
CAPITULO 1

INGENIERÍA DEL CONOCIMIENTO

- **Características Generales de los Sistemas Expertos**
 - **Análisis de la Viabilidad de un Sistema Experto**
 - **Organización General de un Sistema Experto**
 - **Fases de la Adquisición del Conocimiento**
 - **Técnicas de Extracción del Conocimiento**
 - **Método Estructurado de Adquisición del Conocimiento**
 - **Resumen**
 - **Textos Básicos**
-

1. INGENIERÍA DEL CONOCIMIENTO

Al principio de este texto nos habíamos planteado la siguiente cuestión: ¿qué es la inteligencia artificial, una ciencia o una ingeniería? Decíamos también que, como ciencia, la IA trata de desarrollar el vocabulario y los conceptos que permiten ayudar a comprender —y en ocasiones a reproducir— comportamiento inteligente. Por el contrario, como ingeniería, la IA trata de definir y de formalizar un conjunto de métodos que nos permitan adquirir conocimiento de alto nivel, y representarlo según un esquema computacionalmente eficaz, para resolver problemas difíciles en dominios de aplicación concretos. Podríamos resumir estas ideas considerando que la inteligencia artificial, como ciencia, desarrolla modelos; mientras que la inteligencia artificial, como ingeniería —en adelante *ingeniería del conocimiento*— aplica dichos modelos para tratar de resolver problemas intelectualmente complicados. En este contexto encontraremos que los programas resultantes de la aplicación de técnicas de ingeniería del conocimiento —*Sistemas Expertos*— exhiben ciertas características estructurales diferenciales, que describiremos a continuación.

1.1. Características Generales de los Sistemas Expertos

Recordemos que los sistemas expertos son programas inteligentes diseñados para asistir a los expertos humanos en dominios del mundo real, limitados en extensión, pero intelectualmente difíciles. Los sistemas expertos tratan de modelizar en un programa el conocimiento y el modo de razonar de los expertos humanos. Por ello, los sistemas expertos no tienen por qué proponer las mismas soluciones a los mismos problemas. Tampoco se les debe exigir que proporcionen "la mejor solución", basta con que la solución propuesta sea aceptable.

Desde una perspectiva estructural, los sistemas expertos no sólo representan al dominio que tratan de modelizar, también deben conservar representaciones de su propia estructura interna y de su funcionamiento. Esta última característica —que denominamos *autoconocimiento*— es la que permite a los sistemas expertos justificar sus conclusiones, explicar sus procesos de razonamiento, e incrementar dinámicamente el conocimiento que poseen.

Para conseguir avances en los puntos anteriormente expuestos, la ingeniería del conocimiento ha sugerido —y sigue sugiriendo— arquitecturas que separan claramente los conocimientos del dominio de los mecanismos de inferencia y control.

Otro aspecto relativo a los sistemas expertos, y que ya ha sido mencionado, es la importancia decisiva de la experiencia a la hora de resolver un problema. Esta experiencia —que formalizábamos como *conocimiento heurístico*— está íntimamente relacionada con los conocimientos y métodos de actuación de un verdadero experto humano. El grado de experiencia de un profesional cualquiera, en un dominio concreto, suele repercutir en la capacidad del individuo en cuestión para resolver ciertos problemas, en los que otros individuos de la misma profesión fracasan, o no tienen tanto éxito.

En cualquier caso, y aparte de todas las consideraciones anteriores, hay que destacar el papel preponderante de los dominios de aplicación, y de las tareas para las cuales se precisan conocimientos y experiencia. No todos los dominios ni todas las tareas son igualmente apropiadas para la construcción de sistemas expertos. Muchos problemas pueden ser resueltos utilizando técnicas de programación convencional; otros, sin embargo, requieren la aplicación de técnicas de ingeniería de conocimiento.

Atendiendo a los diferentes dominios que un ingeniero del conocimiento puede encontrarse, y considerando *idóneas* aquellas tareas para cuya ejecución y/o resolución se requiere experiencia —en los términos anteriormente expuestos—, podemos ya señalar algunos problemas tipo para los cuales podría ser deseable —e incluso conveniente— el diseño y desarrollo de un sistema experto:

- interpretación de información
- predicción
- pronóstico y prevención
- diseño
- planificación
- monitorización y supervisión
- ayuda a la decisión
- enseñanza asistida por ordenador
- control
- aprendizaje

Estas suelen ser tareas típicas que aconsejan el empleo de técnicas de ingeniería del conocimiento para construir programas inteligentes, en casi cualquier dominio de aplicación.

