay que interpretar y cómo se interpreta;
- cómo describir las tareas a diagnosticar.

Por tanto, siguiendo esta propuesta, la parte fundamental del proceso de desarrollo de un SI<sup>2</sup>A<sup>2</sup> consiste en particularizar para el dominio concreto los elementos de cada nivel del metamodelo de integración. Dicha particularización forma los Modelos de Observación, Interpretación y Tareas que empleará el Marco de Desarrollo para conseguir automáticamente la comunicación entre el Sistema Interactivo y el Sistema

de Ayuda al Aprendizaje. La comunicación con el Sistema Interactivo se modelará utilizando los elementos del Nivel de Observación mientras que, el Nivel de Diagnóstico encapsulará la relación con el Modelo de Tareas del Sistema de Aprendizaje. En resumen, la Metodología está formada por la descripción del metamodelo de integración y se completa con el proceso a seguir para utilizarlo. En el capítulo siguiente se presenta el contenido de la Metodología, mientras que la descripción del Marco de Desarrollo se expone en el capítulo 4.

## *CAPÍTULO 3*

# ***METODOLOGÍA DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup>***

---

En este capítulo se presenta una Metodología de propósito general para el desarrollo de Sistemas Interactivos Inteligentes de Ayuda al Aprendizaje de tareas procedimentales. Tal y como se ha señalado anteriormente, la Metodología está formada por un metamodelo de integración que contiene los elementos que hay que particularizar para construir un SI<sup>2</sup>A<sup>2</sup> en un dominio concreto y el proceso para llevarlo a cabo.

El capítulo comienza delimitando el alcance de la Metodología seguido de una descripción general de la misma para aportar una visión de conjunto. Posteriormente se presentan en varias secciones los tres niveles del metamodelo de integración y se continúa con la descripción del procedimiento de uso del mismo para construir SI<sup>2</sup>A<sup>2</sup> de forma sistematizada. El capítulo finaliza con un apartado de conclusiones en el que se expone cómo la Metodología resuelve los problemas expuestos en el capítulo anterior.

### **3.1 ALCANCE DE LA METODOLOGÍA**

En última instancia, el objetivo de un SI<sup>2</sup>A<sup>2</sup> es conseguir que el alumno transfiera con éxito las habilidades adquiridas durante el entrenamiento asistido por computador al entorno real, lo que depende en buena medida de un correcto diseño educativo. Para ello, se deben diseñar tareas que



permitan observar la destreza del alumno sobre los objetivos de aprendizaje definidos (Bauer, 2002; Shute, 2004), lo que puede llegar a ser muy complicado en algunos dominios. Por ejemplo, en el dominio de la cirugía no existe un acuerdo sobre qué medidas objetivas son adecuadas para valorar la destreza de los alumnos (Rosenthal et al., 2006). En este sentido, la Metodología no puede garantizar que los SI<sup>2</sup>A<sup>2</sup> construidos vayan a ser efectivos para el aprendizaje independientemente del dominio. Sin embargo, es muy importante conseguir que el proceso de construcción de nuevos SI<sup>2</sup>A<sup>2</sup> sea menos costoso. Así, se incentivan los estudios que buscan dichas medidas objetivas en dominios como la cirugía para lograr finalmente la efectividad (Schijven y Jakimowicz, 2003).

Siguiendo estas consideraciones, la Metodología se centra en sistematizar la construcción de SI<sup>2</sup>A<sup>2</sup> orientados a la tarea, que puedan capturar, interpretar y diagnosticar la actividad del alumno para extraer conclusiones sobre sus destrezas. En algunos casos, la experiencia docente permitirá construir un SI<sup>2</sup>A<sup>2</sup> con criterios formativos ya validados en el dominio de aprendizaje, mientras que en otros casos servirá como herramienta para diseñarlos y validarlos. De cualquier forma, el objetivo de la Metodología implica tener en cuenta distintos tipos de dominios con toda la problemática expuesta en el capítulo anterior.

El alcance de la Metodología se centra en dos líneas principales. En la primera, consideramos el desarrollo de SI<sup>2</sup>A<sup>2</sup> como la integración de cualquier Sistema Interactivo diseñado para practicar tareas procedimentales y un Sistema de Ayuda al Aprendizaje sin concretar. Siguiendo este planteamiento, la Metodología debe limitarse a representar la comunicación entre dos Sistemas Abstractos, es decir, se han abstraído las características comunes de los Sistemas Interactivos y los requerimientos de los Sistemas de Ayuda al Aprendizaje y se ha modelado la comunicación entre ambos (Figura 3.1). Mediante esta abstracción se evitan restricciones sobre las arquitecturas internas de los sistemas a integrar para no limitar el abanico de SI<sup>2</sup>A<sup>2</sup> abarcados por la Metodología.

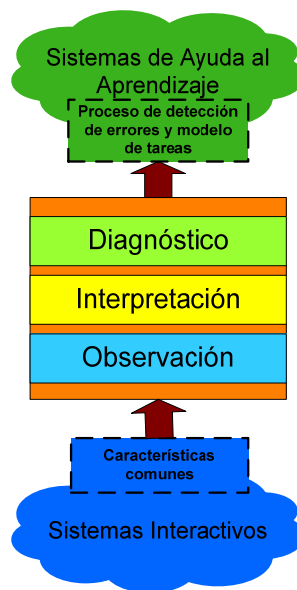


Figura 3.1: Abstracción de sistemas a integrar y modelado de la comunicación

La segunda línea resulta de la evolución de la primera. Al abstraer las características de los Sistemas Interactivos, hay que evitar simplificarlas en exceso para no perder capacidad de obtener información desde distintos tipos de Sistemas Interactivos. Resumiendo, en la Metodología se contemplan:

- Sistemas de actividad continua y/o basada en eventos.
- Entornos realistas o simplificados.
- Entornos autónomos y reactivos.

De todas formas, hay que señalar que lo habitual es que los SIA<sup>2</sup> no sean estrictamente de un tipo u otro, sino que presenten combinaciones de estas características. En cualquier caso, la meta es conseguir que la Metodología mantenga una visión unificada del proceso de integración (Figura 3.2).

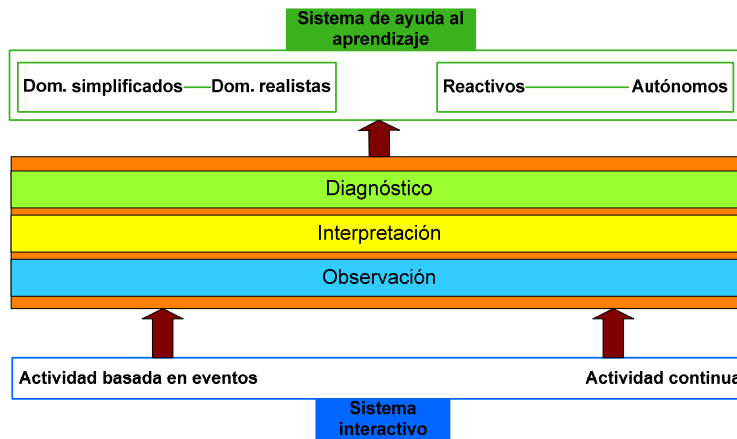


Figura 3.2: Visión unificada del proceso de integración

## 3.2 DESCRIPCIÓN GENERAL

La Metodología representa el proceso de construcción de SI<sup>2</sup>A<sup>2</sup> mediante un metamodelo que describe los elementos necesarios para integrar un Sistema Interactivo con un Sistema de Ayuda al Aprendizaje. El metamodelo de integración se divide en tres niveles de abstracción (Figura 3.3). Cada uno de ellos describe de forma genérica un conjunto de elementos que hay que particularizar para cada SI<sup>2</sup>A<sup>2</sup> que se quiera construir. Como resultado de esta particularización se obtienen los Modelos de Observación, Interpretación y Tareas. Estos modelos son la especificación formal que emplea el Marco de Desarrollo para transformar automáticamente los datos capturados del Sistema Interactivo en información válida para el proceso de detección de errores del Sistema de Ayuda al Aprendizaje. Los procesos que hacen uso de los modelos para establecer la comunicación entre los sistemas integrados se describen en el capítulo 4.

Cada nivel del metamodelo de integración está destinado a representar una fase del proceso de comunicación entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje:

- Nivel de Observación: Contiene los elementos necesarios para especificar los hechos observables en el Sistema Interactivo que sean significativos desde el punto de vista de la interpretación de la actividad del estudiante. Dicha especificación contiene la identificación de los

datos que el Marco de Desarrollo tendrá que recoger del Sistema Interactivo para poder extraer las observaciones: eventos de la interfaz, eventos de la simulación y cambios en los atributos de los objetos del entorno (posiciones, velocidades, materiales...).

- Nivel de Interpretación: Los elementos del Nivel de Interpretación sirven para representar los contextos y actividades de interés para el Nivel de Diagnóstico que se puedan producir durante el desarrollo de las tareas. La representación de este nivel hace de eje entre los extremos del SI<sup>2</sup>A<sup>2</sup>: por una parte conoce las observaciones que se pueden obtener del nivel anterior, y por otra, define cómo generar la información que el Nivel de Diagnóstico necesita. Para ello representa la actividad del alumno mediante patrones formados por relaciones entre observaciones definidas en el nivel anterior.
- Nivel de Diagnóstico: El nivel de diagnóstico representa la información necesaria para reconocer, localizar y analizar los errores cometidos por el estudiante mientras lleva a cabo una tarea. Este nivel encapsula el Modelo de Tareas empleado por el proceso de detección de errores para diagnosticar la actividad del estudiante en el Sistema Interactivo. Su especificación parte de la base de que el modelo de tareas a integrar sigue los principios de diseño de DETECTive, que contiene la especificación de: tareas, soluciones correctas e incorrectas, errores típicos, resultados del diagnóstico...

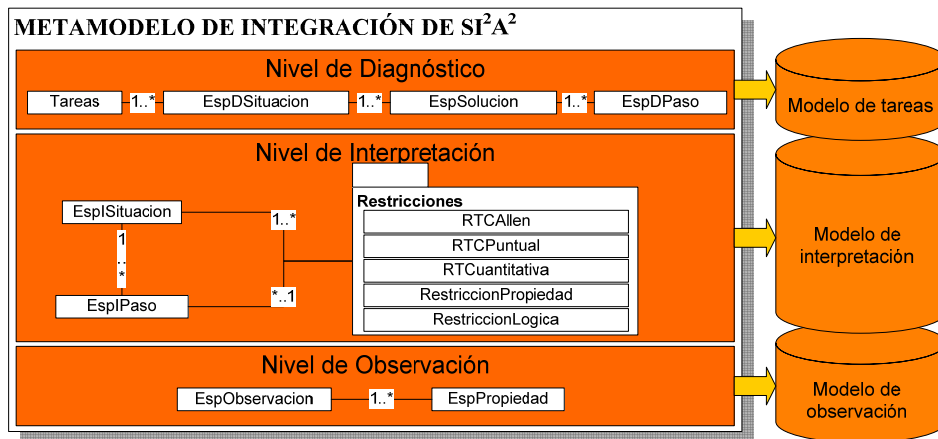


Figura 3.3: Descripción del metamodelo de integración

En las secciones siguientes se presenta cada uno de los niveles del metamodelo de integración. Al inicio de cada sección se describen los objetivos particulares del nivel y, después, se presentan los elementos que lo componen.

### **3.3 NIVEL DE OBSERVACIÓN**

En este apartado se describe el Nivel de Observación del metamodelo de integración de SI<sup>2</sup>A<sup>2</sup>. Se incluyen los objetivos del nivel y la definición de los elementos genéricos que hay que particularizar para generar el Modelo de Observación.

#### **3.3.1 *Objetivos del Nivel de Observación***

El objetivo esencial del Nivel de Observación es proporcionar los elementos necesarios para describir cómo observar en un Sistema Interactivo los hechos que nos interesen. Estos hechos servirán de primitivas para describir la actividad del estudiante en los demás niveles. Por ejemplo, si se quisiera describir en el Nivel de Interpretación que un conductor va a aparcar su vehículo, habría que observar un espacio vacío al que el vehículo se está acercando. Es necesario destacar la importancia del Nivel de Observación. Las carencias en la observación equivaldrían a un instructor despistado que tendría que tomar decisiones educativas sin información o malinterpretándola. Por esta razón, la característica principal que se exige al Nivel de Observación es que tenga flexibilidad para definir observaciones a partir de datos de diferentes características, independientemente de cómo hayan sido generados. Además, hay que tener en cuenta que no se pueden predeterminar las características internas del Sistema Interactivo para que la Metodología de desarrollo sea genérica.

También hay que considerar que la interacción del alumno con el SI<sup>2</sup>A<sup>2</sup> puede realizarse mediante dispositivos de todo tipo: dispositivos de videojuegos, hápticos, sistemas de captura de gestos o movimiento, etc. Se distinguen dos grupos principales de dispositivos en función de los datos que proporcionan: los discretos y los continuos, que generan estados discretos y flujos continuos de datos respectivamente (Sección 1.1.1 de esta memoria) (LaViola, 2001). Esta consideración es relevante para el Nivel de

Observación, ya que para observar la actividad del alumno tiene que poder manejar ambos tipos de datos.

Más aún, los datos generados por el componente encargado de la simulación en el Sistema Interactivo pueden consistir tanto en eventos como en flujos o en objetos complejos que contengan todo tipo de información. Estos datos pueden estar relacionados con la actividad del alumno, por lo que también deben considerarse en la representación de observaciones.

### **3.3.2 Elementos del Nivel de Observación**

El Nivel de Observación especifica la unidad primitiva a partir de la cual se construye el metamodelo de integración: la observación. “Una observación describe un hecho que puede tener lugar en el Sistema Interactivo durante un intervalo de tiempo”. En esta definición destacan dos aspectos principales:

- Una observación describe “un hecho”. Definiéndola así se indica que una observación es considerada como un elemento no subjetivo en el contexto del metamodelo. En el resto de niveles se entenderá que su descripción es independiente del observador y no estará sujeta a interpretación. De esta manera, cuando se detecte una observación durante una sesión de entrenamiento, se dará por supuesto que es fiable. Definiendo así las observaciones se reduce la incertidumbre que tienen que tratar los demás niveles, ya que se dará por sentado que las observaciones que manejan se corresponden con la realidad.
- Una observación “puede tener lugar durante un intervalo de tiempo”. Esta parte de la definición contempla que se puedan extraer observaciones del Sistema Interactivo tanto instantáneas como continuas.

El elemento `EspObservacion` contiene los atributos que especifican qué tiene que hacer el Marco de Desarrollo para extraer la observación a partir de los datos de entrada que obtenga del Sistema Interactivo. Dichos atributos se describen a continuación (Figura 3.4):

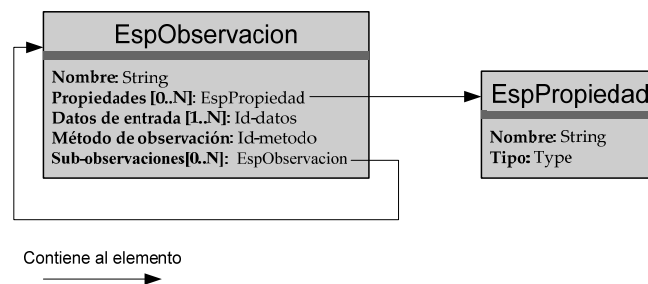


Figura 3.4: Especificación de una Observación y una Propiedad

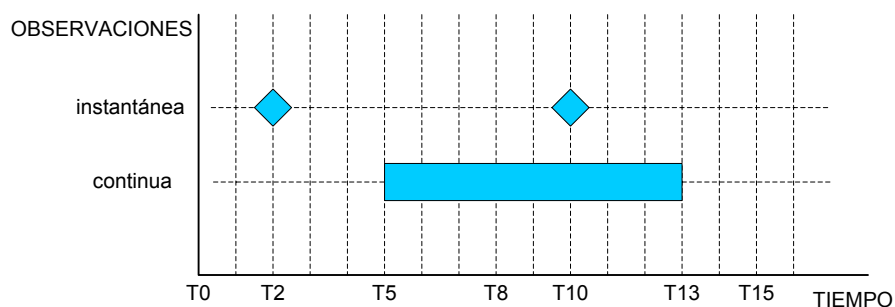
- *Nombre*: Texto que identifica de forma unívoca al hecho que se quiere observar.
- *Propiedades*: Lista de especificaciones de propiedades (EspPropiedad) asociadas a la observación. Cada hecho observado puede incluir opcionalmente una serie de propiedades que describan o cuantifiquen la observación. Por ejemplo, desde un vehículo se ve una señal de Stop, e incluye entre sus propiedades la distancia a la señal.
- *Datos de entrada*: Lista de identificadores de los datos del Sistema Interactivo a partir de los cuales el Marco de Desarrollo deberá extraer la observación. Puede identificar un evento, un flujo de datos, un objeto estructurado...
- *Método de observación*: Identificador del método que a partir de los *datos de entrada* es capaz de determinar si la observación está teniendo lugar en un instante determinado y de calcular el valor de sus propiedades. Siguiendo con el ejemplo anterior, el *método de observación* recogería como datos de entrada la posición del vehículo y la identificación y las posiciones de las señales de tráfico que hay en la vía en la que está. Con estos datos, calcularía si existe alguna señal de Stop dentro de un radio establecido en torno a la posición del vehículo.
- *Sub-observaciones*: Lista de EspObservacion que únicamente es posible extraer si la observación contenedora está teniendo lugar. Por ejemplo, si no se observa que un vehículo se encuentra en una vía de más de un carril, no tiene sentido intentar observar si hay algún vehículo a su alcance en algún carril contiguo. En este ejemplo, la EspObservacion "Vía de varios carriles" incluiría la *Sub-observación* "Vehículo en alcance en carril contiguo". Esta jerarquía actúa de pauta para disminuir el coste

computacional de obtención de observaciones, ya que el Marco de Desarrollo no ejecuta los Métodos de Observación de las Sub-Observaciones cuya EspObservacion contenedora no está teniendo lugar.

Cada propiedad identificada dentro de una EspObservacion se especifica mediante el elemento EspPropiedad que consta de dos atributos:

- *Nombre* de la propiedad.
- *Tipo*: Especifica el tipo de dato (número, texto, objeto...) y sirve para restringir las operaciones entre propiedades.

Tal y como se detallará en la descripción del Marco de Desarrollo (Capítulo 4), hay que señalar que el *Método de observación* de una observación continua se ejecuta en cada ciclo de simulación, y devuelve siempre una respuesta. En el cronograma de la Figura 3.5, se representan con líneas verticales discontinuas todos los instantes (ciclos de simulación) en los que se ejecutan los *Métodos de observación* de dos EspObservaciones: una instantánea y otra continua. Bajo el cronograma se indica la respuesta de los *Métodos de observación* en varios instantes.



T0: Comienzo del proceso de observación

T2: Instante de detección de observación instantánea

T5: Instante de detección de observación continua (1ª vez)

T8: Instante de detección de observación continua (4ª vez)

T10: Instante de detección de observación continua (6ª vez) y de una observación instantánea

T13: Instante en el que se deja de detectar la observación continua

T15: Instante sin observaciones

Figura 3.5: Detección de observaciones



Durante el intervalo de tiempo que se produce la observación continua, ésta puede variar sus características. Por ejemplo, se podría observar que un objeto se aproxima a otro, pero en cada instante la distancia entre ambos irá disminuyendo. Esta variación de las características de la observación a lo largo del tiempo quedará registrada mediante las propiedades de la observación, tal y como se detallará en la descripción del Marco de Desarrollo.

### 3.4 NIVEL DE INTERPRETACIÓN

El Nivel de Interpretación describe de forma genérica cómo reconocer la actividad del alumno en el Sistema Interactivo. Concretamente, los elementos del nivel se utilizarán para describir la actividad que vaya a ser diagnosticada, formando el Modelo de Interpretación.

Según el diccionario de la Real Academia Española (RAE), interpretar es “Concebir, ordenar o expresar de un modo personal la realidad”. En el caso de un SI<sup>2</sup>A<sup>2</sup>, “la realidad” es el Entorno Virtual o real y la interacción del alumno en el mismo a través de la interfaz. Por tanto, el reto de la interpretación consiste en concebir, ordenar o expresar digitalmente lo que sucede en el Entorno Virtual. Es más, hay que saber lo que sucede con la precisión suficiente como para que el diagnóstico pueda determinar su incorrección. Para ello, es fundamental que parte de la interpretación esté dedicada a conocer el contexto en el que actúa el alumno. La RAE define el contexto como “entorno físico o de situación, ya sea político, histórico, cultural o de cualquier otra índole, en el cual se considera un hecho”. Por tanto, para que los resultados de interpretación sean completos hay que reconocer en qué situación o contexto se están produciendo las acciones del alumno.

Es de destacar que la interpretación se hace “de un modo personal”, con lo que ésta no puede considerarse única ni completa, es decir, podrían existir varias interpretaciones válidas que representasen distintas circunstancias de la misma realidad. Más aún, irremediablemente no serían todas las posibles ni podríamos concluir que una de ellas es indiscutiblemente la óptima. Exactamente lo mismo sucede al interpretar los sucesos en un Entorno Virtual. La interpretación será válida o suficientemente completa si sirve para obtener resultados de diagnóstico

ajustados al criterio del instructor. Por esta razón, el Nivel de Interpretación ha sido diseñado buscando flexibilidad para representar la actividad con el criterio que se estime oportuno en cada caso.

El contenido del Nivel de Interpretación se desarrolla siguiendo las consideraciones anteriores. Para exponerlo, se definen en primer lugar sus objetivos particulares. Posteriormente se estudian en detalle los diferentes tipos de elementos Restricción que son fundamentales en este nivel y, finalmente, se presenta el resto de los elementos que componen el nivel con los que hay que crear el Modelo de Interpretación.

### **3.4.1 Objetivos del Nivel de Interpretación**

Son tres los objetivos prioritarios que debe alcanzar el Nivel de Interpretación. El primero de ellos es la **flexibilidad** para representar la actividad en diferentes dominios de forma genérica. Este objetivo implica minimizar, en la medida de lo posible, las restricciones sobre lo que es interpretable y lo que no. Se aborda prioritariamente la representación conjunta de actividad continua y basada en eventos.

El segundo objetivo es conseguir que el Nivel de Interpretación represente **información útil y suficiente para el diagnóstico**. Este objetivo descarta cualquier forma de interpretación que conlleve pérdidas de información para el diagnóstico aunque sea capaz de realizar interpretaciones correctas.

El tercer objetivo planteado es conseguir que el modelo generado en este nivel sea apto para la **interpretar observaciones en tiempo interactivo**<sup>12</sup>. Este objetivo plantea dos retos:

- En primer lugar, la interpretación no tiene que esperar a que una acción haya finalizado para poder determinar qué está sucediendo. Este reto añade complejidad a la interpretación. Por ejemplo, ¿en qué instante un profesor de autoescuela es consciente de que su alumno está intentando adelantar otro vehículo? Por supuesto, no necesita esperar a que termine de adelantar para darse cuenta de que lo está haciendo, pero tampoco lo detecta inmediatamente. De hecho, ¿Cuál es el instante en el que

---

<sup>12</sup> Con la rapidez suficiente para que el usuario que está interactuando con el sistema no perciba retardos que dificulten la interacción

consideramos que comienza el adelantamiento? ¿Y por qué no un milisegundo antes o uno después? El criterio es subjetivo, pero probablemente antes de tener la certeza de que está adelantando, únicamente se tenían indicios. Por tanto, el reto está en obtener interpretaciones sabiendo que la acción ha comenzado con anterioridad, sin perder los datos pasados y sin tener que esperar demasiado tiempo hasta que la interpretación sea fiable.

- En segundo lugar, la interpretación ha de ser suficientemente rápida para funcionar en entornos altamente dinámicos, en los que los datos cambian constantemente. Siguiendo el ejemplo anterior, un adelantamiento practicado en un simulador implica analizar flujos de datos que contienen, por ejemplo, la posición de los vehículos y sus velocidades, y que cambian en cada ciclo de simulación. El análisis de estos flujos tiene que ser suficientemente rápido para evitar que se acumule la información a procesar. Por tanto, al definir los elementos de este nivel hay que evitar que su estructura ralentice el proceso de interpretación.

El Nivel de Interpretación describe cómo se especifican los elementos necesarios para diagnosticar la actividad del alumno a partir de las observaciones definidas en el nivel anterior. Para ello se definen dos elementos básicos: la especificación del paso (EspIPaso) y de la situación (EspISituacion). El primero especifica cómo identificar cada una de las acciones del alumno que se quieran diagnosticar, mientras que con el segundo se describen los posibles contextos en los que se pueden realizar los pasos.

La base de la descripción de estos elementos es la Restricción. Las restricciones definen relaciones entre observaciones, pasos y situaciones. En los siguientes apartados se presentan los tipos de Restricción utilizados: Restricciones Temporales Cualitativas, Cuantitativas, Sobre Propiedades y Lógicas. Posteriormente se describen los elementos principales del Nivel de Interpretación: EspIPaso y EspISituación, y se explicará cómo hacen uso de las Restricciones.

### **3.4.2 Elementos Restricción**

La Restricción permite crear relaciones complejas entre los elementos EspObservacion, EspIPaso y EspISituacion que servirán como herramienta

para describir la actividad del alumno. Tanto las observaciones, como los pasos y las situaciones tienen en común que pueden darse durante un intervalo de tiempo. Por ello, en el contexto de las Restricciones, se utiliza el término intervalo para referirse indistintamente a cualquiera de ellos (Figura 3.6). En el Nivel de Interpretación se combinan distintos tipos de Restricciones para lograr una gran flexibilidad de representación con la que adecuarse a dominios de aprendizaje diferentes.

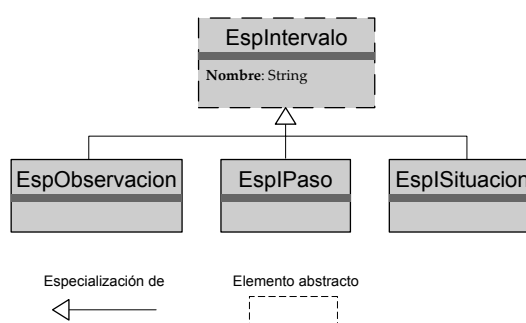


Figura 3.6: Especificación de intervalo

A continuación se presentan las Restricciones Temporales Cualitativas, que fijan el planteamiento de descripción de las acciones continuas. Seguidamente se describen las Restricciones Temporales Cuantitativas que amplían la capacidad de representación temporal de las anteriores. Posteriormente se plantean las Restricciones sobre Propiedades y finalmente se describe la combinación de las anteriores mediante Restricciones Lógicas. En el Anexo B se discuten algunos antecedentes en los que las restricciones temporales han sido relevantes para la interpretación o representación de actividad, y se justifica su uso comparado con otras alternativas.

### 3.4.2.1 Restricciones Temporales Cualitativas

Las restricciones temporales cualitativas sirven para definir relaciones temporales entre intervalos sin necesidad de cuantificar numéricamente los instantes de inicio y fin, ni la duración del intervalo. Se han estudiado dos opciones posibles para representar este tipo de restricciones: el álgebra de intervalos de Allen y el álgebra de puntos. Finalmente, ambas opciones han sido integradas en el Nivel de Interpretación mediante los elementos

### Restricción Temporal Cualitativa Allen (RTCAllen) y Restricción Temporal Cualitativa Puntual (RTCPuntual).

El álgebra de intervalos de Allen (Allen, 1983; Allen, 1984) propone un modelo formal de representación de conocimiento temporal que emplea el intervalo de tiempo como primitiva. Define las relaciones posibles entre intervalos, las operaciones que se pueden realizar con ellos y algunos métodos de razonamiento para extraer conocimiento temporal. El álgebra se basa en 13 relaciones básicas entre intervalos:  $I = \{p, m, o, d, s, f, e, pi, mi, oi, c, si, fi\}$  (Figura 3.7) que se caracterizan por ser: *distintas*, ya que dos intervalos no se pueden relacionar por más de una de las relaciones; *exhaustivas*, porque cualquier par de intervalos se puede describir con alguna de las relaciones; y *cualitativas*, porque para describir las relaciones no es necesario cuantificar numéricamente el inicio, fin ni duración de los intervalos.









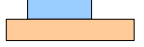
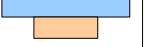
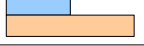

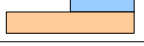
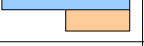

Relación	Simb			Simb	Relación inversa
Precede <i>Precedes</i>	<i>p</i>			<i>pi</i>	Es precedido por <i>Is preceded by</i>
Seguido de <i>Meets</i>	<i>m</i>			<i>mi</i>	Sigue a <i>Is met by</i>
Se solapa con <i>Overlaps</i>	<i>o</i>			<i>oi</i>	Es solapado por <i>Is overlapped by</i>
Durante <i>During</i>	<i>d</i>			<i>c</i>	Contiene <i>Contains</i>
Comienza <i>Starts</i>	<i>s</i>			<i>si</i>	Es comenzado por <i>Is started by</i>
Finaliza <i>Finishes</i>	<i>f</i>			<i>fi</i>	Es finalizado por <i>Is finished by</i>
Igual <i>Equals</i>	<i>e</i>				

Figura 3.7: Las 13 relaciones básicas del álgebra de Allen

Representamos la relación entre dos intervalos  $X$  e  $Y$  con la siguiente notación:  $X[I_1, \dots, I_n]Y$ , donde  $I_1$  e  $I_n$  son relaciones básicas entre intervalos. Por ejemplo,  $X[p]Y$  significa que el intervalo  $X$  precede al  $Y$ . Cuando no se sabe exactamente cual es la relación que va a existir entre dos intervalos, se definen relaciones disyuntivas. Por ejemplo,  $X[p,o]Y$  significa que el intervalo  $X$  precede o se solapa con el  $Y$ .

El Álgebra de Puntos utiliza instantes temporales como primitiva (Vilain y Kautz, 1986; Vilain et al., 1990). En este caso, los instantes

temporales se pueden relacionar mediante tres operadores, de tal forma que siendo  $p_i$  y  $p_j$  dos instantes temporales:

- $p_i < p_j$ : significa que el instante  $p_i$  es anterior al  $p_j$ .
- $p_i = p_j$ : significa que el instante  $p_i$  y el  $p_j$  son simultáneos.
- $p_i > p_j$ : significa que el instante  $p_i$  es posterior al  $p_j$ .

Vilain y Kautz demuestran que algunos problemas típicos del razonamiento temporal son intratables computacionalmente con el álgebra de intervalos, al contrario que con la de puntos. Sin embargo, el álgebra de puntos únicamente puede representar 188 de las  $2^{13}$  relaciones disyuntivas que permite el álgebra de intervalos, lo que se traduce en menor expresividad y capacidad de razonamiento (Ladkin y Maddux, 1989; van Beek y Cohen, 1990). En este sentido, el álgebra de intervalos será aconsejable si se pueden evitar los problemas de intratabilidad computacional.

Teniendo en cuenta que las observaciones son las primitivas de las que parte este nivel y éstas tienen lugar durante intervalos de tiempo, el álgebra de Allen encaja mucho mejor que el álgebra de puntos para representar sus relaciones temporales. Además, las observaciones instantáneas se pueden relacionar con las continuas sin ningún problema, ya que se pueden representar como intervalos de duración cero. Sin embargo, el álgebra de puntos puede ser una herramienta más eficaz para representar algunos casos. Por ejemplo, supongamos la relación entre intervalos  $X[p,m,o,c,fi]Y$ , es decir, el intervalo  $X$  precede, está seguido de, solapa, contiene o es finalizado por el  $Y$  (Figura 3.8). Lo interesante de la restricción es que  $X$  comienza antes que  $Y$ , pero da igual cuando terminen. En este caso, tomando los instantes de inicio de los intervalos es posible utilizar la representación mediante puntos de manera equivalente y más simple que la de intervalos:  $X_{inicio} < Y_{inicio}$ .

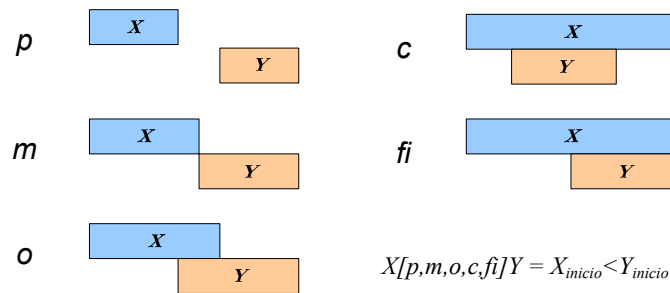


Figura 3.8: Ejemplo de relación álgebra de intervalos — álgebra de puntos

Por estas razones, se ha decidido que el Nivel de Interpretación permita definir restricciones temporales cualitativas basadas tanto en intervalos (Restricción Temporal Cualitativa Allen) como en puntos (Restricción Temporal Cualitativa Puntual). Sin embargo, hay que tener en cuenta que para definir restricciones basadas en puntos, únicamente se pueden emplear los instantes de inicio o fin de los intervalos, pero no puntos intermedios. Por otra parte, existen varias alternativas para combinar ambas representaciones. Algunos autores restringen el álgebra de intervalos al subconjunto transformable en puntos y fusionan ambos en una sola representación de puntos (Kahl et al., 1999; Keretho y Loganantharaj, 1991). Otros, en cambio, definen relaciones específicas entre puntos e intervalos para poder razonar sobre una representación conjunta (Meiri, 1996; Vilain, 1982). Para simplificar, en nuestro caso, hemos optado por evitar las relaciones directas entre intervalos y puntos de forma que una restricción temporal únicamente pueda relacionar o dos intervalos o dos puntos. Se ha preferido esta opción teniendo en cuenta que los puntos se corresponden siempre con el instante de inicio o fin de un intervalo, con lo cual una relación punto-intervalo se podría expresar con varias relaciones punto-punto. De todas formas, no se descarta considerar estas relaciones en extensiones futuras del Nivel de Interpretación.

El elemento Restricción Temporal Cualitativa Allen (RTCAllen) (Figura 3.9) sirve para definir la relación temporal entre dos intervalos y está formada por los siguientes atributos:

- *Intervalo 1*: Instancia de EspObservación, EspPaso o EspSituacion que se relaciona con el *Intervalo 2*.

- *Operadores*: Lista de relaciones básicas de Allen que se pueden dar disyuntivamente entre *Intervalo 1* e *Intervalo 2*.
- *Intervalo 2*: Instancia de EspObservación, EspPaso o EspSituacion relacionado con el *Intervalo 1*.

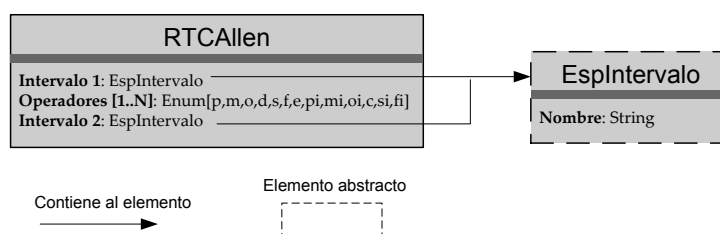


Figura 3.9: Especificación de la Restricción Temporal Cualitativa Allen

El elemento Restricción Temporal Cualitativa Puntual (RTCPuntual) describe la relación cualitativa entre los puntos de inicio o fin de dos intervalos diferentes (Figura 3.10). Para definir las, hay que emplear el elemento Instante Temporal que identifica al punto de inicio o fin de un intervalo. Hay que tener en cuenta que un Instante Temporal es una propiedad de un intervalo. Este detalle es importante para diferenciar un Instante Temporal de un intervalo de duración cero. El Instante Temporal se define mediante los siguientes atributos:

- *Intervalo*: Instancia de EspObservación, EspPaso o EspSituacion del que interesa su instante de inicio o fin.
- *Instante*: Sus valores posibles son "inicio" y "fin".

Una Restricción Temporal Cualitativa Puntual se define mediante los siguientes atributos:

- *Instante 1*: Instante Temporal que se relaciona con el *Instante 2*.
- *Operadores*: Lista de relaciones que se pueden dar entre *Instante 1* e *Instante 2* para considerar que la restricción se cumple. Los valores posibles son: "<" (anterior), "=" (simultáneo), ">" (posterior), "<=" (anterior o simultáneo), ">=" (posterior o simultáneo).
- *Instante 2*: Instante Temporal relacionado con el *Instante 1*.



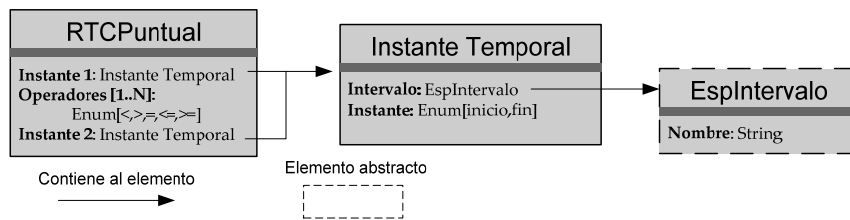


Figura 3.10: Especificación de la Restricción Temporal Cualitativa Puntual

### 3.4.2.2 Restricciones Temporales Cuantitativas

Las Restricciones Temporales Cuantitativas o métricas son aquellas en las que se especifica numéricamente alguna relación temporal entre dos intervalos. Por ejemplo, la diferencia de duración entre dos intervalos, o la cantidad de tiempo transcurrido desde el fin de uno al comienzo de otro, son ejemplos que entran en esta categoría. Estas descripciones hacen referencia a los instantes de inicio y fin de los intervalos, por lo que este tipo de restricciones se suele asociar a representaciones basadas en puntos temporales en lugar de intervalos (Dechter et al., 1991).

En general, una restricción temporal cuantitativa simple sobre dos instantes temporales  $p_i$  y  $p_j$  se expresa con la forma:  $C_{ij} = \{I_1 = [a_1, b_1]\}$ . Los valores  $a_1$  y  $b_1$  forman un intervalo real que delimita la distancia posible entre  $p_i$  y  $p_j$  para que la restricción se cumpla, es decir, se tiene que cumplir:  $a_1 \leq p_j - p_i \leq b_1$ . Además, generalizando esta definición, describimos la restricción temporal cuantitativa disyuntiva, que tendría la forma:  $C_{ij} = \{I_1, \dots, I_k\}$ , tal que  $a_1 \leq p_j - p_i \leq b_1 \vee \dots \vee a_k \leq p_j - p_i \leq b_k$ .

Teniendo en cuenta la definición previa, decidimos incorporar en el Nivel de Interpretación el elemento Restricción Temporal Cuantitativa, con el que se van a crear relaciones entre Instantes Temporales. Con este elemento se podrán definir restricciones tanto simples como disyuntivas. Sin embargo, para componer las restricciones disyuntivas hay que utilizar Restricciones Lógicas, tal y como se expondrá en la sección 3.4.2.4.

Para describir el elemento Restricción Temporal Cuantitativa, se emplean los elementos Duración (Figura 3.11) que especifican el tiempo transcurrido entre dos puntos. El elemento DuraciónInstantes sirve para referirse a la cantidad de tiempo transcurrido entre los atributos *Instante 1* e *Instante 2* que identifican Instantes Temporales de dos intervalos diferentes:

- *Instante 1*: Instante Temporal anterior a *Instante 2*.
- *Instante 2*: Instante Temporal posterior a *Instante 1*.

Para especificar la distancia entre puntos temporales relativos al mismo intervalo se emplea el elemento *DuracionIntervalo*, que muestra únicamente el atributo *Intervalo*:

- *Intervalo*: Instancia de *EspObservación*, *EspPaso* o *EspSituacion* del que se quiere conocer su duración.

Otra posibilidad para definir Restricciones Temporales Cuantitativas es relacionar duraciones de intervalos o entre instantes temporales con duraciones absolutas, por ejemplo, para restringir que entre dos intervalos no transcurran más de 5 segundos. Esta restricción emplea el elemento *DuracionTiempo* que representa una cantidad de tiempo en valor absoluto:

- *Tiempo*: Número entero que representa una cantidad de tiempo. Se consideran las mismas unidades temporales en las que se vaya a calcular la Duración con la que se quiera relacionar.

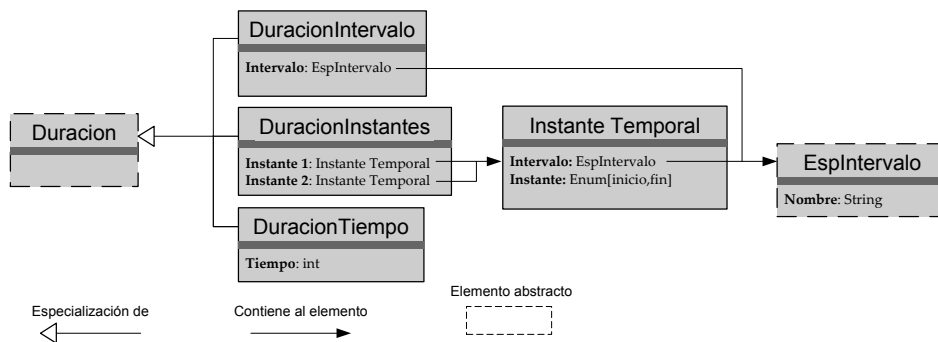


Figura 3.11: Especificación de elementos Duración

Las Restricciones Temporales Cuantitativas se definen mediante los mismos operadores relacionales empleados en el Álgebra de Puntos. En el Álgebra de Puntos se relacionan Instantes Temporales, mientras que estas restricciones relacionan duraciones. De esta forma, el elemento Restricción Temporal Cuantitativa (RTCuantitativa) se compone de los siguientes atributos (Figura 3.12):

- *Duración 1*: Duración de un intervalo o distancia entre dos instantes temporales que se quiere comparar con el atributo *Duración 2*.

- *Operador*: Operador relacional con el que se comparan los atributos *Duración 1* y *Duración 2*. La restricción se evalúa en el orden: *Duración 1* – *Operador* – *Duración 2*. Los valores posibles del operador son: “<” (menor), “=” (igual), “>” (mayor), “<=” (menor o igual), “>=” (mayor o igual). Tanto *Duración 1* como *Duración 2* deben estar calculados en las mismas unidades temporales.
- *Duración 2*: Duración de un intervalo, distancia entre dos instantes temporales o tiempo en valor absoluto que se quiere comparar con el atributo *Duración 1*.

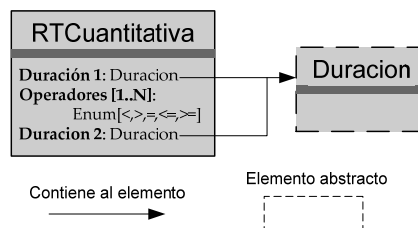


Figura 3.12: Especificación de la Restricción Temporal Cuantitativa

### 3.4.2.3 Restricciones sobre Propiedades

Las Restricciones sobre Propiedades se definen para establecer condiciones sobre los valores de las propiedades de las observaciones durante todo el intervalo de tiempo que duren. Por ejemplo, desde un vehículo se puede observar que hay otro que se le aproxima frontalmente. Si la distancia que los separa es pequeña se interpretaría como una situación de peligro, mientras que una distancia mayor puede que no tenga interés. Para interpretar la situación de peligro se podría definir una Restricción sobre la Propiedad “distancia” de la EspObservacion “aproximación frontal”.

Puesto que estas restricciones se definen sobre Propiedades de Observaciones, es necesario un elemento con el que referirse individualmente a las mismas. Esta es la función del elemento EspPropiedadObservacion que contiene los siguientes atributos (Figura 3.13):

- *Observación*: Contenido del atributo *Nombre* de la EspObservacion a la que se refiere el atributo *Propiedad*.

- *Propiedad*: Contenido del atributo *Nombre* de la *EspPropiedad* contenida en la *EspObservacion* a la se querrá hacer referencia en las restricciones.

