

# HERRAMIENTAS DE AUTOR PARA ENSEÑANZA Y DIAGNÓSTICO: IRIS-D

**B. Ferrero, A. Arruarte, I. Fernández-Castro, M. Urretavizcaya**

Departamento de Lenguajes y Sistemas Informáticos  
Universidad del País Vasco/Euskal Herriko Unibertsitatea  
Apdo. 649 Donostia  
{jibfemab, jiparlaa, jipfecai, jipurlom}@si.ehu.es

## Resumen

Los Sistemas Tutores Inteligentes constituyen un grupo de aplicaciones de enseñanza que promueven un aprendizaje individual y flexible basado en el conocimiento y comportamiento del usuario. Hasta ahora estos sistemas han demostrado su efectividad en diversos dominios. Sin embargo su construcción implica un complejo e intenso trabajo de ingeniería del conocimiento, lo que impide un uso más general y aprovechado. Para eliminar estas trabas se plantea el objetivo de construir herramientas de autor que faciliten la construcción de tutores a personas no expertas, en particular a los propios instructores que dominan una determinada materia de enseñanza.

Este artículo describe la herramienta de autor IRIS-D, la cual permite construir sistemas tutores para varios tipos de dominios a partir de los requisitos identificados por el instructor. También se ilustra su funcionamiento con un ejemplo detallado sobre el dominio procedimental de la máquina-herramienta.

Un aspecto esencial que incide en la capacidad de adaptación al usuario es la posibilidad de diagnosticar su conocimiento. No todas las herramientas de autor proporcionan esta característica. Por ello, en el sistema IRIS-D se ha incorporado un mecanismo de diagnóstico adaptable a diferentes dominios, el sistema DETECTive, que también se detalla en este artículo.

**Palabras clave:** Herramientas de Autor, Diagnóstico Cognitivo, Sistemas Tutores Inteligentes.

## 1. Introducción

El campo de los Sistema Tutores Inteligentes (STIs) se caracteriza por aplicar las técnicas de Inteligencia Artificial (IA) al desarrollo de sistemas de enseñanza asistida por ordenador. Trata de construir *sistemas de enseñanza inteligentes*, donde el término inteligente se asocia a la capacidad de adaptarse dinámicamente al desarrollo del aprendizaje del estudiante. La enseñanza que realiza un STI suele ser individual y se apoya en un proceso interactivo mixto en el que participan tanto el tutor como el alumno compartiendo el objetivo común de desarrollar un aprendizaje efectivo. En general, este

proceso está guiado por el tutor, el cual debe analizar el comportamiento del alumno tanto para conocer su estado de conocimiento como para satisfacer sus requerimientos y así poder determinar y aplicar, en cada momento, las estrategias de enseñanza más adecuadas. Estas estrategias deben resolver una serie de cuestiones prioritarias que son: qué explicar, con qué nivel de detalle, cuándo y cómo interrumpir al alumno y cómo detectar y corregir sus errores.

Durante esta última década los STIs han superado el estatus de prototipo de laboratorio y se empiezan a instalar en aulas y lugares de trabajo, en los que

están demostrando ser muy efectivos [Murray, 99]. Sin embargo, aunque estos sistemas son cada vez más comunes y eficientes, todavía resultan difíciles y caros de construir lo que impide su uso generalizado. Para fomentar la creación y el uso de estos sistemas se plantea el objetivo de construir herramientas de autor que faciliten el desarrollo de STIs a personas sin experiencia en ingeniería del conocimiento. El estudio de estas herramientas surge prácticamente a la vez que los STIs y ya se han construido diferentes prototipos<sup>1</sup>, cada uno con sus características particulares tanto en lo referente al proceso de creación del sistema de enseñanza-aprendizaje como al sistema tutor que generan.

En este artículo describiremos la herramienta de autor IRIS-D, válida para varios tipos de dominios, e ilustraremos su funcionamiento utilizándolo en la generación de un tutor para el dominio procedimental de la máquina-herramienta.

El artículo comienza describiendo las principales características de IRIS-D, a continuación detalla las características del dominio de la máquina-herramienta y el proceso de construcción de un tutor para el mismo utilizando IRIS-D. Seguidamente presentaremos la arquitectura del sistema y finalizaremos con las conclusiones.

## 2. IRIS-D. Descripción general

IRIS-D es un entorno para ayudar a construir sistemas tutores inteligentes. Su utilidad reside en facilitar a profesores no expertos en el campo informático, la construcción de sistemas adaptativos de enseñanza-aprendizaje en dominios concretos en los que son expertos. El proceso que sigue IRIS-D para construir un nuevo tutor consiste en modificar, adaptar y completar una arquitectura de tutor genérica INTZIRI [Arruarte, 98] para cumplir los requerimientos sobre el tutor final deseado.

Las funcionalidades de IRIS-D se pueden resumir en los siguientes aspectos:

- Facilita la adquisición de conocimientos relativos a: la conducta del sistema tutor que se pretende construir, las características propias del dominio de enseñanza-aprendizaje y el tipo de estudiante o usuarios finales del sistema.
- Selecciona o crea la arquitectura concreta del sistema tutor final a partir de la especificación de sus requisitos.

---

<sup>1</sup> Murray (1999) realiza una revisión exhaustiva de los entornos de construcción de STI más relevantes.

- Permite incorporar al sistema tutor el conocimiento para detectar y diagnosticar los errores cometidos por el estudiante y los recursos didácticos necesarios para realizar la función de enseñanza-aprendizaje.
- Construye el sistema tutor final, seleccionando o generando el conjunto de reglas y objetos necesarios en función de los requisitos especificados.

El diagnóstico de los errores cometidos por el alumno durante su aprendizaje es un aspecto problemático que debe tenerse en cuenta durante la construcción de un tutor, ya que la información del diagnóstico es básica para deducir el estado cognitivo del estudiante y tomar las decisiones pedagógicas más adecuadas.

IRIS-D integra el sistema genérico de diagnóstico DETECTive [Ferrero et al., 97] que, igualmente, puede ser adaptado por el instructor a su dominio particular. Así, el tutor resultante es capaz de localizar y evaluar los errores del alumno identificando de forma precisa aquellos que el instructor haya considerado más relevantes por su frecuencia o gravedad. Son pocas las herramientas de construcción de STI que han abordado el aspecto de diagnóstico procedimental desde esta perspectiva ([Blessing, 97] [Munro *et al.*, 97]), por lo que la inclusión de un diagnóstico independiente del dominio da a nuestra propuesta un valor añadido.

A continuación se detallan las características de IRIS-D y se ponen de manifiesto durante la creación de un tutor en el dominio concreto de la máquina-herramienta. Éste es un dominio procedimental muy adecuado para este propósito ilustrativo.

## 3. Un dominio procedimental: la máquina-herramienta

En el ámbito de la formación industrial, la escasez o carestía de recursos físicos junto con el riesgo de accidentes comprometen el proceso de formación del personal y la evaluación de sus competencias profesionales. Como consecuencia, estas actividades se realizan desde planteamientos teóricos, generándose fundadas sospechas sobre el grado de consecución de las capacidades técnicas por parte de los operarios. Nuestro grupo de investigación tuvo la ocasión de conocer en primera persona esta realidad cuando un grupo de instructores del Instituto para la Formación de la Máquina-Herramienta (IMH) de Elgoibar nos propuso crear un sistema de ayuda al profesor, en su tarea de formación y evaluación, en este ámbito industrial.

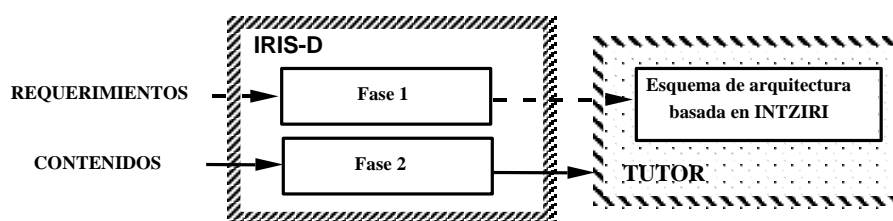


Figura 1: Fases en la construcción de un tutor

Las máquinas-herramienta suelen tener gran tamaño y alto coste; el operario necesita una habilidad concreta para manipularlas correctamente y seleccionar y colocar en la forma adecuada los elementos auxiliares necesarios. Teniendo en cuenta la gran variedad de máquinas-herramienta existentes, decidimos acotar el problema sobre los procesos que usan el torno para mecanizar piezas por arranque de viruta. Así abordamos la construcción del tutor LATHE como sistema capaz de ofrecer al alumno un proceso de formación adaptado a sus capacidades y preferencias personales, y de realizar el diagnóstico y evaluación de sus actividades.