1.2. Análisis de la Viabilidad de un Sistema Experto

Los sistemas expertos contienen información estructurada, razonada y adaptable a cambios sucesivos. Por otra parte, los expertos humanos en un dominio concreto son escasos, difíciles de encontrar, y caros de mantener. Además, podemos considerar que los sistemas expertos *popularizan* el conocimiento de los expertos humanos y reducen el coste de formación de nuevos expertos. Son, por lo tanto, excelentes herramientas de entrenamiento de expertos en formación. Pero la construcción de un sistema experto realmente útil es un proceso largo y costoso. Se abre así una nueva perspectiva de índole económica y social¹¹³, que podemos resumir del siguiente modo: ¿cuándo es realmente viable la construcción de un sistema experto? Waterman trata de responder a esta pregunta proponiendo una metodología de diseño —que nosotros denominamos *análisis de viabilidad de un sistema experto*— y que está basada en el estudio de cuatro características esenciales: Justificación, Posibilidad, Adecuación y Exito. De esta forma, siempre según Waterman:

¹¹³ ... inherente siempre a todo proyecto de ingeniería

- IF: (1) La construcción de un sistema experto está justificada,
and: (2) La construcción de un sistema experto es posible,
and: (3) La construcción de un sistema experto es adecuada,
and: (4) Hay ciertas garantías de éxito tras la construcción de un sistema experto

THEN: La construcción de un sistema experto es viable

Este análisis de viabilidad, que debe ser previo a cualquier intento de desarrollo —y que es frecuentemente olvidado—, trata de identificar dominios, problemas y tareas, en los que verdaderamente merece la pena, al menos, intentar la construcción de un sistema experto. Digamos que, frente a los programas convencionales, cuando un sistema experto es viable, es porque estamos seguros que nuestro sistema experto va a competir con ventaja frente a otros programas desarrollados mediante técnicas convencionales de *ingeniería del "software"*... pero ¿cuándo podemos decir que un sistema experto está justificado, o su construcción es posible, o su construcción es adecuada, o podemos albergar razonables esperanzas de éxito tras su implementación? Seguimos con la metodología de Waterman.

1.2.1 Justificación

Waterman define el criterio de *justificación* basándose en realidades de tipo social, económico o coyuntural, de acuerdo con el siguiente esquema:

- IF: (1) Hay necesidad de experiencia en un entorno hostil,
or: (2) Existe una verdadera carencia de experiencia humana,
or: (3) Se necesita experiencia simultánea en distintos lugares,
or: (4) Se ha detectado pérdida de experiencia humana¹⁴,
or: (5) Hay una alta tasa de recuperación de la inversión,
or: (6) No hay soluciones alternativas,
or: (7) Un enfoque de programación convencional no es satisfactorio

THEN: El desarrollo de un sistema experto está justificado

En la regla anterior, el análisis de viabilidad introduce matices subjetivos. Esto será una constante en el resto de la metodología propuesta, y será labor del ingeniero de conocimiento decidir cuándo, y cuándo no, se verifican las premisas correspondientes.

1.2.2 Posibilidad

El criterio de posibilidad hace referencia a algunos de los aspectos más pragmáticos en el desarrollo de un sistema experto. Más concretamente:

¹⁴ Se entiende en un dominio concreto

- IF: (1) Existen varios expertos cooperativos,
and: (2) Los expertos logran ponerse de acuerdo,
and: (3) Los expertos son capaces de articular sus métodos y procedimientos de trabajo,
and: (4) Las tareas no son excesivamente difíciles,
and: (5) Las tareas están suficientemente estructuradas,
and: (6) El sentido común no es determinante,
and: (7) Se dispone de un número suficiente de casos relevantes para la verificación y posterior validación del producto,
and: (8) Las tareas no son exclusivamente de carácter teórico

THEN: El desarrollo de un sistema experto es posible

Claramente, en esta fase se trata de averiguar si somos, o no, capaces de articular las fuentes de conocimiento, las tareas que hay que resolver, y los medios disponibles.