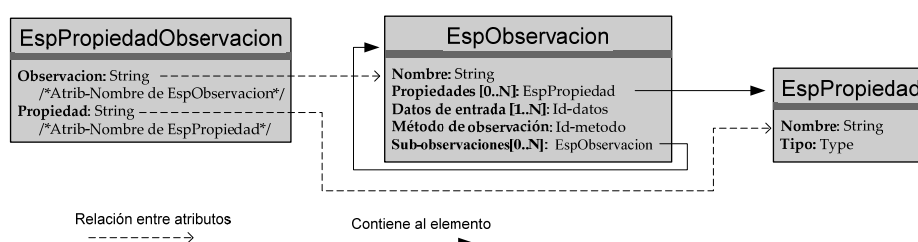


Figura 3.13: Especificación de Propiedad de Observación

Para añadir flexibilidad a la definición de Restricciones sobre Propiedades, se admite el uso de expresiones aritméticas con las que combinar los valores de las propiedades para definir restricciones más complejas. Una Expresión Aritmética se describe mediante los siguientes atributos:

- *Expresión 1*: Expresión Aritmética, Propiedad de Observación o valor de un tipo básico de datos (número, texto) que se relaciona con *Expresión 2* por medio del *Operador*.
- *Operador*: OperadorAritmetico (+,-,\*,÷,%) o identificador de un método externo (OperadorExt) que evalúa la expresión tomando como entrada los resultados de evaluar *Expresión 1* y *Expresión 2*. La expresión se evaluará en el orden: *Expresión 1* – *Operador* – *Expresión 2*. Obviamente, es necesario que el operador empleado sea capaz de tratar los tipos de datos que resulten de evaluar las expresiones.
- *Expresión 2*: Expresión Aritmética, Propiedad de Observación o valor de un tipo básico de datos (número, texto) que se relaciona con *Expresión 1* por medio del *Operador*.

Siguiendo con el ejemplo inicial, si se quisiera definir una Restricción sobre la Propiedad “distancia” de la *EspObservacion* “Aproximación frontal”, podría convenir el uso de expresiones aritméticas. Por ejemplo, si la distancia viene dada en metros y se quiere definir una restricción en centímetros, la restricción puede incluir una *ExpresionAritmetica* para multiplicar la distancia por 100 (Figura 3.14).

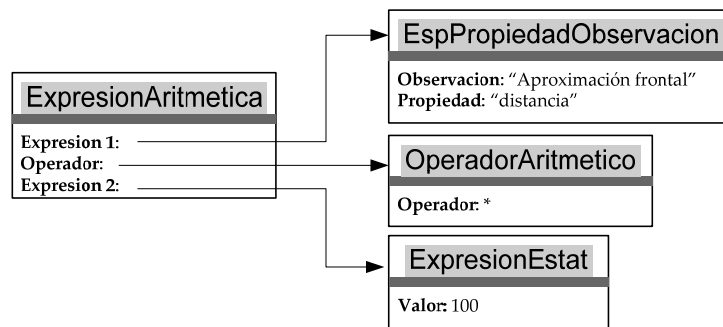


Figura 3.14: Ejemplo de instancias para definir una expresión aritmética

Una Restricción sobre Propiedad, cuando solo incluye dos operandos (RPropRelacional), se describe mediante los siguientes atributos (Figura 3.15):

- *Operando 1*: Expresión sobre la que se quiere aplicar la restricción. Una expresión puede ser una Expresión Aritmética, Propiedad de Observación o valor de un tipo básico de datos (número, texto).
- *Operador*: Operador relacional (<, >, =, <=, >=) que restringe la relación entre los valores de *Operando 1* y *Operando 2*.
- *Operando 2*: Expresión que restringe los valores del *Operando 1*.

Asimismo, es posible definir restricciones n-arias que involucren varias EspPropiedades, para lo cual se hará uso de métodos externos que evaluarán restricciones específicas entre varias Expresiones. El elemento Restricción sobre Propiedad se especializa mostrando los siguientes atributos (RPropExterna):

- *Método*: Identificador del método externo que tomará como entrada el atributo *Expresiones* y devolverá el resultado de evaluar si la restricción que implemente se cumple.
- *Expresiones*: Lista de Expresiones sobre las que se desea verificar la restricción implementada por el *Método*.

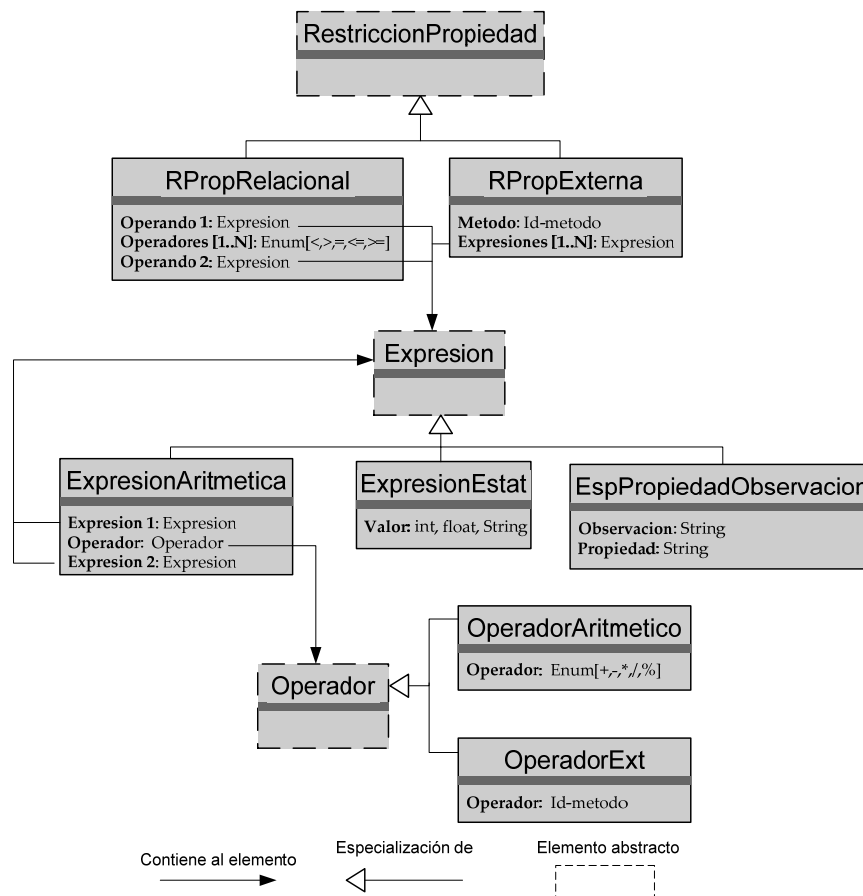


Figura 3.15: Especificación de las Restricciones sobre Propiedades

### 3.4.2.4 Restricciones Lógicas

Las restricciones definidas hasta el momento únicamente permiten relacionar dos intervalos entre sí. Sin embargo, las descripciones necesarias en el nivel de interpretación pueden requerir restricciones más complejas. El mecanismo empleado para conseguirlo consiste en relacionar restricciones creando una red binaria mediante los operadores lógicos: *AND*, *OR*, *NOT*. Por ejemplo, supongamos que queremos representar la restricción ternaria: “el intervalo B se encuentra entre el A y el C”. En tal caso, se emplearía una Restricción Lógica con la que expresar que el intervalo A está seguido del B y el B está seguido del C, es decir,  $A[m]B \text{ AND } B[m]C$ .

Una Restricción Lógica puede relacionar todo tipo de restricciones, incluyendo otras Restricciones Lógicas, con lo que las posibilidades de representación se amplían considerablemente. Mediante estas restricciones se pueden expresar Restricciones Temporales Cuantitativas disyuntivas con la forma:  $C_{ij}=\{I_1=[a_1,b_1], \dots, I_k=[a_k,b_k]\}$  donde  $C_{ij}$  es el conjunto de intervalos reales que limitan la distancia entre los instantes temporales  $p_i$  y  $p_j$  de forma que  $a_1 \leq p_j - p_i \leq b_1 \vee \dots \vee a_k \leq p_j - p_i \leq b_k$ . La restricción se formaría componiendo Restricciones Lógicas:  $(a_1 \leq p_j - p_i \text{ AND } p_j - p_i \leq b_1) \text{ OR} \dots \text{OR } (a_k \leq p_j - p_i \text{ AND } p_j - p_i \leq b_k)$ . También puede combinar restricciones sobre intervalos (RIntervalo), con las que se verifica si la observación, situación o paso especificado en su atributo *intervalo* son detectados en un momento dado (Figura 3.16).

El elemento Restricción Lógica se describe mediante los siguientes atributos (Figura 3.16):

- *Restricción 1*: Restricción temporal cualitativa, cuantitativa, lógica, restricción sobre intervalo o restricción sobre propiedad que se combina con *Restricción 2*.
- *Operador*: "AND", "OR" o "NOT".
- *Restricción 2*: Restricción temporal cualitativa, cuantitativa, lógica, restricción sobre intervalo o restricción sobre propiedad que se combina con *Restricción 1*. En caso de que el *Operador* sea el NOT, este atributo será vacío.

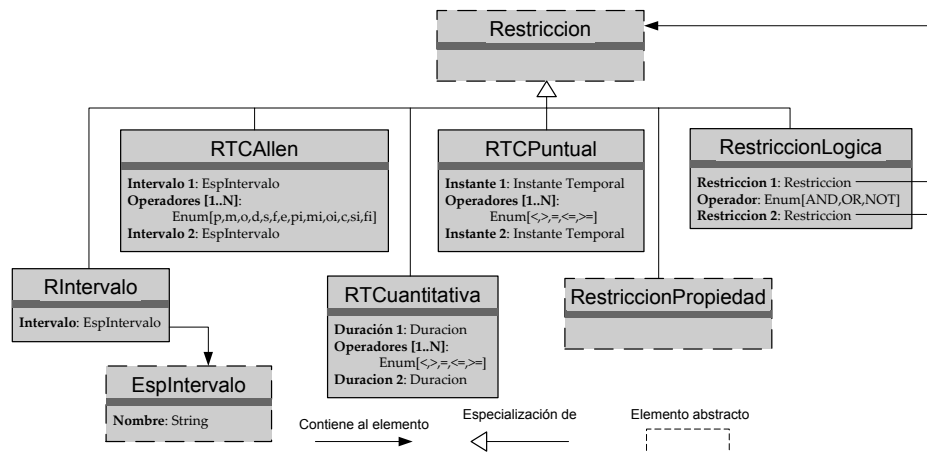


Figura 3.16: Especificación de la Restricción Lógica

Al combinar las restricciones de esta forma, podrían darse ambigüedades que es necesario considerar. Modificando el ejemplo anterior, supongamos que queremos representar la restricción: “el intervalo B se encuentra entre el A y el A”, es decir,  $A[m]B \text{ AND } B[m]A$ . En principio, esta Restricción Lógica es inconsistente porque el intervalo A no puede estar seguido del B y ser seguido por el B al mismo tiempo. Sin embargo, este caso puede darse si interpretamos que el intervalo A aparece dos veces; la primera vez, antes que B; y la segunda, al finalizar B. Para evitar esta problemática, cada aparición de un intervalo que se quiera emplear en las restricciones se especificará explícitamente, dejando la restricción con la forma  $A_1[m]B \text{ AND } B[m]A_2$ . Así, se sabe que B corresponde en ambas restricciones temporales cualitativas con una sola aparición del intervalo, mientras que  $A_1$  y  $A_2$  son dos apariciones. Tal y como se explicará en la sección 3.4.4 sobre el lenguaje del Nivel de Interpretación, la descripción de pasos y situaciones incluye una zona de declaraciones para explicitar las apariciones de los intervalos que se utilicen.

### **3.4.3 Otros elementos del Nivel de Interpretación**

En este apartado se describen los elementos EspIPaso y EspISituación, que son los elementos de interés para el Nivel de Diagnóstico. Estos elementos describen mediante Restricciones cómo se van a interpretar las acciones del alumno y el contexto en el que las lleva a cabo.

#### **3.4.3.1 Especificación del paso**

El elemento EspIPaso representa en el Nivel de Interpretación cada acción diagnosticable del alumno. Sus atributos indican qué restricciones deben verificarse para reconocer de forma segura que el alumno está realizando la acción. Es importante resaltar que las observaciones que definen el paso no pueden depender de que sea un paso correcto o incorrecto. Por ejemplo, en el contexto de la conducción, para definir el paso “Adelantar” no deberíamos incluir entre las observaciones que el vehículo esté en el carril izquierdo. El conductor podría estar adelantando por la derecha, incorrectamente, pero adelantando. Determinar su incorrección es función del diagnóstico, no de la interpretación. Por esta razón hay que ser cuidadosos al definir cada paso utilizando únicamente observaciones representativas de la acción en sí misma.



La especificación de un paso (EspIPaso) se describe mediante los siguientes atributos (Figura 3.17):

- *Nombre*: Identificador único del EspIPaso.
- *Restricciones Generales*: Restricciones entre intervalos que se han de cumplir para interpretar que el alumno está ejecutando el paso. Sólo se podrá confirmar que se está ejecutando si se cumplen todas. En el momento que dejen de cumplirse, el paso se dará por finalizado.
- *Inicio*: Restricciones entre intervalos que al detectarse indicarán el posible inicio del paso. Si no se reconocen indicios que indiquen el comienzo del paso, o si las restricciones de *Inicio* coinciden con las *Restricciones Generales*, este atributo puede quedarse vacío.
- *Ejecución*: Restricciones entre intervalos que al detectarse indican que se puede solicitar diagnóstico. Si el atributo es vacío se pedirá diagnóstico desde el primer momento que se satisfaga alguna parte de las *Restricciones Generales*.
- *Fin*: Restricciones entre intervalos que al detectarse darán por finalizado un paso que se estaba ejecutando. Si se cumplen estas restricciones se dará por finalizado independientemente de que las *Restricciones Generales* se sigan cumpliendo. Si este atributo es vacío el paso se da por finalizado cuando dejan de cumplirse las *Restricciones Generales*.
- *Unidad de tiempo*: Si en alguna restricción se incluye algún dato temporal, este atributo indicará las unidades en las que tiene que expresarse.

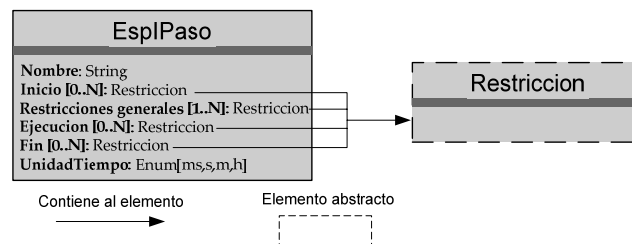


Figura 3.17: Especificación del paso en el Nivel de Interpretación

### 3.4.3.2 Especificación de la situación

Una situación representa un contexto de acción específico que es relevante para el diagnóstico. Dicho contexto está formado por el conjunto de factores que afectan al alumno y condicionan sus pasos. Por esta razón, es necesario identificar dichos factores o circunstancias y así poder relacionarlos con los pasos del alumno. Por ejemplo, supongamos que un alumno conduce un vehículo y de repente se encuentra con otro frontalmente (situación de “Aproximación frontal peligrosa”). Esta situación debería desencadenar que el paso del alumno fuera un “Cambio brusco de dirección”. En este ejemplo, se observa la necesidad de asociar el paso a la situación, ya que el paso “Cambio brusco de dirección” no se puede diagnosticar de la misma manera en una situación de tráfico normal y en la situación “Aproximación frontal peligrosa”. La situación de tráfico normal la definimos como “Situación General”, y representa el contexto por defecto al que se asocian los pasos del alumno cuando no hay ninguna otra situación a la que se puedan asociar. En este sentido, la “Situación General” está definida por defecto en cualquier tarea de cualquier dominio. De esta manera, se puede establecer que el cambio de dirección no está justificado en la “Situación General”, mientras que sí lo está en la “Aproximación frontal peligrosa”, en la que se podría evaluar el control del vehículo, la dirección de giro, etc.

En general, este tipo de ejemplos se observa en entornos autónomos en los que no es posible predecir cuándo ni cuántas veces un elemento autónomo va a provocar una situación. Por tanto, es necesario identificar durante el transcurso de la tarea las situaciones que se están produciendo e interpretar a su vez los pasos que ejecuta el alumno asociados a las mismas.

Para especificar una situación en el Nivel de Interpretación empleamos el elemento `EspISituacion` (Figura 3.18), que se describe mediante los siguientes atributos:

- *Nombre*: Identificador único de la `EspISituacion`.
- *Pasos Posibles*: Identificadores de `EspIPasos` que se pueden producir durante la situación.
- *Restricciones Generales*: Restricciones entre intervalos que se han de cumplir para interpretar que se está dando una situación. Sólo se podrá confirmar que se está produciendo si se cumplen todas las *Restricciones*

*Generales.* En el momento que alguna deje de cumplirse y que ninguno de los *Pasos Posibles* se esté ejecutando, la situación se dará por finalizada.

- *Inicio:* Restricciones entre intervalos que al detectarse indicarán el posible inicio de la situación. Si no se reconocen indicios que indiquen su comienzo, o si las restricciones de *Inicio* coinciden con las *Restricciones Generales*, este atributo puede quedarse vacío.
- *Fin:* Restricciones entre intervalos que al detectarse darán por finalizada una situación que se estaba produciendo. Si se cumplen las restricciones se dará por finalizada independientemente de que las *Restricciones Generales* se sigan cumpliendo. Puede ser un atributo vacío.
- *Unidad de tiempo:* Si en alguna restricción se incluye algún dato temporal, este atributo indicará las unidades en las que tiene que expresarse.

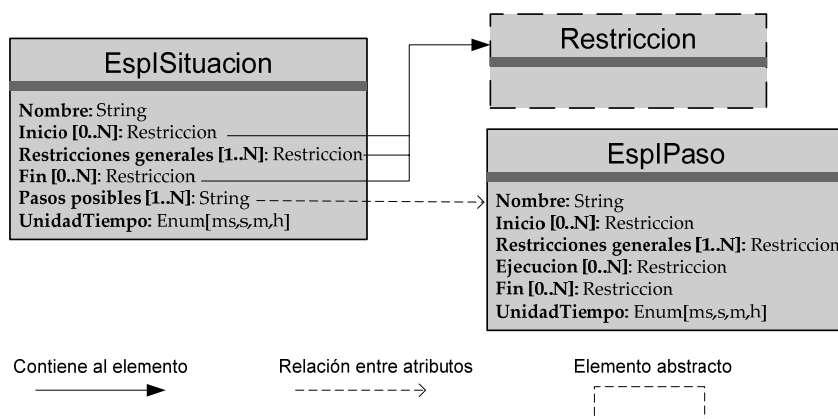


Figura 3.18: Especificación de la Situación en el Nivel de Interpretación

### 3.4.4 El lenguaje del Nivel de Interpretación

Los elementos definidos en el Nivel de Interpretación se emplean en conjunto para describir la actividad del alumno y los contextos en los que se puede encontrar durante la realización de las tareas. Esta descripción particular de cada dominio forma el Modelo de Interpretación. Puesto que es necesario que el modelo sea persistente, se ha definido un lenguaje con el que describirlo y almacenarlo. La gramática completa del lenguaje se

encuentra en el Anexo C en el formato estándar EBNF<sup>13</sup> (International Organization for Standardization, 1996).

En este apartado se exponen ejemplos de un paso y una situación especificados mediante el lenguaje del nivel de interpretación. En el ejemplo de la Figura 3.19 se observa la estructura general del EspIPaso “Arrancar Vehículo” que describe el giro de la llave de contacto para arrancar un vehículo. En su especificación se tiene en cuenta que antes de comenzar el arranque, la llave se encuentra en posición de desbloqueo. El paso comienza cuando se gira la llave hasta la posición de arranque pasando por la de contacto, y termina cuando vuelve a la posición de contacto.

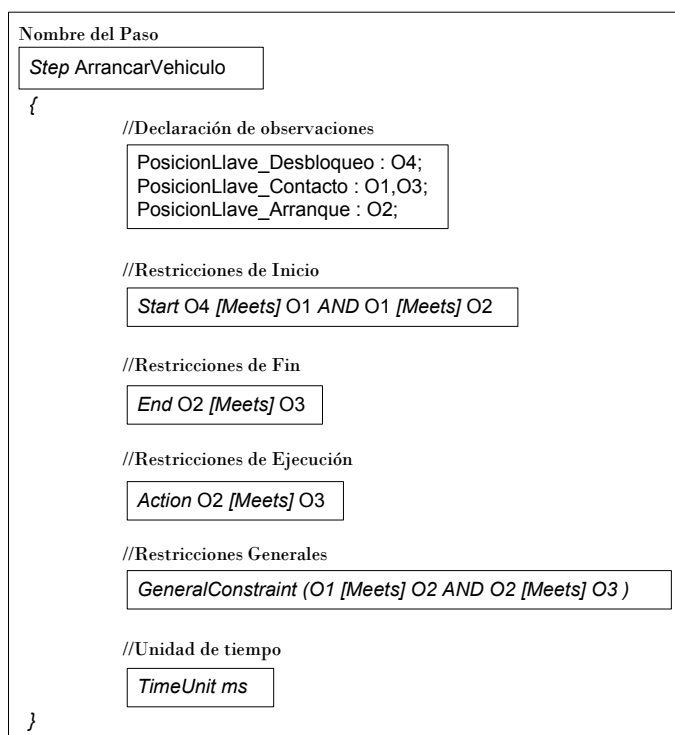


Figura 3.19: Ejemplo de EspIPaso

<sup>13</sup> EBNF (Extended Backus-Naur Form) es un metalenguaje construido para describir formalmente la sintaxis de un lenguaje, de forma compacta y sin ambigüedades, por medio de reglas de producción.

Las palabras y caracteres reservados por el lenguaje se han escrito con letra cursiva. La especificación comienza con la palabra reservada *Step* seguida del nombre del paso, que en este caso es "ArrancarVehiculo". Después, se especifican sus atributos entre llaves ({}), diferenciando varias zonas que se han recuadrado en la figura. La primera es la zona de declaración de observaciones. En este caso se utilizan observaciones por cada posición posible de la llave de contacto (*PosicionLlave\_Desbloqueo*, *PosicionLlave\_Contacto* y *PosicionLlave\_Arranque*). Por cada observación se declaran con nombres distintos tantas apariciones de la observación como veces se tengan que observar en la ejecución del paso (*O1*, *O2*, *O3* y *O4*). Como la llave pasa dos veces por la posición de contacto al arrancar el vehículo, se declaran dos apariciones de la observación *PosicionLlave\_Contacto* (*O1* y *O3*).

Después de la zona de declaración de observaciones se define una zona por cada atributo del elemento *EspIPaso*. Las cuatro primeras corresponden a cada tipo de restricción (*Inicio*, *Fin*, *Ejecución* y *Restricciones Generales*). En cada zona se secuencian todas las restricciones que se quieran definir por cada tipo, comenzando cada una por la palabra reservada que la representa (*Start*, *End*, *Action*, *GeneralConstraint*).

Una secuencia de Restricciones Lógicas que no tuviera sus grupos especificados con paréntesis se leería en el siguiente orden de precedencia de operadores: NOT, AND, OR. Si se quiere definir explícitamente grupos de restricciones lógicas se utilizan paréntesis.

La última zona de la especificación corresponde a la declaración del atributo Unidad de Tiempo, que hace uso de la palabra reservada *TimeUnit*. También se han reservado las abreviaturas de cada unidad de tiempo permitida: *ms*, *s*, *m*, *h*.

La especificación de *EspISituacion* es similar a la del *EspIPaso*. En la Figura 3.20 se muestra como ejemplo la *EspISituacion* "Adelantamiento". El análisis de dicha situación se detalla en el capítulo 5: Aplicaciones de la Metodología y el Marco de Desarrollo (Sección 5.3.4).

```

Situation Adelantamiento
{
    Acercamiento_a_vehículo : O1;
    Vehículo_en_alcance : O2;
    Alejamiento_de_vehículo : O3;

    Start O1 AND O2.Distancia<10
    End O2.Distancia<-10
    GeneralConstraint O1 [Meets] O3 OR (O1 [Precedes] O3 AND Duration(O1.End, O3.Start)<10000)
    PossibleSteps Cambiar_de_carril_hacia_la_derecha, Cambiar_de_carril_hacia_la_izquierda,
                  Dar_intermitente_izquierdo, Dar_intermitente_derecho, Rebasar

    TimeUnit ms
}

```

Figura 3.20: Ejemplo de EspISituacion

### 3.5 NIVEL DE DIAGNÓSTICO

El Nivel de Diagnóstico contiene los elementos que hay que particularizar al construir un SI<sup>2</sup>A<sup>2</sup> para generar el Modelo de Tareas. Este modelo debe describir las tareas que se le van a plantear al alumno de forma que puedan ser diagnosticadas automáticamente. Tal y como se avanzaba en el apartado 2.3, DETECTive (Ferrero et al., 2005) es el punto de partida para considerar el diagnóstico automático en la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup>. El análisis de este sistema genérico de diagnóstico ha contribuido a determinar los objetivos que deben cumplirse en este nivel. Dichos objetivos se exponen en el siguiente apartado, y posteriormente, se presentan los elementos que componen el nivel para alcanzarlos.

#### 3.5.1 Objetivos del Nivel de Diagnóstico

El Nivel de Diagnóstico debe proporcionar los elementos necesarios para describir las tareas que se le van a proponer al alumno. El objetivo de este nivel es proporcionar un conjunto de elementos que permitan encapsular un Modelo de Tareas que cumpla los principios que rigen el diseño de DETECTive: genericidad, usabilidad y capacidad de diagnóstico apropiado y suficiente. Además, planteamos otro principio estratégico que es necesario para que la Metodología permita construir más tipos de SI<sup>2</sup>A<sup>2</sup>: la escalabilidad.

- **Genericidad:** Este principio se sigue desde los dos primeros niveles. En este nivel la genericidad también debe mantenerse evitando dependencias con el dominio.

- Usabilidad: DETECTive favorece la usabilidad mediante una herramienta de autor con la que se facilita la generación del Modelo de Tareas. En el caso del Nivel de Diagnóstico, una herramienta de autor puede variar en función del Modelo de Tareas definitivo y no se aborda en este trabajo, aunque se tiene en cuenta para las líneas futuras.
- Capacidad de diagnóstico apropiado y suficiente: La capacidad de diagnóstico depende principalmente de las técnicas de diagnóstico empleadas. En este sentido, hay que tener en cuenta que en función de las técnicas se impone una organización u otra del modelo de tareas. En este trabajo planteamos la posibilidad de integrar las técnicas de diagnóstico que se crean más apropiadas para cada dominio, pero tratando siempre de preservar la genericidad. Por ejemplo, DETECTive emplea el reconocimiento de planes, la identificación de desviaciones y la verificación de precondiciones sobre los procedimientos del dominio. Para utilizar estas técnicas DETECTive contiene un Modelo de Tareas que se compone de:
  - base de conocimiento de Conceptos, Objetos y Procedimientos del dominio.
  - base de conocimiento que contiene la especificación de las tareas y sus patrones de solución.
  - catálogo de errores que describe los errores típicos que suelen cometer los alumnos.

Si en lugar de las técnicas utilizadas por DETECTive se empleasen otras, la organización del modelo de tareas variaría. Teniendo en cuenta esta consideración, el objetivo del Nivel de Diagnóstico es definir los elementos necesarios para encapsular el Modelo de Tareas apropiado a cada técnica. Visto desde otra perspectiva, los elementos que tienen que formar el nivel han de abstraer la entrada al diagnóstico y el conocimiento interno en el que se va a basar para diagnosticar.

- Escalabilidad: Este objetivo se deriva del anterior. Aparte de no especificar las técnicas de diagnóstico a utilizar, tampoco se pretende fijar cuantas técnicas se han de utilizar. En este sentido, el Nivel de Diagnóstico debe indicar de qué forma se puede adaptar el Modelo de Tareas al incorporar nuevas técnicas, dejando así acotado el proceso de escalado.

### **3.5.2 Elementos del Nivel de Diagnóstico**

Los elementos del Nivel de Diagnóstico acotan las características de los Modelos de Tareas que se pueden encapsular para que sean compatibles con el Nivel de Interpretación. Los elementos de este nivel son abstractos, por lo que para integrar un modelo de tareas externo, como podría ser el de DETECTive, es necesario especializar nuevos elementos que traduzcan los propios del Nivel de Diagnóstico en elementos conocidos por DETECTive. Si no se contase con un Modelo de Tareas ya existente que fuera adecuado para diagnosticar en el dominio del SI<sup>2</sup>A<sup>2</sup>, los elementos del Nivel de Diagnóstico se especializarían incorporando los atributos adicionales requeridos por las técnicas de diagnóstico que se fueran a implementar a medida.

El Nivel de Diagnóstico abstrae: (i) en qué consiste una Tarea, (ii) de qué está compuesta, y (iii) cómo se resuelve. Para ello, se definen los siguientes elementos:

- EspDPaso: Especifica un paso del alumno, que es la unidad mínima diagnosticable. Cada EspDPaso que se defina tiene que corresponderse con un EspIPaso del Nivel de Interpretación haciendo coincidir sus nombres.
- EspDSituacion: La EspDSituacion representa un contexto de diagnóstico. Incluye la especificación de posibles soluciones a la situación y la información (observaciones) que será necesaria para diagnosticar los pasos que se ejecuten.
- EspSolucion: Una EspSolucion define cómo podría resolver el alumno una situación, tanto correcta como incorrectamente. Estará compuesta por EspDPasos, aunque la representación específica que tendrá cada EspSolucion dependerá del Modelo de Tareas encapsulado y de la(s) técnica(s) de diagnóstico utilizada(s).
- Tarea: Describe los objetivos de la tarea, su dificultad, el enunciado que se le va a presentar al alumno, la especificación de situaciones a las que se tendrá que enfrentar para resolverla, etc.
- Restricción: Las Restricciones son los mismos elementos descritos en el nivel de interpretación. Mientras su objetivo en la interpretación es definir cómo se interpretan pasos y situaciones, en el diagnóstico se



incorporan como elemento opcional para restringir en la especificación de soluciones cuándo un paso es correcto o incorrecto.

A continuación se detallan estos elementos.

### 3.5.2.1 Especificación del paso

Un paso, independientemente de que sea instantáneo o continuo, representa la unidad mínima que se va a diagnosticar. El alumno ejecuta un paso para llegar a la solución de aquella situación a la que se está enfrentando, y con ello, contribuye en la resolución total de la Tarea. Como elemento abstracto, los atributos del EspDPaso (Figura 3.21) sirven para identificarlo dentro del dominio y para proporcionar la información necesaria para su diagnóstico:

- *Nombre*: Identificador único del EspDPaso. El nombre que se defina tiene que coincidir con el de algún EspIPaso definido en el Nivel de Interpretación. Mientras los atributos del EspIPaso en la interpretación sirven para definir cómo saber si el alumno está ejecutándolo, sus atributos en el Nivel de Diagnóstico sirven para determinar su corrección.
- *Observaciones*: Identificadores de EspObservaciones que se podrán emplear para determinar la corrección del paso especificado. Su uso concreto dependerá de la técnica de diagnóstico utilizada.

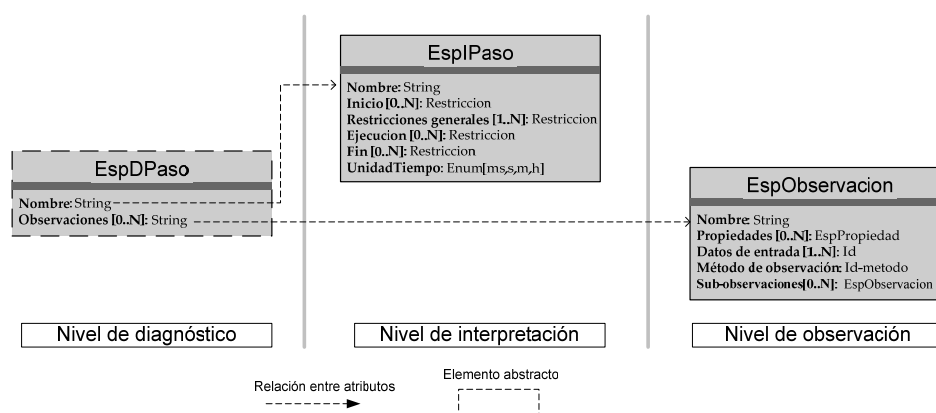


Figura 3.21: Especificación del paso en el Nivel de Diagnóstico

El EspDPaso puede derivarse para incorporar atributos adicionales o para definir la relación con un elemento del Modelo de Tareas encapsulado. Por ejemplo, en el caso del Modelo de Tareas de DETECTive, el EspDPaso

se equipara con el elemento Procedimiento, que contiene una lista de parámetros. Para relacionar ambos, se exige que los nombres de los parámetros se correspondan con alguna de las EspObservaciones indicadas en el EspDPaso. Asimismo, el Procedimiento incorpora un conjunto de precondiciones y postcondiciones para establecer cuando puede realizarse el Procedimiento y qué consecuencias provoca su ejecución. Ambas estarían compuestas de EspObservaciones.

### 3.5.2.2 Especificación de la situación

La EspDSituacion representa el contexto en el que se van a diagnosticar los pasos que realice el alumno cuando se enfrente a ella. Mediante este elemento se tiene en cuenta que la corrección o incorrección de un paso depende de las circunstancias en las que se ha ejecutado. Así, al definir cómo se diagnostica un paso no es necesario considerar todas las circunstancias posibles en las que se puede ejecutar, sino que cada particularidad se puede representar mediante una situación diferente. De esta forma, si se crea una EspDSituacion en la que se puede ejecutar un paso previamente especificado, no es necesario modificar la definición original del EspDPaso para tener en cuenta las nuevas circunstancias. En este sentido, las situaciones juegan un papel importante para facilitar la generación del Modelo de Tareas. Más aún, al definir nuevas Tareas se pueden emplear EspDSituaciones definidas previamente sin tener que partir de cero o contando únicamente con la definición de los pasos.

La información con la que se diagnostican los pasos ejecutados por el alumno cuando aborda una EspDSituacion se representa mediante el elemento EspSolucion. Hay que tener en cuenta que una situación se puede afrontar de distintas maneras, por lo que a una situación se le pueden asociar varias EspSoluciones, tanto correctas como incorrectas.

Además de incorporar posibles EspSoluciones, la EspDSituacion también incorpora un atributo *Observaciones* con el mismo objetivo que el elemento EspDPaso. Las observaciones aportarán información que caracterizará la situación para diagnosticar los pasos.

La EspDSituacion (Figura 3.22), como elemento abstracto, está formada por los siguientes atributos:

- *Nombre*: Identificador único de la situación. Tiene que coincidir con el nombre de alguna de las EspISituaciones definidas en el Modelo de Interpretación.
- *Observaciones*: Identificadores de EspObservaciones que se podrán emplear para determinar la corrección del paso. Su uso concreto dependerá de la técnica de diagnóstico utilizada.
- *Soluciones*: Lista de EspSoluciones tanto correctas como incorrectas que se emplearán para diagnosticar las acciones del alumno cuando se enfrente a la situación.

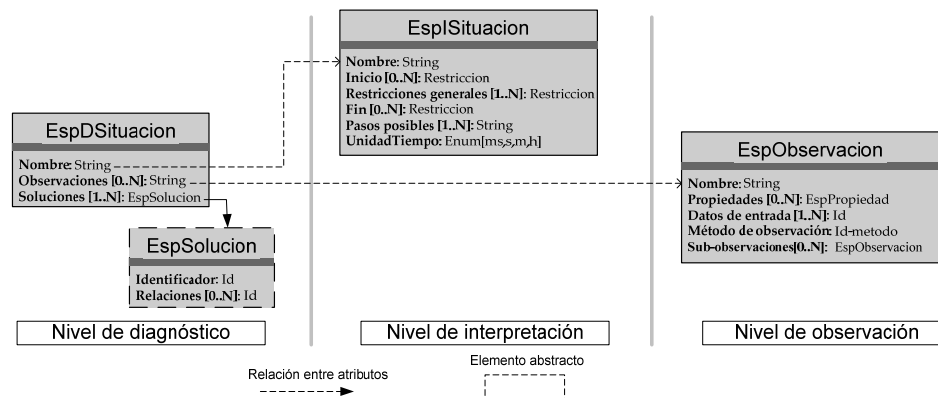


Figura 3.22: Especificación de la situación en el Nivel de Diagnóstico

Al igual que el elemento EspDPaso, la EspDSituacion también se puede derivar para incorporar atributos adicionales que sean necesarios en función del Modelo de Tareas encapsulado. Por ejemplo, se podrían definir precondiciones para limitar en qué casos el alumno se puede enfrentar a la situación. En cualquier caso, las precondiciones tendrían que estar formadas por EspObservaciones.

### 3.5.2.3 Especificación de la solución

El elemento EspSolucion encapsula una forma de resolver una situación aplicando EspDPasos. Hay que remarcar que la EspSolucion no tiene por qué representar una forma correcta de resolver la situación. Se podrían definir soluciones incorrectas para representar los errores típicos que cometen los alumnos y así obtener diagnósticos más precisos.

Los EspDPasos de la EspSolucion estarán organizados en función del Modelo de Tareas encapsulado y los requisitos de la técnica de diagnóstico empleada, por lo que el elemento EspSolucion se especializará para cumplir con los mismos. Además, hay que considerar que podría emplearse una combinación de técnicas para diagnosticar los pasos asociados a una misma situación. Por ejemplo, DETECTive puede utilizar tres técnicas para los casos en que una sola no sea capaz de obtener un resultado de diagnóstico adecuado. Si se utilizan varias técnicas pueden existir dependencias entre las soluciones, por lo que es necesario relacionarlas entre sí. Así, el elemento EspSolucion abstracto únicamente contiene los siguientes atributos:

- *Identificador* único de la solución.
- *Relaciones*: Identificadores de EspSoluciones con las que existe alguna dependencia. Su uso concreto depende de la técnica de diagnóstico.

Aunque en el caso general una especialización de EspSolucion debe contener EspDPasos, en otros podría ser útil formarla a partir de EspDSituaciones. Por ejemplo, se le podría pedir al alumno que resuelva una Tarea afrontando una serie de situaciones en un orden concreto. En este caso se definiría una EspSolucion asociada a la "Situación General" en la que se represente el orden correcto entre situaciones.

#### **3.5.2.4 Especificación de la Tarea**

La Tarea es el elemento principal del Nivel de Diagnóstico. Cada Tarea se define para que el alumno adquiriera unas destrezas concretas, es decir, se plantea de acuerdo con una serie de objetivos de aprendizaje. En función de cómo resuelva el alumno la Tarea se podrá determinar sus progresos de aprendizaje. Para ello, la Tarea incorpora toda la información necesaria para diagnosticar automáticamente la actividad del alumno. Esta información consiste en el conjunto de EspDSituaciones que tiene que afrontar el alumno junto con sus correspondientes EspSoluciones. Por tanto, al menos debe existir una situación que se resuelva siguiendo una serie de pasos.

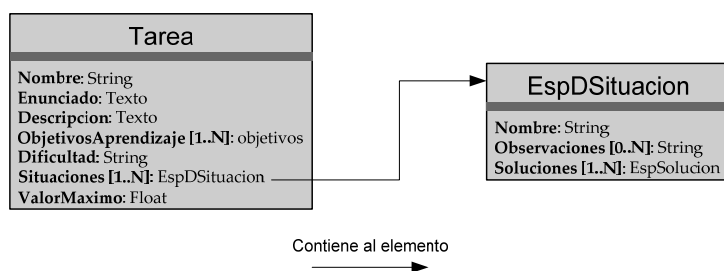


Figura 3.23: Especificación de la Tarea

El elemento Tarea (Figura 3.23) se define mediante los siguientes atributos:

- *Nombre* de la tarea.
- *Enunciado*: Planteamiento de la tarea que se le comunicará al alumno.
- *Descripción*: Explicación de los contenidos asociados a la tarea y a su solución.
- *Objetivos de aprendizaje*: Objetivos de aprendizaje definidos por el instructor que se asocian a la tarea. Su contenido no tiene relevancia para la creación del SI<sup>2</sup>A<sup>2</sup>, pero son útiles para el instructor desde el punto de vista pedagógico.
- *Dificultad* del ejercicio.
- *Situaciones*: Lista de EspDSituaciones a las que se podría enfrentar el alumno mientras resuelve la tarea.
- *Valor máximo*: Puntuación que se otorgará a la resolución óptima de la tarea.

### 3.5.2.5 Elementos Restricción

Los elementos Restricción en el Nivel de Diagnóstico son los mismos que los definidos en el Nivel de Interpretación (Sección 3.4.2). Sin embargo, su utilización en este nivel es opcional y además tiene objetivos diferentes. Mientras las Restricciones se emplean en la interpretación para definir cómo reconocer pasos y situaciones, en el Nivel de Diagnóstico se pueden emplear para desarrollar a medida el proceso de detección de errores si no se dispone de uno válido ya existente. En este caso las Restricciones

servirían de herramienta para definir soluciones. A continuación se indican algunos de los casos para los que pueden resultar de utilidad:

- Restricción de orden entre EspDPasos: Mediante las restricciones temporales cualitativas (apartado 3.4.2.1) se puede indicar con precisión la relación de orden entre dos pasos de una solución, más allá de las relaciones simples de precedencia que representaría una secuencia simple de pasos.
- Restricción de distancia entre EspDPasos: Mediante restricciones temporales cuantitativas (apartado 3.4.2.2) se puede indicar el tiempo máximo o mínimo que puede transcurrir entre dos pasos para considerar satisfactoria la solución del alumno.
- Condiciones sobre EspDPasos: Mediante restricciones sobre propiedades (apartado 3.4.2.3) se puede indicar la corrección o incorrección de un paso en función de que las propiedades de alguna observación asociada al paso se ciñan a algún rango de valores.
- Las restricciones lógicas (apartado 3.4.2.4) siguen siendo válidas en este nivel para generar restricciones que combinen cualquiera de las anteriores.

Obviamente, la utilidad de las Restricciones en el Nivel de Diagnóstico depende de la técnica de diagnóstico que se vaya a emplear, por lo que el uso de este elemento es opcional. Sin embargo, se ha creído conveniente incorporarlas a este nivel por las posibilidades que pueden ofrecer.

### **3.6 UNA VISIÓN CONJUNTA Y ESTRUCTURADA DE LOS MODELOS DE OBSERVACIÓN, INTERPRETACIÓN Y TAREAS**

A lo largo de este capítulo se han presentado los elementos de cada nivel del metamodelo de integración. Para construir un SI<sup>2</sup>A<sup>2</sup> hay que particularizar estos elementos para el dominio de trabajo específico, creando así los Modelos de Observación, Interpretación y Tareas. En esta sección se muestra una visión conjunta de los modelos para proporcionar una perspectiva general de su estructura.

En la Figura 3.24 se muestra un ejemplo genérico de los tres modelos y de las relaciones principales que existen entre ellos. Observando la figura desde la parte superior, comprobamos que las Tareas contienen las EspDSituaciones a las que se va a enfrentar el alumno para resolverlas. Además, las EspDSituaciones pueden reutilizarse entre distintas Tareas. Cada EspDSituación contiene un conjunto de EspSoluciones que representan la información necesaria para diagnosticar los pasos que realice el alumno cuando las aborde.

Siguiendo con el Modelo de Interpretación, los EspIPasos y EspISituaciones definidos representan alguno de los EspDPasos y EspDSituaciones del Modelo de Tareas. Hay que tener en cuenta que una EspISituación puede representar más de una EspDSituación siempre que estén contenidas en tareas diferentes. Esta posibilidad resulta adecuada si las soluciones a una situación se quieren valorar de forma distinta en función de los objetivos de la tarea. Finalmente, tanto los EspIPasos como las EspISituaciones del Modelo de Interpretación son representados mediante conjuntos de Restricciones sobre las EspObservaciones del Modelo de Observación.