A continuación se detalla el proceso de construcción de este tutor utilizando la herramienta de autor IRIS-D descrita en el apartado anterior.

#### 4. Proceso de construcción de un tutor utilizando IRIS-D

El proceso de construcción de un tutor utilizando el entorno IRIS-D consta de dos fases (Figura 1): Fase 1 de especificación de requerimientos y Fase 2 de especificación de contenidos. Durante la Fase 1 se adapta una arquitectura genérica según los requerimientos especificados por el instructor obteniéndose la plantilla o esquema de arquitectura del nuevo tutor. Estos requerimientos se centran en aspectos del dominio de enseñanza-aprendizaje y en la caracterización tanto del estudiante como del tutor. Durante la Fase 2 se establecen los contenidos de los módulos seleccionados o adaptados en la primera fase. Algunos de estos contenidos son especificados directamente por el instructor <sup>34ej</sup>: contenidos del dominio, formas de presentación y evaluación, etc., mientras que otros son generados automáticamente por IRIS-D <sup>34ej</sup>: reglas para refinar el Plan Instruccional.

#### 4.1. Fase 1 de IRIS-D. Identificación y especificación de requerimientos

##### 4.1.1. Identificación de requerimientos

En general, el diseño de cualquier asignatura o curso implica planificar por anticipado el trabajo que han de realizar tanto el profesor como el alumno para conseguir una enseñanza efectiva. Una buena planificación debe contestar a las cuestiones de *qué, cómo y cuándo* enseñar además de *qué, cómo y cuándo* evaluar (Coll, 1987). La respuesta a estas cuestiones define los aspectos a considerar: el estudiante y su entorno, los objetivos a alcanzar, los contenidos y su secuenciación, los recursos didácticos utilizados y los medios de evaluación. En los programas de ordenador en general, y en la construcción de un tutor automático en particular, este proceso de planificación se liga con la fase de *análisis*, que tradicionalmente se desarrolla a mano. El análisis requerido para construir un tutor utilizando IRIS-D afecta a tres componentes de un STI: módulo del dominio, modelo del estudiante y componente pedagógico; IRIS-D no ofrece soporte para esta fase.

Para analizar la materia de enseñanza-aprendizaje el apoyo básico son los libros de texto, que proporcionan una visión clara de los contenidos a transmitir, su posible secuenciación y diferentes formas de presentación y evaluación.

Respecto al análisis del tipo de estudiante al que se dirige el nuevo tutor y su forma de instrucción, es la experiencia práctica del instructor, además de la bibliografía existente al respecto, la mejor guía para su desarrollo.

A continuación enumeramos los requisitos de LATHE identificados en la fase de análisis para cada componente del tutor. En esta tarea los profesores del IMH han participado de forma muy activa.

## Requisitos sobre la materia de enseñanza-aprendizaje

Construir un tutor capaz de evaluar procedimientos requiere considerar dos niveles en la representación del dominio: nivel funcional y nivel pedagógico. El **nivel funcional** representa los elementos del dominio que se manipulan y modifican durante la resolución de problemas. El **nivel pedagógico** representa tanto los contenidos y habilidades que se quieren transmitir al estudiante como las relaciones necesarias entre los contenidos para que la enseñanza sea efectiva, es decir en la organización pedagógica de la materia de enseñanza.

El dominio de la máquina herramienta, concretamente el uso del torno para mecanizar piezas por arranque de viruta, es claramente procedimental<sup>1</sup> y necesita ser descrito tanto a nivel pedagógico como funcional.

Las principales características de este dominio a nivel pedagógico son las siguientes:

- Incluye dos tipos de Unidades Básicas de Aprendizaje, concretamente *Conceptos*, o contenidos teóricos, por ejemplo “partes principales de un torno” y *Procedimientos*, o contenidos prácticos, por ejemplo “cilindrar en desbaste”.
- En general, las habilidades u Objetivos Instruccionales que se desarrollarán en el estudiante son *Conocimiento* y *Aplicación*. El primero se refiere a la memorización específica de contenidos, por ejemplo “memorizar el orden de ejecución de los pasos en un procedimiento”. La segunda se refiere a la transferencia de lo aprendido a la resolución de problemas, por ejemplo, que el estudiante “ejecute todos los pasos necesarios para realizar la operación de cilindrado en desbaste”. Esta habilidad sólo se desarrollará para procedimientos.
- Las relaciones entre contenidos utilizadas por el tutor serán *prerrequisito*, *siguiente*, *es-un y parte-de*. Las dos primeras se refieren a relaciones de secuenciación entre contenidos; las dos últimas a relaciones de pertenencia o composición y clasificación.
- Como formas de presentación el tutor utilizará *textos*, *imágenes* y *videos*, y como formas de evaluación *test*, *ejercicios de*

---

<sup>1</sup> El aprendizaje de un dominio procedimental supone conocer cómo hacer algo, cómo ejecutar un conjunto de acciones ordenadas dirigidas a la consecución de una meta.

*ordenación y ejercicios procedimentales o procedurales.*

El dominio a nivel funcional incluye:

- Los objetos manipulables del dominio, es decir todos aquellos objetos físicos que el alumno puede modificar con la ejecución de los diferentes procedimientos.
- Las acciones que el estudiante puede realizar para manipular estos objetos.

## Requisitos sobre el tipo de estudiante

En cuanto a la población objetivo del sistema, y teniendo en cuenta la edad media de los estudiantes del IMH, asumiremos un estudiante adulto con escasos conocimientos en informática. Se desea además:

- Que el tutor se adapte al tipo de estudiante (hemos identificado tres perfiles básicos: *novel*, *medio*, *experto*), a su motivación (*baja*, *media*, *alta*) y a su preferencia sobre la duración de la sesión.
- El tutor deberá ser capaz de atender los siguientes objetivos del estudiante: *intervenciones de control (dormir la sesión, continuar con la sesión, finalizar la sesión)* y *peticiones asociadas al desarrollo de la sesión (solicitar la resolución de un ejercicio, estar de acuerdo/en desacuerdo con la intervención del tutor, solicitar la repetición de un ejercicio y solicitar un nuevo ejercicio)*.

## Características del Componente Pedagógico

Se han identificado las siguientes características:

- En la instrucción, el sistema utilizará un *método guiado* en el que el tutor dirige la toma de decisiones durante la sesión de enseñanza, asociado a una iniciativa compartida entre el tutor y el estudiante.
- Se considera interesante que el nuevo tutor disponga de varios recursos de motivación: *efectos sonoros*, *mensajes de llamada de atención* y *mensajes para felicitar o animar al usuario*.

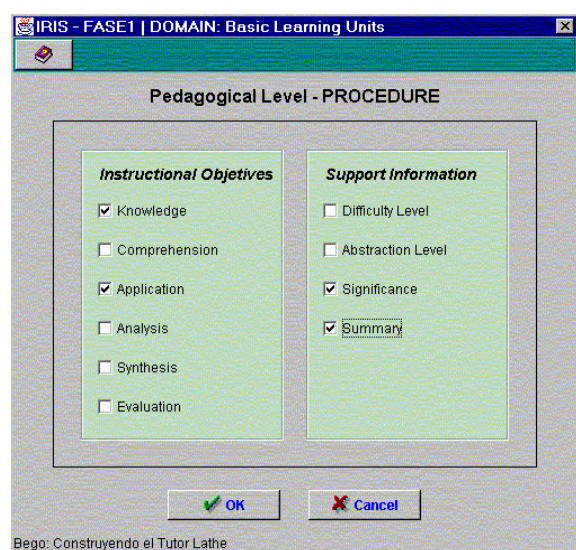
### 4.1.2. Especificación de requerimientos

Una vez caracterizados los componentes del tutor en la fase de análisis, el instructor debe especificarlos como requerimientos a través de las ventanas

gráficas que IRIS-D ofrece durante la Fase 1. Nos centraremos en el proceso de caracterización del tutor LATHE en lo que respecta al dominio, estudiante y módulo pedagógico.

El dominio se describe a nivel pedagógico mediante cuatro clases de información: Unidades Básicas de Aprendizaje (UBA), Objetivos Instruccionales<sup>2</sup> (OI), relaciones entre contenidos y meta-información para caracterizar el dominio de forma general. A continuación se comenta cada una de ellas.

IRIS-D permite representar cualquier dominio instruccional en función de cuatro clases de UBAs: conceptos, procedimientos, principios y hechos. Para representar el dominio procedimental de LATHE utilizaremos conceptos y procedimientos. A su vez cada UBA identificada por el instructor, debe acompañarse de sus objetivos instruccionales o habilidades y la información de soporte que utilizará el tutor final.



