



Frames

Aspectos básicos

Sistemas Expertos

4/09/12

Frames

Aspectos básicos

Definición

Es una colección de atributos que definen el estado de un objeto y su relación con otros objetos.



Es una estructura de datos que sirve para representar objetos que son comunes a una situación dada, como los estereotipos (ejemplos típicos).

Suspendisse potenti.

Estructura para organizar el conocimiento con énfasis en el conocimiento por omisión.

El término de *Frame* (marco) se utiliza para varias formas de representación del conocimiento:

- Los Frames se utilizan preferentemente para la representación de conocimientos susceptibles de división en unidades, describibles mediante la indicación de cualidades.
- Los Frames son una especie de formulario, un “armazón de datos”, que puede contener las cualidades de una unidad de conocimiento.

Son bastante útiles para simular conocimiento de sentido común (área de muy difícil manejo en Computación)

Representa conocimiento relacionado con un tema concreto que cuenta con mucho conocimiento predeterminado

Es análogo a un registro (estructura en C) (slots y fillers).

“Añadido a cada Frame hay varios tipos de información. Parte de esta información hace referencia a cómo utilizar el frame; otra se refiere a lo que uno pueda esperar en segundo lugar. Y otra a su vez indica qué hacer si tales esperanzas no son confirmadas” (Marvin Minsky, 1974)

Un Frame es, por lo tanto, la división de objetos o de situaciones en sus componentes.

Cada marco representa una clase de elementos de la misma manera que un nodo en una red semántica

Componentes de un frame

Un Frame se divide en campos, llamados *Slots* (descriptores). Un Slot se define también como un *Atributo*. Un Slot describe una cualidad de una unidad de conocimiento.

El slot nos da un lugar para colocar sistemáticamente un componente de nuestras experiencias anteriores con relación a las clases de elementos representados

Se les pueden dar valores y tipos default

Un Slot contiene, por su parte, las llamadas *Facets* (facetas). El tipo y la cantidad de Slots y de Facets dependen de la variante de Frame. En la mayoría de ellas hay, al menos, una Facet para su definición y una para su valor. Las demás se utilizan, por ejemplo, para fijar limitaciones para posibles valores de Slot.

Cómo puede representarse el conocimiento mediante Frames?

Con un *Frame* se describe una unidad de conocimiento. Para ello hay que definir unos *Slots* apropiados con los que se puedan definir las cualidades de esta unidad de conocimiento. Los *Slots* suelen designarse por un *Facet* con el nombre del *Slot*. La cantidad de *Slots* de un *Frame* puede elegirse de forma variable en la mayoría de los lenguajes de representación del conocimiento orientado a *frames*. La cantidad y la importancia de cada *Facet* vienen normalmente fijadas por el lenguaje de representación del conocimiento, que permite aplicar de forma más eficiente los mecanismos de valoración.

Los valores de la Facet pueden estar asignados previamente o estar vacíos. En el transcurso de la valoración del conocimiento, pueden llenarse las Facets vacías.