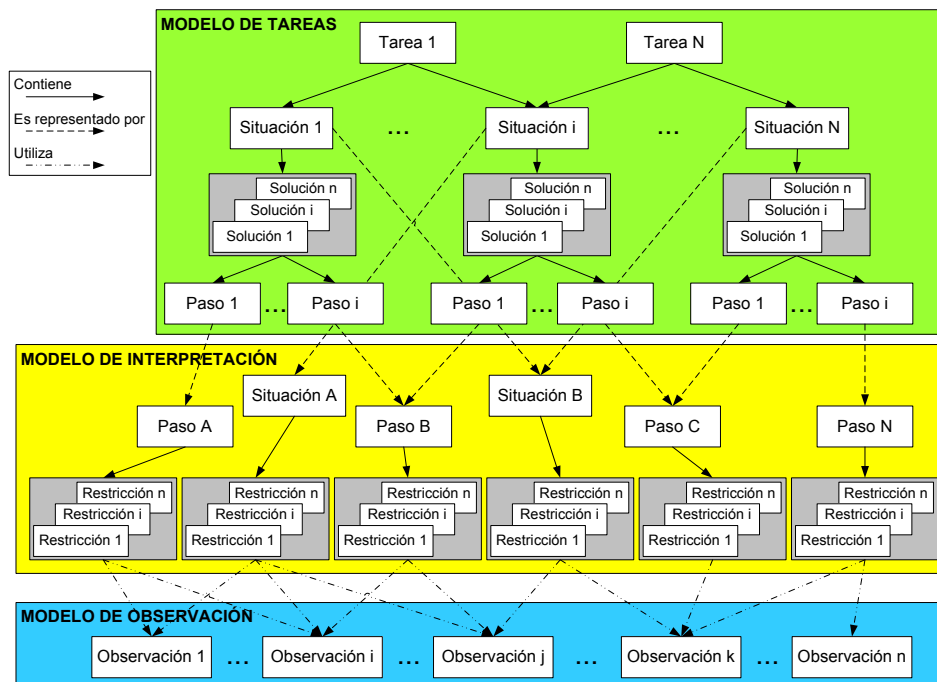


Figura 3.24: Ejemplo genérico de Modelos de Observación, Interpretación y Tareas

Estos tres modelos son la base de la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup>. Una vez definidos, el Marco de Desarrollo que se presentará en el siguiente capítulo se encargará de la comunicación automática entre el Sistema Interactivo y el sistema de diagnóstico encapsulado. Así, cualquier SI<sup>2</sup>A<sup>2</sup> que se vaya a construir integrará un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje siguiendo el esquema: (i) observar que está sucediendo en el entorno interactivo, (ii) interpretar lo que se ha observado y (iii) analizar si es correcto o incorrecto para determinar si el alumno está adquiriendo las destrezas deseadas. Dada la importancia de los modelos, en la siguiente sección se presenta el procedimiento general de desarrollo de SI<sup>2</sup>A<sup>2</sup>, en el que se consideran las implicaciones de los modelos sobre el resultado final del SI<sup>2</sup>A<sup>2</sup>.



### 3.7 PROCESO DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup>

En esta sección se presenta un proceso genérico de desarrollo de SI<sup>2</sup>A<sup>2</sup>. Las fases del proceso asociadas a la Metodología corresponden a la generación de los Modelos de Tareas, Interpretación y Observación específicos del dominio de aprendizaje. Además se muestran las relaciones con otras fases del desarrollo del SI<sup>2</sup>A<sup>2</sup> que, aunque no forman parte de la Metodología, se ven afectadas por la misma (Figura 3.25). El orden establecido para completar el proceso metodológico es el siguiente:

- 1) Generar Modelo de Tareas.
- 2) Generar Modelo de Interpretación.
- 3) Generar Modelo de Observación.

Estas fases están relacionadas con flechas de doble sentido, ya que los procesos asociados a la generación de cada modelo incorporan sub-fases para verificar periódicamente la coherencia del modelo que está siendo generado respecto a los demás.

A continuación se describe el objetivo y las relaciones existentes entre todas las fases del proceso de desarrollo de SI<sup>2</sup>A<sup>2</sup>, y en los siguientes apartados se presentan los procesos asociados a cada fase de la Metodología:

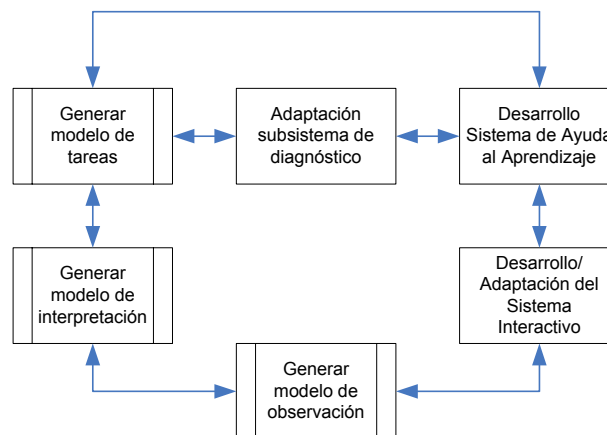


Figura 3.25: Proceso de desarrollo de SI<sup>2</sup>A<sup>2</sup>

- **Generación del Modelo de Tareas:** Tal y como se detallará en la siguiente sección, la generación del Modelo de Tareas implica la descripción de pasos posibles, situaciones y soluciones. Este proceso implica que los elementos del Nivel de Diagnóstico se habrán especializado para integrar los conocidos por el Modelo de Tareas del Subsistema de Diagnóstico que se haya decidido integrar en la fase “Adaptación del Subsistema de Diagnóstico”. Tal y como se estableció en los requerimientos de partida de este trabajo, el Modelo de Tareas que se integre hace de nexo con el resto de componentes del Sistema de Ayuda al Aprendizaje del SI<sup>2</sup>A<sup>2</sup>.
- **Generación del Modelo de Interpretación:** La generación de este modelo depende directamente del proceso de generación del Modelo de Observación. Tal y como se explicará en la sección 3.7.2, podría darse el caso de que alguno de los EspIPasos o EspISituaciones no se pudieran interpretar. En tal caso, habría que revisar el Modelo de Tareas para definir soluciones alternativas que no impliquen el uso de dicho paso o situación. Si no fuera posible, el experto docente tendría que valorar las consecuencias de suprimirlo y definir tareas alternativas que no lo incluyan.
- **Generación del Modelo de Observación:** Este proceso implica desarrollar los métodos de observación que requiere el elemento EspObservacion. Al diseñarlos se determina qué datos de entrada les hace falta obtener del Sistema Interactivo para extraer las observaciones. Por tanto, hay que examinar si éste es capaz de comunicarlos para determinar las adaptaciones que necesitaría en caso de que no lo fuera. Por ejemplo, si en una tarea hubiera que conocer de forma realista la fuerza ejercida por el alumno para interactuar con un objeto, el Sistema Interactivo tendría que contar con un dispositivo háptico para poder comunicar dicho dato. Si originalmente no lo tuviera habría que incorporarlo. Es importante resaltar la necesidad de definir un Modelo de Observación lo más completo posible. Si se obvia alguna observación importante, la interpretación sería deficiente y el diagnóstico se alejaría de la realidad. Posteriormente implicaría la adaptación del Sistema Interactivo para poder extraer la observación. Sin embargo, si el Modelo de Observación es muy completo se facilita la generación de nuevas tareas una vez construido el SI<sup>2</sup>A<sup>2</sup> sin necesidad de adaptar ni implementar ningún módulo del SI<sup>2</sup>A<sup>2</sup>.

- **Adaptación del Sistema Interactivo:** Además de la relación con el Modelo de Observación, el desarrollo del Sistema Interactivo podría relacionarse con el Sistema de Ayuda al Aprendizaje. Si el Sistema de Ayuda al Aprendizaje necesita comunicarse con el alumno, es necesario dotar al Sistema Interactivo de las capacidades de presentación de información que sean necesarias. En los casos más sencillos, se le podría presentar un texto o audio de ayuda y en los más complejos, se podría pedir una demostración.
- **Adaptación del Sistema de Ayuda al Aprendizaje:** El experto docente determinará una parte de sus características. Sin embargo, la integración dentro del SI<sup>2</sup>A<sup>2</sup> está centrada en la comunicación con el subsistema de diagnóstico. Principalmente se le pedirá al Sistema de Ayuda al Aprendizaje que tome sus decisiones a partir de los resultados de diagnóstico de cada paso o de la tarea. Además se le comunicaría la información adicional que necesitase para contextualizar los resultados, como la situación en la que los pasos se llevan a cabo y las observaciones que la caracterizan (López-Garate et al., 2008a; López-Garate et al., 2008b). Empleando DETECTive como subsistema de diagnóstico, esta fase se facilita, ya que DETECTive es integrable en Sistemas Tutores Inteligentes que se generen con la herramienta de autor IRIS (Ferrero et al., 2001).
- **Adaptación del subsistema de diagnóstico:** Esta fase conlleva la integración del Modelo de Tareas definitivo mediante la especialización de los elementos abstractos del Nivel de Diagnóstico del metamodelo. Esta fase está relacionada con la generación del Modelo de Tareas, ya que el diseño de soluciones para las situaciones determina la forma en la que se tienen que diagnosticar y, por tanto, determina si el subsistema de diagnóstico a integrar es adecuado y si requiere alguna adaptación para que el diagnóstico funcione adecuadamente. También se relaciona con otros componentes del Sistema de Ayuda al Aprendizaje, porque estos deben tomar sus decisiones en función de los resultados de diagnóstico. En este trabajo hemos tomado DETECTive como referencia para el diagnóstico. Este entorno encaja en la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup>, permite diagnosticar en un amplio rango de dominios procedimentales y además puede ser integrado con Sistemas Tutores Inteligentes. En la actualidad DETECTive no puede diagnosticar

actividad continua, aunque se está trabajando en las extensiones necesarias para que sea posible.

### **3.7.1 Generación del Modelo de Tareas**

El proceso de generación del Modelo de Tareas consiste en particularizar para el dominio de aprendizaje los elementos del Nivel de Diagnóstico del metamodelo de integración. Por consiguiente, las sub-fases de este proceso dependen en última instancia del subsistema de diagnóstico que se haya integrado y de la organización de su propio Modelo de Tareas. Dicho modelo no está incluido en la Metodología, y por tanto no es posible detallar cómo se debe generar. Sin embargo, su compatibilidad con la Metodología está predeterminada por los elementos del Nivel de Diagnóstico y las relaciones que existen entre ellos. Por esta razón, el proceso de generación del Modelo de Tareas incluye tanto sub-fases para particularizar dichos elementos, como para completarlos de acuerdo a los requisitos del subsistema de diagnóstico (Figura 3.26). En líneas generales, es un proceso iterativo en el que hay que definir para cada tarea:

- 1) Las situaciones con las que se puede encontrar el alumno al enfrentarse a la Tarea, especificando *observaciones* asociadas, y el resto de atributos de cada EspDSituación.
- 2) Soluciones para cada situación: Soluciones correctas e incorrectas de la situación e identificación de técnicas de diagnóstico.
- 3) Pasos que realizará el alumno para resolver cada situación: Descripción general del paso en el dominio y contextualización dentro de las soluciones en las que se integre.

A medida que se va completando el Modelo de Tareas irán apareciendo EspDSituaciones y EspDPasos que habían sido definidos previamente en otras Tareas o EspDSituaciones. Por esta razón, en el procedimiento se indica la necesidad de comprobar para cada nuevo paso o situación si estos existían anteriormente. En tal caso, antes de reutilizarlos hay que verificar si la definición previa encaja en el contexto de la nueva tarea o situación. El motivo principal por el que podría no encajar es que la definición previa fuera incompleta. Por ejemplo, podría faltar alguna observación necesaria para diagnosticar un paso. Una vez identificada la causa de la incoherencia, se optaría por unificar ambas definiciones o por crear una nueva.

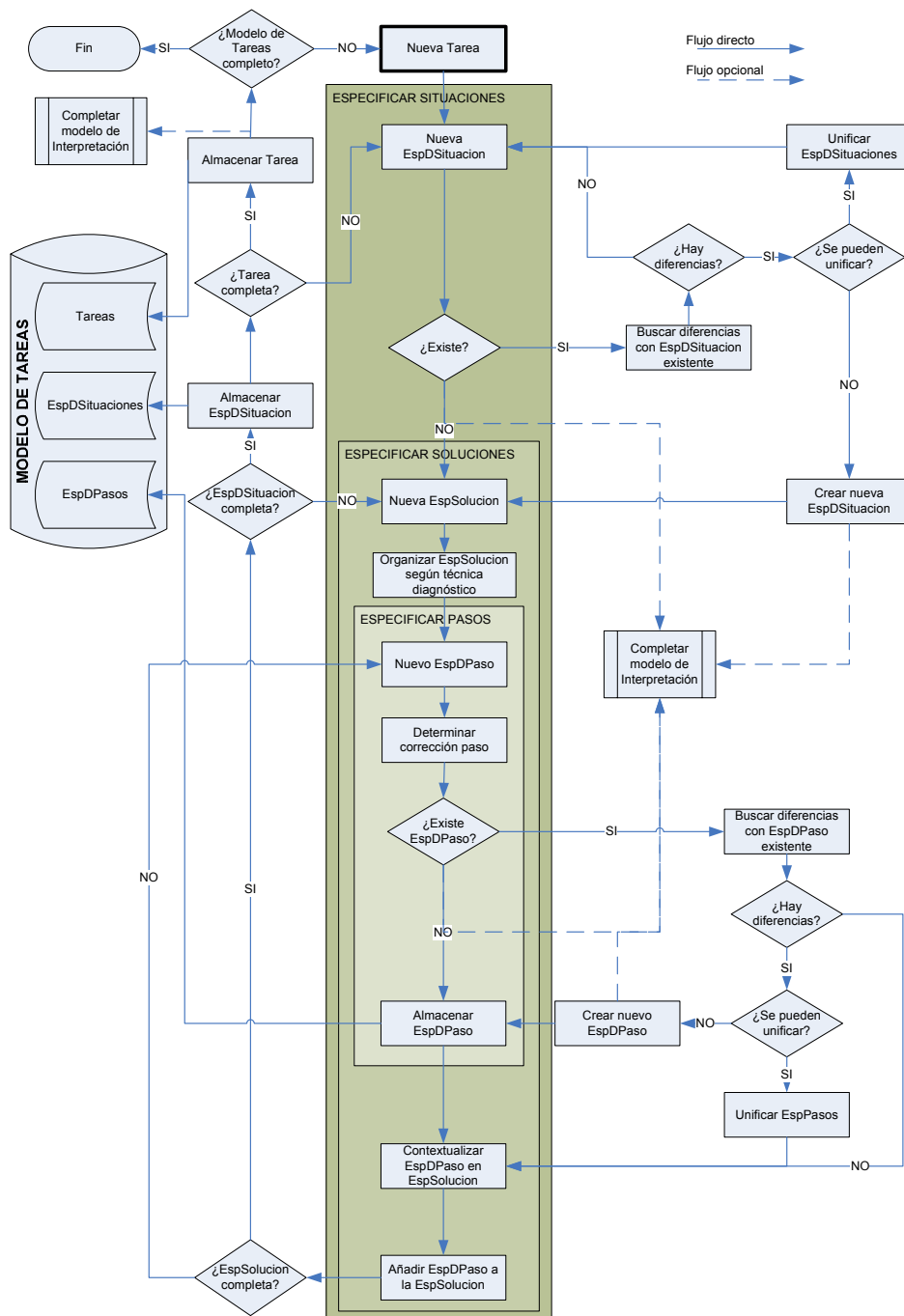


Figura 3.26: Proceso de generación del Modelo de Tareas

Aunque el proceso está centrado en generar el Modelo de Tareas completo, no es recomendable acabarlo antes de iniciar el Modelo de Interpretación. Hay que tener en cuenta que cada EspDSituacion y EspDPaso del Modelo de Tareas tiene que estar vinculado con un EspIPaso o EspISituacion del Modelo de Interpretación. Tal y como se indicará en la siguiente sección, habría que prescindir de los pasos o situaciones que no se pudieran interpretar y analizar el impacto de esta eliminación en el Modelo de Tareas. Obviamente, cuantas más veces hubiera sido reutilizado el elemento, mayor sería el impacto. Para minimizar el riesgo, este procedimiento indica mediante líneas discontinuas un flujo opcional que recuerda la necesidad de completar regularmente el Modelo de Interpretación.

### **3.7.2 Generación del Modelo de Interpretación**

El proceso de generación del Modelo de Interpretación especifica las sub-fases a seguir para definir cómo se interpretan los pasos y situaciones empleados en el Modelo de Tareas (Figura 3.27). El procedimiento es el mismo para cada EspISituacion o EspIPaso, y se centra en la definición de restricciones. El proceso consiste en:

- 1) Crear un EspISituacion o EspIPaso por cada EspDSituacion y EspDPaso definidos en el Modelo de Tareas. Para ello hay que definir el valor de sus atributos, que principalmente consisten en Restricciones.
- 2) Crear las Restricciones que especifican el EspIPaso o la EspISituacion: implica identificar las observaciones que habrá que extraer del Sistema Interactivo y las relaciones que se pueden dar entre ellas cuando el alumno realiza el paso o se produce la situación.
- 3) Analizar si será posible definir las EspObservaciones identificadas al crear las Restricciones. Así se concreta la lista de EspObservaciones que hay que generar siguiendo el proceso de generación del Modelo de Observación (Sección 3.7.3).

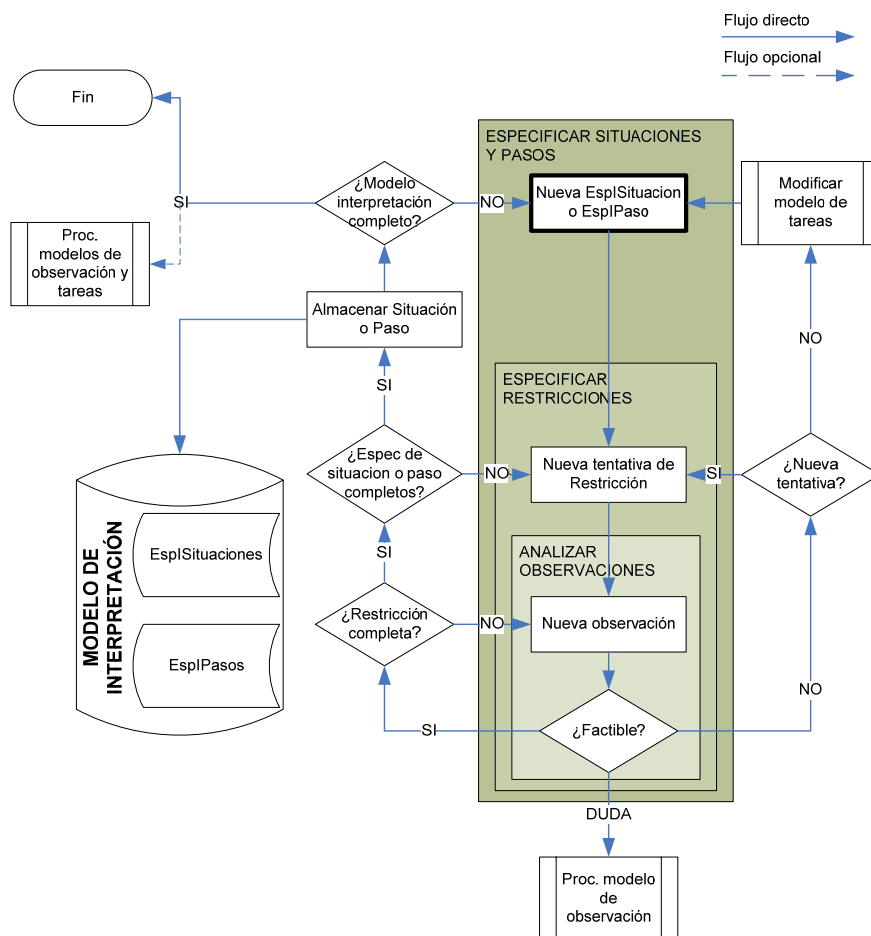


Figura 3.27: Proceso de generación del Modelo de Interpretación

Las principales dificultades para realizar con éxito el proceso están ligadas a la creación de las Restricciones. A veces, puede no ser obvio identificar las relaciones más adecuadas entre observaciones para especificar el paso o la situación. En estos casos aumenta la probabilidad de definir, por error, Restricciones inútiles. Si se produjera este error podría resultar problemático, ya que probablemente no se detectaría hasta que el SI<sup>2</sup>A<sup>2</sup> estuviera desarrollado. Por esta razón, es conveniente identificar a priori las situaciones y pasos más complejos para diseñar varias tentativas de Restricción que integren observaciones variadas. El objetivo es que se puedan rediseñar las Restricciones una vez que el SI<sup>2</sup>A<sup>2</sup> haya sido desarrollado. Si aun así se cometiera un error, el coste de la solución estaría

asociado a la creación de nuevas observaciones y a la dificultad de extraerlas del Sistema Interactivo. Definiendo varias tentativas no se elimina el riesgo, pero sí se reduce el impacto.

Además de los errores que se pueden producir al diseñar Restricciones, hay que tener en cuenta los problemas que pueden darse al utilizar nuevas observaciones, ya que algunas pueden ser imposibles de obtener del Sistema Interactivo. En algunos casos evitarlo es obvio, por ejemplo, no se puede definir una observación para representar que *“un conductor está mirando al espejo retrovisor”* si el vehículo no cuenta con un sistema de cámaras capaz de analizar la dirección de la mirada. Sin embargo, en otros casos podría ser más complejo detectar las dificultades. Por ejemplo, si hubiera que observar al alumno manipulando algún objeto de forma muy precisa, el Sistema Interactivo necesitaría un sistema de detección de colisiones avanzado capaz de identificar las zonas de contacto exactas entre objetos. Estos sistemas tienen un alto coste computacional y, por tanto, no siempre es posible integrarlos en un Sistema Interactivo.

Al utilizar una nueva observación es preciso analizar si será factible extraerla del Sistema Interactivo para desechar la Restricción en caso negativo. Si surgen dudas, la recomendación es iniciar el proceso de generación del Modelo de Observación para tratar de detectar los posibles problemas.

### **3.7.3 Generación del Modelo de Observación**

Este proceso define las sub-fases necesarias para especificar las observaciones identificadas en los Modelos de Interpretación y Tareas. Extraer una observación del Sistema Interactivo consiste en determinar si el hecho representado es observable o no, y en tener actualizado el valor de sus propiedades. El proceso se centra principalmente en concretar los métodos de observación que extraen las observaciones a partir de un conjunto de datos de entrada provenientes del Sistema Interactivo (Figura 3.28).



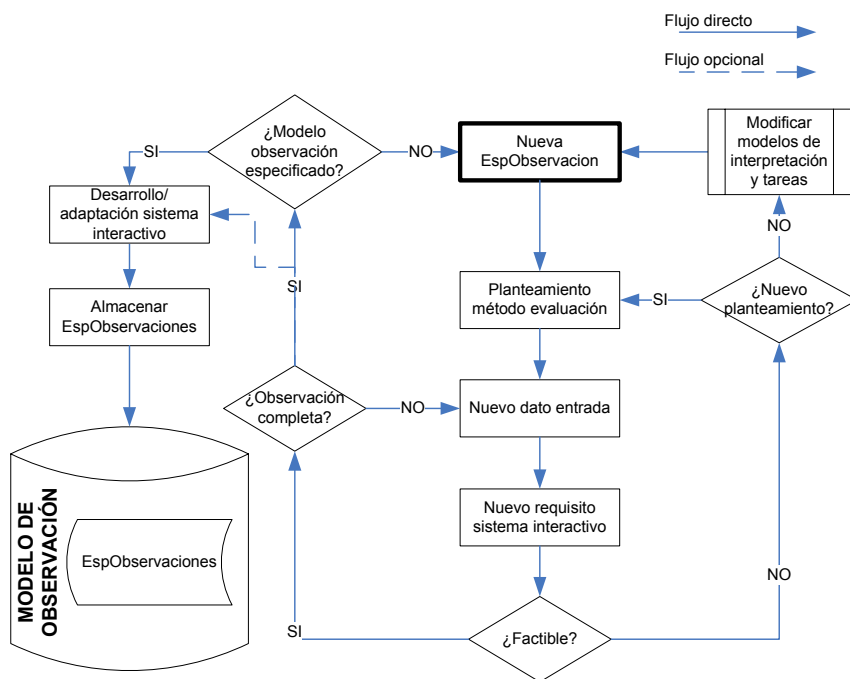


Figura 3.28: Proceso de generación del Modelo de Observación

El proceso consiste en completar para cada EspObservación su atributo *método de observación*, lo que implica identificar su atributo *datos de entrada* y analizar si es factible obtenerlos del Sistema Interactivo. Más detalladamente, el proceso para generar cada EspObservación consiste en:

- 1) Plantear el *método de observación*: Se consideran los siguientes puntos:
  - El método responde a la pregunta de si la observación es observable en un instante específico de tiempo, pero no tiene que considerar cuál fue o será su respuesta en instantes anteriores o posteriores. La gestión de la duración de la observación no es responsabilidad del *método de observación* sino del Marco de Desarrollo.
  - La respuesta del método debe ser objetiva, es decir, no debería ser discutible. Si no fuera objetiva habría que definir alguna observación alternativa o aceptar las consecuencias de que haya casos de observaciones discutibles. Por ejemplo, una observación sin detectar podría implicar que no se llegase a conocer la ejecución de un paso, o que fuera diagnosticado de forma incorrecta.

- Un *método de observación* puede extraer más de una observación. En lugar de utilizar un método por cada observación, podría ser más eficiente utilizar uno sólo para extraer varias. Por ejemplo, con un único método se puede determinar si dos objetos se están acercando o alejando, aunque haya definida una observación de “Acercamiento” y otra de “Alejamiento”.
  - El *método de observación* depende de los *datos de entrada* que se obtienen del Sistema Interactivo, por lo que se puede dejar su implementación para la fase de adaptación del Sistema Interactivo.
- 2) Definir los *datos de entrada*: Una vez que el *método de observación* está planteado hay que especificar los *datos de entrada* que tiene que proporcionar el Sistema Interactivo. La definición de un nuevo dato de entrada implica lo siguiente:
- Comprobar que el dato no es prescindible: es mejor minimizar el número de cálculos aunque, en algunos casos, introducir redundancia puede estar justificado para asegurar que la observación se extrae correctamente.
  - Comprobar si es necesario filtrar el dato: en caso de que el dato de entrada contenga ruido (sección 2.1.4) podría ser necesario filtrarlo.
  - Comprobar si se necesita el histórico de valores del dato: el *método de observación* podría necesitar valores pasados de los *datos de entrada*. Por ejemplo, determinar en un instante si un objeto se está acercando a otro implica conocer las posiciones anteriores de los objetos para calcular la dirección de movimiento y una posible intersección. Si hacen falta datos pasados se debe estimar en qué cantidad y su antigüedad, para evitar que un exceso de datos repercuta negativamente en el rendimiento del *método de observación*.
  - Comprobar si es factible obtenerlo del Sistema Interactivo: es necesario estimar la dificultad de generar cada dato en el Sistema Interactivo. Si generarlo no fuera posible, habría que valorar el impacto que tendría eliminar el dato o la observación sobre los Modelos de Interpretación y Tareas.
- 3) Añadir requisito al Sistema Interactivo: Al definir el *método de observación* y los *datos de entrada* hay que verificar si le exigen algún requisito al Sistema Interactivo, es decir, hay que comprobar que pueda generar y comunicar los *datos de entrada*.

### 3.8 DISCUSIÓN GENERAL

A lo largo de este capítulo se ha expuesto la Metodología de propósito general para el desarrollo de SI<sup>2</sup>A<sup>2</sup>. La base en la que se fundamenta es el metamodelo de integración, que describe de forma genérica cómo integrar un Sistema Interactivo con un Sistema de Ayuda al Aprendizaje. En los puntos siguientes se resume cómo sus características dan respuesta a los problemas que implica la integración, tal y como se describieron en el capítulo 2:

- Integración con entornos autónomos o reactivos (Sección 2.1.1): El tipo de entorno afecta a la integración de dos formas. Por una parte, influye para mantener la coherencia de datos entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje. Por otra parte, afecta a la representación de la actividad procedimental. El Nivel de Observación marca los principios para que el Marco de Desarrollo mantenga la coherencia de datos: actualizar en cada instante qué se ha empezado a observar, qué se sigue observando y qué se ha dejado de observar. Estos principios son válidos tanto para entornos autónomos como reactivos, con lo que la Metodología cumple, en este aspecto, el objetivo de ser independiente del dominio de aprendizaje y de la arquitectura del Sistema Interactivo.

Las diferencias entre entornos autónomos y reactivos también afectan a la representación de la actividad procedimental. La Metodología tiene en cuenta que las acciones del alumno pueden responder a la interacción con elementos autónomos, por lo que no será posible definir a priori todos los pasos que dará el alumno durante toda la tarea. Para estos casos, la solución viene a través de los elementos EspDSituacion y EspISituacion de los Niveles de Diagnóstico e Interpretación. La EspDSituacion incluye el conjunto de soluciones que podría ejecutar el alumno para afrontar la situación, independientemente de si se va a producir por interactuar con un elemento autónomo. Además, en el Nivel de Interpretación, la EspISituacion especifica cómo reconocer que se está produciendo la interacción con el elemento autónomo, sin necesidad de predeterminedar cuándo ni cuántas veces.

- Integración con Sistemas Interactivos de actividad continua y/o basada en eventos (Sección 2.1.2): El metamodelo de integración está diseñado para fusionar la especificación de elementos continuos e instantáneos en sus tres niveles. En primer lugar, el Nivel de Observación modela los hechos que se producirán en el entorno a lo largo del tiempo considerando que son intervalos. La diferencia entre un hecho continuo y otro instantáneo es la duración que tendrá el intervalo, aunque la EspObservacion no necesita distinguirlos. En segundo lugar, el Nivel de Interpretación sirve para modelar el significado de las observaciones desde que empiezan a producirse hasta que acaban, independientemente de su duración. En este sentido, las Restricciones Temporales Cualitativas son una herramienta fundamental para modelar las relaciones entre observaciones sin considerar la duración de los intervalos. Finalmente, el Nivel de Diagnóstico modela la actividad de forma homogénea, es decir, mediante pasos y situaciones que pueden ser continuos en el tiempo o pueden ser instantáneos.
- Integración con entornos realistas y/o simplificados (Sección 2.1.3): Esta diferencia entre entornos influye en el tipo de datos del Sistema Interactivo que hay que analizar. En un entorno realista el Sistema Interactivo genera flujos de datos que definen el estado del entorno en cada momento. Un sólo dato en un instante no tiene significado, pero conjuntamente determinan las acciones y situaciones en las que se encuentra el alumno. La Metodología permite abordar la integración de Sistemas Interactivos tanto simplificados como realistas gracias a los tres niveles de abstracción del metamodelo. Así, tanto eventos instantáneos como flujos de datos se homogeneizan en el Nivel de Observación y se evita diagnosticar directamente la actividad del alumno a partir de los flujos de datos. Además, como cada nivel tiene sus propios objetivos, se acota la dificultad de generar los modelos y se favorece por tanto la usabilidad de la Metodología.
- Conexión con el Sistema de Ayuda al Aprendizaje (Sección 2.2.2): La Metodología de desarrollo quedaba acotada desde el punto de vista de los Sistemas de Ayuda al Aprendizaje en el contexto de la realización de tareas. En este sentido, el análisis realizado enfocaba la integración de un Sistema de Ayuda al Aprendizaje definiendo el nexo de la Metodología en el proceso de diagnóstico de la actividad. Por tanto, quedaba descartado abordar la integración genérica de otro tipo de

funcionalidades de ayuda al aprendizaje, como la planificación instruccional, la comunicación de *feedback*, el análisis de grupos, etc. Sobre este análisis se decidió emplear el Modelo de Tareas de DETECTive como punto de partida para definir el Nivel de Diagnóstico, aunque buscando facilitar la escalabilidad. Así, la Metodología define un conjunto de elementos abstractos que sirven para acotar y encapsular un Modelo de Tareas que siga los principios de DETECTive. Se entenderá que el modelo se puede encapsular si se puede establecer una asociación entre sus elementos y los del Nivel de Diagnóstico. La Metodología no predetermina las técnicas de diagnóstico que se pueden utilizar, de forma que se puedan introducir las más adecuadas para el dominio del SI<sup>2</sup>A<sup>2</sup>. Para facilitar el trabajo, el Marco de Desarrollo puede integrar DETECTive, al que actualmente se están extendiendo sus capacidades de diagnóstico. Su extensión sigue la misma filosofía, preservando su capacidad de diagnóstico independiente del dominio y su independencia de otras funcionalidades de ayuda al aprendizaje.

La Metodología da respuesta a los puntos anteriores indicando cómo usar el metamodelo de integración para construir los Modelos de Observación, Interpretación y Tareas. Estos modelos forman el eje para comunicar al Sistema Interactivo con el Sistema de Ayuda al Aprendizaje. Para hacer efectiva la comunicación, la Metodología se complementa con un Marco de Desarrollo que controla la comunicación basándose en los modelos. Este Marco de Desarrollo de propósito general integra:

- Los procesos que extraen las observaciones definidas en el Modelo de Observación a partir de los *datos de entrada* producidos en tiempo de ejecución por el Sistema Interactivo.
- El proceso de interpretación de las observaciones, que identifica los pasos realizados por el alumno y las situaciones en las que se encuentra basándose en el Modelo de Interpretación.
- La plantilla para integrar el proceso de diagnóstico que determine, en base al Modelo de Tareas, si los pasos realizados por el alumno son apropiados para afrontar correctamente la situación en la que se encuentra.

El capítulo siguiente presenta el Marco de Desarrollo, haciendo hincapié en los tres subsistemas que integran estos procesos.

## *CAPÍTULO 4*

# *MARCO DE DESARROLLO DE SI<sup>2</sup>A<sup>2</sup>*

---

En este capítulo se presenta el Marco de Desarrollo de propósito general que implementa la comunicación automática entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje que se desee integrar. La base de la comunicación son los Modelos de Observación, Interpretación y Tareas definidos según la Metodología presentada en el capítulo anterior. Para ello, el núcleo del Marco de Desarrollo está formado por tres subsistemas que están ligados directamente a los modelos. Cada subsistema se encarga de manejar la comunicación en el nivel de abstracción definido por su modelo.

El capítulo comienza realizando una descripción general del Marco de Desarrollo. Posteriormente se presenta su arquitectura general y se continúa con la descripción de los Subsistemas de Observación, Interpretación y Diagnóstico. La descripción del Subsistema de Observación se centra en el proceso de generación de observaciones; la del Subsistema de Interpretación en los procesos de evaluación de restricciones; y finalmente la del Subsistema de Diagnóstico en la plantilla de integración de componentes de diagnóstico externos.

## 4.1 DESCRIPCIÓN GENERAL DEL MARCO DE DESARROLLO

En el capítulo anterior se presentó una Metodología que organiza en tres niveles genéricos el proceso de comunicación entre un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje. La Metodología define elementos para especificar cómo observar los hechos que suceden en el Sistema Interactivo (EspObservacion), cómo interpretar los pasos realizados por el alumno (EspIPaso) y el contexto en el que los ejecuta (EspISituacion), y cómo diagnosticarlos (EspDPaso y EspDSituacion). Al particularizar estos elementos para el dominio de aprendizaje, quedan definidos los Modelos de Observación, Interpretación y Diagnóstico que sirven de entrada al Marco de Desarrollo, concretamente a los Subsistemas de Observación, Interpretación y Diagnóstico (Figura 4.1):

- Subsistema de Observación: Se encarga de recibir los datos de entrada generados en el Sistema Interactivo, y de extraer las observaciones especificadas en el Modelo de Observación a partir de los mismos. Mientras extrae las observaciones se las comunica al Subsistema de Interpretación para que estén actualizadas cuando las vaya a utilizar.
- Subsistema de Interpretación: Interpreta en tiempo real las observaciones extraídas por el subsistema anterior para saber a qué situaciones se está enfrentando el alumno y qué pasos está ejecutando. Gestiona la información que va interpretando y se la comunica al Subsistema de Diagnóstico.
- Subsistema de Diagnóstico: Utiliza el Modelo de Tareas para determinar la corrección de los pasos realizados por el alumno al enfrentarse a cada situación. Tanto la información de lo que hace el alumno y el contexto en el que lo hace, como los resultados de diagnóstico son accesibles para el Sistema de Ayuda al Aprendizaje o cualquier otro componente del SI<sup>2</sup>A<sup>2</sup>. Así, la interfaz que ofrece este subsistema para acceder a esta información es el nexo de unión entre el Sistema de Ayuda al Aprendizaje y el Marco de Desarrollo.

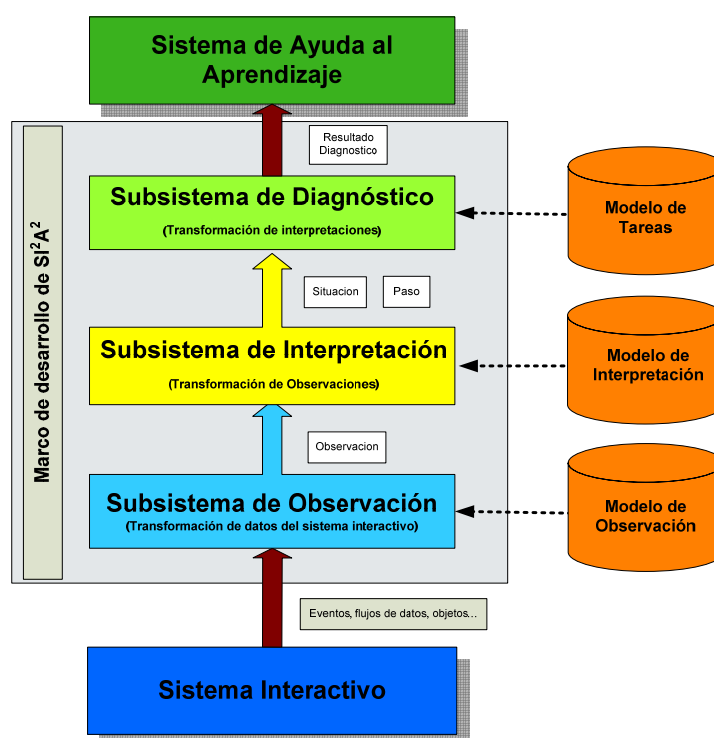


Figura 4.1: Descripción general del Marco de Desarrollo de SI<sup>2</sup>A<sup>2</sup>

Para describir cada subsistema, se presentarán los elementos que generan y los procesos encargados de crearlos y gestionarlos. Se describirá en qué consisten las observaciones y cómo se utiliza el Modelo de Observación para generarlas; en qué consiste el proceso de interpretación, y cómo son los pasos y situaciones que se irán creando; y finalmente, cómo se organiza el Subsistema de Diagnóstico. A continuación se presenta la arquitectura general del Marco de Desarrollo que contiene estos tres subsistemas.

## 4.2 ARQUITECTURA

El Marco de Desarrollo ha sido diseñado siguiendo una arquitectura multiagente, ya que este paradigma facilitaría el escalado del Marco de Desarrollo si las características del SI<sup>2</sup>A<sup>2</sup> así lo requirieran. Los agentes se comunican principalmente siguiendo los protocolos de suscripción, petición y consulta de FIPA, el estándar de interoperabilidad entre agentes



([www.fipa.org](http://www.fipa.org)). A continuación se indican los agentes que componen el Marco de Desarrollo (Figura 4.2):

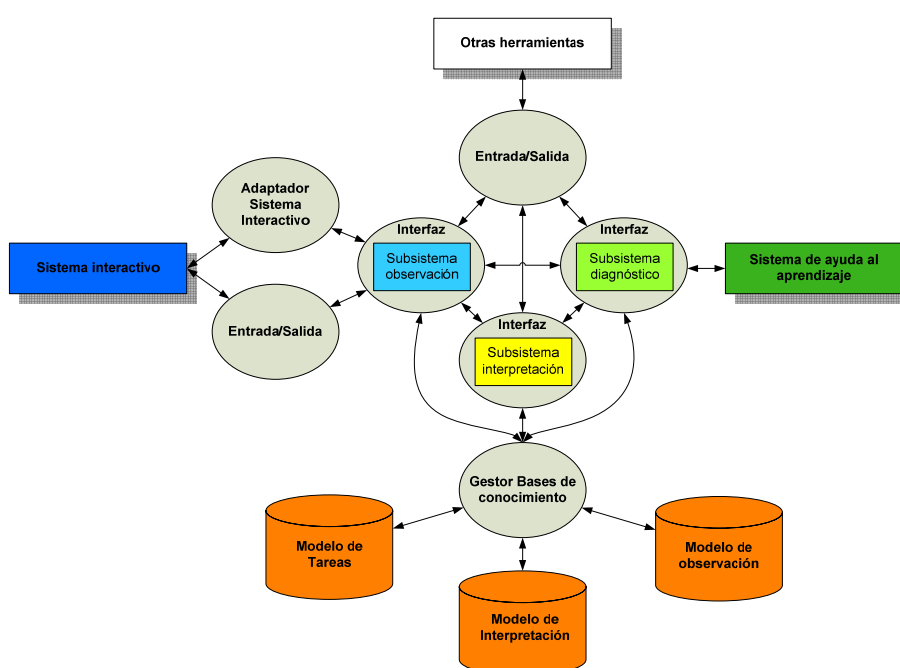


Figura 4.2: Arquitectura multiagente del Marco de Desarrollo

- Agentes Interfaz: de los Subsistemas de Observación, Interpretación y Diagnóstico. El objetivo de estos agentes es encapsular las arquitecturas propias de cada subsistema de forma que la comunicación entre ellos dentro del Marco de Desarrollo o con otros agentes sea homogénea. Inicialmente se plantea únicamente la presencia de un solo agente interfaz para cada subsistema. Sin embargo, para construir SI<sup>2</sup>A<sup>2</sup> colaborativos, se podría utilizar un grupo de subsistemas por cada usuario colaborador, de modo similar a la extensión de STEVE para entornos colaborativos (McCarthy et al., 1999).
- Agentes de Entrada-Salida: los agentes de Entrada-Salida son los encargados de establecer comunicación con la interfaz o interfaces de usuario actuando de intermediarios en la comunicación con el instructor o el alumno. Por una parte reciben de la interfaz los comandos iniciados por el usuario y los distribuyen a los agentes implicados en su ejecución. Por ejemplo comunicarían los comandos de iniciar y finalizar una tarea,

la definición de los parámetros de lanzamiento, etc. Por otra parte, reciben de los demás módulos la información de progreso o las incidencias de la sesión de entrenamiento que se tengan que mostrar en pantalla. Su implementación básica únicamente controla los servicios elementales del Marco de Desarrollo: configuración, inicio, pausa, fin. Al construir un SI<sup>2</sup>A<sup>2</sup>, este agente tiene que ser extendido para manejar los comandos específicos de la nueva aplicación.

- Agente Adaptador del Sistema Interactivo: se encarga de obtener los datos de entrada del Sistema Interactivo especificados por las EspObservacion y de gestionar su envío al Subsistema de Observación. Su misión es empaquetar dichos datos para comunicarlos siguiendo los protocolos del Marco de Desarrollo. Se pueden definir varios agentes de este tipo, sobre todo si la arquitectura del Sistema Interactivo es distribuida. Además de gestionar la comunicación, si hubiera que realizar transformaciones sobre alguno de estos datos, este agente integraría los cálculos. Por ejemplo, se podría realizar un cambio de estructura de datos, un filtrado si tuvieran ruido, o incluso se podrían calcular nuevos datos de entrada a partir de otros ya recogidos del Sistema Interactivo. En un esquema típico, la recogida y envío de datos se integraría en el ciclo de simulación tal y como se indica en la Figura 4.3.

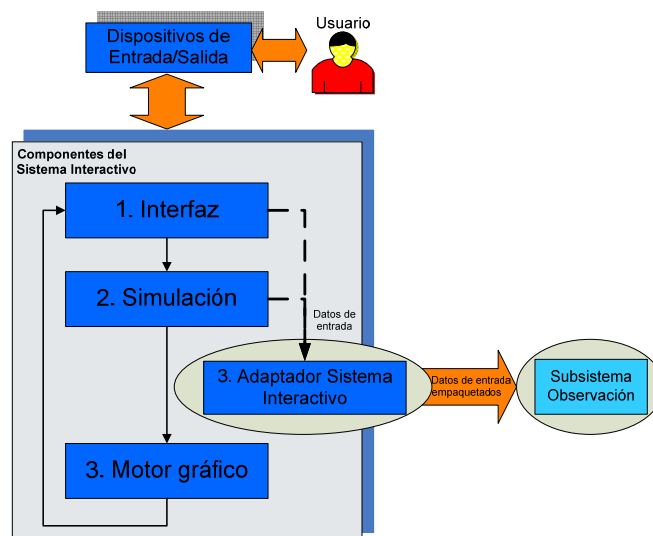


Figura 4.3: Integración del Adaptador del Sistema Interactivo en el ciclo de simulación

- **Agente gestor Bases de Conocimiento:** Este agente gestiona dos tipos de información persistente. Por una parte, manipula los Modelos de Observación, Interpretación y Tareas, ofreciendo servicios de lectura y escritura a los demás agentes. Los servicios de lectura son para que cada agente pueda solicitar los datos que necesita durante las sesiones de entrenamiento, mientras que los de escritura pueden servir para una herramienta de autor que ayude a generar los modelos (Martín et al., 2004). Por otra parte, este agente también gestiona la información generada por cada subsistema durante las sesiones de entrenamiento. Esta información tiene que estar disponible para una posible revisión posterior por parte del instructor, o para que otro proceso la pueda utilizar.