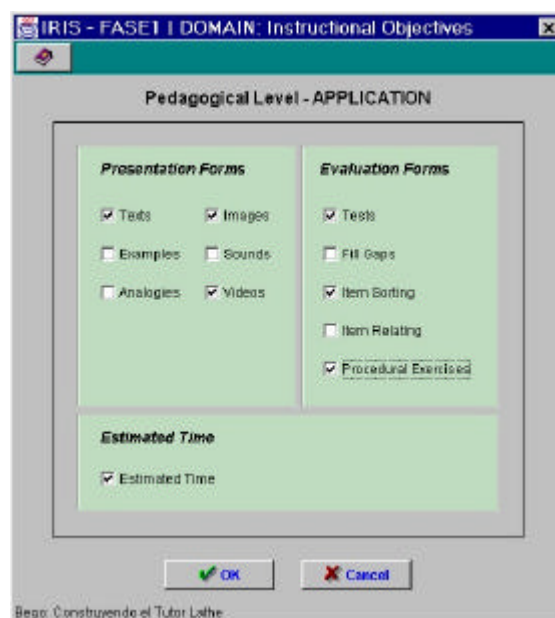
**Figura 2. Especificación de la UBA Procedimiento**

La Figura 2 muestra la especificación de una UBA *Procedimiento*. Se han seleccionado los objetivos instruccionales *Conocimiento* y *Aplicación* y como información de soporte únicamente se han especificado las características *Importancia* (Significance) y *Resumen* (Summary). De este modo se asegura que cualquier procedimiento concreto del

<sup>2</sup> Los OI que recoge nuestra herramienta son los identificados en la taxonomía de Bloom [Bloom et al., 56]

dominio podrá estudiarse desde las perspectivas *conocimiento* y *aplicación*. De forma equivalente se definirá la UBA *Concepto*.

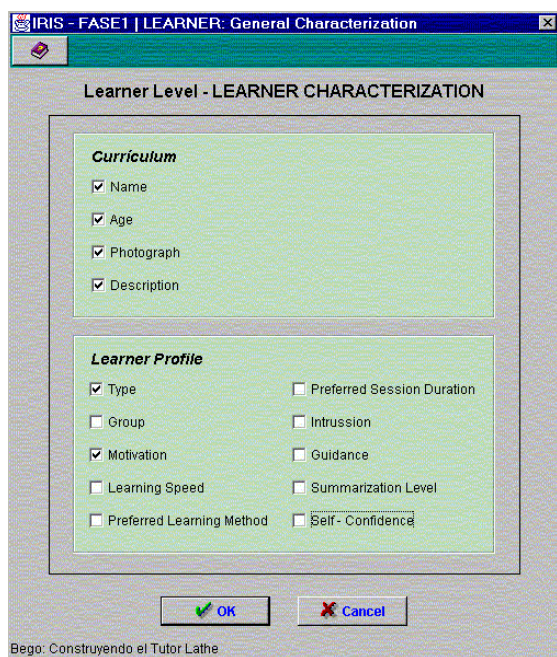
Para cada OI seleccionado, deben indicarse las formas de presentación y evaluación que se utilizarán. La Figura 3 muestra que, en general, para el objetivo instruccional *Aplicación* se utilizan textos (*texts*), imágenes (*images*) y videos (*videos*) como formas de presentación y tests (*tests*), ejercicios de ordenación (*item sorting*) y ejercicios procedimentales (*procedural exercises*) como formas de evaluación. Es posible además asociar el tiempo medio de adquisición de ese objetivo. De manera análoga se especifica el OI *Conocimiento* para el cual se eligen los textos como forma de presentación y los tests como forma de evaluación.



**Figura 3. Especificación del tipo de objetivo instruccional *Aplicación***

Una vez finalizada la especificación de las UBAs y los OIs se determinan las *relaciones estructurales* y *de requisito* entre contenidos.

Finalmente se especificará la información necesaria para caracterizar el dominio de forma general. IRIS-D permite definir dos atributos para justificar el dominio, por ejemplo frente al currículum del estudiante o su plan de estudios, relacionándolo con otras materias que el estudiante conoce, etc. Estos atributos se proporcionan por defecto y son: *objetivo general del dominio* y *relevancia del dominio*.

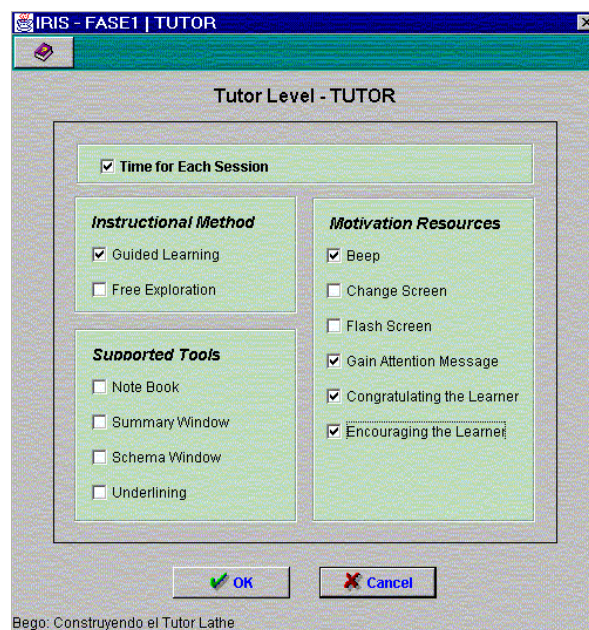


**Figura 4. Caracterización del alumno**

Una vez especificado el dominio a nivel pedagógico, se completan las características generales del futuro usuario. En particular su *currículum vitae*, el *perfil del alumno* y los *objetivos del alumno*. Los dos primeros se definen desde la ventana de la Figura 4. El *currículum* describe las características físicas y profesionales del alumno que el tutor tendrá en cuenta al generar sus sesiones de enseñanza. El *perfil de aprendizaje* recoge las características de aprendizaje que el tutor utilizará para seleccionar las acciones instruccionales y las formas de comunicación con el estudiante. De igual modo se introducirán los *objetivos del alumno* que describen aquellos tipos de interacciones del alumno a los que el tutor será capaz de dotar de significado y que constituyen la base de sus decisiones tutoriales sobre procesos cognitivos, acciones instruccionales y aspectos de comunicación. Estos objetivos están separados en dos grupos dependiendo de su funcionalidad: *Objetivos de Control de la Sesión* y *Objetivos de Desarrollo de la Sesión*.

Tras finalizar la especificación del estudiante, se definen los requerimientos del Componente Pedagógico. Éstos se utilizan para seleccionar aquel conocimiento general de planificación directamente basado en las características del tutor que se quiere construir. La arquitectura genérica INTZIRI ya incluye un conocimiento general de planificación, el

cual se va adaptando según los requisitos especificados para el dominio y el estudiante, teniendo también en cuenta las siguientes características: *tiempo para la sesión*, *métodos instruccionales*, *recursos de motivación* y *herramientas de soporte* (Figura 5).



**Figura 5. Especificación de los requerimientos del Componente Pedagógico**

Con los requerimientos especificados, IRIS-D crea la arquitectura esqueleto del nuevo tutor y se puede iniciar ya el proceso de definición de contenidos.

#### 4.2. Fase 2 de IRIS-D. Creación de contenidos

En esta fase se completan los distintos componentes del tutor. Centrándonos en el módulo del dominio, y por tratarse de un dominio procedimental, es necesario crear los contenidos tanto del nivel pedagógico como del funcional, además de establecer las relaciones existentes entre los elementos definidos en ambos niveles. La definición a nivel funcional es necesaria siempre que se quiera realizar un proceso de detección y diagnóstico de errores en ejercicios procedimentales, es decir, en ejercicios que requieran ser ejecutados por el sistema para simular paso a paso la resolución del problema.