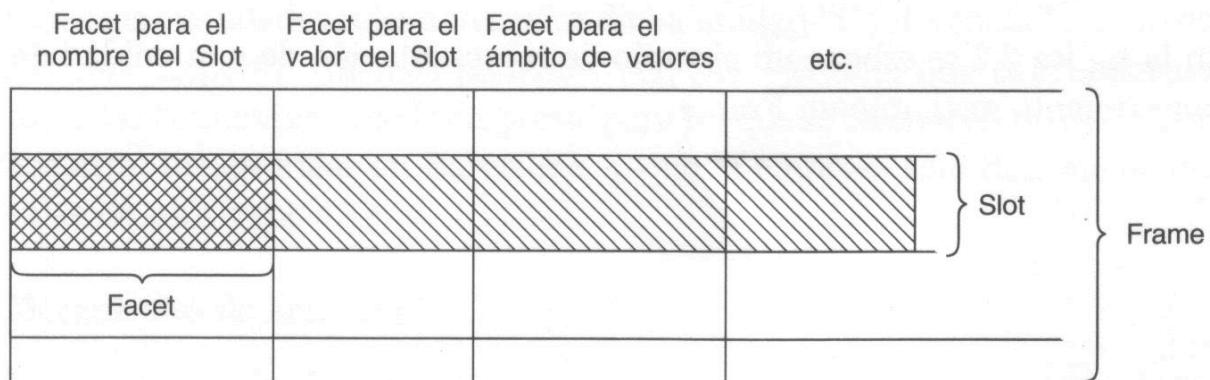


Figura 2.1 Esquema de un Frame

Frame: Conferencia

| Nombre del Slot | Valor del Slot | Ambito de valores |
|-----------------|----------------|---|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | | Cualquiera de las dos: [Sala de conferencia V, Sala de conferencia E] |
| Tema | | Texto |
| Participantes | | Nombres |

Figura 2.2 Ejemplo de representación de una unidad de conocimiento mediante un Frame con el nombre de *Conferencia*

Con la definición de la unidad de conocimiento “Conferencia” de la figura 2.2 hemos fijado ya un cierto conocimiento. De esta forma relacionamos todas las cualidades relevantes para dicha unidad “Conferencia” dentro del Sistema Experto. Además podemos ver, por ejemplo, que prácticamente todas las conferencias de la empresa, para las que se ha concebido el Sistema Experto, tienen lugar en Barcelona, y que la empresa sólo dispone de dos salas de conferencia.

- La información varía dependiendo del contexto
- El frame también cuenta con fillers.
- Los fillers (rellenos) pueden ser valores, como una propiedad en la ranura del nombre o un rango de valores, como la ranura de tipos

- Se pueden tener fillers (rellenos) que sean resultado de un procedimiento adjunto (hacen llamadas a proc)
 - Es-necesario proc que se ejecutarán cuando se necesita un valor de relleno pero no hay ninguno presente o el valor por default no es adecuado
 - Si-añadido se ejecuta cuando se agrega un valor de una ranura
 - Si-es-eliminado se ejecuta cada vez que se elimina un valor de una ranura. Generalmente cuando un valor es obsoleto

Mecanismo de herencia

Mecanismos de herencia

La mayoría de los lenguajes de representación del conocimiento orientados a Frames permiten estructurar jerarquías para las unidades de conocimiento representadas. Con los Frames se pueden crear relaciones de clase/subclase/instancias. Un Frame, especificado como clase, puede ceder contenidos de las Facets de sus Slots (en especial los valores de Slot) a subclases. Las instancias son los elementos más inferiores de la jerarquía. Los Frames que actúan como instancias, sólo pueden heredar.

Los mecanismos de herencia son una característica muy importante de las formas de representación del conocimiento orientadas a Frames. Debido a las múltiples variantes de mecanismos de herencia, no podemos entrar en este capítulo con mayor detalle sobre formas especiales. En el capítulo 4 presentamos algunos mecanismos concretos de herencia.

En las figuras siguientes 2.3 y 2.4 se clarifica mediante un ejemplo la idea de la formación de clase y la herencia.

Frame: Conferencia

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | | de los disponibles [Sala V, E] |
| Tema | | Texto |
| Participantes | | Nombres |

Frame: Conversación sobre Distribución

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|------------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | Sala de conferencias V | de los disponibles [Sala V, E] |
| Tema | Distribución | Texto |
| Participantes | | Nombres |

Frame: Conversación sobre desarrollo

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|------------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | Sala de conferencias V | de los disponibles [Sala V, E] |
| Tema | Desarrollo | Texto |
| Participantes | | Nombres |

Leyenda:

Sólo debe especificarse los valores de Facets subordinados. Los demás valores se definen automáticamente por herencia.

Figura 2.3 Ejemplo de representación de conocimiento por clases y con ayuda de la

Como se muestra en el ejemplo de la figura 2.3, podemos representar la agenda de conferencias de una empresa.

La clase “superior” viene descrita por el Frame Conferencia. Esta clase posee dos subclases, la distribución y el desarrollo.

Con los mecanismos de herencia antes mencionados se traspasan los valores de Facet definidos en la clase superior a las subclases. Para la definición de las subclases indicamos sólo los datos adicionales específicos. En el ejemplo dado, está prescrita la definición fija de las salas de conferencia para los distintos tipos de conferencias. Si hace falta, podemos modificar también esta ocupación previa de Facet por la valoración del conocimiento.