En los siguientes apartados se describen los Subsistemas de Observación, Interpretación y Diagnóstico.

### **4.3 SUBSISTEMA DE OBSERVACIÓN**

El Subsistema de Observación es el encargado de extraer observaciones a partir de los datos de entrada comunicados desde el Agente Adaptador del Sistema Interactivo. Para describir el subsistema, esta sección se divide en dos apartados. En el primero se presenta el Modelo Dinámico de Observación, que contiene la declaración de la clase Observación. El subsistema crea una nueva instancia de esta clase cada vez que extrae alguna de las observaciones especificadas en el Modelo de Observación. En el apartado posterior, se describe el proceso de extracción de observaciones, así como su gestión a lo largo del tiempo.

#### **4.3.1 Modelo Dinámico de Observación**

El Modelo Dinámico de Observación contiene los elementos con los que representar las observaciones que se van extrayendo durante la sesión de entrenamiento. Dichas observaciones son utilizadas por los Subsistemas de Interpretación y Diagnóstico mientras se van generando. Lo importante de la Observación es que se va construyendo a lo largo del intervalo de tiempo en el que se está produciendo, sin necesidad de esperar a que termine. Así, durante el intervalo, contiene el momento en el que comenzó a observarse y se indica que no ha finalizado. Cuando termina, la Observación está descrita

por los instantes en que comenzó y dejó de observarse. Una Observación implementa la interfaz intervalo que contiene los siguientes atributos (Figura 4.4):

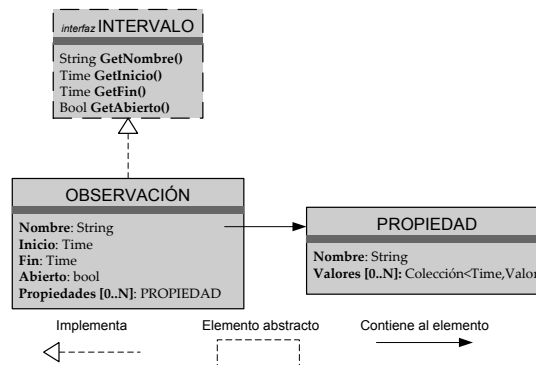


Figura 4.4: Definición de Observación y Propiedad

- *Nombre*: Identificador que se corresponde con el nombre de la EspObservación que lo especifica.
- *Inicio*: Marca temporal que representa el instante en el que se ha empezado a detectar la observación.
- *Abierto*: Sus valores posibles son verdadero y falso. Si es verdadero indica que la observación no ha finalizado y por tanto en el siguiente ciclo de simulación podría seguir observándose.
- *Fin*: Marca temporal que, si *Abierto* es falso, representa el instante en el que la observación dejó de detectarse. Si *Abierto* es verdadero, representa el instante actual de la simulación, indicando que la observación se puede seguir detectando.
- *Propiedades*: Lista de objetos que contienen los valores recogidos para cada Propiedad durante el intervalo de tiempo *Inicio-Fin*. La lista tiene que contener tantos elementos, coincidentes en nombre, como el atributo *Propiedades* de la EspObservación que lo especifica.

Una Propiedad viene especificada por los siguientes atributos:

- *Nombre* de la propiedad.
- *Valores*: Lista de valores de la propiedad asociados a los puntos temporales en los que han sido calculados.

### 4.3.2 Proceso de extracción de observaciones

El Subsistema de Observación ejecuta en cada ciclo de simulación el mismo proceso de extracción de observaciones que se describe en el pseudocódigo indicado más abajo. Este proceso recorre la lista de especificaciones de observaciones y ejecuta sus métodos de observación para comprobar si se están produciendo en ese instante (línea 5). Además, se comprueba si en los instantes anteriores también se estaba observando (línea 6). Con esta información, el resto del proceso está encargado de crear las Observaciones cuando se detecta alguna nueva (línea 15), de extender la duración de los intervalos de las ya existentes (línea 22), y de terminarlas cuando dejan de observarse (líneas 37-39).

Además, cuando se detecta una observación, independientemente de que se estuviera observando anteriormente, se procede a intentar extraer las sub-observaciones que tuviera especificadas (línea 26). De igual manera, cuando las observaciones dejan de detectarse, además de finalizar la propia Observación, se procede a finalizar sus sub-observaciones (línea 40).

```

1   ExtraerObservaciones(espObservaciones)
2   {
3       Para cada esp en espObservaciones
4       {
5           Observado = esp.MetodoObservacion();
6           ob = observActuales[esp.Nombre];
7           si(Observado)
8               //La observación especificada por esp
9               //es observable en el instante actual
10          {
11              Si ob es nulo
12                  //esp no se estaba observando
13                  //en los instantes anteriores
14                  {
15                      ob = nueva Observacion;
16                      observActuales.Añadir(ob);
17                  }
18              Si no
19                  //esp se estaba observando en los
20                  //instantes anteriores
21                  {
22                      ob.Fin = instante actual;
23                  }
24              ob.AñadirPropiedades
25                  (esp.GetValorPropiedades());
26          ExtraerObservaciones

```

```
27         (esp.Sub-observaciones);
28     }
29     Si no
30     //La observación que especifica esp no
31     //es observable en el instante actual
32     {
33         Si ob no es nulo
34         //esp se estaba observando
35         //en los instantes anteriores
36         {
37             ob.Abierto = false;
38             ob.Fin = instante actual;
39             observActuales.Quitar(ob);
40             CerrarSubobservaciones(esp);
41         }
42     }
43 }
44 }
```

Cuando termina el proceso de extracción de observaciones en un ciclo de simulación, se tiene una visión actualizada de todo lo observable en el entorno virtual. Más aún, lo que hasta el momento era un conjunto de datos que únicamente tenían significado en el instante que fueron generados, ha pasado a dar significado a intervalos de tiempo. Gracias a esta información temporal, se va construyendo un primer nivel de conocimiento formado por el conjunto de observaciones que están activas en cada momento. Estas observaciones le sirven de unidad primitiva al Subsistema de Interpretación para determinar lo que está haciendo el alumno y en qué contexto.

#### 4.4 SUBSISTEMA DE INTERPRETACIÓN

El Subsistema de Interpretación es el encargado de determinar si el alumno se está enfrentando a alguna EspISituación y si está ejecutando algún EspIPaso definido en el Modelo de Interpretación. Además, se encarga de comunicar al Subsistema de Diagnóstico las conclusiones a las que va llegando mientras interpreta para que éste pueda ir diagnosticando.

En el siguiente apartado se describen los elementos Paso y Situación que genera este subsistema para comunicárselos al Subsistema de Diagnóstico. En el apartado posterior se describe el proceso de interpretación.

#### 4.4.1 Modelo Dinámico de Interpretación

El Modelo Dinámico de Interpretación contiene los elementos Paso y Situación. Estos elementos se van generando a medida que el proceso de interpretación va determinando si un paso o una situación han comenzado, continúan o han finalizado. Las instancias de Pasos y Situaciones las recibe el Subsistema de Diagnóstico mientras se están produciendo para realizar el diagnóstico.

Tanto el Paso del alumno como la Situación (Figura 4.5) implementan la interfaz Intervalo y contienen los siguientes atributos:

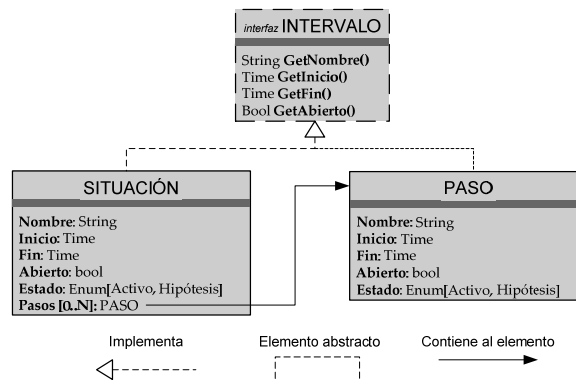


Figura 4.5: Definición de Situación y Paso

- *Nombre*: Identificador que se corresponde con el nombre del EspIPaso o EspISituación que lo especifica.
- *Inicio*: Marca temporal que representa el instante en el que se ha empezado a interpretar el paso o situación.
- *Abierto*: Sus valores posibles son verdadero y falso. Si es verdadero indica que el intervalo no ha finalizado y por tanto en el siguiente ciclo de simulación podría seguir interpretándose.
- *Fin*: Marca temporal que, si *Abierto* es falso, representa el instante en el que el intervalo dejó de interpretarse. Si *Abierto* es verdadero, representa el instante actual de la simulación, indicando que el intervalo se puede seguir interpretando.
- *Estado*: Sus valores posibles son "Activo" o "Hipótesis". El proceso de interpretación crea una instancia de Paso o Situación en estado

“Hipótesis” cuando tiene indicios de que se está produciendo. Una vez se confirma la interpretación cambia el estado a “Activo”. En caso de que estuviera confirmada desde el principio, la instancia de Paso o Situación se crea directamente en estado “Activo”.

Además, la Situación incluye un atributo adicional:

- *Pasos*: Referencia a los Pasos que se estén ejecutando mientras la Situación se está produciendo. Solo se incluyen los Pasos especificados en el atributo *Pasos Posibles* de la EspISituacion que especifica la Situación.

#### **4.4.2 Proceso de interpretación**

El Subsistema de Interpretación decide qué EspIPasos está realizando el alumno y a qué EspISituaciones se está enfrentando a partir de las Observaciones generadas por el Subsistema de Observación y del Modelo de Interpretación. Para conseguirlo, el proceso se compone de dos partes. Por un lado, se evalúa en cada ciclo si las Observaciones que existen hasta ese ciclo cumplen total o parcialmente las Restricciones que contienen cada EspIPaso y EspISituacion. Por otro lado, tomando como entrada el resultado de la evaluación de restricciones, se decide si el paso o situación se está produciendo. Estas dos partes del proceso se describen en los siguientes apartados.

##### **4.4.2.1 Evaluación de restricciones**

Dado un conjunto de Observaciones, la evaluación de restricciones busca el intervalo de tiempo durante el que las restricciones se cumplen, intentando determinar cuanto antes si se están cumpliendo o no. Los resultados posibles que puede ofrecer el proceso de evaluación de restricciones en un ciclo son los siguientes:

- Verdadero: La Restricción se cumple por completo.
- Falso: La Restricción no se cumple.
- Esperar\_Datos: A falta de alguna Observación, las existentes no impiden que se cumpla la Restricción.
- No\_Datos: No existen Observaciones que estén referenciadas en la Restricción.



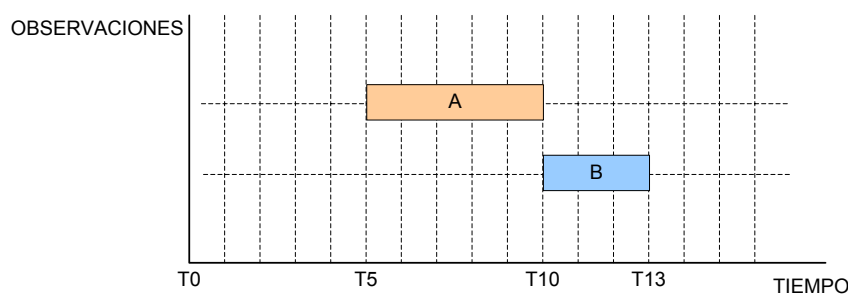
- **Esperar\_Final:** Existen todas las Observaciones necesarias para evaluar la Restricción y no impiden que se cumpla. Sin embargo, hasta que una de las Observaciones esté cerrada no se podrá evaluar.

Si la Restricción contiene alguna observación continua, su evaluación no es inmediata. Por esta razón, el proceso incorpora una memoria que indica el estado de la evaluación de cada Restricción y de las Observaciones implicadas en la misma.

A continuación se describe el proceso de evaluación de restricciones en función del tipo de Restricción: RTCAllen, RTCPuntual, RTCuantitativa, RestriccionPropiedad, RestriccionLogica.

#### ➤ Evaluación de Restricción Temporal Cualitativa Allen

Al evaluar una RTCAllen, pueden darse varias posibilidades, tal y como puede observarse en el ejemplo de la Figura 4.6. En este ejemplo, se quiere evaluar si el intervalo *A* está seguido del *B* ( $A[m]B$ ). Dependiendo del ciclo en el que se esté ejecutando la evaluación de restricciones se da una respuesta diferente:



T0: Comienzo del proceso de evaluación de la RTCAllen:  $A[m]B$

T0-T4: No hay Observaciones. Resultado = No\_Datos

T5-T9: Falta la observación B. A no impide que se cumpla la Restricción. Resultado = Esperar\_Datos

T10-T12: La Restricción se cumple. Resultado = Verdadero

T13-...: No hay Observaciones. Resultado = No\_Datos

Figura 4.6: Ejemplo de evaluación de RTCAllen

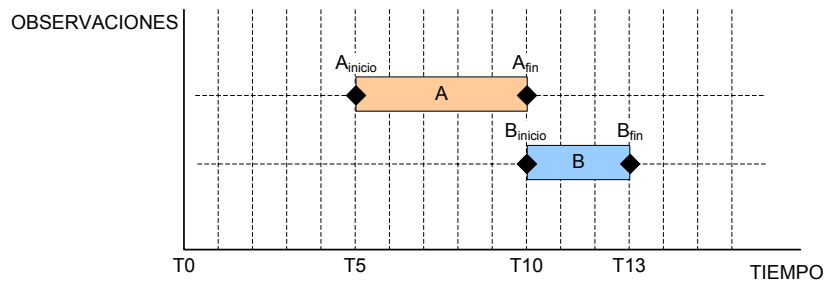
- Hasta el ciclo 5 no existen Observaciones que cumplan la Restricción, lo que significa que el paso o situación que contiene la Restricción no se está produciendo. El resultado de la evaluación es "No\_Datos".

- Desde el ciclo 5 hasta el ciclo 9 únicamente se tiene constancia de la Observación *A* y, obviamente, aún no se sabe si aparecerá la *B* en ciclos posteriores. Por tanto, el resultado de la evaluación es “Esperar\_Datos”.
- Entre los ciclos 10 y 12 la Restricción se cumple, con lo que la evaluación de restricciones devuelve el resultado “Verdadero”.
- Desde el ciclo 13, en el que *B* deja de ser observable, la Restricción ya no se cumple, y el resultado vuelve a ser “No\_Datos”.
- Si no hubiera existido la Observación *A*, entre los ciclos 10 y 12 el resultado habría sido “Falso”, ya que no habría habido posibilidad de que posteriormente se cumpliera.

Este ejemplo únicamente muestra los resultados de evaluar a lo largo del tiempo una RTCAllen que solo contiene el operador *meets*. En el Anexo C se incluyen las reglas de evaluación de RTCAllen para el resto de operadores mediante una tabla con las posibles combinaciones entre dos Observaciones (abiertas/cerradas, observadas/no observadas) y cada operador de Allen.

### ➤ Evaluación de Restricción Temporal Cualitativa Puntual

La evaluación de una RTCPuntual puede devolver los mismos resultados que una RTCAllen. En la Figura 4.7 se muestra un ejemplo en el que se evalúa la Restricción:  $A_{inicio} < B_{fin}$



T0: Comienzo del proceso de evaluación de la RTCPuntual:  $A_{inicio} < B_{fin}$

T0-T4: No hay Observaciones. Resultado = No\_Datos

T5-T9: Falta la observación B.  $A_{inicio}$  no impide que se cumpla la Restricción. Resultado = Esperar\_Datos

T10-T12: La Restricción se cumple. Resultado = Verdadero

T13-...: No hay Observaciones. Resultado = No\_Datos

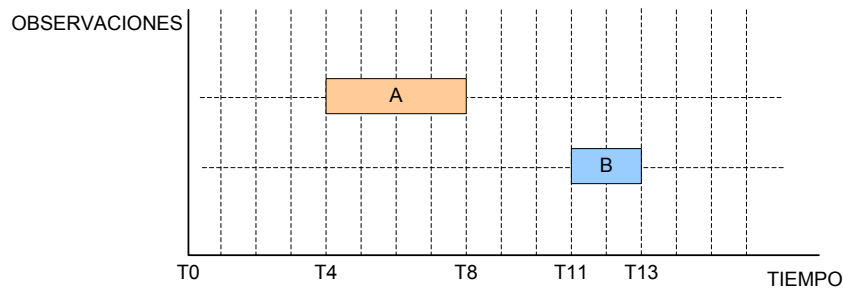
Figura 4.7: Ejemplo de evaluación de RTCPuntual

- Hasta el ciclo 4 no existen Observaciones, con lo que la evaluación de la Restricción devuelve "No\_datos".
- Desde el ciclo 5 al 9 se conoce el instante de inicio de la Observación *A*, pero como no se sabe si la Observación *B* aparecerá, el resultado es "Esperar\_Datos".
- Desde el ciclo 10 al 12, una vez que ha aparecido la Observación *B* ya se sabe que el fin de la Observación *B* es posterior al inicio de la *A*. Por tanto el resultado de la evaluación es "Verdadero".
- Desde el instante 13 en que deja de observarse la *B*, el resultado vuelve a ser "No\_Datos".
- Si no hubiera existido la Observación *A*, al aparecer la *B* en el ciclo 10, el resultado de la evaluación habría sido "Falso".

Las RTCPuntual se definen únicamente mediante los operadores "<", ">", "<=", ">=", "=". Sin embargo, existen más relaciones teniendo en cuenta las combinaciones que se dan por Observaciones abiertas/cerradas y observadas/no observadas. En el Anexo C se incluyen las reglas de evaluación de las combinaciones posibles.

#### ➤ Evaluación de Restricción Temporal Cuantitativa

Las Restricciones Temporales Cuantitativas se evalúan calculando en cada instante las Duraciones que tiene contenidas y comparándolas según el operador relacional definido. Mientras van llegando las Observaciones podrían faltar datos para calcular alguna de las Duraciones, por lo que no se podría decidir si se verificará la Restricción y habría que esperar a tener todos los datos. Este caso se da en el ejemplo de la Figura 4.8, en el que se intenta evaluar la RTCuantitativa:  $Duracion(A_{fin}, B_{inicio}) \geq Duracion(B)$ . Esta Restricción representa que el tiempo transcurrido desde que acaba la Observación *A* hasta que comienza la *B* tiene que ser mayor que la duración de la Observación *B*.



T0: Comienzo del proceso de evaluación  $Duracion(A_{fin}, B_{inicio}) \geq Duracion(B)$

T0-T3: No hay observaciones. Resultado = No\_Datos

T4-T10: Falta la observación B. A no impide que se cumpla la restricción. Resultado = Esperar\_Datos

T11-T12: B no termina y la restricción se cumple. Resultado = Esperar\_Final

T13: La restricción se cumple. Resultado = verdadero

T13-...: Las observaciones cesan. Resultado = No\_datos

Figura 4.8: Ejemplo de evaluación de RTCuantitativa

- Hasta el ciclo 4 no se dispone de Observaciones, por lo que el resultado de la evaluación es "No\_Datos".
- Entre los ciclos 4 y 10 se dispone de la Observación A. Sin embargo, no es posible calcular el tiempo transcurrido entre  $A_{fin}$  y  $B_{inicio}$ , ni la duración de B por no disponer de esta última. Por tanto, el resultado de la evaluación es "Esperar\_Datos".
- Desde el ciclo 11, ya se dispone de las dos Observaciones para comparar las Duraciones contenidas en la Restricción. Entre los ciclos 11 y 12 la duración de B cumple momentáneamente la Restricción, pero no se sabrá si lo hace hasta que ésta finalice, con lo que el resultado hasta el ciclo 12 es "Esperar\_Final".
- En el ciclo 13 la Observación B finaliza y se cumple que el tiempo transcurrido entre  $A_{fin}$  y  $B_{inicio}$  es mayor que la duración de B. Como el estado de la evaluación de la Restricción era "Esperar\_Final", se devuelve el resultado "Verdadero". Además, este caso tiene una peculiaridad añadida, y es que cuando por fin se cumple la Restricción, ya han acabado las Observaciones. Por tanto, en el mismo ciclo, se informa de que ya no existen Observaciones añadiendo "No\_Datos" al resultado.

- En los ciclos posteriores al 13, el resultado es únicamente “No\_Datos”.
- Si la Observación  $B$  hubiera llegado al ciclo 15 o posterior, el resultado habría pasado de “Esperar\_Final” a “Falso”, ya que su duración ya superaría la  $Duracion(A_{fin}, B_{inicio})$ .
- Si la Restricción fuera la inversa:  $Duracion(A_{fin}, B_{inicio}) \leq Duracion(B)$  los primeros ciclos de  $B$  no se cumpliría la Restricción, pero en ciclos posteriores podría acabar cumpliéndose, con lo que también se devolvería momentáneamente “Esperar\_Final” hasta que la  $B$  finalizase o se cumpliera.

El procedimiento general para evaluar cualquier RTCuantitativa consta de dos fases:

- En la primera fase se intenta calcular cada una de las Duraciones por separado. Al intentarlo puede ocurrir que falte alguno de los instantes de inicio o fin. También puede ser que la Observación que lo contenga siga abierta y por tanto el último instante conocido de la Observación cambie en los ciclos posteriores. Teniendo en cuenta estas posibilidades, el cálculo de una Duración devolverá “Verdadero”, “Falso”, “No\_Datos”, “Esperar\_Datos” o “Esperar\_Final” junto a la duración calculada en caso de que haya sido posible.
- En la segunda fase se intenta comparar las dos duraciones. Si ambas duraciones se han podido calcular, se comparan utilizando el operador relacional contenido en la Restricción. El resultado es “Verdadero” o “Falso” dependiendo de si se cumple o no. En caso de que alguna de las Duraciones no se pueda calcular, el resultado de la evaluación se decide en función de la combinación de resultados devueltos en la primera fase. Las combinaciones posibles, así como el algoritmo del cálculo de duraciones se presentan en el Anexo C.

#### ➤ Evaluación de Restricción sobre Propiedad

La evaluación de Restricciones sobre Propiedad no devuelve de resultado ni “Esperar\_Datos” ni “Esperar\_Final”, ya que el valor de la propiedad en un ciclo, o cumple la restricción o no la cumple, pero no es necesario esperar a datos futuros para poder comprobarlo.

En caso de que falte alguna observación para poder evaluar la Restricción sobre Propiedad el resultado es “No\_Datos”.

### ➤ Evaluación de Restricción Lógica

Las Restricciones Lógicas contienen dos Restricciones que se combinan entre sí mediante un operador lógico (*AND*, *OR*, *NOT*). Como dichas Restricciones hijas pueden ser a su vez Restricciones Lógicas, forman una estructura de árbol (Figura 4.9). Para evaluar una Restricción Lógica se comienza evaluando sus restricciones hijas y se propaga el resultado de combinarlas a su Restricción padre hasta llegar a la Restricción raíz. La combinación de resultados (“Verdadero”, “Falso”, “No\_Datos”, “Esperar\_Datos” y “Esperar\_Final”) mediante cada operador lógico se presenta en el Anexo C.

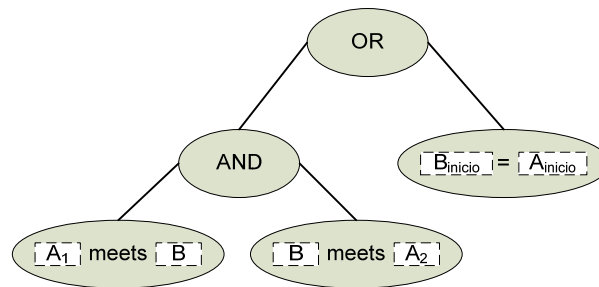


Figura 4.9: Ejemplo de árbol de Restricciones Lógicas

Si un árbol de Restricciones Lógicas contiene varias RTCAllen unidas por el operador *AND*, hay que tener en cuenta que pueden existir relaciones de precedencia entre ellas que estén implícitas en el árbol. Si no se tuvieran en cuenta estas relaciones implícitas se obtendrían resultados incorrectos. Por ejemplo, supongamos la Restricción:  $T = A_1[m]B \text{ AND } B[m]A_2$ , donde  $A_1$  y  $A_2$  son dos apariciones de la EspObservacion  $A$  (Figura 4.10). Dadas las observaciones mostradas en el cronograma, la Restricción se debería evaluar con el resultado de “Verdadero”. Sin embargo, siguiendo la estructura de árbol para evaluarla pasaría lo siguiente:

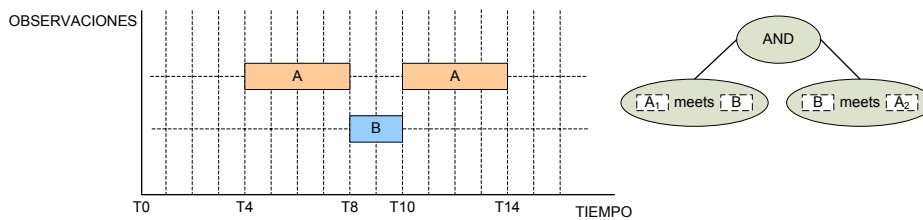


Figura 4.10: Ejemplo de relaciones temporales implícitas entre Observaciones

- En el ciclo 4 se observa la primera aparición de  $A$ . Tal y como se explicaba anteriormente, cada Restricción hija de la Restricción Lógica se intenta evaluar independientemente. Por tanto,  $A_1[m]B$  devolvería "Esperar\_Datos", ya que la Observación  $A$  se asociaría con la EspObservacion  $A_1$  y aún no existe la Observación  $B$ . Pero además, la evaluación de  $B[m]A_2$  devolvería "Falso", porque la observación  $A$  también se asociaría con  $A_2$  y esta Restricción ya no podría cumplirse. Combinando ambos resultados, la Restricción Lógica se evaluaría como "Falso".
- En el ciclo 8 cada RTCAllen se evaluaría al revés que en el ciclo 4. La primera aparición de  $A$  ya habría sido descartada por lo que  $A_1[m]B$  devolvería "Falso" y  $B[m]A_2$  devolvería "Esperar\_Datos". De nuevo la Restricción Lógica se evaluaría como "Falso".

Este tipo de problemas se puede evitar si se conocen las relaciones de precedencia que existan entre las RTCAllen del árbol. Para ello, es necesario preprocesar el árbol y compactar las RTCAllen. El proceso de compactado sustituye el árbol o sub-árbol de elementos RTCAllen por un único nodo equivalente que contendrá una Red de Satisfacción de Restricciones Temporales que se define a continuación.

Una Red de Satisfacción de Restricciones Temporales entre intervalos (RSRT) es una forma clásica de representar un conjunto de relaciones temporales de Allen mediante un grafo. En una RSRT los nodos del grafo se corresponden con los intervalos, mientras que las aristas que los unen son las relaciones temporales de Allen que restringen su relación. Formalmente, el grafo RSRT tiene la forma:  $RSRT = \{n, a\}$ , donde  $n$  es el conjunto de intervalos:  $n = [i_1, \dots, i_n]$ , y  $a$  es el conjunto de aristas:  $a = [R_{11}, \dots, R_{1n}, \dots, R_{ij}, \dots, R_{n1}, \dots, R_{nm}]$ . Cada arista  $R_{ij}$  identifica los nodos relacionados  $i$  y  $j$  y contiene la lista de relaciones básicas de Allen  $I = \{p, m, o, d, s, f, e, pi, mi, oi, c, si, fi\}$  que se pueden dar entre ellos:  $R_{ij} = R(i, j) = [I_1, \dots, I_n]$ . Por ejemplo, la Restricción  $T = A_1[m]B \text{ AND } B[m]A_2$  se transformaría en la red de restricciones  $G = \{[A_1, B, A_2], [(R_{A_1B}=m), (R_{B A_2}=m)]\}$  (Figura 4.11).

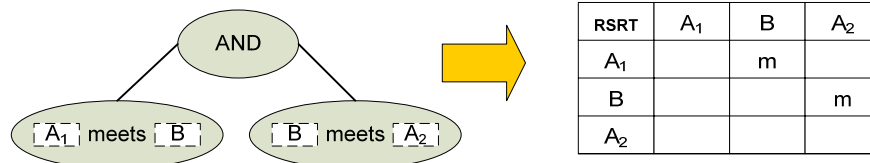


Figura 4.11: Conversión Restricción Lógica – Red de Satisfacción de Restricciones Temporales

Mediante la RSRT se puede obtener información de precedencia entre observaciones contenidas en Restricciones distintas que no se dispone mediante la organización de árbol de las Restricciones Lógicas. Esta información se utiliza para evitar asociar Observaciones que no se esperan a Restricciones que sólo pueden cumplirse más tarde. Siguiendo con el ejemplo, al transformar el árbol en una RSRT inferimos que la relación entre  $A_1$  y  $A_2$  es de precedencia (Figura 4.12). Por tanto,  $A_1[m]B$  tiene que cumplirse antes de intentar asociar una Observación  $A$  a  $A_2$ . Volviendo al cronograma de la Figura 4.10:

- Entre los ciclos 4 y 7,  $A$  es observable y  $B$  no lo es, con lo que  $A_1[m]B$  se evalúa con el resultado “Esperar\_Datos”. Como existe una relación de precedencia con  $A_2$ , ésta última no se asocia a ninguna Observación y  $B[m]A_2$  devuelve “No\_Datos”. La conjunción de ambos resultados es “Esperar\_Datos”. Además, una vez que  $A_1$  ya ha sido asignado, se memoriza para no intentar reasociarlo a otras apariciones de  $A$ .
- En los ciclos 8 y 9, tanto  $A_1$  como  $B$  han sido observados, lo que permite intentar observar  $A_2$ . Puesto que el resultado de comprobar  $A_1[m]B$  es “Verdadero” y en esos ciclos no existe  $A_2$ , el resultado seguiría siendo “Esperar\_Datos”.
- En el ciclo 10 vuelve a observarse  $A$ , por lo que esta vez el intervalo se asigna a  $A_2$ . Así, la evaluación de  $B[m]A_2$  devuelve “Verdadero”. Al combinar este resultado con el resultado “Verdadero” que quedó memorizado al verificar  $A_1[m]B$  el resultado global entre los ciclos 10 y 13 es también “Verdadero”.



RSRT	A <sub>1</sub>	B	A <sub>2</sub>
A <sub>1</sub>		m	
B			m
A <sub>2</sub>			

RSRT	A <sub>1</sub>	B	A <sub>2</sub>
A <sub>1</sub>		m	p
B	mi		m
A <sub>2</sub>	pi	mi	

Figura 4.12: Inferencia de precedencias entre Observaciones

Las Redes de Satisfacción de Restricciones Temporales entre intervalos permiten inferir relaciones de precedencia entre observaciones para evitar problemas como el expuesto anteriormente. Puesto que se pueden dar situaciones similares entre Restricciones Temporales Cualitativas Puntuales, también se realiza un proceso de compactado de árboles o sub-árboles de Restricciones Lógicas que contengan RTCPuntuales, creando RSRT entre Puntos. De esta forma, un  $RSRTP = \{n, a\}$ , donde  $n$  es el conjunto de puntos o instantes temporales:  $n = [i_1, \dots, i_n]$ , y  $a$  es el conjunto de aristas:  $a = [R_{11}, \dots, R_{1n}, \dots, R_{ij}, \dots, R_{n1}, \dots, R_{nm}]$ . Cada arista  $R_{ij}$  identifica los nodos relacionados  $i$  y  $j$  y contiene la lista de relaciones  $I = \{<, >, <=, >=, =\}$  que se pueden dar entre ellos:  $R_{ij} = R(i, j) = [I_1, \dots, I_n]$ .

#### 4.4.2.2 Interpretación de Pasos y Situaciones

El proceso de interpretación es el encargado de determinar en cada instante si cada paso y situación especificados se está produciendo. Hay que tener en cuenta que en algunos momentos no se dispondrá de toda la información para saber si se está produciendo o no. Por lo tanto, la interpretación debe gestionar a lo largo del tiempo en qué estado se encuentra cada paso y situación. Los estados posibles en los que se puede encontrar son:

- Inactivo: El Paso o la Situación no se está produciendo.
- Activo: El Paso o la Situación se está produciendo.
- Hipótesis: Las Observaciones que se van extrayendo no permiten confirmar que el Paso o Situación esté activo, pero en ciclos posteriores podrían hacerlo.

El proceso general que el subsistema de interpretación ejecuta en cada ciclo consiste en comprobar para cada Paso y Situación si hay que cambiarlas de estado. Para ello, se evalúan las Restricciones contenidas en cada atributo de los EspIPaso y EspISituación: *inicio*, *restricciones generales* y *fin*. En función

del estado en el que esté el paso o la situación se comprueban unas restricciones u otras y se decide el cambio de estado dependiendo del resultado de la evaluación: "Verdadero", "Falso", "Esperar\_Datos", "Esperar\_Final", "No\_Datos". Las decisiones se toman siguiendo el diagrama de estados de la Figura 4.13.

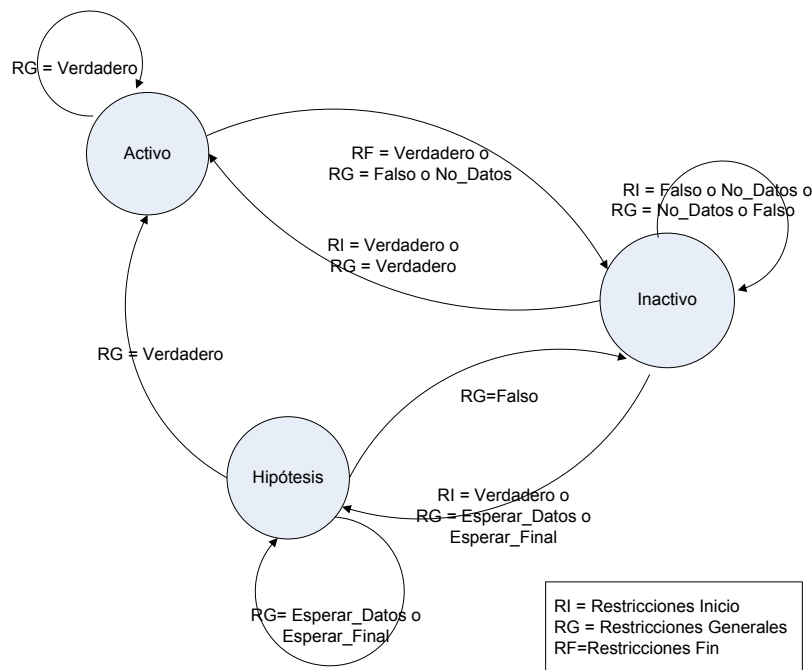


Figura 4.13: Diagrama de estados de la interpretación

Además de interpretar en que estado está cada Paso, se decide cuándo notificárselo al subsistema de diagnóstico. El momento en el que se produce la notificación depende de si el EspIPaso tiene definido su atributo restricciones de *ejecución*. En caso de que así sea, hay que empezar a evaluar las restricciones de *ejecución* cuando se crea una instancia de Paso y entra en estado "hipótesis" o "activo". Sólo cuando se cumplen se le notifica al Subsistema de Diagnóstico. Si el EspIPaso no tuviera definidas las restricciones de *ejecución*, el Subsistema de Diagnóstico sería informado en el momento de creación de la instancia del Paso, tanto en estado "hipótesis" como en "activo". Al Subsistema de Diagnóstico también se le informa del estado del Paso para que lo tenga en cuenta al diagnosticar. Finalmente,

cuando el Paso cambia al estado “inactivo” también se notifica para que finalice su diagnóstico.

Cada vez que se informa al Subsistema de Diagnóstico de la ejecución de un paso, éste se intenta relacionar con alguna Situación que esté “activa” o en estado “hipótesis”. Solo se relacionan con aquellas Situaciones que lo incluyan en el atributo *Pasos Posibles* de su EspISituacion. Si no hubiera ninguna Situación posible en alguno de esos estados, el Paso se asociaría a la “Situación General”. Una vez creados y asociados los Pasos y las Situaciones, el Subsistema de Diagnóstico sabe lo que está haciendo el alumno y el contexto en el que lo hace, con lo que procede a diagnosticar.

## 4.5 SUBSISTEMA DE DIAGNÓSTICO

El Subsistema de Diagnóstico es el encargado de detectar los errores que comete el alumno mientras realiza la tarea. No se especifica la arquitectura interna del subsistema, ya que se pretende flexibilizar la integración de varias técnicas de diagnóstico. En su lugar, se ha diseñado una plantilla que define cómo tiene que ser el proceso de diagnóstico para que encaje en la Metodología (Figura 4.14).

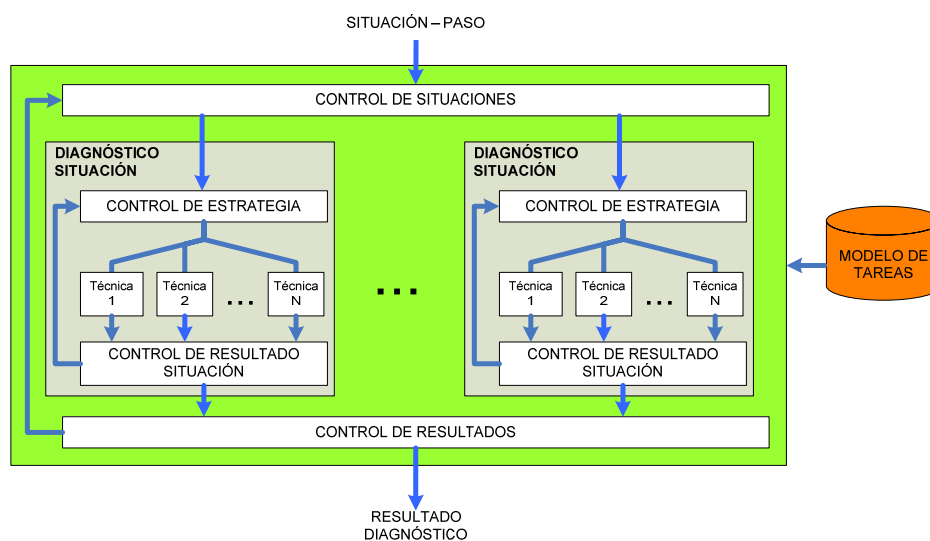


Figura 4.14: Plantilla de adaptación del Subsistema de Diagnóstico

La plantilla se ha diseñado para integrar componentes de diagnóstico siguiendo las pautas indicadas a continuación:

- Los Pasos se diagnostican en el contexto de la Situación o Situaciones en que se hayan producido, teniendo en cuenta que podrían producirse simultáneamente.
- Se permite el uso de varias técnicas de diagnóstico, para poder escoger las que mejor se adecuen a cada Situación.
- Se permite el uso de diferentes estrategias de diagnóstico. Las estrategias serán las que determinen qué técnicas de diagnóstico hay que utilizar en cada momento.

Estas pautas son visibles explícitamente en el proceso de diagnóstico general que define la plantilla. El proceso consta de las siguientes fases:

- 1) Recibir del Subsistema de Interpretación los Pasos que está ejecutando el alumno y las Situaciones a las que estén asociados que se estén produciendo.
- 2) Solicitar al Subsistema de Observación las Observaciones que hagan falta para diagnosticar cada Paso. Dichas Observaciones son las especificadas en el atributo *Observaciones* que tienen tanto el *EspDPaso* como la *EspDSituacion*. Una vez que comienza el proceso de diagnóstico del Paso se van actualizando sus valores hasta que finaliza.
- 3) El componente de Control de Situaciones decide qué componentes de Diagnóstico de Situación van a diagnosticar cada Paso. Se permite integrar varios para poder utilizar estrategias diferentes en función de la Situación o Situaciones a las que estén asociados. En caso de que varias de estas Situaciones se produjeran simultáneamente, cada Paso se diagnosticaría a la vez por varios componentes de Diagnóstico de Situación.
- 4) El componente de Control de Estrategia establece la estrategia de elección de técnicas más adecuada para diagnosticar cada Paso. Para tomar la decisión cuenta con las Observaciones asociadas al Paso y a la Situación.
- 5) El componente de Control de Estrategia decide mediante qué técnicas se va a diagnosticar cada Paso en función de la estrategia seleccionada. Por

ejemplo, podrían utilizarse varias técnicas secuencialmente o simultáneamente.

- 6) Se diagnostica el Paso con cada Técnica utilizando una Solución específica. Algunas técnicas posibles son el reconocimiento de planes (Genesereth, 1982; Kautz y Allen, 1986) o el modelado con restricciones (Ohlsson, 1994).
- 7) El componente de Control de Resultado por Situación decide qué se hace con los resultados de diagnóstico proporcionados por cada técnica. Por ejemplo, si una de las técnicas no ha sido capaz de ofrecer un resultado, habría que avisar al Control de Estrategia para que decida qué hacer: utilizar una técnica distinta, informar al alumno o al instructor... En cambio, si en vez de diagnosticar secuencialmente con cada técnica se diagnosticase con varias a la vez habría que fusionar los resultados.
- 8) El componente de Control de Resultados decide qué se hace con los resultados de cada Paso diagnosticado. Se podrían devolver todos, se podrían fusionar, se podría elegir uno solo o se podrían enviar al Control de Situación para volver a diagnosticar con la nueva información que proporcionan todos los resultados parciales.

Después de completar el proceso en cada ciclo, el Subsistema de Diagnóstico devuelve un Resultado por cada Paso diagnosticado. Dicho Resultado lo puede utilizar un Sistema de Ayuda al Aprendizaje para decidir sus propias acciones pedagógicas.

#### **4.5.1 Integración de DETECTive como Subsistema de Diagnóstico**

Tal y como se analizó en el Apartado 2.3 el sistema de diagnóstico DETECTive presenta características idóneas para diagnosticar la actividad procedimental independientemente del dominio, por lo que ha sido adaptado para formar parte del Marco de Desarrollo. Teniendo en cuenta las características propias de DETECTive, las fases del proceso de diagnóstico definidas por la plantilla se concretan de la siguiente manera (Figura 4.15):

- 1) Recepción del Paso: DETECTive no considera la ejecución simultánea de varios Pasos, por lo que cada vez que se recibe un Paso se diagnostica

de inmediato sin posibilidad de recibir otro a la vez. Además, todos los Pasos van asociados a la "Situación General", sin posibilidad de que concurren otras situaciones.

- 2) Solicitar Observaciones al Subsistema de Observación: Estas observaciones son utilizadas por DETECTive para almacenar los valores de los Parámetros del Paso, y para representar los objetos del Sistema Interactivo.
- 3) El componente de Control de Situaciones no tiene ninguna función, ya que todos los Pasos tienen que ir asociados a la "Situación General".
- 4) La estrategia de selección de técnicas es estática, por lo que esta fase no se aplica con DETECTive.

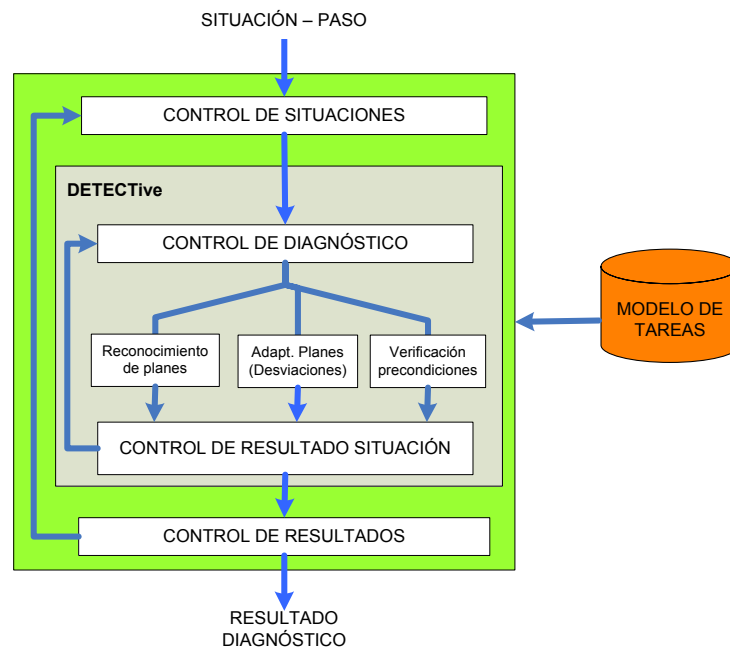


Figura 4.15: Adaptación de DETECTive a la plantilla del Subsistema de Diagnóstico

- 5) El componente de Control de Estrategia escoge la técnica con la que se va a diagnosticar el Paso siguiendo la estrategia siguiente: el diagnóstico de la tarea comienza realizando un *reconocimiento de planes*. Si el Paso no se ajusta a ninguno de los planes definidos en las Soluciones, se intenta *adaptar el plan* con alguna desviación. En caso de que tampoco así se

podiera diagnosticar, abandonaría el diagnóstico basado en planes e intentaría diagnosticar *verificando las precondiciones* generales del Paso en el dominio.