1.2.3 Adecuación

Averiguar si la construcción de un sistema experto verifica el criterio de adecuación implica investigar tres frentes, todos ellos vinculados al tipo de problema que queremos resolver. Específicamente, el esquema que debemos seguir es el siguiente:

- IF: (1) La naturaleza del problema aconseja el desarrollo de un sistema experto,
and: (2) La complejidad del problema aconseja el desarrollo de un sistema experto,
and: (3) El alcance del problema aconseja el desarrollo de un sistema experto

THEN: El desarrollo de un sistema experto es adecuado

Cada una de las tres características anteriores —naturaleza, complejidad y alcance— se deben investigar por separado. De este modo:

- IF: (1) El producto desarrollado cubre necesidades a largo plazo,
and: (2) Las tareas involucradas no requieren investigación básica,
and: (3) Las tareas requieren manipulación simbólica,
and: (4) Las tareas requieren soluciones heurísticas

THEN: La naturaleza del problema aconseja la construcción de un sistema experto

Por otra parte:

- IF: (1) Las tareas no son demasiado fáciles,
and: (2) El conocimiento necesario aconseja la definición de más de una base de conocimientos,
and: (3) Es posible planificar efectos

THEN: La complejidad del problema aconseja la construcción de un sistema experto

Finalmente:

- IF: (1) Las tareas tienen valor práctico,
and: (2) Las tareas tienen un "tamaño" manejable,
and: (3) No es previsible que el producto quede inmediatamente obsoleto

THEN: El alcance del problema aconseja la construcción de un sistema experto

1.2.4 Exito

Es el último de los criterios que hay que investigar antes de decidir sobre la viabilidad de un sistema experto. En este caso el estudio debe centrarse casi exclusivamente en aspectos económicos, políticos y de mercado. Este hecho marca una de las diferencias, hasta ahora no confesada, entre la IA como ciencia y la IA como ingeniería. Así, mientras la ciencia se preocupa de la creación y difusión de conocimientos, la ingeniería se preocupa de construir productos rentables. En concreto:

- IF: (1) Se efectúa una transferencia de tecnología adecuada,
and: (2) Los directivos están mentalizados y tienen expectativas realistas,
and: (3) Hay cambios mínimos en los procedimientos habituales,
and: (4) Los usuarios finales no rechazan categóricamente la tecnología de los sistemas expertos,
and: (5) Los resultados no dependen de vaivenes políticos,
and: (6) El dominio es relativamente estable,
and: (7) Los objetivos están adecuadamente definidos

THEN: Hay ciertas garantías de éxito tras la construcción de un sistema experto

Todo este análisis de viabilidad puede ser representado mediante un circuito inferencial muy sencillo, que los autores dejan como ejercicio para los lectores estudiosos.

En cualquier caso, lo que Waterman no dice es qué pasa si alguno de los requisitos del análisis de viabilidad no es satisfecho. Muchas veces, la gran subjetividad de los criterios empleados hace que, independientemente de los resultados del análisis, se acometa el desarrollo del correspondiente sistema experto. Ello es, sin duda, encomiable desde una perspectiva intelectual,... ¡aunque quizás no tanto desde una óptica ingenieril!

Por último, una cuestión interesante, que los autores también proponen como ejercicio, es la realización del análisis de viabilidad de un sistema experto destinado a realizar análisis de viabilidad de sistemas expertos¹¹⁵.

¹¹⁵ ¡A fin de cuentas la metodología ha sido expuesta en forma de reglas de producción!

1.3. Organización General de un Sistema Experto

Una vez investigada la viabilidad del sistema experto conviene recordar brevemente la arquitectura típica de este tipo de sistemas¹¹⁶, que —como ya sabemos— consta de los siguientes grandes bloques:

Bases de conocimientos
Motor de inferencias
Memoria activa

A estos tres grandes bloques añadiremos un cuarto, el de *interfaces con el usuario*, del que en gran medida puede llegar a depender el éxito de nuestro sistema experto.

Recordemos también que la memoria activa contiene la descripción completa del estado actual del sistema durante un proceso de ejecución, por lo que durante la fase de desarrollo —hacia la que va dirigida este capítulo— podemos prescindir de su descripción pormenorizada.

1.3.1 Bases de Conocimientos

En las bases de conocimientos debemos considerar la inclusión y articulación de tres módulos que incorporan diferentes tipos de conocimiento:

Conocimiento declarativo o descriptivo
Conocimiento operativo o de acción
Metaconocimiento

El conocimiento declarativo está siempre referido a los elementos descriptivos del dominio de discurso, contemplados desde una perspectiva estática. Podemos incluir en esta categoría a los siguientes elementos: objetos del universo, relaciones estáticas entre objetos, definiciones, vocabulario y hechos. También podemos incluir aquí otros elementos más complejos como hipótesis, suposiciones, restricciones y taxonomías.