Mecanismo de valoración para Frames

A parte de los mecanismos de herencia, los Frames, tal como los hemos descrito hasta ahora, ofrecen sólo la posibilidad de representar un conocimiento estático, declarativo.

Para la valoración del conocimiento representado por Frames, hay que relacionar entre sí reglas y procedimientos y la forma en que lo hagamos dependerá del mecanismo de valoración sobre el que se base.

Presentaremos aquí dos posibilidades a título de ejemplo:

- En los Frames existen Slots asociados con procesos. Estos pueden incluso ser heredables en algunos lenguajes de representación del conocimiento. Los procesos se ejecutan cuando se dan ciertas condiciones, por ejemplo, por el acceso de lectura o escritura en un Slot determinado. Por su parte, los procesos pueden a su vez acceder a Slots leyendo o escribiendo, o incluso definir nuevos Frames.
- Adicionalmente a los Frames, existen reglas, en las que podemos fijar distintas acciones. Pueden, por ejemplo, acceder a los Slots leyendo o escribiendo o definir nuevos Slots o Frames.

En la figura 2.4 damos un ejemplo de lo que podría haber pasado con los Frames de la Figura 2.3 durante la valoración del conocimiento.

En el ejemplo de la figura 2.4 se determinó en la valoración (por ejemplo mediante reglas), que la Sra. Madrigal tiene que estar presente en ambas reuniones, introduciendo este hecho como característica para la clase Conferencia. Además se descubrió que se había celebrado una conferencia en concreto (una reunión sobre distribución) y que debía ser tratada. A estos efectos se creó un nuevo Frame (una instancia) con el nombre de Reunión Distribución N° 1 y se llenó con los contenidos correspondientes.

Frame: Conferencia

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|----------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | | de los disponibles [Sala V, E] |
| Tema | | Texto |
| Participantes | Sra. Madrigal | Nombres |

Frame: Conversación sobre Distribución

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|------------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | Sala de conferencias V | de los disponibles [Sala V, E] |
| Tema | Distribución | Texto |
| Participantes | Sra. Madrigal | Nombres |

Frame: Conversación sobre desarrollo

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|------------------------|--------------------------------|
| Fecha | | Secuencia [xx-xx-xx] |
| Hora | | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | Sala de conferencias E | de los disponibles [Sala V, E] |
| Tema | Desarrollo | Texto |
| Participantes | Sra. Madrigal | Nombres |

Frame: Conferencia

| Nombre del Slot | Valores del Slot | Ambito de valores |
|-----------------|--|--------------------------------|
| Fecha | 22.02.88 | Secuencia [xx-xx-xx] |
| Hora | 10:15 | Secuencia [xx:xx] |
| Lugar | Barcelona | Valor constante |
| Local | Sala de conferencias V | de los disponibles [Sala V, E] |
| Tema | Distribución | Texto |
| Participantes | Sra. Madrigal Sr. Martí Pol Sr. Monzó | Nombres |

Leyenda:

Sólo se determinaron e introdujeron los valores de Facet en negrita en el momento de la valoración. Los demás valores estaban ya definidos o fueron heredados

Figura 2.4 Ejemplo de la utilización de Frames durante la valoración

Con las reglas podemos representar aspectos procedurales de una base de conocimientos (por ejemplo: “¿Cómo se pueden determinar hechos a partir de otros hechos ya conocidos?”, o “¿Qué hay que hacer en determinadas circunstancias?”) en una forma no procedural (declarativa). No se determina aquí particularmente cómo hay que ejecutar el proceso (procedural) en las consultas, sino sólo qué *acciones* deben llevarse a cabo al aparecer determinadas *condiciones*. La aplicación de esta información para la realización de ejecuciones reales por “encadenamiento” de reglas se transfiere a procesadores generales (es decir, independientemente de la aplicación) –también llamados “*reasoning engines*”– que pueden trabajar según distintas estrategias de resolución de problemas.