- 6) Se diagnostica el Paso con la Técnica decidida en la fase anterior. Para realizar el diagnóstico, cada técnica cuenta con las Soluciones del EspDPaso que necesite: un conjunto de planes en caso del *reconocimiento* y de la *adaptación de planes*, y un conjunto de restricciones lógicas en caso de la *verificación de precondiciones*.
- 7) El componente de Control de Resultado por Situación solo reenvía el resultado de diagnóstico al Control de Estrategia en caso de que el diagnóstico no haya sido posible. De esta forma, se puede intentar diagnosticar de nuevo con otra técnica. Si el diagnóstico ha sido posible, DETECTive devuelve el Resultado: CORRECTO, ERROR o NO-OPTIMO. En caso de que el Resultado sea distinto de CORRECTO, se incluye la información sobre el error cometido. Distingue las siguientes categorías de errores: Error-Preestablecido, cuando los parámetros del Paso no coinciden; Error-Desviación, cuando el alumno ha seguido alguno de los planes desviación; y Error-Violación de Condición, cuando no se cumplen las precondiciones del Paso.
- 8) El componente de Control de Resultados devuelve directamente el Resultado obtenido de la fase anterior.

## 4.6 CONCLUSIONES

En este capítulo se ha presentado el Marco de Desarrollo encargado de establecer la comunicación entre el Sistema Interactivo y el Sistema de Ayuda al Aprendizaje. El núcleo del Marco de Desarrollo lo forman los Subsistemas de Observación, Interpretación y Diagnóstico. Dichos subsistemas demuestran que los modelos generados a partir del metamodelo de integración son válidos para automatizar la comunicación mientras se está realizando la actividad.

El Subsistema de Observación se encarga de transformar los datos instantáneos del Sistema Interactivo en conocimiento temporal. Este conocimiento consiste en un conjunto de Observaciones que se van construyendo mientras se están produciendo. Además, el Subsistema de

Observación abstrae la arquitectura interna del Sistema Interactivo evitando tener que conocerla para poder construir el SI<sup>2</sup>A<sup>2</sup>.

El Subsistema de Interpretación demuestra que el Modelo de Interpretación sirve para determinar qué Pasos y Situaciones se están produciendo en el Sistema Interactivo a partir de un conjunto de Observaciones.

El Subsistema de Diagnóstico se encarga de determinar la corrección de las acciones del alumno y comunicar sus resultados al Sistema de Ayuda al Aprendizaje. El subsistema se adapta a cada SI<sup>2</sup>A<sup>2</sup> a partir de una plantilla que define cómo realizar el diagnóstico aplicando las estrategias que se consideren más oportunas en cada situación. Además, la plantilla permite escalar el subsistema encapsulando los componentes que se estimen necesarios para utilizar las técnicas de diagnóstico más adecuadas. Concretamente, se ha integrado con el componente de diagnóstico DETECTive, de forma que sus capacidades de diagnóstico son aprovechadas dentro del Marco de Desarrollo.

Para validar el funcionamiento del Marco de Desarrollo y los principios establecidos por la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> se ha realizado una serie de experimentos que se presentan en el siguiente capítulo.





## *CAPÍTULO 5*

# ***APLICACIONES DE LA METODOLOGÍA Y EL MARCO DE DESARROLLO***

---

En este capítulo se presentan las aplicaciones y experimentos realizados para validar la Metodología y el Marco de Desarrollo. La validación busca demostrar que a partir de los datos generados por los Sistemas Interactivos es posible generar Observaciones con las que interpretar los Pasos y Situaciones que se quieren diagnosticar. En primer lugar se describen los criterios de validación, y posteriormente se presentan las aplicaciones realizadas. Por una parte se ha escogido el dominio de los procesos de fabricación con máquina herramienta, y por otra, el de la conducción de vehículos profesionales.

### **5.1 CRITERIOS DE VALIDACIÓN DE LA METODOLOGÍA Y EL MARCO DE DESARROLLO**

A la hora de definir el criterio de validación se ha tenido en cuenta la problemática analizada en el capítulo 2. En dicho capítulo se estudiaban las características más relevantes de los Sistemas de Ayuda al Aprendizaje y de los Sistemas Interactivos con el objetivo de generalizar la construcción de  $SI^2A^2$  independientemente del dominio. Basándonos en ese estudio, la validación ha consistido en realizar experimentos construyendo  $SI^2A^2$  que muestran diferentes características: dominios de actividad continua e

instantánea; sistemas interactivos realistas y simplificados; y entornos autónomos y reactivos.

Los experimentos realizados ponen de manifiesto la validez de la Metodología propuesta para construir SI<sup>2</sup>A<sup>2</sup> que exhiban cualquiera de estas características. El primer dominio trata el aprendizaje de procesos de fabricación mediante máquina herramienta. El SI<sup>2</sup>A<sup>2</sup> construido para este dominio es un sistema simplificado y reactivo en el que se realizan tanto acciones continuas como instantáneas. El segundo dominio es el de la conducción de vehículos. En este caso el SI<sup>2</sup>A<sup>2</sup> es un sistema altamente realista, en el que las acciones son mayoritariamente continuas y además están influidas por vehículos autónomos.

En la siguiente sección se describe la construcción del SI<sup>2</sup>A<sup>2</sup> de procesos de fabricación por máquina herramienta y posteriormente el SI<sup>2</sup>A<sup>2</sup> de conducción de vehículos.

## **5.2 APLICACIÓN DE LA METODOLOGÍA EN EL DOMINIO DE LA MÁQUINA HERRAMIENTA**

En esta sección se describe el SI<sup>2</sup>A<sup>2</sup> construido para el aprendizaje de procesos de máquina herramienta. La sección comienza con la descripción general del dominio de la máquina herramienta, ilustrado mediante un ejemplo de tarea de mecanizado con fresadora. Posteriormente se presenta VIRTOOL (Ustarroz et al., 2004), el SI<sup>2</sup>A<sup>2</sup> para aprendizaje de procesos de máquina herramienta desarrollado siguiendo la Metodología presentada en el capítulo 3. Los demás apartados describen cómo se ha aplicado la Metodología utilizando el mismo ejemplo de mecanizado con fresadora. Se explica cómo se extraen Observaciones a partir de los datos de entrada del Sistema Interactivo, y cómo se interpretan los Pasos realizados por el alumno para diagnosticarlos.

### **5.2.1 Tareas de mecanizado con máquina herramienta**

Una máquina herramienta está destinada a moldear o transformar un material sólido, principalmente metales, para conseguir un producto con la forma deseada. Existen múltiples técnicas de fabricación, aunque en este contexto nos referiremos exclusivamente a las máquinas herramienta que trabajan por arranque de viruta. Esta técnica consiste en utilizar

herramientas de corte con las que se van eliminando partes de material a una pieza en bruto hasta darle la forma final deseada.

Existen distintos tipos de máquina herramienta por arranque de viruta en función del movimiento de corte que realizan (Figura 5.1) (Mateos y Mateos, 1999). Los movimientos pueden ser de desplazamiento o de rotación, tanto de la pieza como de la herramienta. Independientemente de cuáles sean los movimientos, cuando la herramienta colisiona con la pieza se produce el arranque de la viruta. Por ejemplo, las fresadoras rotan la herramienta a altas revoluciones mientras la pieza se desplaza; los tornos rotan la pieza y desplazan la herramienta; y otras máquinas como las cepilladoras y las limadoras trabajan por traslación de la pieza y de la herramienta respectivamente.



Figura 5.1: Tornos y fresadoras convencionales y CNC

Además de por el movimiento de corte, las máquinas herramienta se clasifican en máquinas convencionales y de control numérico (CNC). Las convencionales se controlan manualmente, mediante palancas y manivelas. En cambio, las máquinas de control numérico integran un computador con el que se programan los movimientos de mecanizado para que se ejecuten automáticamente.

Para explicar en qué consiste un proceso de mecanizado se va a utilizar como ejemplo la fabricación de una pieza con una fresadora vertical

convencional (Figura 5.2). La fresadora utilizada en el ejemplo permite desplazar la pieza en los ejes X, Y, Z por el movimiento de los carros transversal, longitudinal y vertical. Además, el cabezal donde se monta la herramienta se puede desplazar verticalmente, tal y como indican las flechas verdes de la figura.

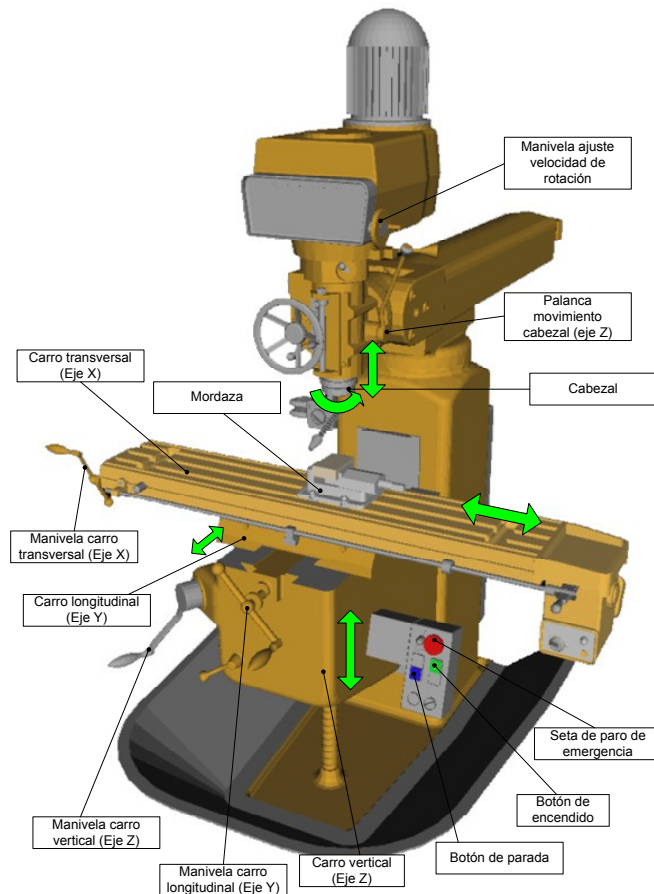


Figura 5.2: Fresadora vertical convencional

En la Figura 5.3 se muestra el plano de la pieza que se quiere fabricar con la fresadora, su representación en tres dimensiones y la pieza de material en bruto. El proceso para fabricarla implica una planificación paso a paso de las operaciones que hay que llevar a cabo (Figura 5.4). En el ejemplo se muestran los siguientes grupos de operaciones:

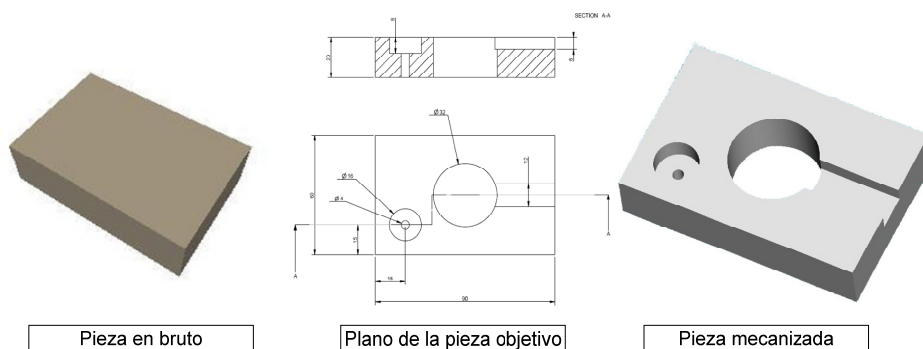


Figura 5.3: Ejemplo de pieza mecanizada

- Preparación de la pieza en bruto: Consiste en colocar la mordaza sobre la mesa y amarrar con ella la pieza en bruto.
- Preparación de la herramienta: Consiste en montar y apretar la herramienta a un portaherramientas compatible, ajustando el voladizo<sup>14</sup> al mínimo posible. Posteriormente se coloca el conjunto en el cabezal de la fresadora.
- Mecanizar: Existen diversas operaciones de mecanizado en función de la herramienta utilizada y la forma de penetrar en la pieza. En cada operación se verifica una serie de parámetros que tienen que ser adecuados para que el mecanizado se realice correctamente, sin deformar ni romper la herramienta o la pieza, y de forma que la pieza resultante coincida con el plano objetivo. Algunos de los parámetros que se tienen en cuenta son: la velocidad de rotación de la herramienta, la velocidad de avance<sup>15</sup>, y la profundidad de pasada<sup>16</sup>. Para conseguir la pieza del ejemplo se deben ejecutar las siguientes operaciones de mecanizado:
  - Planear: Quitar una capa fina de material para alisar la superficie de la pieza en bruto.
  - Taladrar: Agujerear la pieza con una broca.
  - Realizar punto de centrado: marcar con una broca fina los puntos donde se va a taladrar.

<sup>14</sup> Voladizo: longitud de la herramienta que sobresale del portaherramientas.

<sup>15</sup> Velocidad de avance: velocidad relativa entre pieza y herramienta.

<sup>16</sup> Profundidad de pasada: profundidad de la capa de material arrancada.

- Ranurar: Realizar una ranura o canal en la pieza.
- Desmontaje de la herramienta: Consiste en soltar y desmontar la herramienta del portaherramientas y el portaherramientas de la fresadora.

Realizando los Pasos en el orden indicado en el plan (Figura 5.4) se llega a fabricar la pieza objetivo siempre que se ajusten los parámetros adecuadamente. A la vista de las características de los procesos que se abordan en este dominio, la construcción de un SI<sup>2</sup>A<sup>2</sup> para su aprendizaje implica definir tareas que se puedan diagnosticar utilizando la técnica de reconocimiento de planes y verificando los parámetros de cada Paso. Dado que DETECTive puede diagnosticar de esta manera, ha entrado a formar parte de VIRTOOL, que se describe en el siguiente apartado.

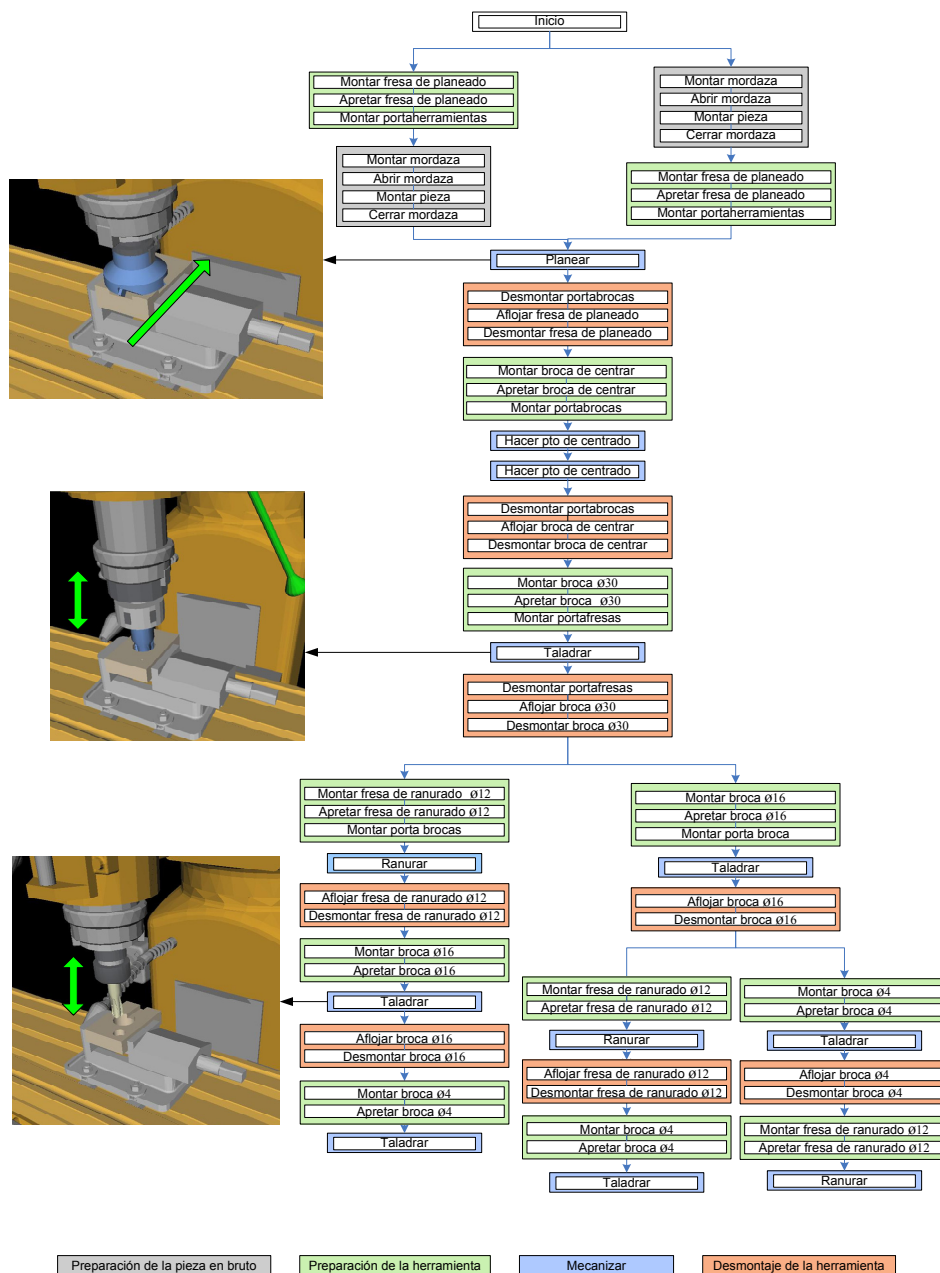


Figura 5.4: Ejemplo de mecanizado con fresadora



### **5.2.2 Descripción de VIRTOOL**

En el contexto del proyecto europeo VIRTOOL (CRAFT-1999-70292) (Lozano et al., 2004) se diseñó y desarrolló un Sistema de Ayuda al Aprendizaje de procesos de máquina herramienta (convencional y de control numérico) utilizando gráficos interactivos 3D y técnicas de Realidad Virtual. El software consta de cuatro aplicaciones: Virtool-Machine, Virtool-Master, Virtool-Generic y Virtool-Case Generator. Las tres últimas se utilizan en una fase de preproceso para crear nuevas máquinas herramienta, las librerías de elementos que se utilizan con ellas (herramientas, portaherramientas, etc.) y las tareas que tiene que resolver el alumno.

Virtool-Machine (Figura 5.5) utiliza los datos generados en la fase de preproceso para proponer al alumno Tareas de mecanizado con alguna máquina herramienta. Cada Tarea incluye el plano de la pieza objetivo que tiene que fabricar el alumno, la máquina herramienta y la librería de accesorios necesaria. Adicionalmente, se asocia un conjunto de recursos didácticos en forma de material multimedia para ayudar al alumno cuando lo necesite. Se le pueden proponer varios tipos de Tarea:

- Tarea guiada: Virtool-Machine le indica al alumno qué tiene que hacer paso a paso.
- Tarea evaluada: Se evalúa cada paso del alumno y se le indican sus errores. En caso de errores generales la comunicación se realiza mediante mensajes de texto, mientras que en otros más específicos se simulan las consecuencias del error. Por ejemplo, si el alumno deja la herramienta con demasiado voladizo, al poner la máquina en marcha se observa y se escucha la vibración de la herramienta.
- Tarea libre: Los errores no se le comunican al alumno mientras realiza la tarea salvo que sean demasiado graves como para continuar la simulación, y al finalizar se le muestra un informe con la evaluación de su solución.

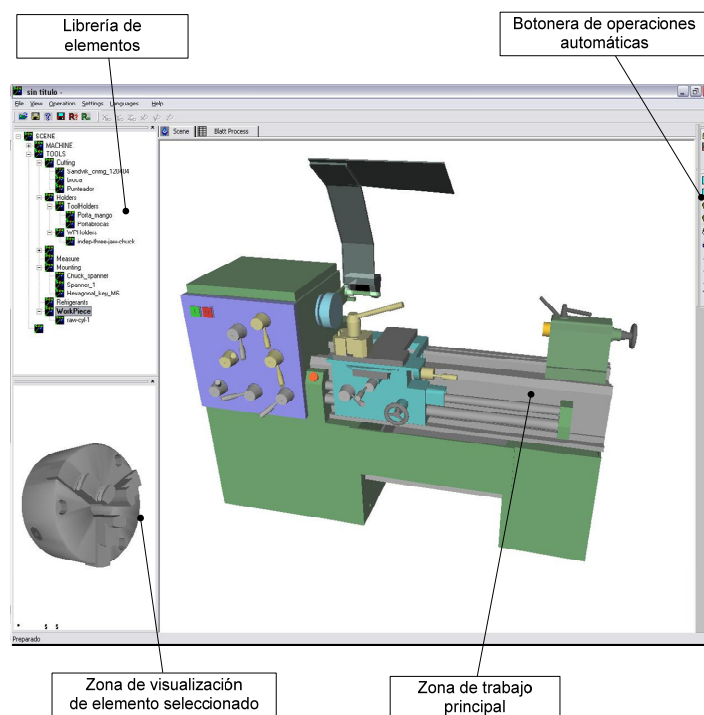


Figura 5.5: Interfaz de Virtool-Machine

El alumno utiliza una combinación de métodos de interacción 2D y 3D para resolver las Tareas de mecanizado con Virtool-Machine. Además, la forma de interactuar con Virtool-Machine no cambia en función de la máquina herramienta. Por esta razón, en los siguientes apartados se utilizará el ejemplo de mecanizado con fresadora descrito en el apartado anterior para explicar qué observaciones se extraen, cómo se interpretan para determinar qué pasos está realizando y cuándo se hace. Específicamente se presenta el proceso de extracción de observaciones y la interpretación de pasos que engloban las operaciones más representativas en Virtool-Machine: el montaje de herramientas y el mecanizado de la pieza.

### 5.2.3 Montaje de herramienta: Extracción de Observaciones e Interpretación de Pasos

Montar una herramienta en un portaherramientas implica realizar varias acciones en la interfaz de Virtool-Machine. Cada una de esas acciones produce cambios que es necesario observar para interpretar exactamente

cuál es el paso que está realizando. Para explicar el proceso, en primer lugar se van a enumerar las acciones que realiza el alumno para montar una fresa de ranurado en un portaherramientas. Después se describen las Observaciones que se van extrayendo, y finalmente se presentan las Restricciones con las que interpretar los Pasos.

#### ➤ Acciones en Virtool-Machine

Una operación de montaje de herramienta implica una serie de acciones sobre varios elementos de la interfaz. En la Figura 5.6 se indican numeradas todas las interacciones del alumno con cada elemento de la interfaz y se describen a continuación:

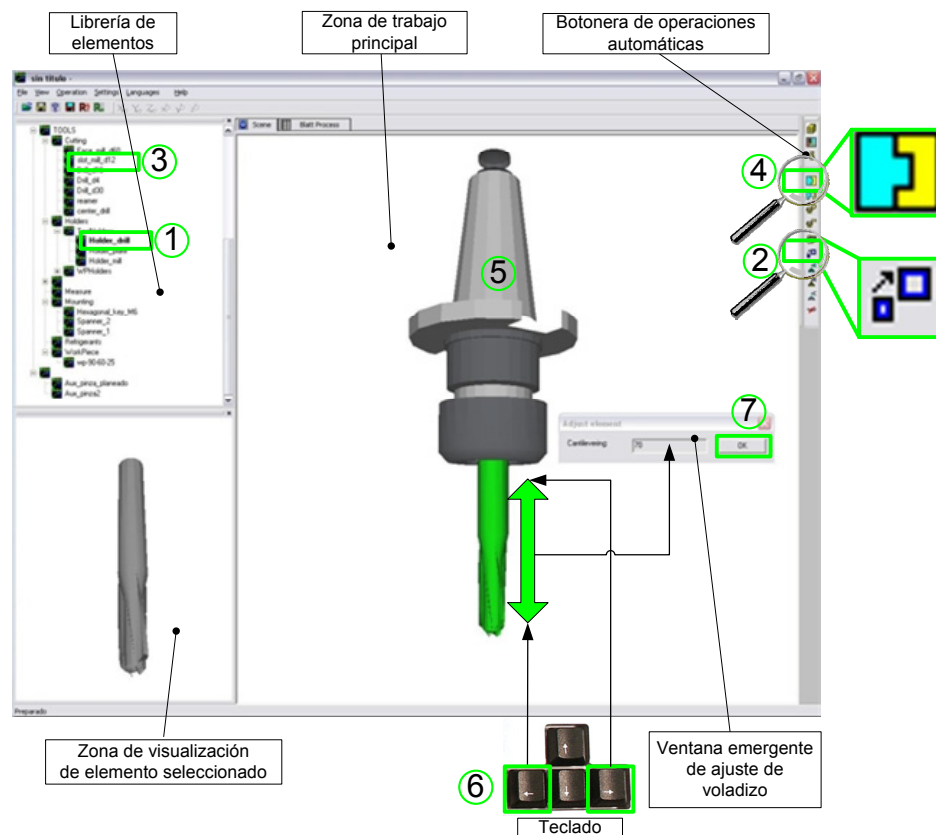


Figura 5.6: Acciones para montar una herramienta

- 1) Seleccionar portaherramientas: Hacer clic sobre el elemento "Portaherramientas" en la librería de elementos. El elemento seleccionado se muestra en la "Zona de visualización de elemento".
- 2) Zoom portaherramientas: Al iniciar Virtool-Machine la "Zona de trabajo principal" muestra la fresadora. Para realizar el montaje de la herramienta es necesario que en lugar de la fresadora se muestre el portaherramientas. Para ello, se pulsa el botón "Zoom elemento seleccionado" de la botonera de operaciones.
- 3) Seleccionar herramienta: Hacer clic sobre el elemento "Fresa de ranurar" en la librería de elementos. La fresa se muestra en la "Zona de visualización de elemento".
- 4) Montar: Pulsar el botón de operación de montaje en la botonera de operaciones.
- 5) Clic portaherramientas: Hacer clic sobre el portaherramientas en la "Zona de trabajo principal". Mediante esta acción se indica que la fresa de ranurar seleccionada mediante la acción 3 se quiere montar (acción 5) en el portaherramientas sobre el que se ha hecho clic. La herramienta se coloca automáticamente en la posición de montaje correcta a falta de ajustar el voladizo.
- 6) Cambiar voladizo: Mediante las flechas izquierda y derecha del teclado la herramienta sale o entra en el portaherramientas dejando más o menos voladizo. El valor exacto del voladizo se muestra en la ventana emergente de "Ajuste de voladizo".
- 7) Bloquear voladizo: Una vez conseguido el voladizo deseado, hay que pulsar el botón OK en la ventana emergente de "Ajuste de voladizo" para no cambiarlo más.

#### ➤ Extracción de Observaciones

Las Observaciones que se consiguen mientras el alumno interactúa con Virtool-Machine proceden de varias fuentes: la librería de elementos, la zona de trabajo principal, la botonera de operaciones y el estado de la simulación (Figura 5.5). En este contexto interesan las siguientes observaciones que se especifican a continuación junto con sus propiedades:

- *Elemento de Librería Seleccionado (ELS)*: Cuando algún elemento de la librería está seleccionado se puede detectar esta observación que incluye las siguientes propiedades:
  - Nombre: del elemento seleccionado. Por ejemplo: “Broca d16mm”
  - Tipo: de elemento. Por ejemplo, “Herramienta de corte”, “Herramienta de montaje”, “Pieza”, “Portaherramientas”...
- *Elemento de Zona principal Seleccionado (Ezs)*: Cuando se hace clic sobre algún elemento de la Zona de trabajo principal”, éste queda seleccionado y se puede detectar esta observación. Generalmente el elemento seleccionado suele ser alguna parte de la máquina herramienta, como alguna manivela. En este caso la observación contiene una única propiedad:
  - Nombre: del elemento seleccionado.
- *Herramienta Activa (HA)*: Esta observación se puede detectar cuando la herramienta que se va a utilizar para mecanizar está montada en el portaherramientas. Las propiedades de esta observación que interesan para la operación de montaje son:
  - Nombre: de la herramienta.
  - Voladizo: longitud de la herramienta que sobresale del portaherramientas.
- *Operación*: Esta observación se detecta cuando el alumno ha hecho clic en algún botón de la “Botonera de operaciones automáticas”. Contiene la siguiente propiedad:
  - Nombre: de la operación. Algunas operaciones automáticas posibles son: “Montar”, “Desmontar”, “Apretar”, “Aflojar”...
- *Voladizo Bloqueado (VB)*: Esta observación se detecta cuando el alumno pulsa el botón “OK” en la ventana emergente “Ajuste de voladizo”. En este caso la observación no contiene propiedades.

### ➤ Interpretación de Pasos

Mientras el alumno realiza todas las acciones para montar la herramienta, las observaciones aparecen y desaparecen, y sus propiedades cambian de valor. La conjunción de Observaciones da lugar a la interpretación de dos pasos: “Montar herramienta de corte” y “Ajustar voladizo”. Para poder interpretarlos definimos dos EspIPaso (Figura 5.7).

Ambos pasos son instantáneos, por lo que no tiene sentido definir otras restricciones aparte de las generales. Para especificar el paso “Montar herramienta de corte” se definen dos Restricciones Lógicas, que combinan dos Restricciones sobre Propiedad y una RIntervalo que contiene una EspObservacion. El paso “Ajustar Voladizo”, por su parte, únicamente requiere extraer la Observación “Voladizo bloqueado”, lo que se representa mediante una RIntervalo.

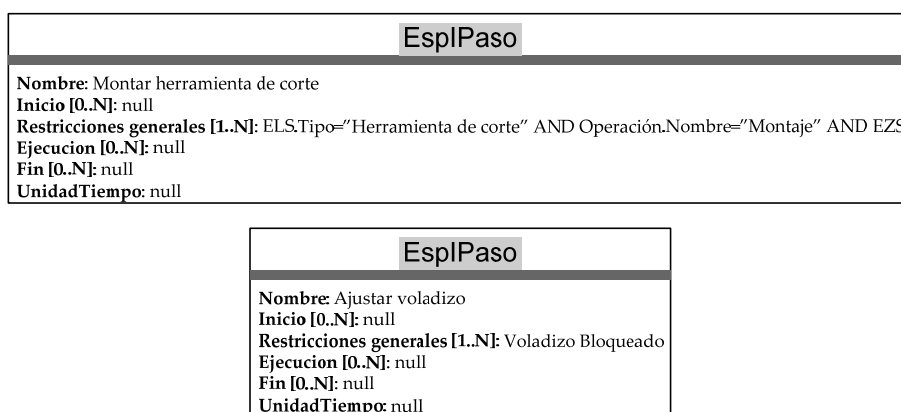


Figura 5.7: Especificación de los pasos: “Montar herramienta de corte” y “Ajustar voladizo”

Estos pasos no son interpretables desde la primera acción del alumno. En la Tabla 5.1 se muestra el proceso completo de interpretación siguiendo el mismo orden de ejecución de acciones que se indicó al comienzo de esta sección. En las filas se registra qué Observaciones son detectadas al ejecutar cada acción, indicando “No Observado” cuando la Observación no es detectable, o el valor de sus propiedades en caso contrario. En la última columna se muestra el nombre del paso que se haya podido interpretar a partir de las Observaciones disponibles tras cada acción.

La tabla muestra que en las cuatro primeras acciones no se interpreta ningún Paso. Al ejecutar la acción 5 “clic en el portaherramientas” aparece la Observación “Elemento de Zona principal Seleccionado” (EZS). Como además el “Elemento de Librería Seleccionado” (ELS) es de tipo “Herramienta de corte” y el nombre de la Operación activa es “Montar”, se cumplen las Restricciones Generales del paso “Montar herramienta de corte”, y por tanto, es interpretado. De igual manera, al ejecutar la acción 7 (pulsar el botón “OK” de la ventana emergente “Ajuste de voladizo”) se

extrae la Observación “Voladizo Bloqueado”. Por tanto, se cumple la Restricción para interpretar el paso “Ajustar voladizo”. El diagnóstico del paso utiliza la propiedad “Voladizo” de la Observación “Herramienta activa” para comprobar si el valor del voladizo está dentro de límites aceptables.

ACCIÓN	OBSERVACIONES					INTER- PRETA- CIÓN
	ELS (N:Nombre T:Tipo)	EZS (N:Nombre T:Tipo)	HA (N:Nombre)	Operación (N:Nombre)	VB	
1 Seleccionar portahtas	N: Portabrocas T: Portaherramientas	No Observado	No Observado	No Observado	No Observado	
2 Zoom portahtas	N: Portabrocas T: Portaherramientas	No Observado	No Observado	No Observado	No Observado	
3 Seleccionar hta.	N: Fresa de ranurar T: Herramienta de corte	No Observado	No Observado	No Observado	No Observado	
4 Montar	N: Fresa de ranurar T: Herramienta de corte	No Observado	No Observado	N: Montaje	No Observado	
5 Clic Portahtas	N: Fresa de ranurar T: Herramienta de corte	N: Portabrocas T: Portaherramientas	No Observado	N: Montaje	No Observado	Montar herramienta de corte
6 Cambiar voladizo	N: Fresa de ranurar T: Herramienta de corte	No Observado	N: Fresa de ranurar Voladizo: Null	No Observado	No Observado	
7 Bloquear voladizo	N: Fresa de ranurar T: Herramienta de corte	No Observado	N: Fresa de ranurar Voladizo: 70	No Observado	Observado	Ajustar voladizo

**Abreviaturas:**

ELS: Elemento de Librería Seleccionado; EZS: Elemento de Zona Principal Seleccionado; HA: Herramienta Activa; VB: Voladizo Bloqueado

Tabla 5.1: Evolución de observaciones durante una operación de montaje.

El resto de operaciones automáticas que se pueden realizar con Virtool-Machine son muy similares al montaje de la herramienta. También se interpretan pasos instantáneos mediante Restricciones sobre Propiedad de las observaciones “Elemento de librería Seleccionado” y “Operación”. Por lo general se interpretan pasos distintos creando Restricciones a partir de las mismas Observaciones. Por ejemplo, para interpretar el paso “Apretar herramienta” se especifica que la propiedad “Nombre” de la Observación “Operación” tiene que ser “Apretar”.

Además de estos ejemplos de pasos instantáneos, en Virtool-Machine también se realizan pasos continuos. En la siguiente sección se presenta el caso concreto del paso continuo: “Ranurar”.

### **5.2.4 Ranurado de una pieza: Extracción de Observaciones e Interpretación de Pasos**

En esta sección se describen los procesos de extracción de observaciones y de interpretación asociados a un paso continuo: "Ranurar". Este paso consiste en mecanizar un canal o ranura en la pieza. El ranurado es un paso continuo al igual que el resto de operaciones de mecanizado, y dura todo el intervalo de tiempo en el que la herramienta colisiona con la pieza estando en marcha la máquina. En primer lugar se van a enumerar las acciones que realiza el alumno para ranurar. Después se describen las observaciones que se extraen durante el proceso y finalmente se presentan las restricciones con las que interpretar los pasos.

#### ➤ **Acciones en Virtool-Machine**

Una operación de ranurado requiere tener montada en la fresadora una fresa de ranurar con el portaherramientas adecuado. Para explicar las acciones que implica ranurar se dará por supuesto que la máquina ya ha sido preparada siguiendo el plan de ejemplo que se ha venido utilizando en este capítulo. En la Figura 5.8 se muestran numeradas las interacciones del alumno con la interfaz y se describen a continuación:

- 1) Mover manivelas en X, Y, Z hasta colocar la herramienta cerca del punto (X,Y) donde hay que empezar a ranurar. Ajustar la posición del carro vertical (Z) para que la profundidad de la ranura corresponda exactamente con la planeada. Mover una manivela implica dos acciones: seleccionar la manivela deseada haciendo clic sobre ella y utilizar las flechas izquierda y derecha del teclado para mover la manivela en el sentido deseado.
- 2) Clic botón de encendido de la fresadora: Al pulsarlo comienza a rotar la herramienta. Si la herramienta no se hubiera apretado se soltaría del portaherramientas y finalizaría el ejercicio. En caso de que el voladizo fuera excesivo la herramienta vibraría pero se podría continuar la tarea.
- 3) Seleccionar la manivela de ajuste de velocidad de rotación de herramienta.
- 4) Utilizar las flechas izquierda y derecha del teclado para ajustar la velocidad. Cada pulsación supone una variación de 23 rpm (revoluciones por minuto).



- 5) Mover la manivela del carro Y para que la pieza avance hacia la herramienta y ésta vaya creando la ranura. Hay que terminar la ranura al llegar al agujero del centro de la pieza, tal y como se observa en la figura.
- 6) Mover la manivela del carro vertical (eje Z) para sacar la herramienta de la pieza.

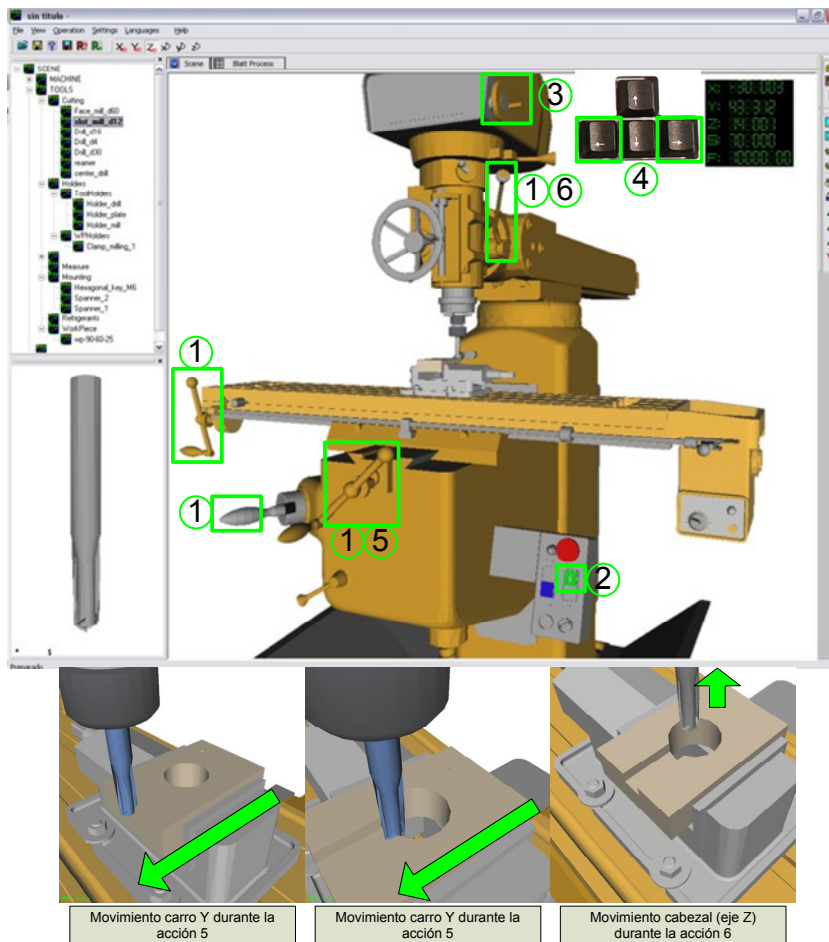


Figura 5.8: Acciones para ranurar

### ➤ Extracción de Observaciones

Las Observaciones que se extraen durante el mecanizado de una pieza proceden principalmente del estado de la fresadora y de la interacción entre la herramienta y la pieza. Las Observaciones necesarias son:

- *Herramienta Activa*: Esta Observación se detecta cuando hay una herramienta montada para mecanizar. A las propiedades “Nombre” y “Voladizo” descritas en el apartado anterior hay que añadir las siguientes:
  - Posición: Coordenadas X, Y, Z de la herramienta respecto del cero pieza<sup>17</sup>.
  - Tipo: Categoría de herramienta de corte. Por ejemplo, “fresa de ranurar”, “broca” o “fresa de planear”.
- *Fresadora*: Esta Observación se detecta en todo momento, y sirve para representar el estado de la máquina. En el contexto de la operación de ranurado interesan las siguientes propiedades:
  - Encendido: Indica si la máquina está encendida o apagada, por lo que los valores posibles de esta propiedad son “Verdadero” o “Falso”.
  - Velocidad de rotación: Velocidad a la que gira la herramienta cuando la fresadora está encendida.
- *Desplazamiento*: Esta Observación se puede detectar cuando alguno de los carros o el cabezal de la herramienta se están moviendo y representa el hecho de que exista un desplazamiento relativo entre la pieza y la herramienta. Por tanto, para poder detectarla es necesario que tanto la pieza como la herramienta estén montadas. La propiedad que interesa de esta Observación es la siguiente:
  - Dirección: indica el eje de desplazamiento (X, Y o Z), teniendo en cuenta que no se puede mover más de un carro simultáneamente.
- *Colisión*: Esta Observación se puede detectar cuando la pieza y la herramienta están colisionando.

---

<sup>17</sup> Cero pieza: Origen de coordenadas de la pieza establecido por el operario antes de mecanizar que representa el punto de referencia para medir los desplazamientos relativos entre herramienta y pieza.

- Profundidad: profundidad de material que se está arrancando de la pieza.

### ➤ Interpretación de Pasos

Cualquier operación de mecanizado se interpreta a partir del paso básico “Mecanizar”, es decir, arrancar material de la pieza. La operación específica (ranurar, planear, taladrar...) se especifica por la forma de mecanizar, más concretamente, por la herramienta utilizada y la dirección de mecanizado. Por tanto, para interpretar un ranurado se definen dos EspIPaso: “Mecanizar” y “Ranurar” (Figura 5.9).

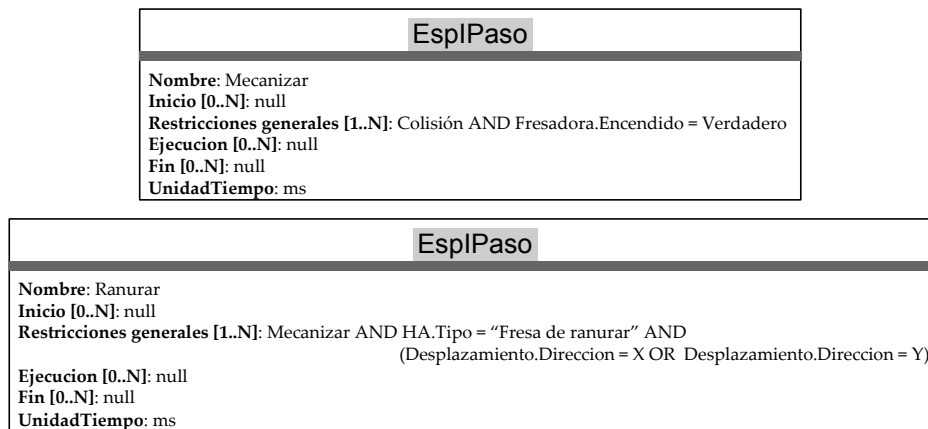


Figura 5.9: Especificación de los pasos: “Mecanizar” y “Ranurar”

Basar la especificación de los pasos de mecanizado en el paso “Mecanizar” tiene dos ventajas principales. La primera y más obvia es la reutilización del EspIPaso, ya que se evita replicar sus Restricciones en todos los pasos de mecanizado específicos. La segunda ventaja queda patente al intentar interpretar un paso que el alumno ha hecho mal. Por ejemplo, supongamos que el alumno intenta hacer una ranura con una broca de taladrar. En este caso no estaría claro si el alumno se ha equivocado de herramienta o de eje de movimiento, pero lo cierto es que está mecanizando y su paso no se corresponde con ningún mecanizado específico. No está ranurando porque la herramienta no es la adecuada, pero tampoco está taladrando porque el movimiento no es vertical. Interpretando el paso “Mecanizar” se permite que el subsistema de diagnóstico determine el error: si el plan indica que el paso correcto era

“Ranurar” significará que el alumno ha elegido mal la herramienta. Si el paso correcto hubiera sido “Taladrar”, se habría equivocado de manivela y estaría desplazando el carro X o el Y en lugar del Z.