Por otra parte, como parte integrante del módulo de conocimientos operativos del sistema debemos considerar las siguientes entidades: procesos y *demons*, reglas, heurísticas y ejemplos. Todos estos elementos describen al dominio de discurso desde una perspectiva dinámica.

Finalmente, el metaconocimiento —que definiremos como *conocimiento sobre conocimiento*— es un tipo de conocimiento operativo que, a nivel local, permite controlar el funcionamiento del sistema. Un ejemplo típico de metaconocimiento son las *metarreglas* o elementos de control con estructura de reglas, que se utilizan para regular procesos. Por ejemplo :

¹¹⁶ Hay otras arquitecturas igualmente válidas, que no estudiaremos dado el carácter introductorio de este capítulo.

IF: Hay más de una regla activada
THEN: Ejecutar primero la que haya sido activada más recientemente

1.3.2 Motor de Inferencias

El segundo gran bloque de construcción en los sistemas expertos es el motor de inferencias, del que ya hemos dicho que consta básicamente de un *intérprete* y de un *módulo de control*. Más específicamente, deberemos definir e implementar en el motor de inferencias estructuras que nos permitan ejecutar algunas —o todas— de las siguientes tareas:

Definición del tipo de encadenamiento y gestión del mismo
Mecanismos de unificación, emparejamiento e interpretación
Gestión de prioridades, agendas y pizarras¹¹⁷
Modelos y esquemas de razonamiento del sistema
Cálculos

Algunas de las estructuras y tareas que acabamos de mencionar han sido ya descritas en este texto. No obstante el lector estudioso encontrará en la bibliografía propuesta abundante material con el que profundizar en lo ya explicado, y con el que iniciarse en lo que no lo ha sido.

1.3.3 Interfaces

Este es el tercer gran módulo al que debemos prestar atención durante la fase de ingeniería del conocimiento. Su misión es la de permitir que el sistema experto interactúe con el usuario y con el mundo exterior. Dentro de este módulo podemos distinguir tres subsistemas genéricos: *subsistemas de usuario*, *subsistemas de explicación* y *subsistemas de actualización del conocimiento*.

Los subsistemas de usuario permiten la interacción uni- o bidireccional del sistema experto con el usuario o con los dispositivos de entrada de información y datos. Típicamente esta interacción se consigue a través de menús, gráficos, rutinas de adquisición de datos que actúan sobre sistemas conectados a instrumentos, accesos a bases de datos, etc.

Por otra parte, los subsistemas de explicación facilitan el seguimiento de los procesos inferenciales realizados por el sistema. Este seguimiento suele ser requerido en momentos precisos mientras tiene lugar un proceso consultivo determinado, aunque también puede ser utilizado para la validación y depuración del producto desarrollado. Un subsistema típico de explicación debe ser capaz de responder adecuadamente a cuestiones como las siguientes: ¿cómo? ¿por qué? y "justifica". Los subsistemas de explicación utilizan con profusión el *autoconocimiento* del sistema.

¹¹⁷ Las pizarras son estructuras de control que nos permiten organizar de manera relativamente sencilla grandes programas de IA.

Por último, los subsistemas de actualización del conocimiento deben posibilitar la adquisición de nuevo conocimiento, y la constante actualización del conocimiento ya existente en el sistema. Típicamente esta tarea se facilita considerablemente con el empleo de procesadores de lenguaje natural, o la utilización de herramientas de ingeniería del conocimiento —"Shells"—.

1.4. Fases de la Adquisición del Conocimiento

Ya hemos mencionado que no podemos considerar que algo muestra un comportamiento inteligente si no utiliza de manera eficaz y eficiente un conjunto mínimo de conocimientos. En este sentido, lo desarrollado hasta ahora corresponde al dominio de los llamados *problemas juguete*, que son útiles para ayudarnos a comprender los principios básicos de la inteligencia artificial, pero que no nos permiten formarnos una idea de lo que realmente subyace tras los sistemas expertos. Al respecto, disponemos ya de una estructura, de una arquitectura, y de unas técnicas de búsqueda de soluciones, de representación del conocimiento, y de razonamiento. Sin embargo, nos falta todavía dotar a los elementos anteriores de contenido, es decir, de *conocimiento*.

La adquisición del conocimiento —en el ámbito estricto de la ingeniería del conocimiento— es, sin duda, uno de los "cuellos de botella" de la inteligencia artificial, hasta el punto que —dado el espectacular desarrollo de las herramientas de que se dispone hoy día para la construcción de sistemas expertos (i.e., "Shells")—, el problema de la ingeniería del conocimiento es el problema de la adquisición del conocimiento.