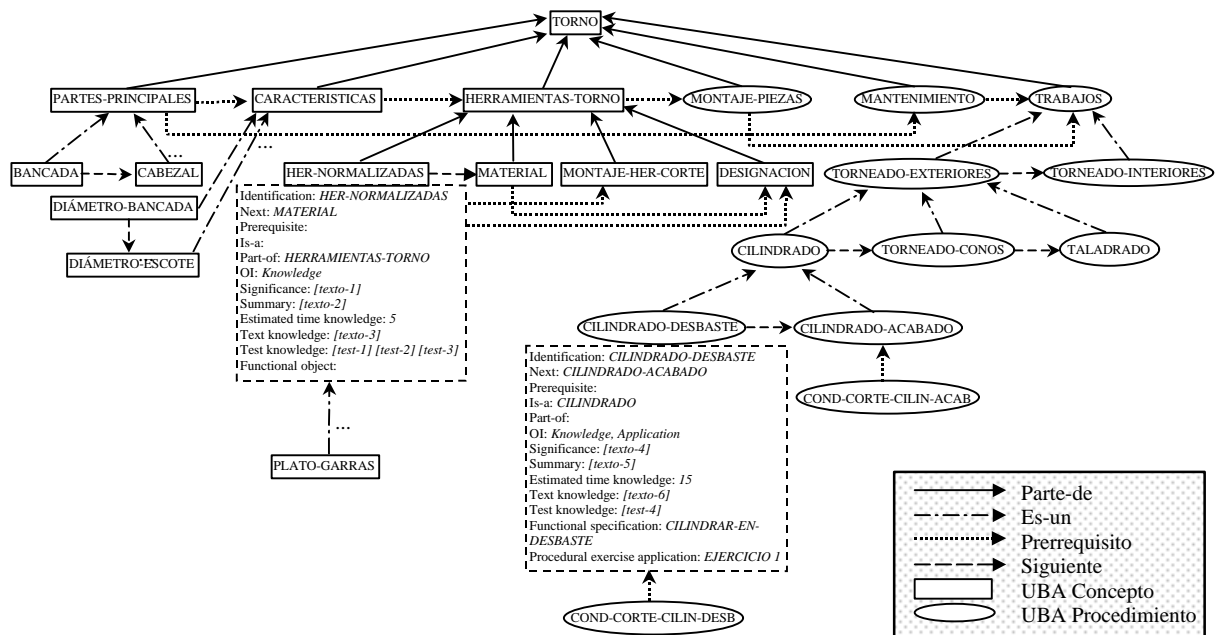


Figura 6. Grafo del dominio pedagógico

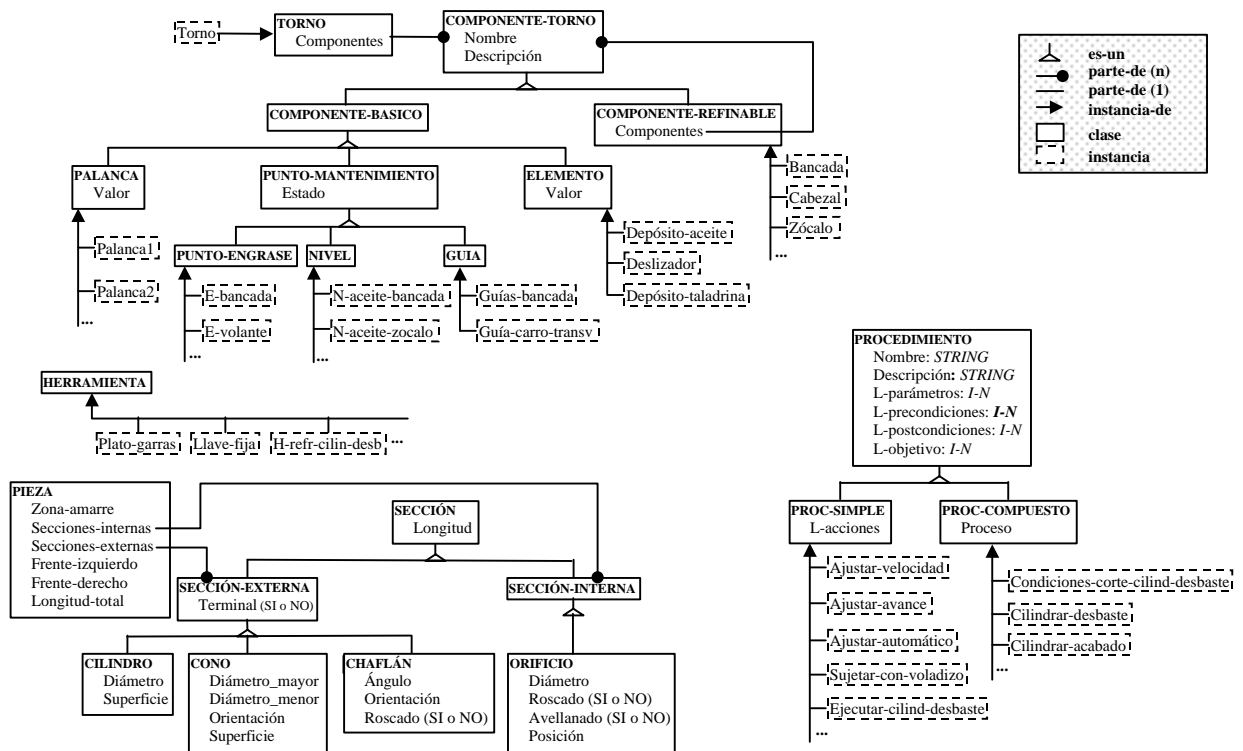


Figura 7. Jerarquía para la descripción funcional del dominio

Desde una perspectiva pedagógica, el instructor debe describir los contenidos correspondientes a las UBAs para cada uno de los objetivos instruccionales determinados en la primera fase junto con sus relaciones. Además deberá suministrar al sistema las formas de presentación y evaluación concretas que darán soporte a las sesiones de enseñanza.

La Figura 6 muestra un fragmento de la red de contenidos del dominio pedagógico identificada en la fase de análisis. En ella se recogen parte de los conceptos y procedimientos del dominio de la máquina-herramienta junto con la perspectiva evolutiva del aprendizaje definida por las relaciones de secuenciación correspondientes.

La definición a nivel funcional conlleva identificar las clases e instancias que pueden manipularse durante la resolución de los ejercicios, y definir las relaciones entre ellas; éstas pueden ser: *es-un*, *parte-de* e *instancia-de*. IRIS-D se apoya en la herramienta de adquisición de conocimiento GBC [Pérez-Colmenar et al., 99] para facilitar esta actividad.

La Figura 7 representa la jerarquía parcial de clases e instancias identificada en el nivel funcional del dominio de la máquina-herramienta. En ella se describen los distintos elementos que el alumno puede manipular en la resolución de ejercicios procedimentales: el TORNO y sus partes principales y las HERRAMIENTAS y PIEZAS cuya mecanización se planteará como ejercicio. En este

nivel también se realiza la descripción funcional de los distintos PROCEDIMIENTOS que pueden ejecutarse durante un proceso de mecanizado.

Para ilustrar el uso de IRIS-D durante la definición de contenidos mostraremos dos ejemplos diferentes: el primero de ellos corresponde a la creación del concepto *herramientas normalizadas* del torno (Figura 8), el segundo a la creación del procedimiento *cilindrar en desbaste*.

Para definir la UBA concepto *herramientas normalizadas*, se muestra al instructor la ventana de la Figura 8 en la que deberá rellenar todos los atributos determinados previamente por IRIS-D a partir de la información introducida durante la fase de especificación de requerimientos. En el caso de LATHE, los atributos son:

- Identification:** HER-NORMALIZADAS
- Next:** MATERIAL
- Prerequisite:**
- Is-a:**
- Part-of:** HERRAMIENTAS-TORNO
- OI:** Knowledge
- Significance:** [texto-1]
- Summary:** [texto-2]
- Estimated time knowledge:** 5
- Text-knowledge:** [texto-3]
- Test-knowledge:** [test-1] [test-2] [test-3]
- Functional-especification:**

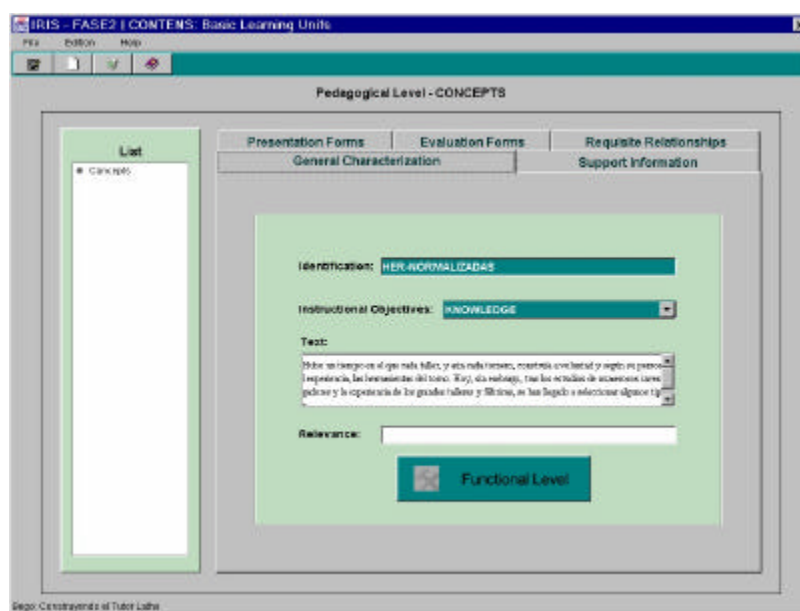


Figura 8. Creación de una UBA concepto



En la parte inferior de la ventana de la Figura 8 aparece un botón que permite acceder al nivel funcional para establecer las relaciones entre la UBA que se está definiendo y los correspondientes objetos físicos. En este caso como el concepto sólo tiene el OI *conocimiento* (knowledge) su descripción se circunscribe al plano pedagógico. Este botón no aparece para dominios puramente declarativos puesto que no necesitan definición funcional.