En la Figura 5.10 se muestra un cronograma con la evolución de las Observaciones mientras el alumno ejecuta las acciones enumeradas al principio de este apartado. En el eje de ordenadas aparecen las Observaciones, representándose además en color azul los intervalos correspondientes a los pasos interpretados a partir de las mismas. Puesto que la duración de las Observaciones no es relevante la escala temporal no se especifica. Lo importante del cronograma son las relaciones entre las acciones del alumno y las Observaciones, y cómo se interpretan los pasos al verificarse las Restricciones. En el interior de las Observaciones se muestra el valor de sus propiedades durante el intervalo.

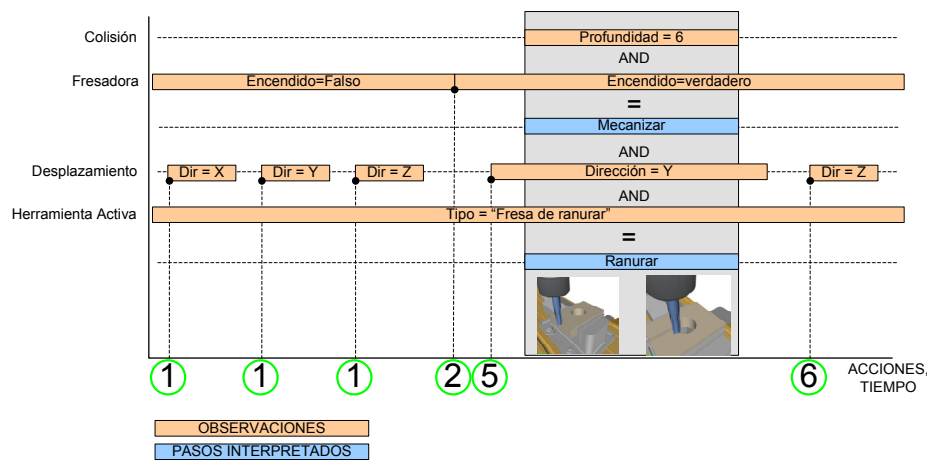


Figura 5.10: Evolución de Observaciones durante una operación de ranurado

La primera acción consiste en ubicar la pieza en el punto donde tiene que empezar a penetrar la herramienta para ranurar. Para ello, el alumno interactúa con las manivelas de la fresadora que mueven los tres carros (Acción 1), extrayéndose las Observaciones de “Desplazamiento” en cada eje. En el cronograma se realiza en el orden X, Y, Z, aunque podría haberse realizado en cualquier otro. Cuando el alumno acciona el botón de encendido de la fresadora (Acción 2), la propiedad “Encendido” de la Observación “Fresadora” cambia de valor. Después se realizan las acciones 3 y 4 con las que se ajusta la velocidad de rotación de la herramienta, pero como no aportan nada a la interpretación no se incluyen en el cronograma.

Posteriormente el alumno mueve la manivela del eje Y (Acción 5) para dirigir la pieza hacia la herramienta en la dirección que desea realizar la ranura. Sin embargo, no se interpreta nada hasta que la herramienta colisiona con la pieza. La zona marcada en color gris corresponde con el intervalo de tiempo en el que se cumplen las Restricciones de los pasos "Mecanizar" y "Ranurar". Haciendo una lectura vertical, comprobamos que "Colisión" AND "Fresadora.Encendido=Verdadero" dan lugar a la interpretación del paso "Mecanizar". Además, "Mecanizar" AND "Desplazamiento.Dirección=Y" AND "HerramientaActiva.Tipo=Fresa de ranurar" dan lugar a la interpretación del paso "Ranurar". Como el paso "Ranurar" es más específico que "Mecanizar", el subsistema de interpretación determina que "Ranurar" es el paso que tiene que ser diagnosticado. Finalmente, la acción 6 "Mover la manivela del carro vertical" consigue alejar la pieza de la herramienta.

### **5.2.5 Conclusiones**

La aplicación de la Metodología en VIRTOOL demuestra que es válida para construir SI<sup>2</sup>A<sup>2</sup> de diferentes características. En primer lugar, se ha verificado que se pueden emplear Sistemas Interactivos simplificados como Virtool-Machine. Además, mediante los dos ejemplos descritos se ha mostrado que en un mismo SI<sup>2</sup>A<sup>2</sup> se pueden realizar tanto pasos continuos como instantáneos, lo que implica que la Metodología tiene que considerar ambos ineludiblemente. Siguiendo las pautas de observación, interpretación y diagnóstico que marca la Metodología se ha demostrado que los pasos continuos y los instantáneos se pueden tratar de forma homogénea satisfactoriamente. Así, en ambos ejemplos se ha partido de un conjunto de Observaciones extraídas de Virtool-Machine y se han empleado Restricciones sobre Propiedades y Restricciones Lógicas para interpretarlos, dando lugar a Pasos que son diagnosticados independientemente de su duración. Los Pasos interpretados se asocian siempre a la "Situación General", ya que en Virtool-Machine no aparece ningún elemento autónomo que condicione los Pasos del alumno para resolver las Tareas. En este sentido, también queda demostrado que la Metodología es apta para entornos reactivos.

El dominio de la máquina herramienta no es suficientemente completo para validar el resto de aportaciones de la Metodología. Por esta razón se ha experimentado en el dominio de la conducción para demostrar:

la validez de la Metodología en entornos autónomos, su compatibilidad con Sistemas Interactivos realistas, su capacidad para tratar acciones simultáneas y la idoneidad de las Restricciones Temporales en el Nivel de Interpretación. Por último, y no menos importante, al emplear la Metodología en el dominio de la conducción queda demostrada su independencia del dominio de aprendizaje, de la arquitectura del Sistema Interactivo y del Sistema de Ayuda al Aprendizaje.

### **5.3 APLICACIÓN DE LA METODOLOGÍA EN EL DOMINIO DE LA CONDUCCIÓN**

En esta sección se presentan los experimentos realizados en el dominio de la conducción para validar la Metodología. En primer lugar se describen las características del dominio y se enfocan los experimentos, que se centran en las maniobras de cambio de carril y de adelantamiento. Posteriormente se describe cada uno de los experimentos y se finaliza resumiendo las conclusiones extraídas de las pruebas en este dominio.

#### **5.3.1 El dominio de la conducción**

Conducir un vehículo es una tarea que requiere destrezas diferentes para desempeñarla correctamente. Michon (1985) identifica tres clases de procesos que hacen falta para conducir:

- los procesos operativos para manipular los controles del vehículo.
- los procesos tácticos determinan la interacción con el entorno y con los demás vehículos.
- los procesos de estrategia corresponden a otros razonamientos y planificaciones de más alto nivel.

Conducir implica utilizar conjuntamente las tres clases de procesos en la mayoría de situaciones.

La gran cantidad de factores que implican estos procesos es una de las causas por las que formalizar computacionalmente de forma realista el dominio de la conducción es muy complicado. Existen aproximaciones desde distintos ámbitos de investigación, como la simulación de vehículos autónomos (Al-Shihabi y Mourant, 2001; Reece y Shafer, 1991), el estudio del comportamiento del conductor (Oliver y Pentland, 2000; Zou y

Levinson, 2006), el reconocimiento de maniobras en dispositivos embarcados (Oliver y Pentland, 2000), la monitorización inteligente de tráfico (Heintz et al., 2007) y también la formación de conductores (Weevers et al., 2003). En general, en cada trabajo se persigue modelar alguna característica concreta de la conducción. En algunos casos se centran en la síntesis de los procesos operativos que mantienen al vehículo en la vía, mientras que en otros buscan modelar o reconocer situaciones entre varios vehículos.

En general, suele ser objeto de interés el estudio de las maniobras de cambio de carril y de adelantamiento (Nigro y Rombaut, 2003; Salvucci, 2006). Estas maniobras también son adecuadas para demostrar las características de la Metodología que no se han cubierto en el dominio de la máquina herramienta. El cambio de carril es un paso continuo, y para interpretarlo se necesitan Restricciones Temporales. El adelantamiento es una Situación compleja en la que intervienen muchos factores y que implica ejecutar varios Pasos simultáneamente en función de la interacción con otros vehículos autónomos. Además, se ha empleado un simulador de conducción de camiones altamente realista para realizar las pruebas, con lo que la validez de la Metodología para construir SP<sup>2</sup>A<sup>2</sup> con Sistemas Interactivos realistas también queda demostrada. En la sección siguiente se describen las características del simulador.

### **5.3.2 Descripción del simulador de conducción**

Los experimentos realizados en el dominio de la conducción se han llevado a cabo utilizando el simulador de conducción de camión y autobús que se muestra en la Figura 5.11. El simulador está formado por una cabina de camión real que está montada sobre una plataforma dinámica de 6 grados de libertad que se encarga de transmitir la sensación de movimiento al conductor. Además, tres pantallas gigantes y un sistema de audio envolvente proporcionan tanto inmersión visual como auditiva.

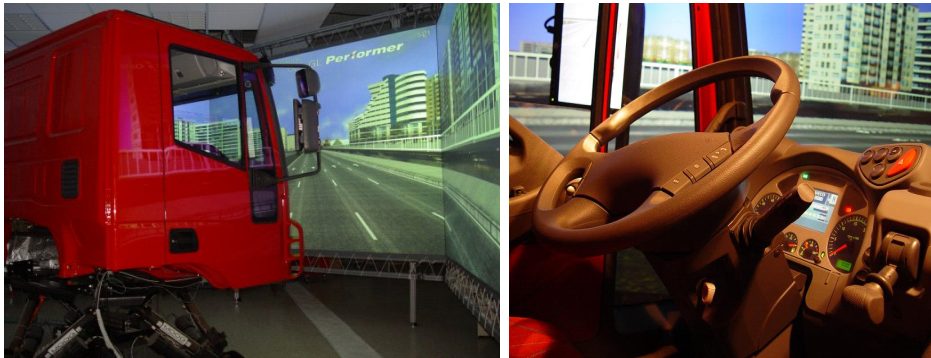


Figura 5.11: Exterior e interior del simulador de camión y autobús

El instructor sigue las sesiones de entrenamiento desde el Puesto del Instructor (Figura 5.12) mientras el alumno conduce en el simulador. El instructor propone al alumno tareas para demostrar sus habilidades en diferentes entornos: conducción por ciudad, por autopista, etc. También puede modificar en tiempo real las condiciones de la simulación, por ejemplo, la densidad de tráfico, averías del vehículo, pinchazos, etc.



Figura 5.12: Puesto del instructor

La arquitectura del simulador está distribuida en 8 PCs que integran múltiples componentes encargados de cada parte de la simulación: tráfico autónomo, dinámica del vehículo, sonido, control de plataforma, visualización... Estos módulos se comunican a través de sockets para sincronizar los datos que generan y que determinan el estado de la simulación en cada momento. Dichos datos son accesibles desde el Marco de Desarrollo para generar Observaciones a partir de ellos. Por ejemplo, se



dispone de la posición del camión y de los demás vehículos en la carretera, la velocidad y demás variables que determinan el estado del camión, el estado de la señalización y de la carretera, etc. En los ejemplos de los siguientes apartados se describe en mayor profundidad como se usan estos datos para crear Observaciones con las que interpretar y diagnosticar un cambio de carril y un adelantamiento.

### 5.3.3 Cambio de carril: Extracción de Observaciones e Interpretación de Pasos

En esta sección se presentan los procesos de extracción de observaciones y de interpretación asociados al cambio de carril. Esta maniobra se representa como un paso continuo que comienza cuando el vehículo inicia la aproximación al carril contiguo y no finaliza hasta que ha traspasado completamente la línea de separación. En primer lugar se describen las Observaciones que se extraen durante el proceso y posteriormente se presentan las Restricciones con las que interpretar el Paso.

#### ► Extracción de Observaciones

En la Figura 5.13 se muestra un doble cambio de carril. Primero hacia la izquierda y después hacia la derecha. Observando la figura se puede comprobar que para interpretar que el conductor realiza un cambio de carril lo interesante es conocer las distancias a los carriles contiguos. Dichas distancias se obtienen del módulo del simulador que calcula la dinámica del camión.

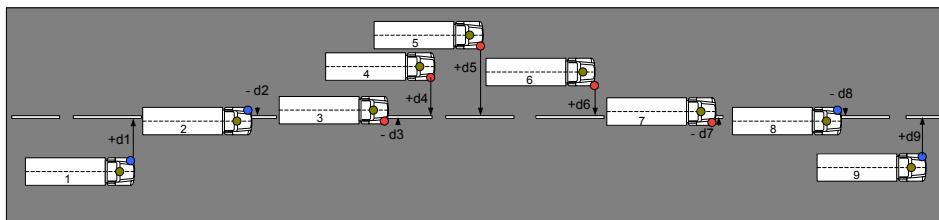


Figura 5.13: Cambio de carril hacia la izquierda y hacia la derecha

Se ha decidido medir la distancia al carril contiguo desde el lado del camión más cercano al carril hasta la línea de separación de carriles. Así, la distancia se mide desde el punto azul cuando el centro de la cabina, representado por el punto verde, está en el carril derecho, y desde el rojo

cuando está en el izquierdo. Además, las distancias calculadas hasta la línea de separación de carriles tienen magnitud y sentido, que dependen del carril en el que se encuentre el camión. Entonces, cuando el punto verde y el punto desde el que se calcula la distancia (azul o rojo) están en el mismo carril, las distancias calculadas son positivas (posiciones 1,4,5,6,9), mientras que cuando están en carriles distintos son negativas (posiciones 2,3,7,8). En la Figura 5.14 está representada la evolución de las distancias a cada carril que se han capturado en el simulador durante dos cambios de carril a izquierda y derecha. La distancia al carril izquierdo está representada con una línea azul, y como la carretera sólo tiene dos carriles, únicamente existen valores cuando el vehículo está en el carril derecho. De la misma manera, la distancia al carril derecho representada por la línea rosa sólo aparece cuando el camión está en el carril izquierdo. Se han separado mediante líneas discontinuas los intervalos numerados que representan las mismas posiciones del camión en la figura anterior.

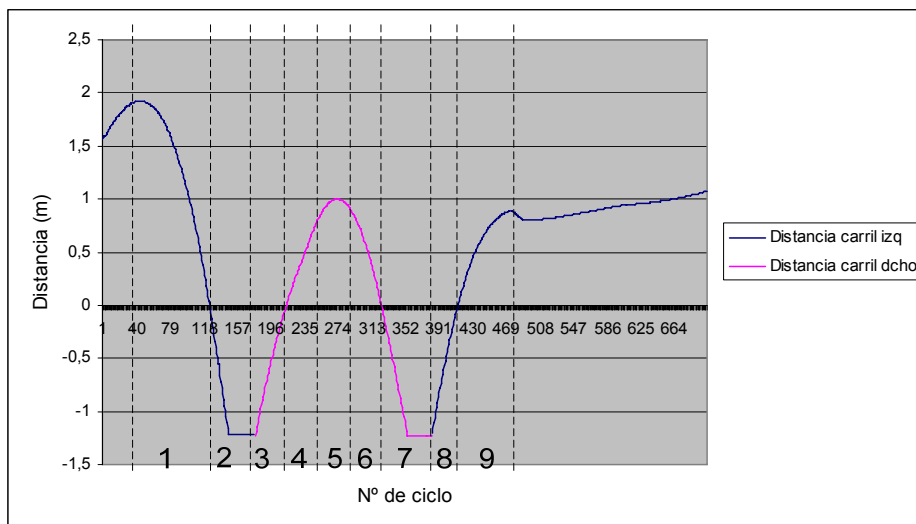


Figura 5.14: Evolución de las distancias a los carriles izquierdo y derecho durante las maniobras de cambio de carril

Considerando la forma de la curva, se han definido las siguientes observaciones necesarias para caracterizar el cambio de carril en toda su duración:

- *Acercamiento a carril izquierdo*: Esta observación se extrae durante el intervalo 1, en el que el vehículo está en el carril derecho y tiende a acercarse al izquierdo. El método de observación comprueba que la pendiente media de la curva de distancia al carril izquierdo es alta y las distancias son positivas y decrecientes.
- *Invasión carril izquierdo*: Esta observación se extrae durante el intervalo 2, en el que menos de la mitad del camión ha penetrado en el carril izquierdo. El método de observación comprueba que la pendiente media de la curva de distancia al carril izquierdo es alta y las distancias son negativas y decrecientes.
- *Alejamiento carril derecho*: Esta observación se extrae durante el intervalo 3, en el que la mayor parte del camión ya ha penetrado en el carril izquierdo y su movimiento hará que lo invada del todo. El método de observación comprueba que la pendiente media de la curva de distancia al carril izquierdo es alta y las distancias son negativas y crecientes.
- *Vehículo carril izquierdo*: Esta observación se extrae durante el intervalo 4, cuando el vehículo tiende a espaciarse más del carril izquierdo del que procede. El método de observación comprueba que la pendiente media de la curva de distancia al carril izquierdo es alta y las distancias son positivas y crecientes.

En el intervalo 5, correspondiente al periodo en el que el camión permanece a una distancia estable del carril derecho, no se ha definido ninguna observación. Los métodos de observación distinguen este intervalo del resto porque la pendiente media de la curva es significativamente más baja en este caso. Además de estas observaciones, es de reseñar que se han definido otras cuatro observaciones complementarias que se asocian al cambio de carril hacia la derecha: *Acercamiento a carril derecho*, *Invasión carril derecho*, *Alejamiento carril izquierdo* y *Vehículo carril derecho*. Todas estas Observaciones sirven para interpretar la ejecución de los pasos “Cambiar de carril hacia la izquierda” y “Cambiar de carril hacia la derecha”, cuya especificación se describe en el siguiente apartado.

#### ➤ Interpretación de Pasos

Las observaciones enumeradas en el apartado anterior se utilizan para especificar los EspIPaso de la maniobra de cambio de carril (Figura 5.15).

Las cuatro primeras observaciones que se han descrito se emplean para interpretar el cambio de carril a la izquierda y las complementarias para el cambio a la derecha.

EspIPaso
<p><b>Nombre:</b> Cambiar de carril hacia la izquierda  <b>Inicio [0..N]:</b> Acercamiento a carril izquierdo [Meets] Invasión carril izquierdo  <b>Restricciones generales [1..N]:</b> Invasión carril izquierdo [Meets] Alejamiento carril derecho  <b>Ejecucion [0..N]:</b> Invasión carril izquierdo [Meets] Alejamiento carril derecho  <b>Fin [0..N]:</b> Alejamiento carril derecho [Meets] Vehículo carril izquierdo  <b>UnidadTiempo:</b> ms</p>
EspIPaso
<p><b>Nombre:</b> Cambiar de carril hacia la derecha  <b>Inicio [0..N]:</b> Acercamiento a carril derecho [Meets] Invasión carril derecho  <b>Restricciones generales [1..N]:</b> Invasión carril derecho [Meets] Alejamiento carril izquierdo  <b>Ejecucion [0..N]:</b> Invasión carril derecho [Meets] Alejamiento carril izquierdo  <b>Fin [0..N]:</b> Alejamiento carril izquierdo [Meets] Vehículo carril derecho  <b>UnidadTiempo:</b> ms</p>

Figura 5.15: Especificación de los pasos: “Cambiar de carril hacia la izquierda” y “Cambiar de carril hacia la derecha”

En estos EspIPaso se hace uso de sus atributos *Inicio*, *Ejecución* y *Fin* además de las *Restricciones Generales*. Cada restricción se va cumpliendo en distintos momentos del proceso de interpretación, tal y como se observa en el cronograma de la Figura 5.16. En el eje de ordenadas se representan las observaciones que se van extrayendo. En el de abscisas se indican las posiciones del vehículo en cada fase del cambio de carril hacia la izquierda, siguiendo la misma numeración del apartado anterior. Además, se ha sombreado el fondo con distintos colores para identificar los cambios de estado del proceso de interpretación del paso. La interpretación del paso “Cambiar de carril hacia la derecha” es equivalente al del cronograma, sustituyendo las observaciones por sus complementarias.

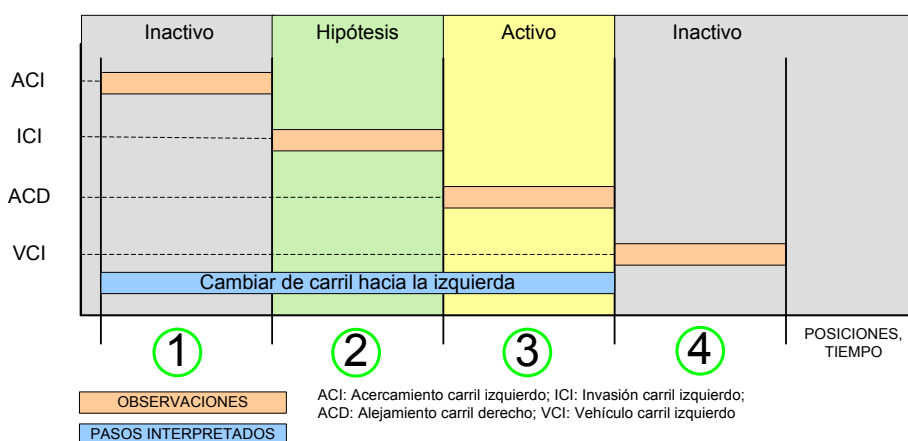


Figura 5.16: Evolución de la interpretación del Paso: "Cambiar de carril hacia la izquierda"

Cuando aparece la primera observación, "Acercamiento a carril izquierdo", todavía faltan datos para saber si se cumplirá la restricción de *Inicio*, por lo que la interpretación permanece en estado "Inactivo". Cuando se extrae la observación "Invasión carril izquierdo" ya se cumplen las restricciones de *Inicio*, lo que indica el posible comienzo del paso desde el instante que se extrajo la primera observación. Sin embargo, no se puede asegurar que se está ejecutando hasta que no se cumplen las *Restricciones Generales*. Concretamente, el vehículo podría acercarse e invadir el carril contiguo, pero si no llegase a traspasarlo no se podría considerar como un cambio de carril. Al cumplirse la restricción de *Inicio* se comprueban las *Restricciones Generales*. Como únicamente se dispone de la observación "Invasión carril izquierdo", se decide establecer el estado de "Hipótesis". Una vez que aparece la observación "Alejamiento carril derecho" ya está confirmado el cambio de carril y se establece el estado "Activo". Como además se cumple la restricción de *Ejecución* se solicita diagnóstico utilizando los datos recogidos desde que comenzó el acercamiento al carril izquierdo. Finalmente, cuando se extrae la observación "Vehículo carril izquierdo", ya se cumple la restricción de *Fin* y se da el paso por finalizado.

Los EspIPaso se pueden reutilizar para definir otros pasos nuevos, por ejemplo un doble cambio de carril. Si en una vía de tres carriles el vehículo se encuentra en el carril más interior y hace una sola maniobra para ir hasta el exterior, es conveniente diferenciarla de dos maniobras consecutivas de cambios de carril. Para ello, se define el EspIPaso "Doble

cambio de carril hacia la derecha” especificando dos apariciones del paso “Cambiar de carril hacia la derecha”. Para interpretarlo, únicamente es necesario definir en sus *Restricciones Generales*: “Cambiar de carril hacia la derecha 1” [Overlaps] “Cambiar de carril hacia la derecha 2”. Este paso se puede ejecutar en Situaciones diferentes. Por ejemplo, en una Situación de “Circulación en rotonda” el alumno puede ejecutarlo para salir de la rotonda cruzando dos carriles en una sola maniobra. Asimismo, puede realizarlo en una Situación de “Salida de autopista”. Teniendo esto en cuenta, el paso se diagnostica en función de la Situación en la que se lleva a cabo.

Además de reutilizar los EspIPasos para construir otros nuevos, los cambios de carril también forman parte de situaciones complejas, como puede ser una situación de adelantamiento.

### **5.3.4 Adelantamiento: Extracción de Observaciones e Interpretación de Pasos**

En esta sección se describen los procesos de extracción de observaciones e interpretación de pasos asociados a la maniobra de adelantamiento de un vehículo. A través de esta maniobra se ponen de manifiesto algunas capacidades de la Metodología que no se han empleado en el resto de ejemplos del capítulo. Por ejemplo, demuestra la capacidad para tratar entornos autónomos mediante el elemento Situación. Además, el adelantamiento permite comprobar que reutilizando observaciones, pasos y situaciones ya especificados se simplifica la construcción de otros nuevos más complejos. También sirve para demostrar que la definición de pasos que se pueden dar simultáneamente (ej: cambio de carril y dar intermitentes) no sólo es posible sino que es inmediato.

En el primer apartado se describe la maniobra, identificando los pasos y situaciones que se han definido para modelarla. Posteriormente se presentan las observaciones implicadas y finalmente se describe el proceso de interpretación.

#### **➤ Descripción de la maniobra de adelantamiento**

En este apartado se describe una maniobra típica de adelantamiento de un vehículo en una vía de doble sentido. No se han incluido observaciones sobre el tipo de vía o la interacción con más de un vehículo para no saturar

el ejemplo con observaciones que no son determinantes para la interpretación. Definir una maniobra de adelantamiento implica tener en cuenta todo lo que sucede desde la aproximación al vehículo que se pretende adelantar hasta después de rebasarlo. Además de considerar el rebasamiento es necesario identificar los posibles cambios de carril, el uso de los intermitentes, etc. Por ejemplo, Nigro y Rombaut (2003) proponen un método para reconocer adelantamientos en autopista a partir de los datos proporcionados por un conjunto de sensores. En su propuesta definen la maniobra como una secuencia de diez estados que hay que alcanzar en el siguiente orden: Esperar para adelantar, intento de adelantamiento, comienzo de cambio de carril hacia la izquierda, cruzar línea discontinua izquierda, terminar cambio de carril hacia la izquierda, rebasar, finalizar rebasamiento, comienzo de cambio de carril hacia la derecha, cruzar línea discontinua derecha y terminar cambio de carril hacia la derecha. Si los estados se alcanzasen en otro orden no se reconocería el adelantamiento. La representación de Nigro y Rombaut se podría reproducir mediante una RTCAllen, transformando la secuencia de estados en observaciones que estarían unidas mediante el operador *Meets*.

En su propuesta ya se intuye la dificultad de describir de forma realista el adelantamiento. Seis de los estados están asociados a los cambios de carril hacia la izquierda y hacia la derecha. Sin embargo, no se considera que el vehículo pueda estar en el carril izquierdo antes del adelantamiento o que después de adelantar no se realice el cambio de carril a la derecha. Más aún, si se realizase un adelantamiento incorrecto por la derecha, no sería interpretado. Para evitar que se pierda información es necesaria una especificación más completa del adelantamiento, como la que se propone a continuación.

La maniobra de adelantamiento se puede realizar de distintas maneras, lo que equivale a identificar varias soluciones. Por este motivo es más adecuado definir una EspDSituación "Adelantamiento", que contenga soluciones para diagnosticar la maniobra con y sin cambios de carril, teniendo en cuenta el uso de los intermitentes, etc. En la Figura 5.17 se muestra una solución correcta diagnosticable mediante la técnica de reconocimiento de planes. Aunque obviamente el diagnóstico completo de la maniobra implicaría comprobar cuándo se mira a los espejos retrovisores, este paso no se ha incluido en esta aproximación inicial. Para tenerlo en cuenta sería necesario disponer de un sistema de cámaras en el simulador

capaz de hacer el seguimiento de la mirada del conductor, lo cual se plantea en las líneas futuras de este trabajo. Una vez se pueda observar e interpretar los pasos “Mirar espejo izquierdo o derecho”, se incorporarían a los planes de solución.

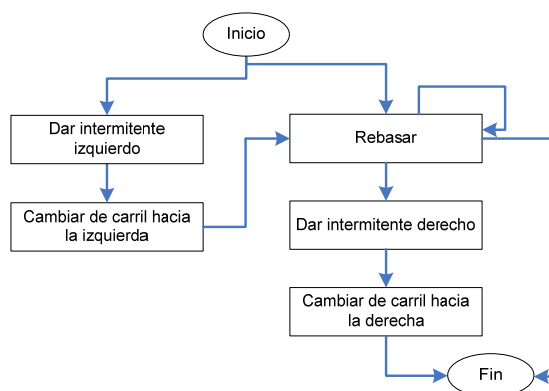


Figura 5.17: Plan de solución para la situación de “Adelantamiento”

El plan da entidad propia a los pasos “Cambiar de carril”, “Dar intermitentes” y “Rebasar”, posibilitando el diagnóstico de cada parte de la maniobra por separado. Además, permite reconocer el adelantamiento con y sin cambios de carril, lo que no sería posible realizando la interpretación basada en estados secuenciales de Nigro y Rombaut. Más allá, el diagnóstico puede detectar desviaciones de orden entre pasos, como cambiar de carril antes de dar los intermitentes. También puede restringir el tiempo transcurrido entre el cambio de carril y el rebasamiento, y además, como el elemento *EspDSituacion* permite especificar varias soluciones correctas o incorrectas, se puede definir un plan alternativo para diagnosticar un adelantamiento por la derecha.

Esta forma de modelar la maniobra de adelantamiento es posible gracias a la separación entre los niveles de diagnóstico e interpretación. Como los pasos y la situación son interpretados individualmente, se pueden generar todas las soluciones necesarias para diagnosticar sin necesidad de conocer las relaciones entre las observaciones implicadas. Más aún, el diagnóstico de los pasos es específico para cada situación, sin necesidad de cambiar la forma de interpretarse. Los aspectos más relevantes de la interpretación y la extracción de observaciones se exponen en los siguientes apartados.



### ➤ Extracción de Observaciones

Las observaciones principales implicadas en la maniobra de adelantamiento están asociadas a la presencia y relación con otros vehículos. La más importante es "Vehículo en alcance", de la que dependen otras cuatro sub-observaciones: "Acercamiento a vehículo", "Alejamiento de vehículo", "Vehículo delante" y "Vehículo detrás". Las sub-observaciones solo se intentan extraer cuando la observación de la que dependen ha sido extraída y se refieren siempre al mismo vehículo. Estas EspObservacion se describen a continuación:

- *Vehículo en alcance*: Esta observación se extrae por cada vehículo que esté circulando en la misma vía y en el mismo sentido que el camión a una distancia reducida. Por tanto, quedan excluidos de esta observación los vehículos estacionados, los que circulen por otras vías, aunque sean cercanas, y los que vayan en sentido contrario. Estos vehículos se representan por otras observaciones. Las propiedades de la EspObservacion son las siguientes:
  - Identificador: del vehículo en alcance.
  - Carril: identificador del carril por el que circula el vehículo en alcance.
  - Distancia: entre el vehículo en alcance y el camión. Esta distancia se calcula proyectando la posición de los vehículos sobre el eje central de la carretera. Así, la distancia obtenida es independiente del carril por el que estén circulando los vehículos. Si el vehículo en alcance está por delante del camión la distancia es positiva, si no, es negativa.
- *Acercamiento a vehículo*: Esta sub-observación se extrae cuando la distancia al vehículo en alcance es positiva y tiende a disminuir, es decir, el camión va más rápido que el otro vehículo y se le acerca por detrás.
- *Alejamiento de vehículo*: Esta sub-observación se extrae cuando la distancia al vehículo en alcance es negativa y tiende a disminuir, es decir, el camión va más rápido que el otro vehículo y se va distanciando por delante.
- *Vehículo delante*: Esta sub-observación se extrae cuando el morro del camión ha alcanzado al vehículo objetivo pero sin llegar a sobresalir.

- *Vehículo detrás*: Esta sub-observación se extrae cuando el morro del camión sobresale del morro del vehículo en alcance y el final del camión no sobresale.

Además de estas observaciones se utilizan otras para registrar los cambios de estado de los intermitentes, además de las asociadas al diagnóstico (ej: señalización horizontal y vertical de la vía). Sin embargo, no son relevantes para la interpretación, por lo que no se especifican.

#### ➤ Interpretación de Situación y Pasos

La situación de adelantamiento engloba todo aquello que sucede desde que el camión comienza a acercarse a un vehículo en alcance hasta que se aleja. Para representar esta definición se especifica la EspISituacion de la Figura 5.18.

EspISituacion
<p><b>Nombre:</b> Adelantamiento</p> <p><b>Inicio [0..N]:</b> Acercamiento a vehículo AND Vehículo en alcance.Distancia&lt;10</p> <p><b>Restricciones generales [1..N]:</b> Acercamiento a vehículo [Meets] Alejamiento de vehículo OR (Acercamiento a vehículo [Precedes] Alejamiento de vehículo AND Duracion(Acercamiento a vehículo.Fin, Alejamiento de vehículo.Inicio)&lt;10000)</p> <p><b>Fin [0..N]:</b> Vehículo en alcance.Distancia&lt;-10</p> <p><b>Pasos Posibles:</b> Cambiar de carril hacia la derecha, Cambiar de carril hacia la izquierda, Dar intermitente izquierdo, Dar intermitente derecho, Rebasar</p> <p><b>UnidadTiempo:</b> ms</p>

Figura 5.18: Especificación de la situación “Adelantamiento”

Como se observa, las restricciones de *Inicio* contienen una Restricción sobre Propiedad que indica que el comienzo de la situación de adelantamiento se produce cuando el camión se acerca y está suficientemente próximo al vehículo objetivo. Las *Restricciones Generales* especifican que se está en situación de adelantamiento si el camión se acerca e inmediatamente se aleja del vehículo objetivo, es decir, si el adelantamiento es fluido. Además, en la rama *OR* de las *Restricciones Generales* se tiene en cuenta que desde que el camión se acerca hasta que se aleja puede estar durante un periodo de tiempo a la par del vehículo objetivo sin movimientos claros de acercamiento o alejamiento. Dicho periodo de tiempo se limita a una duración máxima para no interpretar como adelantamientos los casos en los que el camión se acerca inicialmente

al vehículo objetivo pero después se queda circulando a distancia constante sin terminar la maniobra. El final de la situación se produce cuando el camión se aleja lo suficiente del vehículo objetivo o por el desistimiento en la maniobra después de haber comenzado a alejarse, lo que implicaría que las *Restricciones Generales* se dejarían de cumplir.

En la situación de adelantamiento que se ha especificado se indican los pasos incluidos en la solución de ejemplo propuesta en el apartado anterior. En dicha solución se reutilizan los pasos de cambio de carril que ya habían sido especificados. También se emplean los pasos referentes a dar alguno de los intermitentes. Su especificación es trivial ya que únicamente contienen la observación que recoge el evento de activación de la palanca de intermitentes. Finalmente, se incluye el paso “Rebasar”, que es interpretado cuando el camión pasa de estar por detrás del vehículo a estar por delante (Figura 5.19).

EspIPaso
<b>Nombre:</b> Rebasar <b>Inicio [0..N]:</b> <b>Restricciones generales [1..N]:</b> Vehículo delante [Meets] Vehículo detrás <b>Ejecucion [0..N]:</b> <b>Fin [0..N]:</b> <b>UnidadTiempo:</b>

Figura 5.19: Especificación del paso “Rebasar”

A lo largo de la situación de adelantamiento se puede realizar cualquiera de los pasos previstos en momentos distintos. El subsistema de diagnóstico determinará si el momento y la forma de realizar cada paso es el adecuado, independientemente de si el paso es continuo o instantáneo o de si hay varios pasos y situaciones desarrollándose simultáneamente. Para conseguirlo, el subsistema de interpretación gestiona en cada ciclo de simulación el estado de interpretación de todos los pasos y situaciones. Así, controla toda la actividad simultánea que se produzca tanto de situaciones como de pasos continuos e instantáneos. En el cronograma de la Figura 5.20 se ha representado la evolución de la interpretación de la situación de adelantamiento, así como de cada paso.

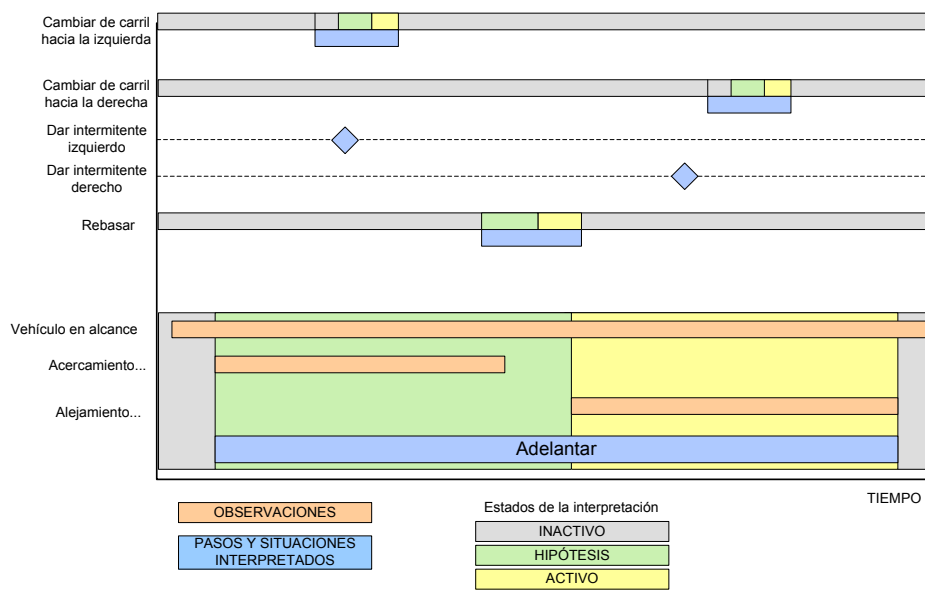


Figura 5.20: Evolución de la situación "Adelantamiento"

En la parte superior del cronograma se muestra la evolución de la interpretación de cada paso. Los pasos continuos, correspondientes a los cambios de carril y al rebasamiento, cambian del estado inactivo al de hipótesis, después al activo y de nuevo al inactivo cuando finalizan, de la misma forma que se describió en la sección anterior. En cambio, el momento en el que se interpretan los pasos instantáneos correspondientes a dar los intermitentes se marcan con rombos. En este ejemplo se ha representado la ejecución del paso "Dar intermitente izquierdo" durante la ejecución del cambio de carril en lugar de antes de la maniobra, lo que se consideraría un error.

En la parte inferior del cronograma se muestra la evolución de las observaciones de acercamiento y alejamiento al vehículo en alcance. Cuando comienza la observación de acercamiento se establece la hipótesis de la situación de adelantamiento (indicado con el fondo de color verde). Durante ese intervalo los pasos ejecutados se diagnostican en el contexto de todas las situaciones activas y en hipótesis a las que estén asociados. En este caso típico de adelantamiento, se diagnosticarían en el contexto de la "Situación General" además de la de "Adelantamiento". Si se quisieran considerar otras maniobras de adelantamiento más complicadas habría que especificar más situaciones, como "Intento de adelantamiento" o

“Adelantamiento de larga duración”. Estas situaciones contendrían en sus soluciones las reacciones específicas que se esperan del conductor para afrontarlas.

### **5.3.5 Conclusiones**

Los experimentos realizados en el dominio de la conducción que se han presentado a lo largo de la sección han servido para probar algunas capacidades concretas de la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> que no se abarcaron en el dominio de la máquina herramienta.

Al aplicar la Metodología en un dominio radicalmente distinto al de VIRTUOL se ha verificado que es independiente del dominio de aprendizaje. Además, se ha demostrado que permite construir SI<sup>2</sup>A<sup>2</sup> con Sistemas Interactivos altamente realistas como el simulador de conducción de camiones empleado. Más aún, la integración es independiente de la arquitectura interna del simulador, con el único requisito de que éste tenga la capacidad de generar y comunicar los datos de entrada de los que extraer observaciones.

Se han especificado pasos continuos para demostrar la validez de las Restricciones Temporales en el Nivel de Interpretación. Se han representado la maniobra de cambio de carril y el rebasamiento de un vehículo. En ambos casos su interpretación comienza generando observaciones a partir de flujos de datos y relacionándolas a través de Restricciones Temporales para determinar que paso representan.

Se ha modelado una situación típica de adelantamiento para demostrar la validez de la Metodología cuando aparecen elementos autónomos en el Sistema Interactivo. El modelado mediante pasos y situaciones tiene múltiples ventajas. En primer lugar, el Subsistema de Interpretación gestiona concurrentemente la interpretación de todos los pasos y situaciones, lo que posibilita de forma natural la ejecución de pasos simultáneos. En segundo lugar, mediante la separación de los Modelos de Interpretación y Tareas se facilita la representación de variaciones en la forma de adelantar. Por ejemplo, los planes de solución tienen en cuenta que no siempre se producen cambios de carril al adelantar y que se pueden dar adelantamientos incorrectos por la derecha. Esta flexibilidad contrasta con la propuesta de Nigro y Rombaut para representar la misma maniobra mediante una secuencia de estados inalterable.

Los experimentos realizados en el dominio de la conducción se han orientado a validar características concretas de la Metodología. Debido a los resultados prometedores obtenidos en este trabajo se está desarrollando en la actualidad un SI<sup>2</sup>A<sup>2</sup> completo para la formación de conductores profesionales. En dicho proyecto se están llevando a cabo ampliaciones en el simulador de camión para extraer nuevas observaciones. También se está extendiendo DETECTive para diagnosticar actividad continua, y se está trabajando en ofrecer asistencia inteligente multimodal al conductor a partir de los resultados de diagnóstico en tiempo real (López-Garate et al., 2008a; López-Garate et al., 2008b).



## *CAPÍTULO 6*

# ***CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN***

---

En este capítulo se resume el trabajo realizado, presentando los principales resultados y aportaciones obtenidos de la investigación llevada a cabo. Además se exponen las perspectivas de continuación del trabajo presentado.

### **6.1 RESUMEN**

En este proyecto se ha profundizado en la problemática inherente a la construcción de Sistemas Interactivos Inteligentes de Ayuda al Aprendizaje (SI<sup>2</sup>A<sup>2</sup>) de tareas procedimentales. Dada la complejidad del proceso de creación de este tipo de sistemas, el objetivo abordado ha sido facilitar la integración de funcionalidades educativas en todo tipo de Sistemas Interactivos diseñados para la práctica de tareas procedimentales. Para dar mayor flexibilidad a las capacidades educativas que puede exhibir el Sistema Interactivo, este proceso de integración se ha planteado evitando predeterminar los componentes concretos de ayuda al aprendizaje que es posible integrar, sus arquitecturas y sus procesos internos.

Se ha presentado un estudio de SI<sup>2</sup>A<sup>2</sup> focalizado en las arquitecturas de integración. A partir de este estudio se han enumerado, descrito y analizado en profundidad las ventajas y desventajas de las estrategias más comunes de construcción de SI<sup>2</sup>A<sup>2</sup>. Este análisis ha permitido valorar las



limitaciones de cada estrategia y de cada arquitectura para estimar su capacidad de adaptación en la construcción de nuevos SI<sup>2</sup>A<sup>2</sup> para diferentes dominios.

A continuación, se ha analizado la problemática clave que surge al abordar el desarrollo de una Metodología de propósito general para la construcción de SI<sup>2</sup>A<sup>2</sup>. Este análisis ha permitido determinar que los nexos entre un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje deben fundamentarse en las capacidades de **observación, interpretación y diagnóstico**. Como consecuencia:

- El nexo genérico con el Sistema Interactivo se debe basar en la extracción de observaciones, permitiendo la independencia de la arquitectura interna y de los dispositivos de interacción.
- El nexo con un Sistema de Ayuda al Aprendizaje se debe fundamentar en la comunicación de los resultados de diagnóstico.

Considerando las conclusiones extraídas de estos estudios previos, se ha diseñado una Metodología para guiar el diseño y desarrollo de nuevos SI<sup>2</sup>A<sup>2</sup> a partir de la integración de un Sistema Interactivo y un Sistema Ayuda al Aprendizaje, independientemente de su arquitectura y características particulares. Además se ha establecido el Marco de Desarrollo para la comunicación automática de un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje.

## 6.2 RESULTADOS Y APORTACIONES

El objetivo planteado inicialmente en este trabajo ha sido satisfecho, permitiendo establecer las siguientes conclusiones:

- Se ha diseñado una Metodología para guiar el diseño y desarrollo de nuevos SI<sup>2</sup>A<sup>2</sup>. La Metodología se basa en un metamodelo de integración que abstrae el proceso de integración en tres Niveles: **Observación, Interpretación y Diagnóstico**. Cada nivel contiene los elementos necesarios para modelar: *qué observar en el Sistema Interactivo* (Observaciones); *cómo interpretar las Observaciones para determinar lo que hace el alumno y el contexto en el que lo hace* (Pasos y Situaciones); y *cómo determinar si lo que hace es correcto o incorrecto* (Soluciones).