Dado un dominio cualquiera, no podemos pretender resolver el problema de la adquisición del conocimiento sentando a un grupo de expertos y diciéndoles: ¡Venga, contadme todo lo que sabéis sobre este problema...! Muy al contrario, hay que tener bien presente que el objetivo final es la creación de un modelo computacional cualitativo de comportamiento inteligente, en un dominio de aplicación concreto. Para ello hay que unificar terminología. Como paso previo al establecimiento de una metodología específica de adquisición del conocimiento, definiremos los siguientes conceptos:

Dominio: Área de aplicación sobre la que queremos construir nuestro sistema (e.g., enfermedades infecciosas)

Tareas: Problemas que se van a presentar en el dominio, y que tendremos que resolver (e.g., tratamiento terapéutico de la meningitis)

Métodos: Estrategias y/o procedimientos de resolución de los problemas planteados.

Sobre todos y cada uno de los puntos anteriores tendremos que definir nuestra metodología de adquisición del conocimiento, teniendo en cuenta que:

modelizar un problema implica extraer conocimiento del mundo exterior, articularlo, estructurarlo, y traducirlo computacionalmente. No se modeliza un problema simplemente transfiriendo experiencia y conocimientos.

las bases de conocimientos son modelos del dominio de aplicación. En ellas se incluyen entidades relevantes, procesos, y estrategias de resolución de problemas.

el conocimiento de los expertos, tal y como ellos nos lo comunican es incompleto, aproximado, y poco operativo computacionalmente hablando.

En base a los argumentos anteriormente expuestos, nuestra labor de adquisición del conocimiento debe seguir una metodología concreta que observe las siguientes fases: *conceptualización, formalización, elicitación, operacionalización, verificación y revisión.*

Conceptualización

La conceptualización se refiere al dominio de aplicación y a las tareas que pretendemos resolver. Esta fase nos va a permitir identificar los elementos clave, las relaciones, los procesos, y otras entidades del dominio que son relevantes en la construcción de un sistema experto concreto. Para conceptualizar correctamente un dominio tendremos que efectuar un análisis, completo y detallado, de las tareas identificadas, y tratar de caracterizar qué es lo que hay que resolver, cuáles son las entradas disponibles, cuáles son las salidas deseadas, y qué tipo de información necesitamos para resolver los problemas. Además, tendremos que comprender, pensando en su posterior articulación, los métodos y técnicas apropiadas para la resolución de los problemas del dominio.

Un ejemplo muy simplificado de conceptualización podría ser el siguiente:

<u>Dominio:</u>	Enfermedades infecciosas
<u>Tarea:</u>	Clasificación de cierta sintomatología en base a las categorías diagnósticas conocidas
<u>Entradas:</u>	Conjunto de síntomas
<u>Salidas:</u>	Una lista de posibles diagnósticos, ordenados según su relación con los síntomas
<u>Tipo de Información:</u>	Asociaciones empíricas entre evidencias y categorías diagnósticas

Formalización

Podemos definir la formalización como el proceso de construcción de representaciones simbólicas, que nos permitan "traducir" los resultados de la conceptualización —hasta ahora de carácter abstracto—, en algo que pueda ser implementado en una máquina. Así, la formalización de conceptos implica asimilar las entidades clave, las relaciones, los métodos, etc., y encontrar un procedimiento —estructurado o lógico—, para su posterior representación computacional (e.g., "frames", reglas, predicados, funciones,...) En definitiva, aquí se trata de decidir qué tipo de

estructura es más útil para representar cada una de las entidades relevantes identificadas tras el proceso de conceptualización (e.g., representar las asociaciones empíricas del ejemplo anterior mediante reglas que relacionen síntomas con enfermedades,...)

Elicitación

Esta fase es la primera "bajada a la arena" en la adquisición del conocimiento. Básicamente, la elicitación es un proceso de extracción del conocimiento de los expertos humanos, que se realiza de una forma estructurada y consistente con el proceso de conceptualización. La elicitación supone, en primer lugar, diseñar algún tipo de estrategia estructurada de interlocución con los expertos.

Ello puede obligarnos a identificar previamente distintas categorías de información, y establecer unos formatos adecuados a la lógica de los procesos con los que nos enfrentamos. A continuación, procede el diseño y materialización de alguna estructura física (e.g., formularios, interfaces,...), con la que llevar a cabo la elicitación del conocimiento. La Figura 1.1 muestra un documento típico de elicitación, que también puede ser utilizado para la ulterior validación de nuestro sistema inteligente.