Es posible crear varios tipos de ejercicios de evaluación para cada UBA y OI. La Figura 9 muestra el aspecto de la ventana para definir un ejercicio de tipo test. En ella se observan dos tests ya definidos, Test-1 y Test-2, y un tercero que se está definiendo (Test-3); se indica la pregunta que se plantea al alumno, y la lista de posibles respuestas con su correspondiente valoración (consideramos que es posible que haya respuestas correctas e incorrectas con diferentes notas asociadas).

Continuando con el proceso de definición del dominio se describe el procedimiento *cilindrar en desbaste* a nivel pedagógico y funcional. Los

atributos que se han seleccionado en el nivel pedagógico son:

**Identification:** CILINDRADO-DESBASTE  
**Next:** CILINDRADO-ACABADO  
**Prerequisite:**  
**Is-a:** CILINDRADO  
**Part-of:**  
**OI:** Knowledge, Application  
**Significance:** [texto-4]  
**Summary:** [texto-5]  
**Estimated time knowledge:** 15  
**Text-knowledge:** [texto-6]  
**Test-knowledge:** [texto-7]  
**Text-application:** [texto-8]  
**Images-application:** [image1]  
**Video-application:**  
**Funcn.-specif.:** CILINDRAR-EN-DESBASTE  
**Test-application:** [test-4]  
**Sorting-application:** [sort-1] [sort-2]  
**Proced. exercise application:** EJERCICIO 1

Sus valores se introducen de la misma forma descrita para el concepto *herramientas normalizadas*.

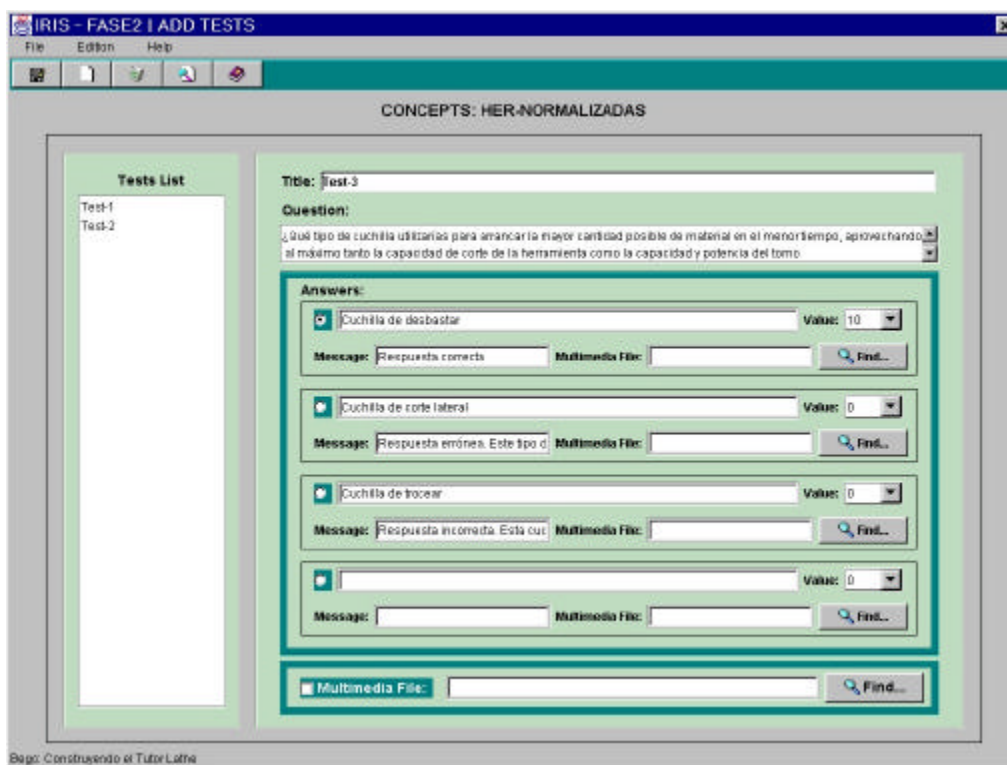
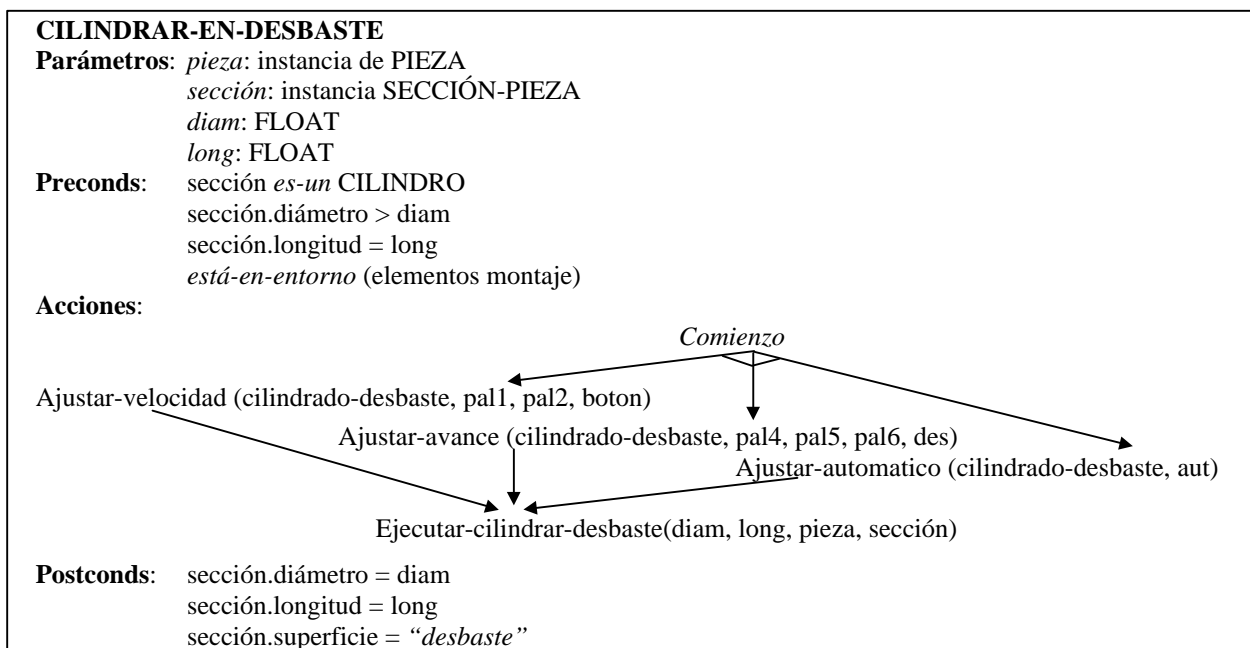


Figura 9. Pantalla para la definición de un test

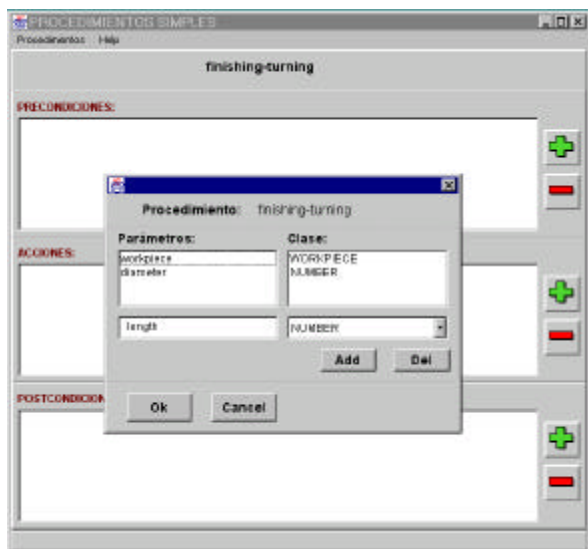
A nivel funcional, un procedimiento se identifica mediante un nombre y una serie de parámetros (ver Figura 7). Además, sus *precondiciones*, las *acciones* que ejecuta (provocando cambios en algunos objetos) y las *postcondiciones* que se verifican tras su ejecución, definen su comportamiento. La ejecución de las acciones y la evaluación de las precondiciones y postcondiciones se realizan sobre un *entorno* global compuesto por una selección de objetos del dominio identificados en el nivel funcional. En nuestro caso éste lo formarán la pieza que se está mecanizando, el torno y las herramientas. De esta forma, un *estado* del entorno global se define mediante un conjunto de tuplas [objeto-atributo-valor] susceptibles de ser modificadas y evaluadas durante la ejecución de un ejercicio. El orden de ejecución de las acciones de un procedimiento se representa mediante una estructura arborescente ordenada, en la que cada nodo describe la acción a realizar y su relación directa con aquellas que podrían ejecutarse a continuación. Distinguimos tres clases de relaciones entre nodos: *secuencia*, *disyunción* y *conjunción* (sería la establecida en la Figura 10), con su semántica habitual. Cada procedimiento está representado por medio de una estructura de este tipo que posee un nodo raíz conectado con otros nodos a través de un enlace de alguna de las tres clases previamente mencionadas.