- La Metodología integra los procesos a seguir para diseñar y construir un SI<sup>2</sup>A<sup>2</sup>. Se centran en la generación de los Modelos de Observación, Interpretación y Diagnóstico, teniendo en cuenta la problemática y los riesgos principales que hay que evitar durante el diseño.
- Esta Metodología es independiente de los dispositivos de interacción y de las arquitecturas internas de los Sistemas que está integrando. Esto facilita la colaboración de equipos multidisciplinares en la construcción de nuevos SI<sup>2</sup>A<sup>2</sup>, ya que el desarrollador del Sistema Interactivo no necesita conocer los detalles del Sistema de Ayuda al Aprendizaje y el desarrollador de este último tampoco precisa conocer las particularidades del Sistema Interactivo.
- Se ha creado un Marco de Desarrollo para la comunicación automática de un Sistema Interactivo y un Sistema de Ayuda al Aprendizaje. Está compuesto por tres **Subsistemas** que se encargan de la **observación, interpretación y diagnóstico** de la actividad del alumno. La comunicación se basa en los modelos del mismo nombre generados siguiendo la Metodología. La reutilización del Marco de Desarrollo minimiza el tiempo de construcción y, por tanto, el coste del SI<sup>2</sup>A<sup>2</sup> (en comparación con los desarrollos a medida para lograr funcionalidades similares).
- La Metodología y el Marco de Desarrollo creados han sido probados en el dominio de la máquina herramienta y en el de la conducción. Los experimentos realizados han permitido verificar la validez de la propuesta en los siguientes aspectos:
- Tanto la Metodología como el Marco de Desarrollo son independientes del dominio de aprendizaje.
- La Metodología permite representar de forma genérica tanto actividad continua como instantánea en el mismo dominio valiéndose de una combinación de Restricciones: Restricciones Temporales Cualitativas de Allen y Puntuales, Restricciones Temporales Cuantitativas, Restricciones sobre Propiedades y Restricciones Lógicas.
- La Metodología es compatible con entornos autónomos, gracias al elemento *Situación* que permite contextualizar los pasos del alumno provocados como respuesta a las condiciones del entorno impuestas por la entidad autónoma.

- El Marco de Desarrollo permite integrar Sistemas Interactivos simplificados como Virtool-Machine y Sistemas Interactivos altamente realistas como el simulador de conducción de camiones.
- El Marco de Desarrollo puede interpretar situaciones y pasos tanto continuos como instantáneos valiéndose de un proceso de interpretación genérico basado en la evaluación de Restricciones y de Redes de Satisfacción de Restricciones Temporales.
- El Subsistema de Interpretación puede gestionar la ejecución de pasos y la aparición de situaciones simultáneas de forma natural, sin necesidad de ningún control específico, al contrario que otros formalismos clásicos de representación de actividad como STRIPS o el cálculo de situación.

### 6.3 FUTURAS LÍNEAS DE INVESTIGACIÓN

Partiendo de los resultados de este trabajo se proponen cuatro líneas principales de investigación: extensiones de la Metodología y el Marco de Desarrollo, nuevos Sistemas Interactivos, conectividad con Sistemas de Ayuda al Aprendizaje y herramientas de adquisición de conocimiento. Cada una de estas líneas se detalla en los siguientes apartados.

#### ➤ Extensiones de la Metodología y el Marco de Desarrollo

Esta línea contempla la extensión de la Metodología y del Marco de Desarrollo en todos sus niveles para ampliar las capacidades que ya exhiben y para acometer unos nuevos objetivos que se describen a continuación:

- *Subsistema de Observación:* Investigar la reutilización de métodos de observación para agilizar más la construcción de SPA<sup>2</sup>. Por la experiencia obtenida, se ha concluido que la forma de extraer observaciones puede generalizarse en muchos casos. Los datos generados en el Sistema Interactivo que se utilizan habitualmente suelen ser flujos de datos, estados o eventos. Los métodos de observación que se pretende reutilizar identificarían cambios en los valores de los datos de entrada o patrones en su forma de cambiar. Además, en nuestro grupo de investigación del CEIT ya se está estudiando el uso de clasificadores en el campo de la biomecánica para detectar patrones complejos en flujos de datos de los que no es obvio cómo extraer las observaciones (Mena et al., 2008). Finalmente, se

propone incluir métodos de extracción de observaciones utilizando lógica difusa (Castro, 2007). Gracias a la lógica difusa se podrían representar hechos u observaciones que no tienen por qué ser categóricamente ciertas o falsas, sino que representarían valores intermedios.

- *Subsistema de Interpretación*: Estudiar el comportamiento del Subsistema de Interpretación en función del crecimiento de su modelo en dominios complejos. Interesa estudiar la probabilidad de que haya colisión entre interpretaciones, es decir, que dado un conjunto de observaciones aparezcan múltiples hipótesis que no se puedan desambiguar o que se tarde demasiado en hacerlo. Asimismo se estudiará el impacto que puede tener la resolución de ambigüedades sobre el rendimiento del Subsistema de Interpretación y sobre su comportamiento.
- *Subsistema de Diagnóstico*: Desarrollo de un Subsistema de Diagnóstico genérico que solvete las carencias de DETECTive presentadas en el análisis del apartado 2.3 y que encaje en la plantilla del subsistema de diagnóstico del Marco de Desarrollo (sección 4.5). El nuevo subsistema debe facilitar su escalabilidad, además de permitir el diagnóstico de pasos y situaciones continuas y simultáneas, y de manejar interpretaciones en estado de hipótesis. Todo ello, sin perder las ventajas originales de DETECTive: genericidad, usabilidad, y capacidad de diagnóstico apropiada y suficiente.

Otra posible ampliación de este trabajo consistiría en extender el metamodelo de integración para modelar tareas en equipo. Para llevarlo a cabo, se estudiaría un enfoque similar al seguido para extender el modelo de tareas de STEVE (McCarthy et al., 1999), que asocia la identificación del usuario a cada una de sus acciones en el entorno virtual.

#### ➤ **Nuevos Sistemas Interactivos**

Esta línea futura aborda el desarrollo de nuevos SP<sup>2</sup>A<sup>2</sup> que integren Sistemas Interactivos avanzados distintos del simulador de conducción empleado en los experimentos. Una de las motivaciones de este trabajo de tesis viene dada por la evolución de las interfaces de los Sistemas Interactivos y su tendencia a ser cada vez más naturales y realistas. De ahí que se haya abordado el problema de la construcción de SP<sup>2</sup>A<sup>2</sup> considerando el modelado y la interpretación de pasos continuos, además de los

instantáneos. Se han modelado pasos continuos tanto en el dominio de la máquina herramienta como en el de la conducción, pero hay que seguir profundizando en el problema con otras interfaces complejas. Por ejemplo, ya se comenzó a analizar el problema de las interfaces hápticas en el contexto del entrenamiento de operarios para tareas de mantenimiento (Díaz et al., 2006). Sin embargo, sería interesante abordar otros dominios de aprendizaje que den más posibilidades para investigar. Por ejemplo, la formación de cirujanos y el entrenamiento en algunos deportes como el remo (Ruffaldi et al., 2009) conllevan la adquisición de destrezas físicas o manuales más complejas que las tareas de mantenimiento estudiadas.

Además de las interfaces hápticas, otra línea futura abordaría el uso de sistemas de captura de movimiento de cuerpo completo como interfaz para SI<sup>2</sup>A<sup>2</sup>. Los pasos iniciales que ya se han dado en esta dirección en nuestro grupo de investigación han abordado la reconstrucción digital del movimiento del usuario capturado y el reconocimiento de movimientos básicos mediante clasificadores (Ausejo, 2006; Unzueta, 2009; Unzueta et al., 2005; Unzueta et al., 2008). Mediante esta tecnología se puede abordar la formación en dominios que impliquen la adquisición de destrezas físicas, como la esgrima o la danza. Para avanzar en esta línea sería necesario abordar los problemas que puedan surgir por el envío de flujos masivos de datos desde el sistema de captura.

#### ➤ **Conectividad con Sistemas de Ayuda al Aprendizaje**

Esta línea busca maximizar la capacidad de conexión del Marco de Desarrollo con Sistemas de Ayuda al Aprendizaje diferentes. Inicialmente, se cuenta con la experiencia de integración de DETECTive con Sistemas Tutores Inteligentes generados por la herramienta de autor IRIS. Sin embargo, conviene diversificar los componentes de ayuda al aprendizaje con los que se puede conectar. Actualmente se está llevando a cabo en nuestro grupo de investigación un trabajo de tesis en el que se está construyendo un componente de *feedback* capaz de ofrecer asistencia multimodal al alumno mientras realiza una tarea (López-Garate et al., 2008a). Dicho componente decide cuándo y cómo comunicarse con el alumno en función de los resultados de diagnóstico de cada paso que realiza, el histórico de asistencia durante la tarea, los tipos de mensajes disponibles, lo intrusivos que pueden ser para el alumno, etc. Además, el instructor puede personalizar el comportamiento del componente de

*feedback* para que se ajuste a distintas circunstancias: clases para principiantes, clases para expertos, exámenes...

Otra línea de investigación que se plantea abordar en profundidad es la conexión con otras herramientas docentes. Resulta de interés la conexión con MAGADI<sup>18</sup> (Álvarez et al., 2009) por su orientación a entornos de *blended-learning*<sup>19</sup>. Asimismo, merece la pena estudiar la interoperabilidad de MAGADI con el componente de *feedback* multimodal descrito en el párrafo anterior. En esta línea, también se buscarán nexos con otras investigaciones del grupo GaLan que están relacionadas con MAGADI (Martín et al., 2008; Rueda, 2009).

#### ➤ Herramientas de adquisición de conocimiento

Esta línea futura se centra en la investigación sobre herramientas de autor que minimicen el coste de creación de los Modelos de Observación, Interpretación y Tareas. La creación de los modelos puede ser un trabajo muy costoso cuando el dominio es complejo. Por esta razón es beneficioso disponer de herramientas que ayuden a generarlos. DETECTive cuenta con KADI, su propia herramienta de adquisición de conocimiento con la que se genera el modelo de tareas interactivamente (Martín et al., 2004). Sin embargo el modelo de tareas de un SI<sup>2</sup>A<sup>2</sup> puede ser muy extenso y complejo, por lo que conviene automatizar más su generación. En VIRTOOL se comprobó la viabilidad de generar automáticamente el modelo de tareas a partir de demostraciones del experto con la misma interfaz del alumno. De los resultados extraídos se determinó la necesidad de investigar el problema en mayor profundidad, abordando no sólo la generación semiautomática del Modelo de Tareas sino también los de Observación e Interpretación.

---

<sup>18</sup> MAGADI es una herramienta docente desarrollada por el grupo de investigación GaLan de la Universidad del País Vasco

<sup>19</sup> En los entornos de *blended-learning* se combinan instrucción presencial y mediante ordenador



# ANEXO A

## *ESTÁNDAR DE INTEGRACIÓN SCORM- SIMULACIÓN*

---

En el año 2006 el comité de estándares de tecnologías de aprendizaje de IEEE (IEEE Learning Technology Standards Committee, LTSC) y la Organización de estándares de interoperabilidad entre simulaciones (Simulation Interoperability Standards Organization, SISO) formaron un grupo de trabajo conjunto (SCORM-Sim Study Group, [www.sisostds.org](http://www.sisostds.org)) para crear un estándar de integración de simulaciones en objetos de aprendizaje SCORM (Sección 1.2.3). En este comité participan, entre otros, ADL<sup>20</sup> como creadores de SCORM, y empresas desarrolladoras de tecnologías asociadas a SCORM, como MAK Technologies<sup>21</sup>. Además, colaboran varios departamentos militares y gubernamentales de los EEUU, la compañía aeronáutica Boeing, la Agencia Espacial Europea y varios contratistas del departamento de defensa de EEUU dedicados al desarrollo de sistemas de aprendizaje para militares, como Stottler Henke<sup>22</sup>, BBN Technologies<sup>23</sup> y Aptima<sup>24</sup>.

---

<sup>20</sup> ADL: Advanced Distributed Learning Initiative. [www.adlnet.gov](http://www.adlnet.gov)

<sup>21</sup> MAK Technologies: [www.mak.com](http://www.mak.com)

<sup>22</sup> Stottler Henke: [www.stottlerhenke.com](http://www.stottlerhenke.com)

<sup>23</sup> BBN Technologies: [www.bbn.com](http://www.bbn.com)

<sup>24</sup> Aptima: [www.aptima.com](http://www.aptima.com)



El objetivo hacia el que quieren avanzar es la integración de experiencias de aprendizaje basadas en simulaciones con entornos SCORM. El primer paso que han planteado en esa dirección consiste en habilitar a los contenidos y al motor de SCORM para invocarse y comunicarse con simulaciones de una forma estandarizada e interoperable. Dichos estándares “reducirán el coste de integración y permitirán construir herramientas que ahorren tiempo y dinero a los diseñadores y desarrolladores, y ayudarán a los vendedores de simulaciones a desarrollar componentes y simulaciones reutilizables” (Objetivos SCORM-Sim Study Group).

El estándar aún no ha sido finalizado, y los datos disponibles actualmente son sólo un esbozo de las intenciones del grupo. Por ejemplo, en la Figura A.6.1 se muestran algunos focos de interés que identifican en el proceso de integración. Se distinguen tres grupos de elementos:

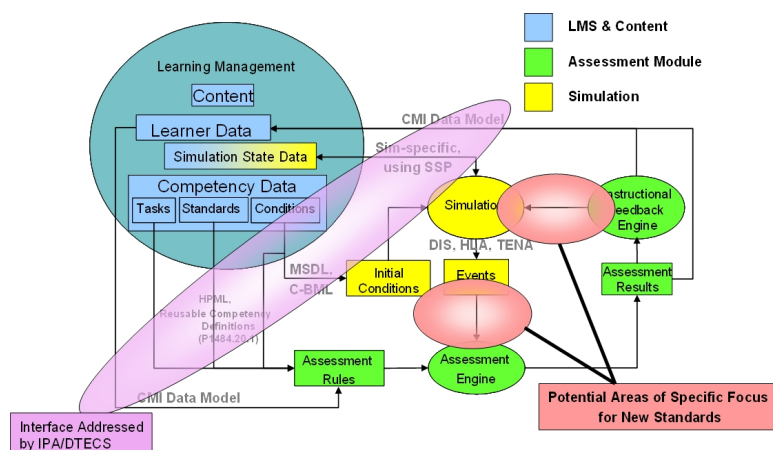


Figura A.6.1: Focos de interés del SCORM-Sim Study Group

- Simulación (elementos amarillos): Inicialmente sólo consideran simulaciones simplificadas que se puedan ejecutar en un navegador web. Por eso, indican que la información que se extrae de la simulación consiste en eventos, sin mención alguna a flujos de datos que irían asociados a pasos continuos. Los aspectos de la simulación que deciden tratar en el estándar tienen que ver con los mensajes de inicialización y control de la simulación. También consideran el almacenamiento del estado de la simulación en el objeto de aprendizaje para restaurar ejercicios inacabados.

- Motor de evaluación (elementos verdes): El grupo de estudio ha determinado la conveniencia de integrar un motor de evaluación genérico, capaz de determinar los progresos del estudiante. Sin embargo, no se indica ninguna característica del mismo. En el mismo grupo de elementos, incorporan un módulo encargado de comunicar feedback al alumno en función de los resultados de evaluación. En este sentido, aunque tampoco se da ningún detalle de dicho componente, su propuesta de independizar simulación, evaluación, feedback y planificación instruccional se parece al planteamiento de este trabajo de tesis.
- Sistema de Gestión de Aprendizaje (*Learning Management System, LMS*) (elementos azules): El LMS es un componente básico en SCORM. Integra el conjunto de funcionalidades diseñadas para gestionar y repartir contenidos educativos, hacer el seguimiento de los progresos de los alumnos y registrar sus interacciones. Su relación con las simulaciones se limita al almacenamiento de los objetos que las encapsulan y al envío de comandos para configurar sus condiciones de inicio.

Además de esta iniciativa conjunta de estandarización, algunos de sus socios han desarrollado sus propias propuestas de integración de simulaciones en SCORM que se recogen en las memorias de actividades del Joint ADL Co-Laboratory ([www.jointadlcolab.org](http://www.jointadlcolab.org)). En general, se centran en la unión del objeto SCORM que contiene la simulación en una arquitectura distribuida de aprendizaje a distancia. La comunicación con el LMS y con otros componentes de ayuda al aprendizaje, como Sistemas Tutores Inteligentes, está fundamentada en un módulo de evaluación que se desarrolla e integra a medida en cada simulación. En este sentido, su propuesta deja sin abordar toda la problemática de construcción del SI<sup>2</sup>A<sup>2</sup> abordada en este trabajo de tesis. Sin embargo, el resto de su arquitectura basada en componentes interoperables puede ser de gran interés para maximizar la conectividad de la Metodología de desarrollo de SI<sup>2</sup>A<sup>2</sup> propuesta en este trabajo con distintas funcionalidades de ayuda al aprendizaje. Por esta razón, el análisis en profundidad de esta arquitectura es un foco de interés para las líneas futuras de este trabajo de tesis.



## *ANEXO B*

# ***REPRESENTACIÓN DE ACTIVIDAD MEDIANTE RESTRICCIONES TEMPORALES***

---

Dada la relevancia de las Restricciones Temporales en el Nivel de Interpretación del metamodelo de integración, en este anexo se describen otros trabajos que las emplean y que sirven para justificar su utilidad en este contexto comparándolas con otras alternativas. En primer lugar se describen los antecedentes relacionados con las Restricciones Temporales Cualitativas y, posteriormente, los relacionados con las Cuantitativas.

### **B.1 ANTECEDENTES DE REPRESENTACIÓN E INTERPRETACIÓN DE LA ACTIVIDAD MEDIANTE EL ÁLGEBRA DE INTERVALOS**

Las Restricciones Temporales Cualitativas se han justificado por la necesidad de describir la actividad continua del alumno. Mediante estas Restricciones, se construyen relaciones complejas entre intervalos que sirven para definir los pasos y situaciones del Nivel de Interpretación (Sección 3.4). En este apartado se profundiza en la validez de esta aproximación examinando algunos antecedentes que relacionan el álgebra de intervalos con la representación o interpretación de actividad. Aunque

presentan diferencias importantes con nuestra propuesta, sus estudios tienen algunos puntos en común que resultan relevantes para este trabajo.

Allen y Ferguson (1994) plantean un formalismo basado en el álgebra de intervalos para describir la actividad. Su objetivo es establecer un marco de representación de tareas de propósito general con el que poder realizar distintos tipos de razonamiento como:

- Planificación: Dada la descripción inicial del mundo y un objetivo, encontrar la secuencia de acciones que llevan a conseguir dicho objetivo.
- Explicación: Dado un conjunto de datos sobre el mundo, encontrar la mejor explicación de los datos. Si los datos se refieren a las acciones de otra entidad y la explicación buscada es el plan seguido para llegar al estado presente, el problema se conoce como reconocimiento de planes.
- Predicción: Dada una descripción del estado actual del mundo, inferir el próximo estado más probable.

Sus objetivos no incluyen la interpretación, por lo que su propuesta no resuelve las necesidades del metamodelo. Sin embargo, su investigación señala algunas características necesarias tanto para la representación como la interpretación de actividad. Sus argumentos apoyan el uso de Restricciones Temporales Cualitativas en el Nivel de Interpretación y además sirven para descartar otras alternativas de representación de actividad. Principalmente se centra en descartar dos formalismos clásicos de representación: el Cálculo de Situación (McCarthy y Hayes, 1969) y STRIPS (Fikes y Nilsson, 1971).

El Cálculo de Situación describe la actividad como composición de estados que se dan en un único instante de tiempo. De hecho, la composición se puede entender como una forma básica de representación basada en puntos temporales. STRIPS por su parte añade a la representación los efectos que las acciones causan sobre los estados. Teniendo en cuenta que, en nuestro caso, el efecto de las acciones del alumno será normalmente determinado por una simulación externa, esta característica de STRIPS no resulta útil para este trabajo. Por otra parte, Allen y Ferguson indican varias carencias de estos formalismos centrándose en las tareas de planificación, explicación y predicción, dando argumentos válidos para descartar ambos para la interpretación:

- 
- Ni STRIPS ni el Cálculo de Situación considera la duración de las acciones. Por tanto, no hay forma de representar relaciones complejas que incluyan acciones simultáneas.
  - No permiten manejar eventos externos, con lo que no podría representar la actividad en entornos autónomos. Tal y como se explicaba en el capítulo anterior, el alumno puede actuar no solo por la planificación que tuviera prevista, sino también en respuesta a la interacción con elementos autónomos.

El álgebra de intervalos resuelve el primer problema tanto para los objetivos de Allen y Ferguson como para el Nivel de Interpretación. El segundo problema no le afecta directamente a este nivel, ya que los eventos externos son considerados en el Nivel de Observación. Por estos motivos, las Restricciones Temporales Cualitativas se ajustan a las necesidades de representación temporal que requiere la interpretación y resuelven los problemas que descartan el Cálculo de Situación y STRIPS para el Nivel de Interpretación.

El trabajo de Allen y Ferguson no contempla la interpretación entre sus objetivos. Sin embargo, analizando el trabajo de Pinhanez (1999) sí que se puede observar la utilidad de los intervalos para interpretar la actividad. Su propuesta está orientada a reconocer la actividad de una persona que está siendo capturada en vídeo. Concretamente, plantea un ejemplo práctico en el que es capaz de reconocer los pasos que lleva a cabo un cocinero mientras prepara una receta. Su representación describe inicialmente cada paso mediante relaciones de Allen entre “detectores” que podrían ser equivalentes a las observaciones que definimos en este trabajo. Posteriormente, simplifica las relaciones de Allen dejando únicamente tres relaciones entre intervalos: Pasado, Presente, Futuro. El conjunto de relaciones forma una red de detectores denominada red-PNF (*Past-Now-Future network*) que describe un paso. Esta simplificación es válida cuando lo único que se persigue es saber si en un instante de tiempo está sucediendo algún intervalo obtenido por los detectores, pero sin necesidad de realizar ninguna predicción sobre lo que pasará a continuación. En nuestro caso, el Nivel de Interpretación sigue la misma filosofía. Sin embargo, reducir de 13 a 3 las posibles relaciones entre intervalos conlleva pérdidas de información sobre la actividad, con lo que la utilidad de su propuesta dependerá del uso que se quiera hacer de los resultados de la

interpretación. En este trabajo, los resultados de la interpretación serán diagnosticados posteriormente, con lo que la pérdida de información no resulta admisible.

En conclusión, el planteamiento de Pinhanez para representar la actividad mediante la simplificación de restricciones temporales cualitativas es adecuado para interpretar siempre que perder información no sea un problema. Puesto que en nuestro caso sí que lo es, no sería adecuado utilizar su simplificación. Además, nuestra propuesta difiere de la suya en que incorpora más tipos de restricciones que enriquecen la capacidad de representación de la actividad del Nivel de Interpretación.

## **B.2 ANTECEDENTES DE REPRESENTACIÓN E INTERPRETACIÓN DE LA ACTIVIDAD MEDIANTE RESTRICCIONES TEMPORALES CUANTITATIVAS**

Las Restricciones Temporales Cuantitativas enriquecen la capacidad para representar e interpretar una actividad respecto a las posibilidades ofrecidas por las Restricciones Temporales Cualitativas. Además, el impacto positivo viene refrendado por antecedentes en los que se han empleado restricciones temporales cuantitativas para interpretación. En este apartado se presentan dichos antecedentes.

Dousson et al. (1993) plantean un sistema de reconocimiento de situaciones basado en restricciones temporales cuantitativas sobre puntos. El sistema recibe como entrada un flujo de eventos marcados temporalmente y busca patrones que coincidan con la representación de situaciones modeladas previamente. En el código bajo este párrafo se muestra un ejemplo de situación que modela el trabajo de un robot que tiene que entrar en una habitación, cargar una máquina y salir. En cuanto la máquina está cargada puede empezar a funcionar. La máquina tiene que estar parada antes de poder cargarla y debe haber condiciones de seguridad en la sala para que el robot pueda realizar cualquier acción.

```
situation RobotLoadMachine {  
  ON(RobotInRoom, e1);  
  OFF(RobotInRoom, e4);  
  ON(MachineLoaded, e2);  
  ON(MachineRunning, e3);
```

```
e1<e2;  
1" <= e3-e2 <=6";  
3" <= e4-e2 <=5";  
FALSE(MachineRunning,e2);  
TRUE(SafetyCondition,e1,e4);  
when recognized  
    emit ON(SuccesfulLoad,e2);  
}
```

Esta representación de situaciones, también llamadas crónicas, se ha empleado para la monitorización y detección de averías de turbinas de gas (Aguilar et al., 1994), el seguimiento de síntomas en pacientes médicos para descubrir posibles enfermedades (Ramaux et al., 1997), y para la monitorización de redes de telecomunicaciones (Dousson, 2002; Dousson y Maigat, 2007).

En resumen, las restricciones temporales cuantitativas son útiles para representar actividad que haya que interpretar, independientemente del dominio. Sin embargo, hay que tener en cuenta que al cuantificar una relación entre dos Instantes Temporales se está definiendo una relación rígida. Por ejemplo, en la crónica anterior se define una restricción de entre 3 y 5 segundos desde que la máquina está cargada hasta que el robot sale de la sala. En principio, no debería haber problemas para determinar las cotas de la restricción, ya que el comportamiento del robot será predecible. Sin embargo, en otros casos, puede ser complicado o imposible describir una acción cuantificando las relaciones entre observaciones. Por esta razón, en el Nivel de Interpretación, las restricciones temporales cualitativas tienen prioridad sobre las cuantitativas, y estas últimas tienen el papel de complementar a las cualitativas, ya que son mucho más flexibles.





## *ANEXO C*

# ***REGLAS DE EVALUACIÓN DE RESTRICCIONES***

---

En este anexo se presentan las reglas definidas para evaluar Restricciones. El anexo comienza con la descripción gráfica de las relaciones temporales que se pueden dar entre dos intervalos cuando alguno de ellos o los dos aún están abiertos y cuando alguno de los dos aún no ha sido observado. Las dos siguientes secciones describen las reglas que determinan cómo se evalúan dichas relaciones temporales mediante  $RTC_{Allen}$  y  $RTC_{Puntual}$ . Seguidamente, se presentan las tablas de reglas para evaluar Restricciones Lógicas, y para finalizar, el último apartado contiene las reglas de evaluación de  $RTC_{Cuantitativa}$  junto al algoritmo de cálculo de Duración del intervalo que separa dos Instantes Temporales.

### **C.1 REPRESENTACIÓN GRÁFICA DE RELACIONES ENTRE INTERVALOS**

La Figura B.7.1 representa las relaciones temporales cualitativas que se pueden dar entre dos intervalos (verde y azul) cuando alguno de ellos o los dos están abiertos en el instante  $t_i$ . Los intervalos abiertos se representan mediante un rectángulo que no tiene cerrado su lado derecho, mientras que los finalizados están completamente cerrados. En las tablas de las dos secciones posteriores se emplea la misma numeración para identificar cada relación temporal.

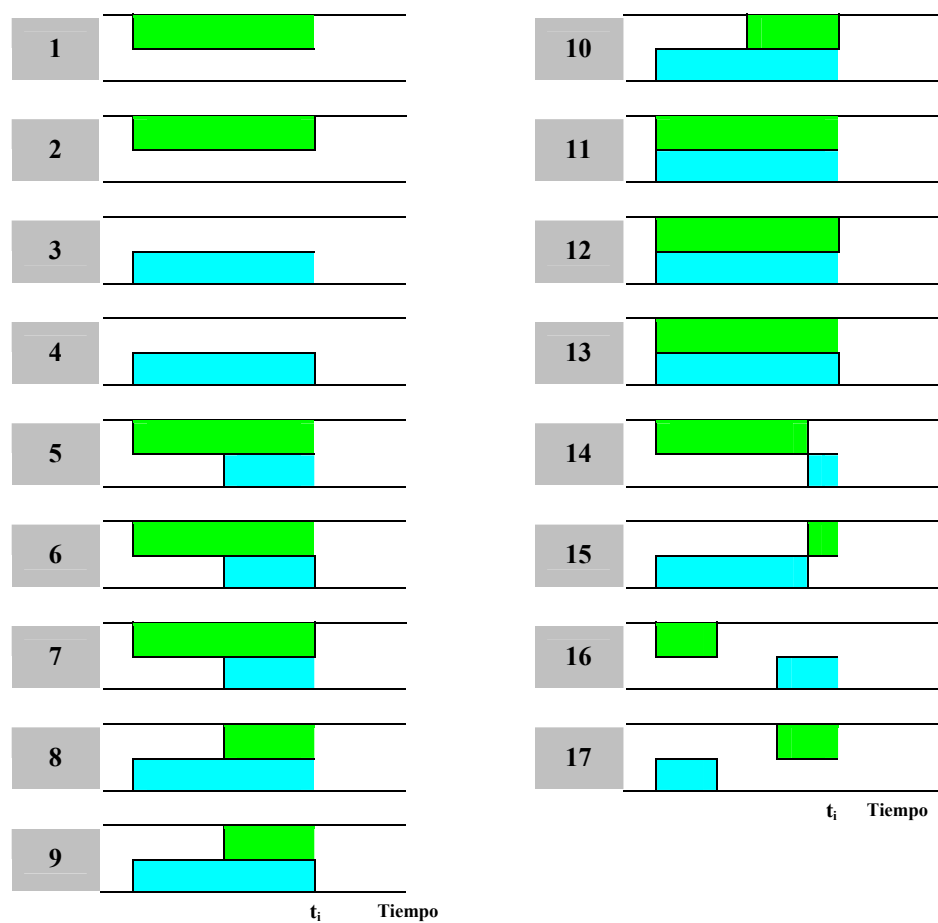


Figura B.7.1: Relaciones temporales entre intervalos abiertos

## C.2 EVALUACIÓN DE RESTRICCIONES TEMPORALES CUALITATIVAS ALLEN

En las Tabla B.1 y B.2 se muestran las reglas de evaluación de RTC<sub>Allen</sub> para todas las relaciones temporales enumeradas en la sección anterior. Las filas muestran el resultado de evaluar cada relación temporal con cada operador básico de Allen. La evaluación de la Restricción se realiza en el orden “Intervalo verde”[Relación Allen]“Intervalo azul”.

	1	2	3	4	5	6	7	8	9
<b>precedes(p)</b>	ED	ED	F	F	F	F	F	F	F
<b>overlaps(o)</b>	ED	F	F	F	EF	F	V	F	F
<b>meets(m)</b>	ED	F	F	F	F	F	F	F	F
<b>starts(s)</b>	F	F	F	F	F	F	F	F	F
<b>finishes(f)</b>	F	F	ED	F	F	F	F	EF	F
<b>equals(e)</b>	F	F	F	F	F	F	F	F	F
<b>during(d)</b>	F	F	ED	F	F	F	F	EF	F
<b>contains (D)</b>	ED	F	F	F	EF	V	F	F	F
<b>is finished by(F)</b>	ED	F	F	F	EF	F	F	F	F
<b>is started by(S)</b>	F	F	F	F	F	F	F	F	F
<b>is overlapped by(O)</b>	F	F	ED	F	F	F	F	EF	V
<b>is met by (M)</b>	F	F	ED	F	F	F	F	F	F
<b>is preceded by(P)</b>	F	F	ED	ED	F	F	F	F	F

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.1: Reglas de evaluación de RTCAllen (relaciones 1-9)

	10	11	12	13	14	15	16	17
<b>precedes(p)</b>	F	F	F	F	F	F	V	F
<b>overlaps(o)</b>	F	F	F	F	F	F	F	F
<b>meets(m)</b>	F	F	F	F	V	F	F	F
<b>starts(s)</b>	F	EF	V	F	F	F	F	F
<b>finishes(f)</b>	F	F	F	F	F	F	F	F
<b>equals(e)</b>	F	EF	F	F	F	F	F	F
<b>during(d)</b>	V	F	F	F	F	F	F	F
<b>contains (D)</b>	F	F	F	F	F	F	F	F
<b>is finished by(F)</b>	F	F	F	F	F	F	F	F
<b>is started by(S)</b>	F	EF	F	V	F	F	F	F
<b>is overlapped by(O)</b>	F	F	F	F	F	F	F	F
<b>is met by (M)</b>	F	F	F	F	F	V	F	F
<b>is preceded by(P)</b>	F	F	F	F	F	F	F	V

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.2: Reglas de evaluación de RTCAllen (relaciones 10-17)

### C.3 EVALUACIÓN DE RESTRICCIONES TEMPORALES CUALITATIVAS PUNTUALES

En las Tabla B.3 y B.4 se muestran las reglas de evaluación de RTCPuntual para cada relación temporal enumerada en la sección B.1. Las filas muestran el resultado de evaluar todas las relaciones temporales con cada restricción posible entre los instantes de inicio y fin de los dos intervalos. Los Instantes Temporales de los intervalos se representan como:

- $I1$  = inicio intervalo verde.
- $F1$  = fin intervalo verde.
- $I2$  = inicio intervalo azul.
- $F2$  = fin intervalo azul.

	1	2	3	4	5	6	7	8	9
$I1 \leq I2$	ED	ED	F	F	V	V	V	F	F
$I1 < I2$	ED	ED	F	F	V	V	V	F	F
$I1 = I2$	F	F	F	F	F	F	F	F	F
$I1 \geq I2$	F	F	ED	ED	F	F	F	V	V
$I1 > I2$	F	F	ED	ED	F	F	F	V	V
$I1 \leq F2$	ED	ED	ED	ED	V	V	V	V	V
$I1 < F2$	ED	ED	ED	F	V	V	V	V	V
$I1 = F2$	F	F	ED	ED	F	F	F	F	F
$I1 \geq F2$	F	F	ED	ED	F	F	F	F	F
$I1 > F2$	F	F	ED	ED	F	F	F	F	F
$F1 \leq I2$	ED	ED	F	F	F	F	F	F	F
$F1 < I2$	ED	ED	F	F	F	F	F	F	F
$F1 = I2$	ED	ED	F	F	F	F	F	F	F
$F1 \geq I2$	ED	F	ED	ED	V	V	V	V	V
$F1 > I2$	ED	F	ED	ED	V	V	V	V	V
$F1 \leq F2$	ED	ED	ED	F	EF	F	V	EF	F
$F1 < F2$	ED	ED	ED	F	EF	F	V	EF	F
$F1 = F2$	ED	F	ED	F	EF	F	F	EF	F
$F1 \geq F2$	ED	F	ED	ED	EF	V	F	EF	V
$F1 > F2$	ED	F	ED	ED	EF	V	F	EF	V

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.3: Reglas de evaluación de RTCPuntual (relaciones 1-9)

	10	11	12	13	14	15	16	17
<b>I1 &lt;= I2</b>	F	V	V	V	V	F	V	F
<b>I1 &lt; I2</b>	F	F	F	F	V	F	V	F
<b>I1 = I2</b>	F	V	V	V	F	F	F	F
<b>I1 &gt;= I2</b>	V	V	V	V	F	V	F	V
<b>I1 &gt; I2</b>	V	F	F	F	F	V	F	V
<b>I1 &lt;= F2</b>	V	V	V	V	V	V	V	F
<b>I1 &lt; F2</b>	V	V	V	V	V	F	V	F
<b>I1 = F2</b>	F	F	F	F	F	V	F	F
<b>I1 &gt;= F2</b>	F	F	F	F	F	V	F	V
<b>I1 &gt; F2</b>	F	F	F	F	F	F	F	V
<b>F1 &lt;= I2</b>	F	F	F	F	V	F	V	F
<b>F1 &lt; I2</b>	F	F	F	F	F	F	V	F
<b>F1 = I2</b>	F	F	F	F	V	F	F	F
<b>F1 &gt;= I2</b>	V	V	V	V	V	V	F	V
<b>F1 &gt; I2</b>	V	V	V	V	F	V	F	V
<b>F1 &lt;= F2</b>	V	EF	V	F	V	F	V	F
<b>F1 &lt; F2</b>	V	EF	V	F	V	F	V	F
<b>F1 = F2</b>	F	EF	F	F	F	F	F	F
<b>F1 &gt;= F2</b>	F	EF	F	V	F	V	F	V
<b>F1 &gt; F2</b>	F	EF	F	V	F	V	F	V

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.4: Reglas de evaluación de RTCPuntual (relaciones 10-17)

## C.4 EVALUACIÓN DE RESTRICCIONES LÓGICAS

En esta sección se presentan las reglas de evaluación de restricciones lógicas. Se presenta una tabla para cada operador lógico (AND, OR, NOT) (Tabla B.5-B.7). Cada una muestra el resultado de evaluar la RestriccionLogica  $R1[operador]R2$  por cada posible evaluación de  $R1$  (filas) y de  $R2$  (columnas).

AND	ED	EF	ND	V	F
ED	ED	ED	F	ED	F
EF	ED	ED	ED	EF	F
ND	ED	ED	ND	ED	F
V	ED	EF	ED	V	F
F	F	F	F	F	F

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos	<b>ND</b>	NoDatos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final		

Tabla B.5: Reglas de evaluación de RestriccionLogica (AND)

OR	ED	EF	ND	V	F
ED	ED	EF	ED	V	ED
EF	EF	EF	EF	V	EF
ND	ED	EF	ND	V	ND
V	V	V	V	V	V
F	ED	EF	ND	V	F

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos	<b>ND</b>	NoDatos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final		

Tabla B.6: Reglas de evaluación de RestriccionLogica (OR)

NOT	ED	EF	ND	V	F
	ED	EF	V	F	V

<b>V</b>	Verdadero	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.7: Reglas de evaluación de RestriccionLogica (NOT)

## C.5 EVALUACIÓN DE RESTRICCIONES TEMPORALES CUANTITATIVAS

El proceso de evaluación de una RTCuantitativa tiene dos fases (Sección 4.4.2.1). En primer lugar se intenta calcular por separado las dos duraciones restringidas, utilizando para ello el algoritmo presentado el siguiente apartado. Después se evalúa la RTCuantitativa en función de las duraciones obtenidas siguiendo las reglas de evaluación recogidas en el apartado posterior.

➤ **Cálculo de duración entre dos instantes temporales**

A continuación se presenta el algoritmo de cálculo de duración entre dos instantes temporales. En el pseudocódigo se repiten dos condiciones con el siguiente significado:

- *Si (existe I.Intervalo)*: Esta condición se cumple cuando el intervalo que contiene el instante temporal ha sido detectado.
- *Si (I.Instante es válido)*: I.Instante es válido si el instante temporal referencia a un inicio de intervalo y el intervalo ha empezado o finalizado. También es válido si I.Instante referencia a un fin de intervalo y éste ha finalizado.

```

1  CalcularDuracion(Instante I1, Instante I2, out: d)
2  {
3      Si (existe I1.Intervalo)
4      {
5          Si (I1.Instante es válido)
6          {
7              Si (existe I2. Intervalo)
8              {
9                  Si (I2.Instante es válido)
10             {
11                 d=I2.Instante-I1.Instante;
12                 return Verdadero;
13             }
14             Si no //I2.Instante no es válido
15             {
16                 d=I2.Instante-I1.Instante;
17                 return Esperar_Final;
18             }
19         }
20         Si no //I2.Intervalo no existe
21         {
22             d=null;
23             return Esperar_Datos;
24         }
25     }
26     Si no //I1.Instante no es válido
27     {
28         Si (existe I2. Intervalo)
29         {
30             Si (I2.Instante es válido)
31             {

```





➤ **Reglas de evaluación de Restricciones Temporales Cuantitativas**

En las Tabla B.8-B.12 se presentan las reglas que determinan el resultado de evaluar una RTCuantitativa que relaciona los resultados posibles del cálculo de dos duraciones: Verdadero+ duración calculada, Falso, No\_Datos, Esperar\_Datos, Esperar\_Final. Se muestra una tabla por cada operador (<, >, <=, >=, =). Cada una muestra el resultado de evaluar la RTCuantitativa *Duracion1[operador]Duracion2* según el resultado del cálculo de Duracion1 (filas) y el de Duracion2 (columnas).