Finalmente, la fase de elicitación concluye con la traducción de la información adquirida en una representación —sobre el papel— estructurada o formal.

NST-EXPERT VALIDATION CHART

To be filled by the nurse		Name: <input style="width: 80px;" type="text"/>	
		Age <input style="width: 80px;" type="text"/>	
		Gestational age <input style="width: 80px;" type="text"/>	
		Test duration (min) <input style="width: 80px;" type="text"/>	
Patient Number <input style="width: 80px;" type="text"/>			
Race <input style="width: 80px;" type="text"/>			
GTPAL <input style="width: 80px;" type="text"/>		Mean baseline (bpm) <input style="width: 80px;" type="text"/>	
Last smoke <input style="width: 80px;" type="text"/>		Coef. Baseline Variation or BPM <input style="width: 80px;" type="text"/>	
Activity level <input style="width: 80px;" type="text"/>			
Drugs <input style="width: 80px;" type="text"/>		Pregnancy Risk Factors: <input style="width: 80px;" type="text"/>	
Number of Accelerations <input style="width: 80px;" type="text"/>		(>10bpm) <input style="width: 80px;" type="text"/>	(>15bpm) <input style="width: 80px;" type="text"/>
		Num. Decelerations <input style="width: 80px;" type="text"/>	Special conditions (e.g. fever) <input style="width: 80px;" type="text"/>
To be filled by physician		MD to enter if DECELERATIONS ARE PRESENT	
The test is <input type="checkbox"/> NORMAL			
<input type="checkbox"/> ABNORMAL			
If test abnormal, reasons <input style="width: 80px;" type="text"/>			
Outcome Expectations: <input type="checkbox"/> GOOD			
<input type="checkbox"/> BAD → If bad, please quantify:			
COMMENTS AND SUPPORTING DATA	Slightly possible	< 25%	<input type="checkbox"/>
	Moderately possible	25-45%	<input type="checkbox"/>
	Quite possible	45-55%	<input type="checkbox"/>
	Very possible	55-75%	<input type="checkbox"/>
	Sure or Almos Sure	>75%	<input type="checkbox"/>
MANAGEMENT RECOMMENDATIONS			
- Repeat the NST. Specify Frequency (eg. 2x/week) _____ <input type="checkbox"/>			
- Perform Another test (eg. Amniocentesis on Maturity, CST, BPP) _____ <input type="checkbox"/>			
- Perform Emergency Cesarean Section _____ <input type="checkbox"/>			
- Labor Induction _____ <input type="checkbox"/>			
- Wait for Spontaneous Labor and Normal Delivery _____ <input type="checkbox"/>			
- Other (Specify) _____ <input type="checkbox"/>			
NAME OF REVIEWER		SIGNATURE	
		DATE	

Figura 1.1. Documento de Elicitación / Validación

Operacionalización

En esta fase se trata de hacer computacionalmente operativos los conocimientos previamente elicitados. Para ello trataremos de encontrar —o de construir— los procedimientos de representación más apropiados que permitan al conocimiento elicitado ejecutar las tareas deseadas. Ello supone:

encontrar el procedimiento computacional que mejor simule las estrategias de resolución descritas por los expertos

seleccionar, adaptar o desarrollar intérpretes

construir prototipos y realizar simulaciones que incluyan diversos modos de interacción con el usuario

Verificación y revisión

Es la última fase del proceso, y con ella se pretende comprobar el funcionamiento de las estructuras implementadas y, si procede, efectuar las correcciones oportunas. Desgraciadamente, constataremos la necesidad de efectuar varias reimplementaciones a medida que el proyecto avanza, según vamos analizando los resultados de las sucesivas verificaciones y revisiones. Hay varios factores que obligan a tales reimplementaciones. En primer lugar, el experto suele cambiar de idea conforme el sistema evoluciona. En segundo lugar, el experto acaba por acostumbrarse al prototipo, y comienza a exigir más cosas. Por otra parte, el ingeniero de conocimiento va aprendiendo sobre el dominio de aplicación, y es él mismo quien sugiere posibles mejoras del sistema. Por último, el ingeniero de conocimiento se familiariza con el entorno de trabajo del sistema, y suele proponer modificaciones acerca de la integración final del sistema.