Por ejemplo, el procedimiento de nombre **CILINDRAR-EN-DESBASTE** (Figura 10) representa una operación de mecanizado cuyos parámetros son la *pieza* y la *sección* de la misma que se va a mecanizar, así como las dimensiones (*long*: longitud, y *diam*: diámetro) que dicha sección deberá tener al finalizar la operación. Las precondiciones indican que este procedimiento únicamente podrá ser ejecutado si la sección a mecanizar es un cilindro cuyo diámetro inicial es mayor que el valor indicado por el parámetro *diam*, su longitud es igual a *long* y los elementos de montaje necesarios para realizar dicha operación han sido seleccionados y colocados en algún paso previo. Por otro lado, las postcondiciones del procedimiento describen el *estado* resultante. En este caso concreto, indican la longitud y el diámetro finales de la pieza.

La ejecución de este procedimiento conlleva realizar una serie de acciones que, a su vez, son procedimientos. Estos son: *ajustar-velocidad*, *ajustar-avance* y *ajustar-automático*. El operario tendrá que realizar estas tres acciones en cualquier orden. Una vez realizados estos pasos, se ejecutará la operación de mecanizado. El procedimiento *Ejecutar-cilindrar-desbaste* simula la realización de esta operación por parte del operario, utilizando el torno.

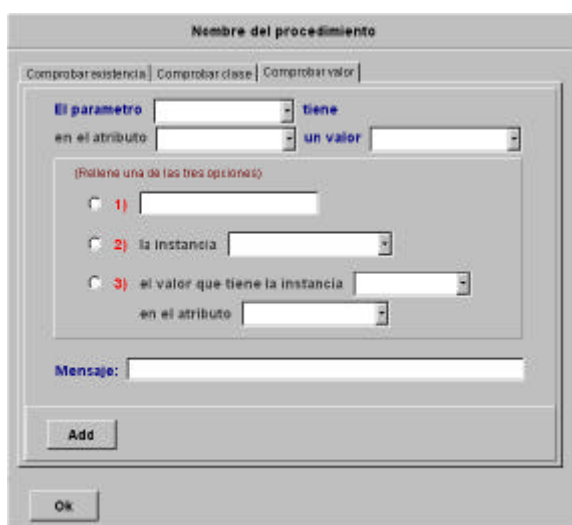


**Figura 10. Procedimiento CILINDRAR-EN-DESBASTE**



**Figura 11. Definición de parámetros de un procedimiento**

La interfaz de IRIS-D cuenta con ventanas que guían al instructor en la definición de un procedimiento. Aquí se muestran algunos ejemplos. La ventana de la Figura 11 aparece al definir un nuevo procedimiento cuando, tras haber introducido su nombre (cilindrado en acabado *-finishing-turning*), se solicitan sus parámetros. Las precondiciones, acciones y postcondiciones, se definen tras pulsar los botones etiquetados con un signo de adición situados en la parte derecha de la ventana.



**Figura 12. Definición de una precondición o postcondición**

La Figura 12 muestra la pantalla para definir una precondición o postcondición. Cada pestaña de la parte superior representa una clase de condición; concretamente en la figura aparece seleccionada la pestaña que permite describir una condición de la clase *comprobar valor*.

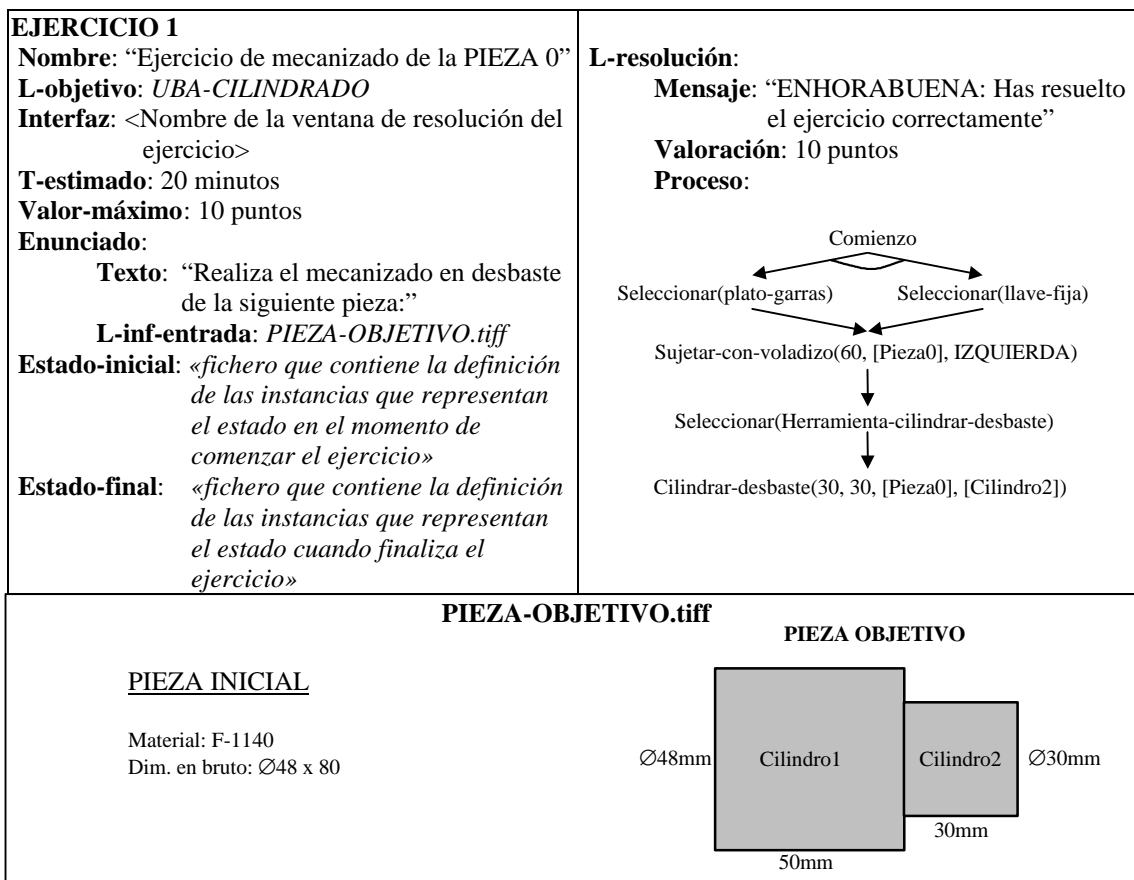
También se facilita un editor de árboles que permite definir la secuencia de ejecución de los procedimientos.

A continuación se define un ejercicio procedimental en el que se solicita al alumno que, partiendo de una pieza cilíndrica inicial, mecanice una determinada sección para obtener otra pieza resultado. La definición de este ejercicio se muestra en la Figura 13.

Un ejercicio procedimental define un estado de inicio sobre un entorno global, y un objetivo final cuya consecución se logrará mediante la aplicación de uno o varios procedimientos de acuerdo a una estrategia previamente elaborada. Por tanto, un ejercicio de esta clase lo definen su *enunciado*, los *procesos de resolución* definidos y la lista de valores que describen el *estado inicial* y el *estado final* del *entorno global*.

El proceso de resolución de un problema se plantea como la ejecución de ciertas acciones. Cada acción se corresponde con un procedimiento del dominio que resuelve parcialmente el problema inicial. Para definir el orden en el que deben ejecutarse estas acciones se utiliza la misma estructura jerárquica definida para los procedimientos.

Además de la información ya comentada, en la definición de un ejercicio procedimental se incluyen los datos necesarios para la evaluación del alumno, por ejemplo el tiempo estimado para la realización del ejercicio (T-estimado), la nota máxima que se puede asociar al mismo (Valor-máximo) y la lista de UBAs a cuya evaluación está asociado (L-objetivo). El enunciado del ejercicio consta de un texto y de la información de entrada, la cual puede contener datos adicionales para completar la exposición del problema que se plantea; en este ejemplo se hace referencia al fichero gráfico *PIEZA-OBJETIVO.tiff* que contiene información sobre la pieza inicial y el plano de la pieza objetivo que se desea obtener. Sólo se ha definido un proceso de resolución para este ejercicio que corresponde a una solución correcta cuya nota son 10 puntos.