<	V	F	ND	ED	EF
V	d1<d2	F	ED	ED	d2==null->EF (d1<d2)->V !(d1<d2)->EF
F	F	F	F	F	F
ND	ED	F	ND	ED	ED
ED	ED	F	ED	ED	ED
EF	d1==null->EF (d1<d2)->EF !(d1<d2)->F	F	ED	ED	EF

<b>ND</b>	No_Datos	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.8: Reglas de evaluación de RTCuantitativa (operador <)

>	V	F	ND	ED	EF
V	d1>d2	F	ED	ED	d2==null->EF (d1>d2)->EF !(d1>d2)->F
F	F	F	F	F	F
ND	ED	F	ND	ED	ED
ED	ED	F	ED	ED	ED
EF	d1==null->EF (d1>d2)->V !(d1>d2)->EF	F	ED	ED	EF

<b>ND</b>	No_Datos	<b>ED</b>	Esperar_Datos
<b>F</b>	Falso	<b>EF</b>	Esperar_Final

Tabla B.9: Reglas de evaluación de RTCuantitativa (operador >)

<=	V	F	ND	ED	EF
V	$d1 \leq d2$	F	ED	ED	$d2 == \text{null} \rightarrow \text{EF}$ $(d1 \leq d2) \rightarrow V$ $!(d1 \leq d2) \rightarrow \text{EF}$
F	F	F	F	F	F
ND	ED	F	ND	ED	ED
ED	ED	F	ED	ED	ED
EF	$d1 == \text{null} \rightarrow \text{EF}$ $(d1 \leq d2) \rightarrow \text{EF}$ $!(d1 \leq d2) \rightarrow F$	F	ED	ED	EF

ND	No_Datos	ED	Esperar_Datos
F	Falso	EF	Esperar_Final

Tabla B.10: Reglas de evaluación de RTCuantitativa (operador &lt;=)

>=	V	F	ND	ED	EF
V	$d1 \geq d2$	F	ED	ED	$d2 == \text{null} \rightarrow \text{EF}$ $(d1 \geq d2) \rightarrow \text{EF}$ $!(d1 \geq d2) \rightarrow F$
F	F	F	F	F	F
ND	ED	F	ND	ED	ED
ED	ED	F	ED	ED	ED
EF	$d1 == \text{null} \rightarrow \text{EF}$ $(d1 \geq d2) \rightarrow V$ $!(d1 \geq d2) \rightarrow \text{EF}$	F	ED	ED	EF

ND	No_Datos	ED	Esperar_Datos
F	Falso	EF	Esperar_Final

Tabla B.11: Reglas de evaluación de RTCuantitativa (operador &gt;=)

=	V	F	ND	ED	EF
V	$d1 = d2$	F	ED	ED	EF
F	F	F	F	F	F
ND	ED	F	ND	ED	ED
ED	ED	F	ED	ED	ED
EF	EF	F	ED	ED	EF

ND	No_Datos	ED	Esperar_Datos
F	Falso	EF	Esperar_Final

Tabla B.12: Reglas de evaluación de RTCuantitativa (operador =)

## *ANEXO D*

# *GRAMÁTICA DEL LENGUAJE DEL NIVEL DE INTERPRETACIÓN*

---

Este anexo contiene la gramática del lenguaje definido en el Nivel de Interpretación del metamodelo de integración (Sección 3.4.4). Mediante este lenguaje se especifican todos los pasos y situaciones que tenga que interpretar el Subsistema de Interpretación. La gramática está definida en el formato estándar EBNF.

```
IKB          : situations steps
              ;

Steps        : step
              | steps step
              ;

Situations   : situation
              | situations situation
              ;

situation    : TokSituation SituationName OBRACE Declarations
              StartConstraints EndConstraints GeneralConstraints PossibleSteps
              TimeUnit EBRACE
              ;

step         : TokStep StepName OBRACE Declarations StartConstraints
              EndConstraints ActionConstraints GeneralConstraints TimeUnit
              EBRACE
              ;

SituationName : Word
              ;

StepName     : Word
              ;
```

```

Declarations      : IntervalName COLON IntervalInstances SEMICOLON
                  | Declarations IntervalName COLON IntervalInstances SEMICOLON
                  ;

IntervalInstances : IntervalInstance
                  | IntervalInstances COMMA IntervalInstance
                  ;

IntervalInstance  : Word
                  ;

StartConstraints  : StartConstraint
                  | StartConstraints StartConstraint
                  ;

StartConstraint   : TokStart Constraint
                  | TokStart StepName
                  ;

EndConstraints    : EndConstraint
                  | EndConstraints EndConstraint
                  ;

EndConstraint     : TokEnd Constraint
                  | TokEnd StepName
                  ;

ActionConstraints : ActionConstraint
                  | ActionConstraints ActionConstraint
                  ;

ActionConstraint  : TokAction Constraint
                  | TokAction StepName
                  ;

GeneralConstraints : TokConstraint Constraint
                  | GeneralConstraints TokConstraint Constraint
                  ;

PossibleSteps     : TokPossibleSteps StepNames
                  ;

StepNames         : StepName
                  | StepNames COMMA StepName
                  ;

TimeUnit          : TokTimeUnitDef Word
                  ;

LogicConstraint   : OPARENTHESIS LogicConstraint EPARENTHESIS
                  | Constraint BoolOper Constraint
                  | NOT Constraint
                  | Constraint BoolOper BoolValue
                  | BoolValue BoolOper Constraint
                  ;

Constraint        : Expression CompOperator Expression
                  | DateTimeExpression CompOperator DateTimeExpression
                  | Duration CompOperator Duration
                  | IntervalInstance OBRACKET AllenOperands EBRACKET
                  | IntervalInstance
                  | LogicConstraint
                  | IntervalInstance
                  | Function
                  ;

```

---

```
AllenOperands      : AllenOperand
                   | AllenOperands COMMA AllenOperand
                   ;

AllenOperand       : Precedes
                   | Overlaps
                   | Meets
                   | Starts
                   | Finishes
                   | During
                   | Equals
                   | Contains
                   | Is_Finished_By
                   | Is_Started_By
                   | Is_Overlapped_By
                   | Is_Met_By
                   | Is_Preceded_By
                   ;

BoolOper          : AND
                   | OR
                   ;

BoolValue         : TRUE
                   | FALSE
                   ;

CompOperator      : GE
                   | LE
                   | EQ
                   | NE
                   | GT
                   | LT
                   ;

Expression        : IntervalProperty
                   | ObservationProperty
                   | FLOAT
                   | INTEGER
                   | Function
                   | ArithmeticExpression
                   ;

ArithmeticExpression : OPARENTHESIS ArithmeticExpression EPARENTHESIS
                    | Expression NumOperator Expression
                    | '-' Expression %prec UMINUS
                    ;

NumOperator       : '+'
                   | '-'
                   | '*'
                   | '/'
                   | '%'
                   ;

Function          : Word OPARENTHESIS parameters EPARENTHESIS
                   ;

parameters       : parameter
                   | parameters COMMA parameter
                   ;

parameter        : IntervalMomentum
                   | IntervalProperty
                   | IntervalInstance
                   | Expression
                   ;
```

```
DateTimeExpression : OPARENTHESIS DateTimeExpression EPARENTHESIS
                   | IntervalMomentum
                   ;

Duration           : TokDuration OPARENTHESIS IntervalMomentum COMMA
                   IntervalMomentum EPARENTHESIS
                   | TokDuration OPARENTHESIS IntervalInstance EPARENTHESIS
                   | INTEGER
                   ;

IntervalMomentum  : OPARENTHESIS IntervalMomentum EPARENTHESIS
                   | IntervalInstance IntervalDTProp
                   ;

IntervalProperty  : IntervalInstance IntervalNumProp
                   ;

ObservationProperty: IntervalInstance DOT Word
                   ;

IntervalDTProp    : DOT StartIntervalProp
                   | DOT EndIntervalProp
                   ;

IntervalNumProp   : DurationIntervalProp
                   | FromBeginningIntervalProp
                   ;

IntervalName      : Word
                   | Word FOURDOTS Word
                   ;
```

## *ANEXO E*

# *ACRÓNIMOS*

---

- ADL: Advanced Distributed Learning Initiative
- AI-CAI: véase ICAI
- AIWBES: Adaptive and Intelligent Web-Based Educational System
- CAI: Computer-Assisted Instruction
- CNC: Computer Numerical Controlled
- DC: Dependencia Conceptual
- EV: Entorno Virtual
- FIPA: Foundation for Intelligent Physical Agents
- IA: Inteligencia Artificial
- ICAI: Intelligent Computer-Assisted Instruction
- LMS: Learning Management System
- LTSC: IEEE Learning Technologies Standards Committee
- RA: Realidad Aumentada
- RM: Realidad Mixta
- RSRT: Red de Satisfacción de Restricciones Temporales
- RSRTP: Red de Satisfacción de Restricciones Temporales entre Puntos
- RV: Realidad Virtual



- SCORM: Sharable Content Object Reference Model
- SI<sup>2</sup>A<sup>2</sup>: Sistema Interactivo Inteligente de Ayuda al Aprendizaje
- SISO: Simulation Interoperability Standards Organization
- STEVE: Soar Training Expert for Virtual Environments
- STI: Sistema Tutor Inteligente
- VA: Virtualidad Aumentada

# *ANEXO F*

## *PUBLICACIONES*

---

En este anexo se indican las publicaciones y comunicaciones a congresos que se han generado durante este trabajo de tesis.

### **F.1 ARTÍCULOS EN REVISTA**

Ustarroz, A., Lozano, A., Matey, L., Siemon, J., Klockmann, D., y Berasategui, M. I., "VIRTOOL - Virtual Reality for Machine-Tool Training", *Mécanique & Industries*, vol. 5, Pp. 207-212, 2004.

Lozano, A., Urretavizcaya, M., Ferrero, B., Fernández-Castro, I., Ustarroz, A., y Matey, L., "Integration of a Generic Diagnostic Tool in Virtual Environments for Procedural Training", en *Lecture Notes in Artificial Intelligence*, vol. 3040, 2004, pp. 666-675. (Indexada)

Díaz, I., Hernantes, J., Mansa, I., Lozano, A., Borro, D., Gil, J. J., y Sánchez, E., "Influence of Multisensory Feedback on Haptic Accessibility Tasks", *Virtual Reality, Special Issue on Multisensory Interaction in Virtual Environments*, vol. 10 (1), Pp. 31-40, 2006.

### **F.2 COMUNICACIONES A CONGRESOS**

Lozano, A., Urretavizcaya, M., Ferrero, B., Fernández-Castro, I., y Matey, L., "Integración de una Herramienta Genérica de Diagnóstico en Entornos

- Virtuales de Entrenamiento Procedimental", en *actas de X CAEPIA - V TTIA*, Pp. 399-408. Donostia, 2003. (Galardonado con mención especial en la categoría de Transferencia Tecnológica en Inteligencia Artificial)
- Lozano, A., Urretavizcaya, M., Ferrero, B., Fernández-Castro, I., y Matey, L., "VIRTOOL-D: Entorno Virtual Educativo para Aprendizaje en Dominios Procedimentales", en *actas de 5º SIIE - Challenges 2003*. Braga, 2003.
- Lozano, A., Ustarroz, A., Matey, L., Unzueta, L., y Sánchez, E., "Virtual reality enhanced with diagnostic capabilities for training machine-tool processes", en *actas de 1st Enactive Workshop*, 2005.
- Unzueta, L., Berselli, G., Cazón, A., Lozano, A., y Suescun, Á., "Genetic Algorithms Applications to the Reconstruction of the Human Motion using a Noninvasive Motion Capture", en *actas de Multibody Dynamics 2005, ECCOMAS Thematic Conference*, Pp. 17. Madrid, 2005.
- López-Garate, M., Lozano-Rodero, A., y Matey, L., "An Adaptive and Customizable Feedback System for Intelligent Interactive Learning Systems", en *actas de 9th International Conference on Intelligent Tutoring Systems (ITS'08)*. Montreal (Canada), 2008.
- López-Garate, M., Lozano-Rodero, A., y Matey, L., "An adaptive and customizable feedback system for VR-based training simulators (short paper)", en *actas de 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Pp. 1635-1638. Estoril (Portugal), 2008.
- López-Garate, M., Lozano-Rodero, A., y Matey, L., "Intelligent feedback for simulation based training", en *actas de 6th International Conference on Education and Information Systems, Technologies and Applications (EISTA 2008)*. Orlando, USA, 2008.

## ***BIBLIOGRAFÍA***

---

- "IEEE Standard for Distributed Interactive Simulation - Application Protocols," en *IEEE Std 1278.1- 1995*, Institute of Electrical and Electronics Engineers, NY, 1995.
- "IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA)—Framework and Rules," en *IEEE Std 1516-2000*, Institute of Electrical and Electronics Engineers, NY, 2000.
- Aggarwal, R., Grantcharov, T., Moorthy, K., Hance, J., y Darzi, A., "A competency-based virtual reality training curriculum for the acquisition of laparoscopic psychomotor skill", *The American Journal of Surgery*, vol. 191 (1), Pp. 128, 2006.
- Aguilar, J., Bousson, K., Dousson, C., Ghallab, M., Guasch, A., Milne, R., Nicol, C., Quevedo, J., y Travé-Massuyès, L., "TIGER: real-time situation assessment of dynamic systems", *Intelligent systems engineering*, Pp. 103-124, 1994.
- Al-Shihabi, T. y Mourant, R. R., "A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators", en *actas de International Conference on Autonomous Agents*, Pp. 286-291, 2001.
- Almond, R., Steinberg, L., y Mislevy, R., "Enhancing the design and delivery of assessment systems: A four-process architecture", *Journal of Technology, Learning, and Assessment*, vol. 1 (5), Pp. 2-63, 2002.
- Álvarez, A., Ruiz, S., Martín, M., Fernandez-Castro, I., y Urretavizcaya, M., "MAGADI: a Blended-Learning Framework for Overall Learning", en *actas de AIED 2009*. Brighton, 2009.
- Allen, J. F., "Maintaining knowledge about temporal intervals," en *Communications of the ACM*, vol. 26, 1983, pp. 832-843.

- Allen, J. F., "Towards a general theory of action and time", *Artificial Intelligence*, vol. 23, Pp. 123-154, 1984.
- Allen, J. F. y Ferguson, G., "Actions and Events in Interval Temporal Logic", *Journal of Logic and Computation*, vol. 4 (5), Pp. 531-579, 1994.
- Anderson, J. R., *The architecture of cognition*. Cambridge, MA: Harvard University Press, 1983.
- Anderson, J. R., *Rules of the mind*. Hillsdale: Erlbaum, 1993.
- Anderson, J. R., Boyle, F., Corbett, A. T., y Lewis, M. W., "Cognitive modeling and Intelligent Tutoring", *Artificial Intelligence*, vol. 42, Pp. 7-49, 1990.
- Anderson, J. R. y Lebiere, C., *The atomic components of thought*. Mahwah, NJ: Erlbaum, 1998.
- Anderson, J. R. y Reiser, B. J., "The LISP tutor", *BYTE*, vol. 10 (4), Pp. 159-175, 1985.
- Andriessen, J. y Sandberg, J., "Where is Education Heading and How about AI?" *International Journal of Artificial Intelligence in Education*, vol. 10, Pp. 130-150, 1999.
- Angros Jr., R., "Learning What to Instruct: Acquiring Knowledge from Demonstrations and Focussed Experimentation," Tesis doctoral, Dpto. University of Southern California, 2000.
- Anzai, Y. y Simon, H. A., "The Theory of Learning by Doing", *Psychological Review*, vol. 86, Pp. 124-140, 1979.
- ARIADNE, "ARIADNE Educational Metadata Recommendation - V2.0," 1998.
- Arruarte, A., Ferrero, B., Fernández-Castro, I., Urretavizcaya, M., y Greer, J., "The IRIS Authoring Tool", en *Authoring Tools for Advanced Technology Learning Environments*, 2003.
- Ausejo, S., "A new robust motion reconstruction method based on optimisation with redundant constraints and natural coordinates," Tesis doctoral, Dpto. Applied mechanics, Tecnun, University of Navarra, 2006.
- Ayala, G. y Yano, Y., "A collaborative learning environment based on intelligent agents", *Expert Systems with Applications*, vol. 14 (1), Pp. 129-137, 1998.
- Baffes, P. y Mooney, R., "Refinement-based student modeling and automated bug library construction", *International Journal of Artificial Intelligence in Education*, vol. 7 (1), Pp. 75-117, 1996.
- Baker, R. S. J. d., Barnes, T., y Beck, J. E., *Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings*. Montreal, Quebec, Canada, 2008.

- 
- Barnhart, B. y Ratvasky, T., "Effective Training for Flight in Icing Conditions", en *actas de IITSEC*. Orlando, FL, 2006.
- Bauer, M., "Using Evidence-Centered Design to Align Formative and Summative Assessment", en *actas de Evidence-centered design approach to creating diagnostic e-assessments (ITS'2002)*, Pp. 87-96, 2002.
- Beck, J., en actas de AAAI Conference on Artificial Intelligence. Workshop on Educational Data Mining. Pittsburgh, Pennsylvania, 2005.
- Beck, J., en actas de ITS'04 Workshop: Analyzing student-tutor interaction logs to improve educational outcomes. Maceió, Brazil, 2004.
- Beck, J., Aimeur, E., y Barnes, T., en actas de American Association for Artificial Intelligence (AAAI) Workshop on Educational Data Mining, 2006.
- Bernold, L. E., Lloyd, J., y Vouk, M., "Equipment operator training in the age of Internet2", en *actas de NIST, 19th International Symposium on Automation and Robotics in Construction*, Pp. 505-510. Gaithersburg, MD, 2002.
- Billinghurst, M., Savage-Carmona, J., Oppenheimer, P., y Edmond, C., "The Expert Surgical Assistant: An Intelligent Virtual Environment with Multimodal Input", en *actas de Medicine Meets Virtual Reality IV*, Pp. 590-607, 1995.
- Bin, M., Zhi-ying, G., y Hua-min, Z., "Development of a plastic injection molding training system using Petri nets and virtual reality", *Journal of Zhejiang University - Science A*, Springer, vol. 7 (3), Pp. 302-308, 2006.
- Blessing, S., "A Programming by Demonstration Authoring Tool for Model-Tracing Tutors", *International Journal of Artificial Intelligence in Education*, vol. 8, Pp. 233-261, 1997.
- Bloom, B. S., Engelhart, M. D., Murst, E. J., Hill, W. H., y Drathwohl, D. R., *Taxonomy of educational objectives. Handbook 1: Cognitive domain*: Longman, 1956.
- Borro, D., Savall, J., Amundarain, A., Gil, J. J., García-Alonso, A., y Matey, L., "A large haptic device for aircraft engines maintainability", *IEEE Computer Graphics and Applications*, vol. 24 (6), Pp. 70-74, 2004.
- Bowman, D. A., Kruijff, E., LaViola, J. J., y Poupyrev, I., *3D User Interfaces: Theory and Practice*: Addison-Wesley/Pearson Education, 2004.
- Bowman, D. A., Kruijff, E., LaViola, J. J., y Poupyrev, I., "An introduction to 3D user interface design", *Presence: Teleoperators and Virtual Environments*, vol. 10 (1), Pp. 96-108, 2001.
- Brown, J. S. y Burton, R. B., "Diagnostic models for procedural bugs in basic mathematical skills", *Cognitive science*, vol. 2, Pp. 155-192, 1978.

- Brown, J. S., Burton, R. R., y Zdybel, F., "A model-driven question-answering system for mixed-initiative computer-assisted instruction", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3 (1), Pp. 248-257, 1973.
- Brusilovsky, P., "Adaptive and Intelligent Technologies for Web-based Education", *Künstliche Intelligenz*, vol. 4, Pp. 19-25, 1999.
- Brusilovsky, P. y Peylo, C., "Adaptive and intelligent web-based educational systems", *International Journal of Artificial Intelligence in Education*, vol. 13, Pp. 156-169, 2003.
- Brusilovsky, P., Sosnovsky, S., y Yudelsohn, M., "Ontology-based Framework for User Model Interoperability in Distributed Learning Environments", *World Conference on E-Learning, E-Learn*, Pp. 2851-2855, 2005.
- Burdea, G. C., "Virtual Reality Systems and Applications", *en actas de Electro'93*, 1993.
- Burdea, G. C. y Coiffet, P., *Virtual Reality Technology*, 2nd edition ed: Hoboken: J. Wiley-Interscience, 2003.
- Burton, R. B., "Diagnosing bugs in a simple procedural skill", *en actas de Intelligent Tutoring Systems*, Pp. 157-183, 1982.
- Burton, R. B. y Brown, J. S., "An investigation of computer coaching for informal learning activities", *International Journal of Man-Machine Studies*, vol. 11, Pp. 5-24, 1979.
- Çakmak, H., Maass, H., y Künapfel, U., "VSOOne, a virtual reality simulator for laparoscopic surgery", *Minimally Invasive Therapy.*, Taylor & Francis, vol. 14 (3), Pp. 134-144, 2005.
- Carbonell, J. y Hood, G., "The world modelers project: objectives and simulator architecture", *Knowledge Representation, Learning And Expert Systems*, Kluwer Academic Publishers Norwell, MA, USA, Pp. 29-34, 1986.
- Carbonell, J. R., "AI in CAI: An Artificial Intelligence approach to Computer-Assisted Instruction", *IEEE Transactions on Man Machine Systems*, vol. 11 (4), Pp. 190-202, 1970.
- Carter, F. J., Schijven, M. P., Aggarwal, R., Grantcharov, T., Francis, N. K., Hanna, G. B., y Jakimowicz, J. J., "Consensus guidelines for validation of virtual reality surgical simulators", *Surgical Endoscopy*, Springer Verlag, vol. 19, Pp. 1523-1532, 2005.
- Carter, F. J., Schijven, M. P., Aggarwal, R., Grantcharov, T., Francis, N. K., Hanna, G. B., y Jakimowicz, J. J., "Consensus guidelines for validation of virtual reality surgical simulators (Update tables validation, Dec 2006)," Work Group for Evaluation and Implementation of Simulators and Skills Training Programmes (EAES) 2006.

- 
- Castro, M. B., "Nueva metodología de control no lineal basada en lógica difusa," Tesis doctoral, Dpto. Applied Mechanics, Tecnun, Universidad de Navarra, 2007.
- Clancey, W. J., "Tutoring rules for guiding a case method dialog", *International Journal of Man-Machine Studies*, vol. 11, Pp. 25-49, 1979.
- Clark, R. E., "When teaching kills learning: Research on mathemathantics", *Learning and instruction: European research in an international context*, Pergamon Press, vol. 2, Pp. 1-22, 1988.
- Conati, C. y Klawe, M., "Socially Intelligent Agents to Improve the Effectiveness of Educational Games." *en actas de AAAI Fall Symposium on Socially Intelligent Agents - The human in the loop*, 2000.
- Cosman, P. H., Cregan, P. C., Martin, C. J., y Cartmill, J. A., "Virtual reality simulators: Current status in acquisition and assessment of surgical skills", *ANZ Journal of Surgery*, Blackwell Synergy, vol. 72 (1), Pp. 30-34, 2002.
- Cox, M. T. y Ram, A., "Choosing learning strategies to achieve learning goals", *en actas de AAAI Spring Symposium on Goal-Driven Learning*, Pp. 12-21, 1994.
- Chan, T. W., "Learning Companion Systems, Social Learning Systems, and Intelligent Virtual Classroom." *International Journal of Artificial Intelligence in Education*, vol. 7 (2), Pp. 125-159, 1996.
- Choquet, C., Luengo, V., y Yacef, K., en actas de Artificial Intelligence in Education (AIED). Usage Analysis in Learning systems Workshop., 2005.
- Dave, R. H., *Developing and Writing Behavioural Objectives*: Educational Innovators Press, 1975.
- de Antonio, A., Ramírez, J., Imbert, R., Méndez, G., y Aguilar, R., "A Software Architecture for Intelligent Virtual Environments Applied to Education", *Revista de la Facultad de Ingeniería, Universidad de Tarapacá*, vol. 13 (1), 2005.
- Dechter, R., Meiri, I., y Pearl, J., "Temporal constraint networks", *Artificial Intelligence*, Elsevier Science Publishers Ltd., vol. 49 (1-3), Pp. 61-95, 1991.
- Díaz, I., Hernantes, J., Mansa, I., Lozano, A., Borro, D., Gil, J. J., y Sánchez, E., "Influence of Multisensory Feedback on Haptic Accesibility Tasks", *Virtual Reality, Special Issue on Multisensory Interaction in Virtual Environments*, vol. 10 (1), Pp. 31-40, 2006.
- Dillenbourg, P., *Collaborative learning: cognitive and computational approaches*. Amsterdam: Pergamon, 1999.
- Dodds, P., *Sharable Content Object Reference Model 2004: 3rd Edition Overview: Advanced Distributed Learning*, 2004.



- Dousson, C., "Extending and Unifying Chronicle Representation with Event Counters", *en actas de 15th ECAI*, Pp. 257-261. Lyon, France, 2002.
- Dousson, C., Gaborit, P., y Ghallab, M., "Situation recognition: Representation and algorithms", *en actas de 13th IJCAI*, Pp. 166-172. Chambéry, France, 1993.
- Dousson, C. y Maigat, P. L., "Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization", *en actas de IJCAI*, Pp. 324-329. Hyderabad, India, 2007.
- Dublin-Core, "DCMI Metadata Terms", DCMI, 2006.
- Ferrero, B., "DETECTive: un entorno genérico e integrable para diagnóstico de actividades de aprendizaje. Fundamento, diseño y evaluación," Tesis doctoral, Dpto. Univ. of the Basque Country UPV/EHU, 2004.
- Ferrero, B., Arruarte, A., Fernández-Castro, I., y Urretavizcaya, M., "Herramientas de autor para enseñanza y Diagnóstico: IRIS-D", *Inteligencia Artificial*, vol. 12, Pp. 13-28, 2001.
- Ferrero, B., Martín, M., Alvarez, A., Urretavizcaya, M., y Fernández-Castro, I., "Authoring and Diagnosis of Learning Activities with the KADDET Environment", *Journal of Universal Computer Science*, vol. 11 (9), Pp. 1530-1542, 2005.
- Fikes, R. E. y Nilsson, N. J., "STRIPS: A new approach to the application of theorem proving to problem solving", *Artificial Intelligence*, vol. 2, Pp. 198-208, 1971.
- Fisher, D. H., "Knowledge acquisition via incremental conceptual clustering", *Machine Learning*, Springer, vol. 2 (2), Pp. 139-172, 1987.
- Friesen, N. y McGreal, R., "Three Objections to Learning Objects", *en Online Education Using Learning Objects*: Routledge, 2004, pp. 70.
- Furness, T. A., "The Super Cockpit and Human Factors Challenges", *en actas de Human Factors Society 30th Annual Meeting*, 1986.
- Genesereth, M. R., "The role of plans in intelligent teaching systems", *en Intelligent Tutoring Systems*, D. Sleeman ed. New York: Academic Press, 1982, pp. 137-156.
- Gibbons, A. S., Nelson, J., Richards, R., y Wiley, D. A., "The nature and origin of instructional objects", *en The Instructional Use of Learning Objects*: AECT, 2002, pp. 58.
- Gopher, D., "The skill of attention control: Acquisition and execution of attention strategies". *Attention and Performance XIV: Synergies in Experimental Psychology, Artificial Intelligence, and Cognitive Neuroscience*, 1993, pp. 290-322.
- Green, M., *Report on Dialogue Specification Tools*: Springer Verlag, 1985.

- 
- Gugerty, L., "Non-diagnostic intelligent tutoring systems: Teaching without student models", *Instructional Science*, vol. 25 (6), Pp. 409-432, 1997.
- Harrow, A. J., A taxonomy of the psychomotor domain: a guide for developing behavioral objectives: David McKay, 1972.
- Heiner, C., Baker, R., y Yacef, K., en actas de ITS'06. Educational Data Mining Workshop. Taiwan, 2006.
- Heiner, C., Heffernan, N., y Barnes, T., en actas de AIED 2007. Educational Data Mining Workshop. Marina del Rey, CA, 2007.
- Heintz, F., Rudol, P., y Doherty, P., "From images to traffic behavior - A UAV tracking and monitoring application", en actas de 10th International Conference on Information Fusion, Pp. 1-8. Quebec City, QC, Canada, 2007.
- Hutchins, M. A., Stevenson, D. R., Gunn, C., Krumpholtz, A., Adriaansen, T., Pyman, B., y O'Leary, S., "Communication in a Networked Haptic Virtual Environment for Temporal Bone Surgery Training", *Virtual Reality*, vol. 9 (2-3), Pp. 97-107, 2006.
- IEEE, "The Learning Object Metadata standard," 2005.
- IMS, P., "IMS Learning Resource Meta-data Specification," IMS 2006.
- International Organization for Standardization, "ISO/IEC 14977: Information Technology - Syntactic metalanguage - Extended BNF," 1996.
- Jensen, R., Marshall, H., Stahl, J., y Stottler, R., "An Intelligent Tutoring System (ITS) for Future Combat Systems (FCS) Robotic Vehicle Command", en actas de I/ITSEC 2003, 2003.
- Jermann, P., Soller, A., y Muehlenbrock, M., "From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning", en actas de Conference on Computer-Supported Collaborative Learning. Maastricht, 2001.
- Joolingen, W. R. v. y Jong, T. d., "Supporting the authoring process for simulation-based discovery learning", en actas de European Conference on Artificial Intelligence in Education, Pp. 66-73. Brna, 1996.
- Kahl, J., Hotz, L., Milde, H., y Wessel, S., "A more efficient knowledge representation for Allen's algebra and point algebra", en actas de 12th international conference on Industrial and engineering applications of artificial intelligence and expert systems: multiple approaches to intelligent systems, Pp. 747-752. Cairo, Egypt, 1999.
- Kautz, H. y Allen, J. F., "Generalized plan recognition", en actas de Fifth National Conference on Artificial Intelligence, Pp. 32-38, 1986.

- Kearsly, G., "Explorations in Learning & Instruction: The Theory into Practice Database," 2001.
- Keretho, S. y Loganantharaj, R., "Qualitative and quantitative time interval constraint networks", *en actas de Proceedings of the 19th annual conference on Computer Science*, Pp. 239-246. San Antonio, Texas, United States, 1991.
- Koschmann, T., CSCL: Theory and practice of an emerging paradigm: Lawrence Erlbaum Assoc., 1996.
- Krathwohl, D. R., Bloom, B. S., y Masia, B. B., Taxonomy of Educational Objectives, the Classification of Educational Goals. Handbook II: Affective Domain. New York: Longman, 1964.
- Ladkin, P. B. y Maddux, R. D., "On binary constraint networks," Kestrel Institute, Palo Alto, CA, USA 1989.
- Laird, J., Newell, A., y Rosenbloom, P., "Soar: An architecture for general intelligence", *Artificial Intelligence*, vol. 33 (1), Pp. 1-64, 1987.
- LaViola, J. J., "A Review of Input and Output Devices for 3D Interaction", *en actas de Siggraph*, 2001.
- Lin, F., Ye, L., Duffy, V., y Su, C., "Developing Virtual Environments for Industrial Training", *Information Sciences*, vol. 140 (1), Pp. 153-170, 2002.
- Liu, A., Tendick, F., Cleary, K., y Kaufmann, C., "A Survey of Surgical Simulation: Applications, Technology, and Education", *Presence (MIT Press)*, vol. 12 (6), 2003.
- Loftin, R. B. y Savely, R. T., "Advanced Training Systems for the Next Decade and Beyond," NASA 1991.
- Loftin, R. B., Wang, L., Baffes, P. T., y Hua, G. C., "General purpose architecture for intelligent computer-aided training," The United States of America as represented by the Administrator of the National Aeronautics and Space Administration, 1994.
- López-Garate, M., Lozano-Rodero, A., y Matey, L., "An Adaptive and Customizable Feedback System for Intelligent Interactive Learning Systems", *en actas de 9th International Conference on Intelligent Tutoring Systems (ITS'08)*. Montreal (Canada), 2008a.
- López-Garate, M., Lozano-Rodero, A., y Matey, L., "An adaptive and customizable feedback system for VR-based training simulators (short paper)", *en actas de 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Pp. 1635-1638. Estoril (Portugal), 2008b.
- Los Arcos, J. L., Muller, W., Fuente, O., Orúe, L., Arroyo, E., Leanizbarrutia, I., y Santander, J., "LAHYSTOTRAIN: Integration of Virtual Environments and

- 
- ITS for Surgery Training", en *actas de ITS 2000*, Pp. 43-52. Montréal, Canada, 2000.
- Lowgren, J., "History, state and future of user interface management systems", *SIGCHI Bull.*, ACM Press, vol. 20 (1), Pp. 32-44, 1988.
- Lozano, A., Urretavizcaya, M., Ferrero, B., Fernández-Castro, I., Ustarroz, A., y Matey, L., "Integration of a Generic Diagnostic Tool in Virtual Environments for Procedural Training", en *Lecture Notes in Artificial Intelligence*, vol. 3040, 2004, pp. 666-675.
- Martín, M., Álvarez, A., Fernández-Castro, I., y Urretavizcaya, M., "Generating teacher adapted suggestions for improving distance educational systems with Sigma", en *actas de International Conference on Advanced Learning technologies (ICALT)*, Pp. 449 - 453. Santander, Spain., 2008.
- Martín, M., Fernández-Castro, I., Urretavizcaya, M., y Ferrero, B., "KADI: A Knowledge Acquisition Tool for Learning Activities", en *actas de Ed-MEDIA'04*, Pp. 889-894. Lugano, Switzerland, 2004.
- Marzano, R. J. y Kendall, J. S., *The New Taxonomy of Educational Objectives*: Corwin Press, 2007.
- Mateos, B. y Mateos, I. J., *Tecnología mecánica*: Servicio de publicaciones. Universidad de Oviedo, 1999.
- Mathan, S. y Koedinger, K., "Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skills", en *actas de Artificial Intelligence in Education*, Pp. 13-20, 2003.
- McCarthy, J. y Hayes, P. J., "Some philosophical problems from the standpoint of artificial intelligence", *Machine Intelligence*, American Elsevier Publishing, vol. 4, Pp. 463-502, 1969.
- McCarthy, L., Stiles, R., Johnson, W. L., y Rickel, J., "Human-systems interaction for virtual environment team training", *Virtual Reality*, Springer London, vol. 4 (1), Pp. 38-48, 1999.
- Meiri, I., "Combining qualitative and quantitative constraints in temporal reasoning", *Artificial Intelligence*, vol. 87 (1-2), Pp. 343-385, 1996.
- Mellet d'Huart, D., "Virtual Environment for Training: An Art of Enhancing Reality," en *Workshops of the Intelligent Tutoring Systems conference*. Donostia (Spain), Biarritz (France), 2002.
- Mena, O., Unzueta, L., Sierra, B., y Matey, L., "Nearest End-Effectors for Real-Time Full-Body Human Actions Recognition", en *actas de V Conference on Articulated Motion and Deformable Objects (AMDO 2008)*, Lecture Notes in Computer Science, Articulated Motion and Deformable Objects, Pp. 269-278. Pto. Andratx, Mallorca, Spain, 2008.

- Merrill, D., "Knowledge Objects", *CBT Solutions*, (March/April), Pp. 11, 1998.
- Michon, J. A., "A critical view of driver behavior models: What do we know, what should we do?" en *Human Behavior and traffic safety*, L. Evans and R. C. Schwing, Eds. New York: Plenum Press, 1985, pp. 485-520.
- Milgram, P. y Kishino, F., "A taxonomy of Mixed Reality visual displays", *IEICE Transactions on Information Systems*, vol. E77-D, 1994.
- Mislevy, R., Steinberg, L., reyer, F. J., lmond, R., y ohnson, L., "A Cognitive Task Analysis, with Implications for Designing A Simulation-Based Performance Assessment," CRESST 487, 1998.
- Mitrovich, A., en actas de Tutorial on Constraint-Based Tutoring Systems: From Theory to autohoring. ITS'2008. Montreal, Canada, 2008.
- Moreno-Ger, P., Burgos, D., Martínez-Ortiz, I., Sierra, J. L., y Fernández-Manjón, B., "Educational game design for online education", *Computers in Human Behavior*, vol. 24 (6), Pp. 25-30, 2008.
- Munro, A., Johnson, M., Pizzini, Q., Surmon, D., Towne, A., y Wogulis, J., "Authoring simulation-centered tutors with RIDES", *International Journal of Artificial Intelligence in Education*, vol. 8 (3-4), Pp. 284-316, 1997.
- Munro, A. y Surmon, D., "Primitive simulation-centered tutor services", en actas de AIED Workshop on Architectures for Intelligent Simulation-based Learning Environments, 1997.
- Murray, T., "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art", *Int. Journal of Artificial Intelligence in Education*, vol. 10 (1), Pp. 98-129, 1999.
- Murray, T., Blessing, S., y Ainsworth, S., *Authoring tools for advanced technology learning environments: Toward cost-effective, adaptive and intelligent educational software*: Kluwer Academic Publishers, 2003.
- Nigro, J. M. y Rombaut, M., "IDRES: A rule-based system for driving situation recognition with uncertainty management", *Information fusion*, Elsevier, vol. 4, Pp. 309-317, 2003.
- Nwana, H. S., "Intelligent tutoring systems: an overview", *Artificial Intelligence Review*, vol. 4 (4), Pp. 251-277, 1990.
- Ohlsson, S., "Constraint-Based Student Modelling", en *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, J. E. M. G. I. Greer, Ed. Berlin: Springer-Verlag, 1994, pp. 167-189.
- Oliver, N. y Pentland, A. P., "Driver Behavior Recognition and Prediction in a SmartCar", en actas de SPIE, *Enhanced and Synthetic Vision 2000*, Pp. 280-290, 2000.

- 
- Ong, J. y Noneman, S., "Intelligent Tutoring Systems for Procedural task Training of Remote Payload Operations at NASA", *en actas de IITSEC 2000*, 2000.
- Ota, D., Loftin, B., Saito, T., Lea, R., y Keller, J., "Virtual reality in surgical education", *Computers in Biology and Medicine*, vol. 25 (2), Pp. 127-137, 1995.
- Patel, A. D., Gallagher, A. G., Nicholson, W. J., y Cates, C. U., "Learning Curves and Reliability Measures for Virtual Reality Simulation in the Performance Assessment of Carotid Angiography", *J Am Coll Cardiol*, vol. 47 (9), Pp. 1796-1802, 2006.
- Pearl, J., Probabilistic reasoning in expert systems: Networks of plausible inference. San Francisco: Morgan Kaufmann Publishers, Inc., 1988.
- Perry, L. D. S., Smith, C. M., y Yang, S., "An investigation of current virtual reality interfaces", *Crossroads*, ACM Press, vol. 3 (3), Pp. 23-28, 1997.
- Pinhanez, C., "Representation and Recognition of Action in Interactive Spaces," Tesis doctoral, Dpto. MIT, 1999.
- Pitrat, J., Métaconnaissance, avenir de L'Intelligence Artificielle. Paris: Her mès, 1991.
- Pizzini, Q., Munro, A., Wogulis, J., y Towne, D., "The Cost-Effective Authoring of Procedural Training", *en actas de Intelligent Tutoring Systems*, 1996.
- Pressman, R. S., Software Engineering: A Practitioner's Approach 6/e: McGraw-Hill, 2005.
- Ragnemalm, E. L., "Student diagnosis in practice; bridging a gap", *User Modelling and User Adapted Interaction*, vol. 5 (2), Pp. 93-116, 1996.
- Ramaux, N., Fontaine, D., y Dojat, M., "Temporal scenario recognition for intelligent patient monitoring", *Artificial Intelligence in medicine. LNAI*, Springer, vol. 1211, Pp. 331-342, 1997.
- Reece, D. A. y Shafer, S., "A computational model of Driving for Autonomous Vehicles," *Transportation Research* 1991.
- Remolina, E., Ramachandran, S., Fu, D., Stottler, R., y Howse, W., "Intelligent Simulation-Based Tutor for Flight Training", *en actas de IITSEC 2004*, 2004.
- Rickel, J. W., "Intelligent Computer-Aided Instruction: A survey organized around system components", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19 (1), Pp. 40-57, 1989.
- Rickel, J. W., Gratch, J., Marsella, S., y Swartout, W., "Steve Goes to Bosnia: Towards a New Generation of Virtual Humans for Interactive Experiences", *en actas de AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2001.

- Rickel, J. W. y Johnson, W. L., "Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control", *Applied Artificial Intelligence*, vol. 13, Pp. 343-382, 1999.
- Richards, R., "Common Cockpit Helicopter Training Simulator", *en actas de AVSIM 2002 Conference & Exhibition*. Stateline, Nevada, 2002a.
- Richards, R., "Principle Hierarchy Based Intelligent Tutoring System for Common Cockpit Helicopter Training", *en actas de Intelligent Tutoring Systems*. Biarritz, 2002b.
- Ritter, S., "Communication, cooperation and competition among multiple tutor agents", *en actas de AIED'97*, 1997.
- Ritter, S. y Koedinger, K., "An architecture for plug-in tutor agents." *Journal of Artificial Intelligence in Education*, vol. 3, Pp. 315-347, 1996.
- Romizowski, A. J., *Designing instructional systems*. New York: Kogan Page London/Nichols Publishing, 1981.
- Rosenthal, R., Gantert, W., Scheidegger, D., y Oertli, D., "Can skills assessment on a virtual reality trainer predict a surgical trainee's talent in laparoscopic surgery?" *Surg Endosc*, vol. 20 (8), Pp. 1286-1290, 2006.
- Rueda, U., "Desarrollo de software reutilizable de Mapas Conceptuales y estudio de su aplicación en contextos de aprendizaje y en los Modelos Abiertos de Estudiante," Tesis doctoral, Dpto. Lenguajes y Sistemas Informáticos, Universidad del País Vasco, 2009.
- Ruffaldi, E., Filippeschi, A., Gonzalez, O. O. S., Frisoli, A., Avizzano, C. A., y Bergamasco, M., "Vibrotactile Perception Assessment for a Rowing Training System", *en actas de IEEE Worldhaptics 2009*, 2009.
- Saito, T., Ortiz, C., y Loftin, R. B., "On the acquisition and representation of procedural knowledge", *en actas de Fifth Annual Workshop on Space Operations Applications and Research*, Pp. 108-117, 1992.
- Salvucci, D. D., "Modeling Driver Behavior in a Cognitive Architecture", *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 48, Pp. 362-380, 2006.
- Schank, R. C., *Conceptual Information Processing*: Elsevier Science, 1975.
- Schijven, M. y Jakimowicz, J., "Virtual reality surgical laparoscopic simulators", *Surgical Endoscopy*, Springer Verlag, vol. 17 (12), Pp. 1943-1950, 2003.
- Self, J. A., "Bypassing the intractable problem of student modelling", *Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*, Pp. 107-123, 1990.

- 
- Self, J. A., "Computational mathematics: the missing link in Intelligent Tutoring Systems research", en *New directions for Intelligent Tutoring Systems*, E. Costa, Ed. Berlin: Springer-Verlag, 1992, pp. 265-352.
- Self, J. A., "The defining characteristics of intelligent tutoring systems research: ITSS care, precisely", *International Journal of Artificial Intelligence in Education*, vol. 10, Pp. 350-364, 1999.
- Self, J. A., "Student models in Computer-aided Instruction", *International Journal of Man-Machine Studies*, vol. 6, Pp. 261-276, 1974.
- Shute, V. J., "Towards Automating ECD-Based Diagnostic Assessments", *Technology, Instruction, Cognition and Learning (TICL)*, vol. 2 (1-2), 2004.
- Simpson, E. J., *The Classification of Educational Objectives in the Psychomotor Domain*. Washington: Gryphon House, 1972.
- Sleeman, D. H. y Brown, J. S., *Intelligent Tutoring Systems*. London: Academic Press, 1982.
- Sosteric, M. y Hesemeier, S., "When is a Learning Object not an Object: A first step towards a theory of Learning Objects," en *The International Review of Research in Open and Distance Learning*, 2002.
- Stiles, R., "Virtual Environments for Training - Final Report," Lockheed Martin 1999.
- Stone, R., "Virtual Reality for Interactive Training: an Industrial Practitioner's Viewpoint", *Int. J. Human-Computer Studies*, vol. 55, Pp. 699-711, 2001.
- Stottler, R., Fu, D., Ramachandran, S., y Vinkavich, M., "Applying a Generic Intelligent Tutoring System (ITS) Authoring Tool to Specific Military Domains", en *actas de I/ITSEC 2001*, 2001.
- Stottler, R. y Jensen, R., "Adding an Intelligent Tutoring System to an Existing Training Simulation", en *actas de I/ITSEC 2002*, 2002.
- Stottler, R. y Panichas, S., "A New Generation of Tactical Action Officer Intelligent Tutoring System (ITS)", en *actas de I/ITSEC 2006*, 2006.
- Stottler, R. y Pike, B., "An Embedded Training Solution: FBCB2/Tactical Decision-Making Tutoring System", en *actas de I/ITSEC 2002*, 2002.
- Stottler, R. y Vinkavich, M., "Tactical Action Officer Intelligent Tutoring System", en *actas de I/ITSEC 2000*, 2000.
- Sutherland, L., Middleton, P., Anthony, A., Hamdorf, J., Cregan, P., Scott, D., y Maddern, G., "Surgical Simulation: A Systematic Review", *Annals of Surgery*, vol. 243 (3), Pp. 291-300, 2006.
- Unzueta, L., "Markerless full-body human motion capture and combined motor action recognition for human-computer interaction," Tesis doctoral, Dpto. Applied Mechanics, Tecnum, University of Navarra, 2009.



- Unzueta, L., Berselli, G., Cazón, A., Lozano, A., y Suescun, Á., "Genetic Algorithms Applications to the Reconstruction of the Human Motion using a Noninvasive Motion Capture", en *actas de Multibody Dynamics 2005, ECCOMAS Thematic Conference*, Pp. 17. Madrid, 2005.
- Unzueta, L., Peinado, M., Boulic, R., y Suescun, Á., "Full-Body Performance Animation with Sequential Inverse Kinematics", *Graphical models*, Elsevier, vol. 70 (5), Pp. 87-104, 2008.
- Ustarroz, A., Lozano, A., Matey, L., Siemon, J., Klockmann, D., y Berasategui, M. I., "VIRTOOL - Virtual Reality for Machine-Tool Training", *Mécanique & Industries*, vol. 5, Pp. 207-212, 2004.
- van Beek, P. y Cohen, R., "Exact and approximate reasoning about temporal relations", *Computational Intelligence*, Blackwell Publishers, Inc., vol. 6 (3), Pp. 132-147, 1990.
- van Merriënboer, J. J. G., *Training Complex Cognitive Skills: A Four-Component Instructional Design Model for Technical Training*: Educational Technology Publications, 1997.
- VanLehn, K., "Student Modelling", en *Foundations of Intelligent Tutoring Systems*, J. J. R. M. C. Polson, Ed.: Lawrence Erlbaum Associates, 1988, pp. 55-78.
- Vigotsky, L. S., *Mind in society: The development of higher psychological processes*: Cambridge MA: Harvard University Press, 1978.
- Vilain, M., "A System for Reasoning about Time", en *actas de 2nd National (US) Conference on Artificial Intelligence*, Pp. 197-201, 1982.
- Vilain, M. y Kautz, H., "Constraint propagation algorithms for temporal reasoning", en *actas de AAAI-86*, Pp. 377-382. Philadelphia, PA, 1986.
- Vilain, M., Kautz, H., y van Beek, P., "Constraint propagation algorithms for temporal reasoning: a revised report", en *Readings in qualitative reasoning about physical systems*: Morgan Kaufmann Publishers Inc., 1990, pp. 373-381.
- Weevers, I., Kuipers, J., Brugman, A. O., Zwiers, J., Dijk, E. M. A. G. v., y Nijholt, A., "The Virtual Driving Instructor Creating Awareness in a Multiagent System", en *Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003, Halifax, Canada, June 11-13, 2003. Proceedings, Lecture Notes in Computer Science*: Springer Berlin / Heidelberg, 2003.
- Wenger, E., *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*: Morgan Kaufmann, 1987.
- Wenzel, B. M., Dirnberger, M. T., Hsieh, P. Y., Chudanov, T. J., y Halff, H. M., "Evaluating Subject Matter Experts' Learning and Use of an ITS Authoring

Tool ", *en actas de 4th International Conference on Intelligent Tutoring Systems*, Pp. 156-165, 1998.

Wescourt, K. T., Beard, M., y Gould, L., "Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation ", *en actas de National ACM Conference*, Pp. 234-240, 1977.

Zeltzer, D., "Autonomy, interaction and presence", *Presence*, vol. 1, Pp. 127-132, 1992.

Zou, X. y Levinson, D., "Modeling pipeline Driving Behaviors: A Hidden Markov Model Approach", *Journal of the Transportation Research Board*, (1980), Pp. 16-23, 2006.