Las reglas trabajan con los objetos de una base de datos o de hechos. Las *acciones* principales son la ampliación de la base de hechos con más hechos, es decir, conocimiento que ayuda a la resolución de un determinado planteamiento. El conocimiento de hechos se representa en ExTool definiendo una terminología y taxonomía. En otras palabras, se fijan los *conceptos* o los símbolos con los que se quiere trabajar y se describen estos conceptos detalladamente respecto a distintas facetas; además se definen ciertas relaciones existentes entre estos conceptos.

En la definición de la terminología se distinguen principalmente tres tipos de objeto:

- Los *objetos* complejos de la aplicación,
- Las *cualidades* de estos objetos,
- Los *valores* que pueden tener estas cualidades.

Estos se representan con tres tipos de objeto correspondientes, definidos como *Frames*, *Atributos* y *Valores*. La función para la definición de la terminología del campo científico está, por lo tanto, en la definición de los distintos Frames, sus atributos y los valores dados a éstos.

Como ejemplo podemos tomar un sistema para el diagnóstico de fallos en un vehículo. Los objetos complejos serían, por un lado, el coche mismo, pero también partes de éste, como el motor, la carrocería y el bastidor. Habría, por lo tanto, que definir cuatro Frames con los nombres *coche*, *motor*, *carrocería* y *bastidor*. A éstos hay que subordinar unos *atributos*. Para el motor podrían ser *cilindrada*, *potencia fiscal*, para la carrocería quizás el *color*, si es de importancia para el problema planteado, etc.

Los posibles valores de estos atributos son en parte numéricos, en parte texto, de naturaleza más bien *simbólica*. Los atributos *simbólicos* son de especial importancia en muchos sistemas basados en el conocimiento. Los símbolos que un atributo de este tipo puede tomar como valor son una parte importante de la terminología específica. En ExTool pueden definirse los símbolos utilizados como valores numéricos independientemente de los atributos especiales en las llamadas *clases de valores*. Para nuestro ejemplo podemos definir la clase de valor *Color* e introducir los valores *rojo, verde, azul,...* Por lo demás podemos clasificar esta clase de valor al atributo *Color* del Frame *Carrocería*.

Con la definición de la terminología en forma de Frames, atributos y valores, no hemos descrito aún del todo el dominio de la aplicación. Faltan datos sobre las relaciones entre los objetos. Especialmente entre los objetos complejos representados por Frames. Una estructura particularmente importante es la composición de objetos complejos con partes que a su vez constan de otras partes. Esta circunstancia no sólo se da en estructuras físicas, como por ejemplo la composición de un coche con carrocería, motor y bastidor, donde el motor a su vez consta de cárter, émbolo, etc. En el diagnóstico médico podría ser útil observar al paciente como el conjunto compuesto por la *anamnesis, resultados de análisis complejos e hipótesis* sobre su enfermedad. Por ello, en ExTool, el medio primario de estructuración de la base de conocimientos es una jerarquía de *totalidad-componentes*.

Para nuestra base de conocimientos de coches, la estructura podría ser como en la figura 4.1:

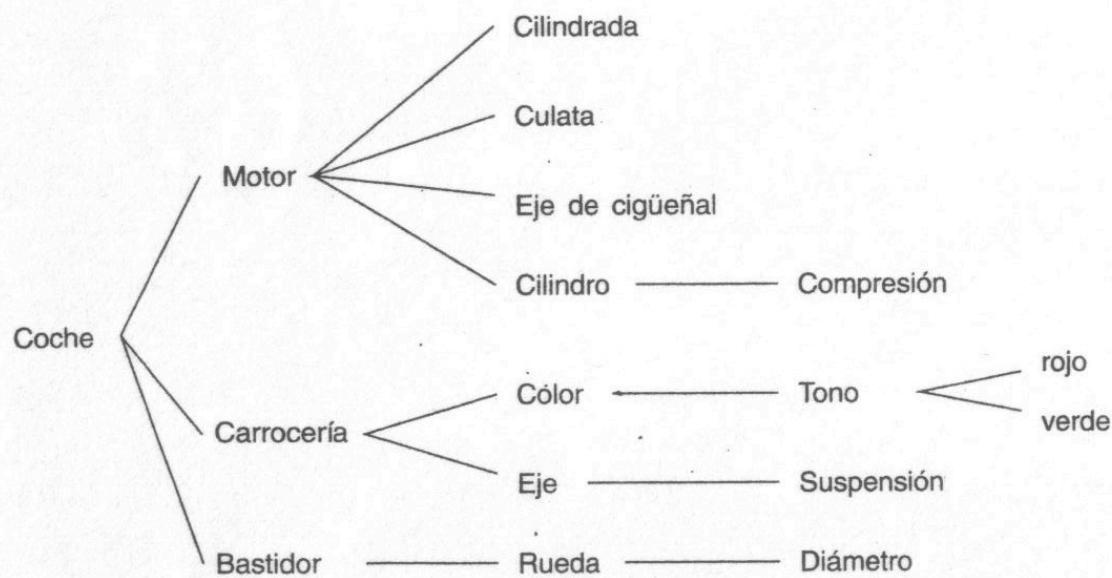


Figura 4.1 Frames, atributos y clases de valores de una base de conocimientos para coches

La estructura de la base de conocimientos estática significa el proyecto de un sistema jerárquico de *Frames* con *atributos* y los *valores*, que pueden tomar estos atributos. Las reglas o conjuntos de reglas pertenecientes a esta base con el conocimiento procedural se subordinan a Frames en particular y se incorporan a la estructura general.

Dicho de forma más precisa, los objetos definidos en la base de conocimientos estática son *Clases* de Frames o de atributos. Es decir, sólo se definen en general aquellas cualidades de los objetos, que poseen ejemplares especiales de éstos. Estos ejemplares reales, llamados también *instancias* se crean dinámicamente durante las *consultas* a través de la *base de conocimientos*. Empezando por la raíz de la jerarquía puede construirse una red de instancias de Frames y atributos, para la que la estructura estática crea una muestra. Normalmente, el árbol de instancias dinámico representa sólo una parte de la estructura estática de clases, ya que sólo se *instancian* los objetos relevantes para la consulta. El desarrollo de la consulta se controla mediante el *disparo* de reglas, que se ejecutan según los distintos mecanismos, en la raíz de la jerarquía, habiendo empezado en las *reglas de inicialización y ejecución* del Frame. Según haga falta, se instancian nuevos Frames

o atributos, de forma que el árbol de la base de datos dinámica crezca de forma continua durante una consulta.

Como otra posibilidad de estructuración de una base de conocimientos estática pueden *especializarse* clases de Frames, de forma que se den relaciones de *clases superiores e inferiores*. De esta forma podrían definirse subclases de *vehículos*, como *turismos* y *furgonetas*. En este caso, la subclase puede *heredar* determinadas cualidades, como reglas o atributos de la clase superior, con lo que nos ahorraremos de definir estas cualidades para cada subclase. En ExTool se da la posibilidad adicional de poder *especializar* también las instancias de la base de datos dinámica de una consulta. Si tenemos, por ejemplo, una base de datos para identificación de animales, y se define en ésta el Frame *Animal* con las subclases *mamífero* y *pájaro*, puede procederse de la forma siguiente: En la consulta se crea primero una instancia de la clase general *Animal*. Se relacionan entonces las reglas para distinguir los mamíferos de los pájaros. Si se dispara una regla que para el atributo *Superficie corporal* pide el valor *con pelo*, en la acción de esta regla se clasificará (especializará) la instancia correspondiente como perteneciente a la clase *mamífero*. En el desarrollo posterior se ejecutan las reglas subordinadas a esta clase de Frame para la determinación de mamíferos (con, dado el caso, más especialidades).

Tras esta visión general observaremos los objetos mencionados con más detalle.

En la definición de un nuevo Frame hay que darle un nombre y fijar el tipo: Por un lado hay que distinguir si puede ser instanciado sólo una vez durante una consulta (*Singleton-Frame*) o si pueden producirse varias instancias según se necesiten (*Multiple-Frame*). En el último caso, las instancias pueden obtener un *Registro de tiempo* adicional, cuando la clasificación temporal tenga importancia (*Timed-Frame*).