En cualquier caso, la verificación y revisión del sistema en desarrollo también debe seguir unas pautas bien definidas. Un procedimiento adecuado para llevar a cabo la verificación y revisión de un sistema inteligente podría ser el siguiente:

construir un primer prototipo que opere siempre con un mismo conjunto de entradas, y refinarlo hasta obtener la respuesta deseada

realizar una primera revisión en la que se le permita al usuario introducir nuevos datos. Ello supone dinamizar al sistema en desarrollo, y permite detectar errores en los procesos de razonamiento implementados.

realizar sucesivas revisiones, según el procedimiento anterior, hasta conseguir un sistema optimizado que sea capaz de trabajar con información y datos reales.

1.5. Técnicas de Extracción del Conocimiento

Una vez descritas las etapas fundamentales del proceso de adquisición del conocimiento, mencionaremos brevemente algunas de las distintas técnicas que se pueden emplear en el proceso de *extracción del conocimiento*.

Observación directa

En realidad, la *observación directa* es una técnica de extracción del conocimiento, previa al diseño real del sistema inteligente. La observación directa

consiste en la simple observación pasiva del modo en que un experto se enfrenta con los problemas del dominio de aplicación. El objetivo es claro: tratar de familiarizar al ingeniero de conocimiento con el dominio de aplicación y con el entorno.

Dissección del problema

Tras la observación directa procede la discusión informal con los expertos sobre un conjunto de problemas representativos del dominio. Este proceso se denomina *dissección del problema*, y el objetivo buscado es tratar de averiguar de qué manera los expertos tienden a organizar sus conocimientos, cómo representan mentalmente sus conceptos y sus ideas, y cómo analizan la información inconsistente, inexacta o imprecisa. Al respecto, podemos formular las preguntas siguientes:

¿Qué características diferenciales tiene este problema concreto frente a otros problemas del dominio?

¿Qué tipo de información es precisa, y qué datos son relevantes en la resolución del problema?

¿Qué tipo de soluciones son adecuadas?

¿Podemos reducir el problema planteado a subproblemas no interactivos?

¿Qué tipo de conocimientos se necesitan para resolver el problema?

¿Qué elementos básicos deben incluirse en una explicación correcta, adecuada, y suficientemente informativa?

El resultado de una dissección del problema bien hecha suele ser la aparición de nuevos términos, de nuevos conceptos, y de nuevas relaciones.

Descripción del problema

Esta técnica de extracción del conocimiento supone describir problemas típicos relacionados con cada categoría importante de respuestas. Se trata de descubrir estrategias y enfoques básicos, de carácter general, para tratar de establecer una organización jerárquica del conocimiento de los expertos.

Análisis

Para aplicar esta técnica se requiere que los expertos resuelvan, en presencia del ingeniero del conocimiento, un conjunto de problemas del dominio. Los problemas planteados deben ser realistas, y el experto comentará todos y cada uno de sus procedimientos de resolución. Las explicaciones deberán ser pormenorizadas, exhaustivas, y detalladas. Por su parte, el ingeniero del conocimiento deberá cuestionar cada paso de la resolución efectuada por el experto, y tratará de generalizar las

conclusiones y metodologías aplicadas, sin que ello suponga pérdida de la calidad inferencial.

Refinamiento

Se puede considerar que el *refinamiento* es un proceso de análisis invertido, en el que el experto plantea problemas al ingeniero de conocimiento para que éste los resuelva. Los problemas planteados deben ser de dificultad creciente, y el experto debe supervisar al ingeniero del conocimiento mientras trata de resolver los problemas. Si ya existe un prototipo operativo, el proceso de refinamiento se repite con el sistema. El objetivo final es la crítica de planteamientos, la búsqueda de soluciones alternativas, y la optimización de los procesos de resolución.

Examen

Esta técnica de extracción supone la revisión microscópica del conocimiento del sistema. Aquí es el experto quien debe analizar todas y cada una de las reglas del sistema, supervisar las estrategias de resolución y, en definitiva, dar el *visto bueno* al prototipo construido. Como resultado colateral, el resultado de un buen examen suele ser el incremento y mejora de las posibilidades de explicación y justificación del sistema experto.

Validación

Más que una técnica de extracción es un proceso completo mediante el cual se pretende comprobar que el sistema experto, considerado globalmente, funciona correctamente frente a problemas reales, y en entornos reales. Por su extraordinaria importancia, la validación de sistemas expertos será tratada en profundidad en un capítulo aparte.