**Figura 13. Definición de un ejercicio procedimental**

#### 4.2. Descripción general de los tutores construidos por IRIS-D

En este apartado describimos brevemente las características compartidas por todos los tutores construidos mediante IRIS-D en cuanto a la modelización del dominio, estudiante, módulo pedagógico y diagnóstico.

El dominio, detallado en secciones anteriores, representa de forma explícita el conocimiento a ser transmitido al estudiante y está descrito tanto a nivel pedagógico como funcional.

El modelo del estudiante, basado en un enfoque *overlay* [Golstein, 82], define las características de aprendizaje individual y la evolución de cada alumno. Está organizado en dos partes que se van modificando a medida que transcurre el proceso de enseñanza-aprendizaje. El *modelo permanente* se mantiene durante todo el proceso de instrucción del alumno y se actualiza sesión a sesión. Contiene información relativa a las características personales y de aprendizaje del alumno, el conocimiento que posee, los errores que ha cometido y la historia de

las sesiones de enseñanza que han tenido lugar. El *modelo temporal* contiene la información referente a la sesión de enseñanza en curso. Al término de cada sesión, el modelo temporal desaparece y con sus contenidos se actualiza el modelo permanente.

En cuanto al conocimiento pedagógico, es el módulo didáctico el encargado de planificar y ejecutar, a través del plan instruccional, la sesión de enseñanza-aprendizaje. Los tutores construidos con IRIS-D integran el modelo cognitivo CLAI [Arruarte *et al.*, 96] y se basan en un proceso de planificación instruccional de mínimo compromiso en el que a priori sólo se planifica la Unidad Básica de Aprendizaje (UBA) inicial. Resulta además completamente oportunista en el sentido de que las intervenciones del alumno pueden ser aceptadas en cualquier momento. La estructura del plan instruccional [Arruarte, 98] determina el ciclo general de actividades del tutor:

- establecer el siguiente contenido de la materia (UBA) que se presentará al estudiante;
- determinar el Objetivo Instruccional (OI) o

habilidad que sobre ese contenido se va a tratar de alcanzar durante la sesión;

- establecer el conjunto de Procesos Cognitivos (PCs) asociado a cada OI;
- refinar los PC en Eventos Instruccionales (EIs);
- refinar los EIs en Acciones Instruccionales concretos que finalmente ejecutará el tutor.

Las *reglas de decisión pedagógica* establecen las ligaduras entre los distintos niveles de planificación estableciendo el plan instruccional en su nivel de decisión pedagógica.

En los tutores generados por IRIS-D, el proceso de diagnóstico de ejercicios procedurales está guiado por el conocimiento del dominio a nivel funcional y por las resoluciones definidas para cada ejercicio. Uno de los fundamentos de este proceso –técnica de *Model-Tracing* [Anderson *et al.*, 90] [Corbett *et al.*, 90]– utiliza los modelos de solución definidos en el ejercicio para contrastar cada paso de la actividad del estudiante produciendo un *Diagnóstico Dirigido por Modelos*. Sin embargo, en general no es posible asegurar que los modelos incluidos abarcan exhaustivamente todas las soluciones de un problema. Es decir, durante el proceso de resolución el alumno puede realizar una acción no incluida en ningún modelo. En estos casos, el sistema abandona definitivamente (para el problema actual) el diagnóstico dirigido y pasa a realizar un *Diagnóstico Basado en el Dominio*. En esta situación el conocimiento del dominio y, en particular, la articulación de los procedimientos por medio de sus pre y post condiciones permiten determinar la corrección o incorrección de cada paso del estudiante según el estado actual de resolución del problema. El seguimiento realizado no puede garantizar un diagnóstico/evaluación exacto en todos los casos. Puede ocurrir que el alumno, ejecute procedimientos para los que se verifican todas las precondiciones pero llegue a una solución no correcta, o diseñe una solución cíclica sin alcanzar un resultado final debido a una incorrecta o incompleta definición del dominio. El sistema almacena este grupo de resoluciones para su posterior revisión por el profesor, que puede utilizarlas para mejorar la cobertura funcional del dominio.

## 5. Arquitectura de IRIS-D

La arquitectura básica del sistema IRIS-D integra cuatro componentes principales (Figura 14):

*Interfaz, Generador de la Estructura del Tutor, Generador de Objetos y Generador de Reglas.*

La Interfaz gestiona el proceso de comunicación del entorno IRIS-D con el exterior, en concreto con el instructor, por lo que interesa una interfaz gráfica fácil de utilizar. El módulo incluye un conjunto de ventanas gráficas desarrolladas en JAVA y su comportamiento, esto es, la secuenciación de ventanas para realizar la tarea global de IRIS-D y las llamadas a sus módulos funcionales. En el ejemplo de creación del tutor definido en la sección anterior se han venido mostrando algunas ellas.

El Generador de la Estructura del Tutor selecciona la arquitectura esqueleto del nuevo tutor. Tomando como base las preferencias del instructor, IRIS-D decide si construirlo a partir de la arquitectura genérica o a partir de la arquitectura asociada a un tutor existente. Todas las posibles arquitecturas iniciales están recopiladas en una biblioteca de *arquitecturas de tutor*. La arquitectura que se ha utilizado como base en el ejemplo desarrollado incluye todos los componentes principales (módulo del dominio, modelo del estudiante, módulo pedagógico y sistema de diagnóstico genérico). Además, este módulo se encarga de comprobar las precondiciones de creación y la completitud de los contenidos de cada fase asegurando así la construcción de un tutor viable. No se debe permitir, por ejemplo, la introducción de ningún concepto particular si la unidad básica de aprendizaje *concepto* no ha sido previamente caracterizada en la fase de especificación de requerimientos.

Una vez que se ha definido una arquitectura inicial, el Generador de Objetos adapta, completa y almacena la estructura declarativa de cada uno de sus componentes. Esta adaptación se realiza de acuerdo con los requerimientos especificados; todas aquellas características de la arquitectura inicial que estén asociadas a requisitos seleccionados por el instructor se mantendrán como pertenecientes al nuevo tutor, mientras que el resto se eliminarán.

Tras adaptar la arquitectura hay que completarla obteniendo la estructura declarativa global – *conjunto de clases e instancias asociado a los diferentes componentes*. Actualmente se identifican y se crean las instancias tanto del dominio funcional como del pedagógico.

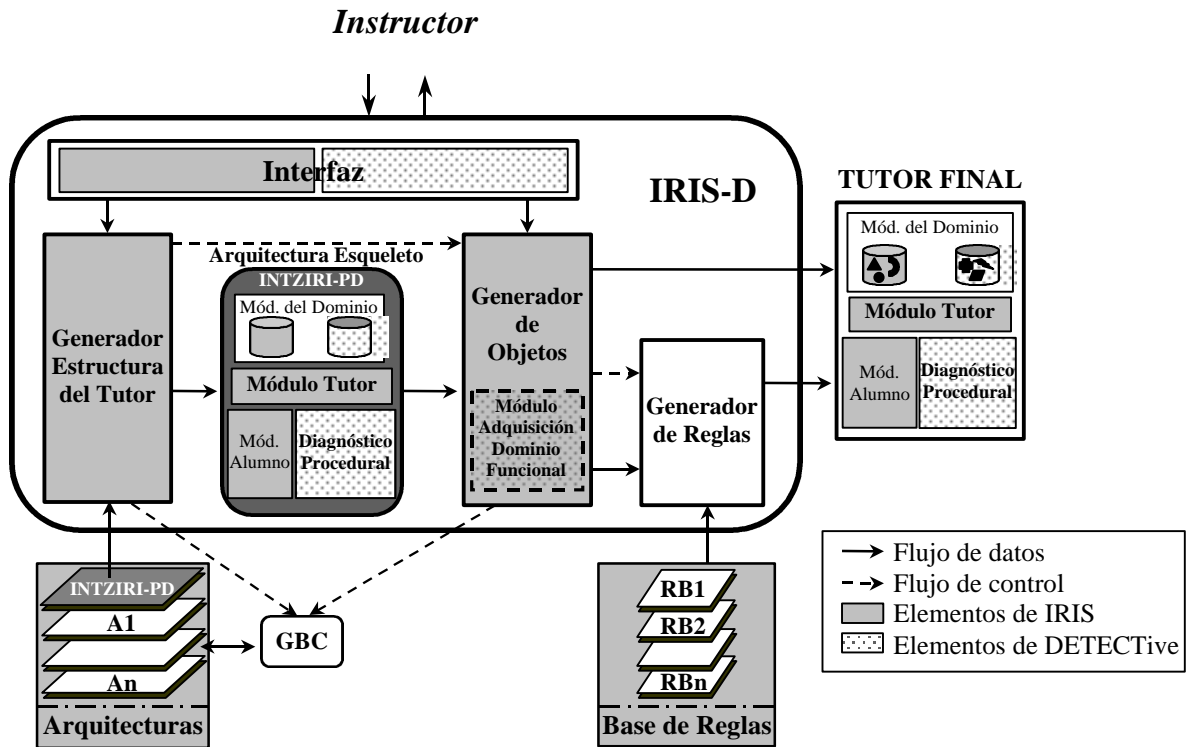


Figura 14. Arquitectura de IRIS-D

Para finalizar, únicamente resta generar o introducir los procesos de decisión adecuados. El Generador de Reglas se encarga de generar o seleccionar, dentro de cada componente del Módulo Pedagógico, el conjunto de reglas de decisión pedagógica necesario para que éste sea operativo. En el prototipo actual de IRIS-D únicamente se incluye el *Selector* con la funcionalidad de restringir o limitar el conjunto de reglas existentes en la jerarquía del kernel inicial en función de los requisitos introducidos. Este módulo utiliza una base de reglas organizada según los diferentes requisitos, con los que mantiene una relación directa según las condiciones utilizadas en ellas.