Pensemos en el marco de un Sistema Experto médico en un paciente con una enfermedad infecciosa. En el paciente se analizan varios cultivos bacteriales. En cada cultivo se descubren varias *bacterias*. El paciente mismo está representado en un *Singleton-Frame*, los cultivos en *Timed-Frames* y las bacterias de los cultivos en simples *Multiple-Frames* (véase figura 4.2).

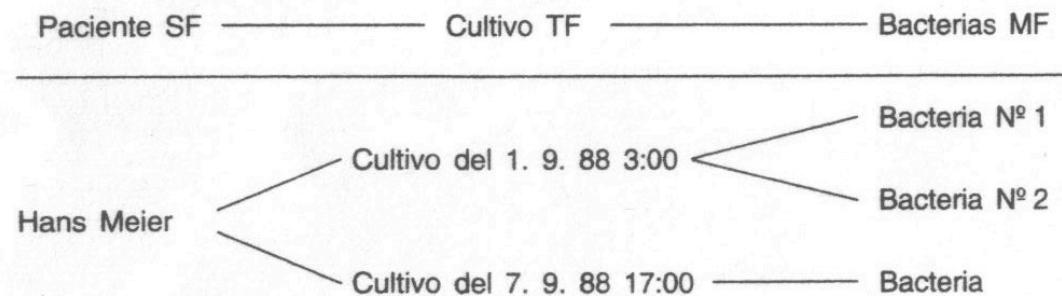


Figura 4.2 Los Frames de un Sistema Experto médico

La figura 4.2 muestra las clases de Frames de la base de conocimientos estática y abajo una posible estructura de instancias de estas clases (base de conocimientos dinámica) para una consulta.

4.2.2.4 Reglas

La definición de reglas se realiza siempre en el contexto de los distintos Frames. La relación que puede tener una regla o conjunto de reglas con “su” Frame, es:

- La regla es una *regla de inicialización* del Frame. Tras la creación de una nueva instancia de esta clase de Frame, se inicializa éste, indicando las *reglas de inicialización* entre otras en la secuencia previamente dada.
- La regla pertenece a las *reglas de ejecución* del Frame. Estas reglas se ejecutan en la secuencia dada, cada vez que se ejecuta el Frame. Una instancia de Frame puede ejecutarse un número variado de veces, el estímulo para ello se produce por las acciones de reglas.
- La regla puede deducir valores para los atributos subordinados al *Frame* y ejecutarlo cuando uno de estos atributos se convierte en objetivo de determinación. Estas reglas reciben el nombre de *Reglas de retroceso*.
- La regla utiliza uno o más atributos del Frame en su premisa y se ejecuta cuando se determina uno de estos atributos. Estas reglas reciben el nombre de *Reglas de avance*.
- Otra forma de clasificación se refiere sólo a conjuntos de reglas. Estas pueden presentarse como métodos del Frame que son llamados como subprogramas. La transmisión de parámetros es aquí opcional.

Las reglas no tienen un *nombre*, sino sólo un *Label*, que al contrario que el nombre no tiene que ser único. Cuando un Label no se define de forma explícita, es generado por el sistema.

A cada Frame se pueden subordinar los ya mencionados *parámetros iniciales* u *opcionales*. Estos son nombres de atributos y subframes, que tras la instanciación o ejecución del Frame son leídos o instanciados. Mientras que los parámetros iniciales deben ser siempre introducidos, el usuario puede elegir los parámetros opcionales que desee introducir, a través de un menú. Dado el caso, puede definirse una jerarquía de menús correspondiente a la jerarquía de Frames, con cuya ayuda el usuario puede elegir los atributos que desee para la introducción de valores existentes en toda la base de conocimientos.

Bibliografía

Giarratano, Joseph and Gary Riley. *Expert Systems : principles and programming.* 3^a. ed.
E.U.A. Thomson International Editors. 1998.

Nebendahal, Dieter, editor. *Sistemas Expertos. Experiencia de la Práctica.* Editorial Marcombo
S.A. Alemania. 1991.

Nebendahal, Dieter, editor. *Sistemas Expertos. Introducción a la técnica y aplicación.*
Editorial Marcombo S.A. Alemania. 1988.