Todas las técnicas de extracción del conocimiento que se acaban de exponer se apoyan en una serie de *herramientas de ayuda* como son:

entrevistas
análisis de tareas y protocolos
clasificación de conceptos

Las *entrevistas* permiten generar conocimientos sobre la terminología del dominio y sobre el universo de discurso que se pretende modelizar, y pueden ser de dos tipos: no estructuradas —o informales—, y estructuradas —con guión—

Por otra parte, con el *análisis de tareas y protocolos* se trata de determinar cuáles son las limitaciones impuestas por la naturaleza del problema, cuál es el conocimiento relevante, y cuál es la estructura genérica de los diversos problemas del dominio. Para lograr los objetivos propuestos hay que investigar el comportamiento del

experto en el marco de las tareas identificadas, analizarlo minuciosamente, y conseguir un conjunto de reglas relevantes.

Finalmente, la *clasificación de conceptos* es útil ya que, en todo proceso inteligente —aparte del conocimiento detallado disponible, estrictamente relativo al dominio— los expertos utilizan un conocimiento más global, de muy alto nivel, y muy estructurado, que les permite optimizar procesos inferenciales complicados. Como consecuencia de ello, los expertos tienden a agrupar la información en clases, y luego tratan de establecer jerarquías entre ellas.

1.6. Método Estructurado de Adquisición del Conocimiento

La discusión efectuada en la sección anterior, bastante abstracta y lenta de digerir, nos va a permitir no obstante proponer una metodología concreta para la adquisición del conocimiento, en la que podemos distinguir las siguientes fases: *inicial*, *metodológica*, *estructurada*, y *evaluación*. Estas cuatro fases constituyen realmente una metodología completa de ingeniería del conocimiento —no sólo de adquisición—, ya que el resultado final obtenido tras su aplicación debería ser un sistema experto perfectamente operativo.

Esquema de la Fase Inicial

Realización de entrevistas no estructuradas ni dirigidas
Obtención de ejemplos para su análisis conjunto por parte de expertos humanos y de ingenieros de conocimiento
Establecimiento de un conjunto inicial de reglas

Esquema de la Fase Metodológica

Estructuración macroscópica del conocimiento obtenido en la fase anterior
Organización global, y tentativa, del sistema en desarrollo
Establecimiento de clases, y clasificación de todos aquellos elementos de información que compartan características

Esquema de la Fase Estructurada

Organización y estructuración microscópica de los conjuntos de información que comparten características
Diseño y construcción de prototipos y módulos individuales
Evaluación y refinamiento, por separado, de todos y cada uno de los prototipos y módulos construidos

Esquema de la Fase de Evaluación

Integración de módulos

Optimización de las estructuras de control
Adecuación de las interfaces y de los mecanismos de explicación y de justificación
Validación del sistema en el laboratorio
Validación del sistema en su entorno real de trabajo
Vuelta atrás

Cuando la vuelta atrás ya no sea necesaria habremos conseguido un sistema experto operativo, listo para su eventual comercialización.

1.7. Resumen

En este capítulo hemos pretendido dar una perspectiva amplia, aunque introductoria, de los problemas fundamentales de la ingeniería del conocimiento. Al respecto, se hace hincapié en la aplicación de los métodos estudiados en temas anteriores para diseñar productos "software" que sean operativos. Dada la naturaleza ingenieril del contenido de este capítulo, comenzamos con un breve recordatorio de la peculiar arquitectura de los sistemas expertos. A continuación se propone un procedimiento para averiguar si la construcción de un sistema experto es viable. Este procedimiento está basado en el análisis de cuatro características —algunas de ellas difícilmente objetivables— que se refieren al dominio de aplicación, a las características de los problemas del dominio, a las características y actitud de los usuarios finales, y a la disponibilidad de tecnología apropiada. Una vez decidida la viabilidad de un sistema experto se describen los requisitos de diseño que hay que contemplar para diseñar y construir cada uno de los grandes módulos de un sistema experto comercial: bases de conocimientos, motor de inferencias, e interfaces. Seguidamente se discute con cierto detalle el proceso global de la adquisición del conocimiento como método de elaboración de un modelo computacional de comportamiento inteligente. Esta fase puede considerarse como el verdadero cuello de botella de la ingeniería del conocimiento. En este sentido, tras efectuar un recorrido por las distintas etapas de la adquisición del conocimiento, hacemos una breve mención a las distintas técnicas de extracción del conocimiento y, finalmente, proponemos un método estructurado para la adquisición del conocimiento.

1.8. Textos Básicos

Hayes-Roth, Waterman, Lenat, "Building expert systems", Addison-Wesley, eds., 1983.

Maté, Pazos, "Ingeniería del conocimiento: diseño y construcción de sistemas expertos", CETTICO, eds., 1988.