## 6. Conclusiones

En este artículo hemos presentado una herramienta de ayuda a la generación de sistemas de enseñanza-entrenamiento llamada IRIS-D y su utilización para construir un tutor concreto en el dominio de la máquina-herramienta, LATHE.

IRIS-D es el resultado de la ampliación del sistema IRIS [Arruarte *et al.*, 97] desarrollado por nuestro grupo de investigación. Con IRIS se pueden generar

tutores para dominios tanto procedimentales como declarativos. Éstos tienen la capacidad de realizar una instrucción adaptada al alumno pero carecen de los componentes necesarios para llevar a cabo el diagnóstico de su actividad durante los procesos de resolución de ejercicios procedimentales. El prototipo actual de IRIS está implementado en CLIPS y JAVA.

Observando la necesidad que tenía IRIS de un sistema de diagnóstico, y tras comprobar que esta carencia no era exclusiva del mismo, abordamos la construcción de un sistema de diagnóstico genérico que fuera fácilmente adaptable a cualquier clase de dominio, y así llegamos a obtener DETECTive [Ferrero *et al.*, 97]. Este sistema, implementado también en CLIPS y JAVA, se planteó inicialmente como una herramienta autónoma capaz de diagnosticar y evaluar los errores cometidos por el alumno durante una sesión práctica. Para comprobar y evaluar su validez, DETECTive se adaptó a distintos dominios procedimentales (derivación, fotografía, máquina-herramienta). Concretamente, FROGALAN [Ferrero *et al.*, 99] es el resultado de la adaptación de DETECTive al dominio de la máquina-herramienta. Durante su diseño e implementación, se ha realizado un proceso de

validación según la técnica de “*verificación piloto*” [Shute *et al.*, 93], [Mark *et al.*, 93]. Este proceso se ha complementado con un estudio de campo realizado con un grupo más amplio de estudiantes de nueva formación en el IMH. De forma general podemos decir que el profesorado se siente satisfecho por los resultados obtenidos. Desde el punto de vista del alumnado, la herramienta se acepta con facilidad y su uso no supone una dificultad añadida a las ya inherentes a los ejercicios propuestos.

Una vez integrado DETECTive en IRIS, nuestro próximo objetivo es doble. Por una parte validar la herramienta IRIS-D en cuanto a las facilidades que proporciona para construir nuevos tutores y, por otra, validar las posibilidades de uso y el éxito de los nuevos tutores generados.

## Agradecimientos

Los trabajos expuestos en este artículo han sido financiados por el Gobierno Vasco (UE-1999-36), la Universidad del País Vasco (UPV/EHU) (UPV 141.226-TA109/99), la Diputación Foral de Guipúzcoa (Dpto. Economía) (IRIS-D; OF237/1999) y el CICYT (TIC99-0252). Los autores agradecen la inestimable colaboración y el apoyo del Instituto de la Máquina Herramienta (IMH) de Elgoibar-Gipuzkoa.

## Referencias

- [Anderson *et al.*, 90] Anderson, J.R., Boyle, C.F., Corbett, A.T., Lewis, M.W. “Cognitive Modelling and Intelligent Tutoring”. *Artificial Intelligence* 42, 1990, pp. 7-49.
- [Arruarte *et al.*, 96] Arruarte, A., Fernández-Castro, I., Greer, J. "The CLAI Model. A cognitive theory of instruction to guide ITS development". *Journal of Artificial Intelligence in Education* 7(3/4), 1996, pp. 277-313.
- [Arruarte *et al.*, 97] Arruarte, A., Fernández-Castro, I., Ferrero, B., Greer, J. "The IRIS Shell: how to build ITSs from pedagogical and design requisites". *International Journal of Artificial Intelligence in Education* 8(3/4), 1997, pp. 341-381.
- [Arruarte, 98] Arruarte, A. "Fundamentos y diseño de IRIS. Un entorno para la generación de *Sistemas de Enseñanza Inteligentes*". PhD Thesis. University of the Basque Country, UPV/EHU, 1998.
- [Blessing, 97] Blessing, S.B. “A Programming by Demonstration Authoring Tool for Model-Tracing Tutors”, *Int. Journal of Artificial Intelligence in Education* 8, 1997, pp. 233-261.
- [Bloom *et al.*, 56] Bloom, B.S., Engelhart, M.D., Murst, E.J., Hill, W.H., Drathwohl, D.R. "Taxonomy of Educational Objectives". Logman, 1956.
- [Corbett *et al.*, 90] Corbett, A.T., Anderson, J.R., Patterson, E.G. “Student Modelling and Tutoring Flexibility in the Lisp Intelligent Tutoring System”. *Intelligent Tutoring Systems: At the crossroads of artificial intelligence and education*, C. Frasson & G. Gautier (Eds.), 1990, pp. 83-106. Norwood, NJ: Ablex.
- [Coll, C., 87] Psicología y Curriculum. Ediciones Paidós, 1987.
- [Ferrero *et al.*, 97] Ferrero, B., Urretavizcaya, M., Fernández-Castro, I. “Un Sistema de Diagnóstico y Evaluación basado en DETECTive”. *Actas de la VII Conferencia de la Asociación Española para la Inteligencia Artificial*, Vicent Botti (Eds.), 1997, pp. 261-270.
- [Ferrero *et al.*, 99] Ferrero, B., Fernández-Castro, I., Urretavizcaya, M., Sánchez, H., “Evaluación de capacidades técnicas de operarios en entornos industriales”. *Actas de la VIII Conferencia de la Asociación Española para la Inteligencia Artificial*, A.M. García Serrano, Ramón Rizo, Serafín Moral, Francisco Toledo (Eds.). Vol. II, IA aplicada a los Sistemas de Atención al Usuario, 1999, pp. 10-17.
- [Ferrero *et al.*, 99] Ferrero, B., Fernández-Castro, I., Urretavizcaya, M., “Diagnostic et évaluation dans les systèmes de «training» industriel. Diagnosis and assessment in industrial training systems”. *Simulation et formation professionnelle dans l'industrie* 6(1), 1999, pp. 189-217.
- [Golstein, 82] Golstein, I.P., "The Genetic Graph: A Representation for the Evolution of Procedural Knowledge". *Intelligent Tutoring Systems*, Sleeman, D. & Brown, J.S. (Eds.), 1982, pp. 51-77, Academic Press, .
- [Mark *et al.*, 93] Mark, M.A., Greer, J.E. “Evaluation Methodologies for Intelligent Tutoring Systems”. *International Journal of Artificial Intelligence in Education* 4(2/3), 1993, pp. 129-153.

- [Munro *et al.*, 97] Munro, A., Johnson, M.C., Pizzini, Q.A., Surmon, D.S., Towne, D.M., Wogulis, J.L. "Authoring Simulation-Centered Tutors with RIDES", *Int. Journal of Artificial Intelligence in Education* 8, 1997, pp. 284-316.
- [Murray, 99] Murray, T. "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art", *Int. Journal of Artificial Intelligence in Education* 10(1), 1999, pp. 98-129.
- [Pérez-Colmenar *et al.*, 99] Pérez-Colmenar, R., Elorriaga, J.A., Fernández-Castro, I., "Utilización de GBC-II en la Adquisición de Conocimiento en el area de los STI". *Actas de la VIII Conferencia de la Asociación Española para la Inteligencia Artificial*, A.M. García Serrano, Ramón Rizo, Serafín Moral, Francisco Toledo (Eds.). Vol. II, IA aplicada a los Sistemas de Atención al Usuario, 1999, pp. 54-61.
- [Shute *et al.*, 93] Shute, V.J., Regian, J.W. "Principles for evaluating Intelligent Tutoring Systems". *Journal of Artificial Intelligence in Education* 4(2/3), 1993, pp. 245-271.