



03062
7
201

Universidad Nacional Autónoma de México

**Sistemas Expertos facultativamente asociados
en red cooperativa
con arquitecturas de pizarrón**

**Una aplicación en la consulta
e interconsulta médica**

Tesis para obtener el grado de
Maestro en Investigación Biomédica Básica

presenta : Pedro Pablo González Pérez

Director : Dr. José Negrete Martínez

México, 1995

FALLA DE IMPRESIÓN

1995



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres
*por el constante y generoso apoyo, cariño y comprensión
que me han brindado*

Quiero expresar mi agradecimiento al Dr. José Negrete Martínez, quien ha sido la máxima inspiración y guía de este proyecto, y por todo lo que ha significado en mi vida profesional el haber trabajado bajo su dirección.

Agradezco a la Dra. Patricia Ostrosky Wegman y al Dr. Francisco Lara Ochoa, tutores de este proyecto de investigación, por sus valiosas contribuciones para el desarrollo del mismo.

Mis agradecimientos para el Dr. Arturo Cuellar Gaxiola, la Dra. Jessica de la Rosa Palacios, el Dr. Dante Aguilar Navarrete y el Dr. Ricardo Rivera Sánchez, quienes proporcionaron el conocimiento médico experto y los métodos de razonamiento médico contenidos en el sistema INTERMED.

Deseo agradecer a la Dra. Hanna Oktaba, al Dr. Enrique Ruiz-Velasco Sánchez, al M.C. Horacio Carvajal Sánchez y al M.C. Francisco Javier Noriega Romero, por sus comentarios, sugerencias y críticas acerca de este material.

Agradezco al Dr. Ariel Jesús Barreiro García, mi gran y estimado amigo, por todos estos años de colaboración y por sus valiosas ideas aportadas para el desarrollo de este proyecto.

Mis agradecimientos para todas aquellas personas que han contribuido con sus comentarios, comprensión, apoyo y camaradería a llevar adelante esta tarea.

Mi último agradecimiento para la tan amada Universidad Nacional Autónoma de México, y para México, mi segunda patria, por haberme brindado esta maravillosa oportunidad.

INTRODUCCIÓN

PRIMERA PARTE

INTRODUCCIÓN A LOS SISTEMAS EXPERTOS

	1
1. Sistemas Expertos	2
¿Qué son los Sistemas Expertos?	2
Principales características de los Sistemas Expertos	3
Representación del Conocimiento	4
La representación del conocimiento con reglas de producción	5
El Antecedente	5
El Consecuente	6
Métodos de inferencia	7
Razonamiento por encadenamiento de reglas "hacia adelante"	8
Razonamiento por encadenamiento de reglas "hacia atrás"	8
Razonamiento Oportunístico	9
Representación y procesamiento de la incertidumbre	9
Funciones de Combinación para factores de certidumbre	10
El factor de certidumbre	10
Cálculo de la certidumbre de proposiciones compuestas	13
Cálculo de la contribución de la regla	15
Cálculo del efecto global de varias reglas	17
Componentes de un Sistema Experto	20
La base de conocimientos	21
El mecanismo de inferencia	22
La memoria de trabajo	22
La interfaz de entrada/salida	23
El módulo de explicaciones	23
El módulo de aprendizaje	23
2. Tipología de Sistemas Expertos	24
Sistemas Expertos para la interpretación	24
Sistemas Expertos para la predicción	25

Sistemas Expertos para el diagnóstico	25
Sistemas Expertos para el diseño	26
Sistemas Expertos para la planeación	26
Sistemas Expertos para el monitoreo	27
Sistemas Expertos para la depuración	27
Sistemas Expertos para la reparación	28
Sistemas Expertos para la instrucción	28
Sistemas Expertos para el control	28
3. Principales Áreas de aplicación de los Sistemas Expertos	30
Administración de información	30
Agricultura	30
Ciencias militares	31
Control de procesos	31
Derecho	31
Electrónica	32
Física	32
Geología	32
Ingeniería	32
Matemáticas	33
Meteorología	33
Química/Bioquímica	33
Sistemas de Computación	33
"ZOOM" sobre un dominio de aplicación	34

SEGUNDA PARTE

SISTEMAS PIZARRÓN

4. El modelo y la estructura de pizarrón para la solución de problemas	48
El modelo de pizarrón	48
La estructura de pizarrón	50
Las fuentes de conocimiento	51
El pizarrón	52
El sistema de control	52
5. La arquitectura de pizarrón	54
La arquitectura de pizarrón del sistema HEARSAY II	54
El dominio de HEARSAY II	54
El modelo HEARSAY II	55
La arquitectura de HEARSAY II	56
El pizarrón	57
Las fuentes de conocimiento	58
El sistema de control	59

La arquitectura de pizarrón del sistema DVMT	60
Un medio ambiente distribuido para el monitoreo de vehiculos	60
La arquitectura de los nodos en DVMT	61
El pizarrón de datos	62
Las fuentes de conocimiento	62
El pizarrón de metas	65
El monitor del pizarrón	65
El planeador	65
La cola de despacho	65
El despachador	65
El pizarrón de consistencias	66
Los resolvedores	66
Las áreas de interés	66
La actividad de los nodos en la solución de problemas	67
La arquitectura de pizarrón para el control	67
El control	67
Componentes de la arquitectura	68
El pizarrón del dominio	68
El pizarrón del control	69
Las fuentes de conocimiento del dominio	71
Las fuentes de conocimiento del control	71
El mecanismo de despacho	71
Otras aplicaciones pizarrón	72
AGE	73
BBI	73
BLOBS	75
CAGE	75
GBB	77
GUARDIAN	78
HASP/SIAP	79
MUSE	79
MXA	81
POLIGON	82
PROTEAN	83
TRICERO	83

TERCERA PARTE

INTERMED : UNA RED DE SISTEMAS EXPERTOS PARA LA CONSULTA E INTERCONSULTA MEDICA

85

6. REDSIEX : Una Red Cooperativa de Sistemas Pizarrón	86
REDSIEX y la solución cooperativa de problemas distribuidos	86
Distribución de la información, procesamiento y control en REDSIEX	88
La arquitectura de los nodos en REDSIEX	89
Los componentes-base de datos del nodo	89

El pizarrón del dominio	90
La base de datos de control	92
La lista de registros de activación de las fuentes de conocimientos	93
El buzón de mensajes de entrada	93
El árbol de búsqueda estructurado	95
Los componentes-programa del nodo y su funcionamiento	96
Las fuentes de conocimiento de razonamiento	96
Las fuentes de conocimiento de comunicación	98
La fuente de conocimiento de procesamiento "backward"	99
La fuente de conocimiento "Interfaz"	101
La fuente de conocimiento "Parada"	102
El planeador local	103
El mecanismo de control	107
Representación y procesamiento de la incertidumbre en REDSIEX	112
La revisión de creencias	114
7. INTERMED : Una Red de Sistemas Expertos para la Consulta e Interconsulta Médica	116
La cooperación en el proceso de diagnóstico médico : la interconsulta médica	116
El diagnóstico médico	116
Diagnóstico diferencial	116
La interconsulta médica	117
Algunos conceptos y términos médicos necesarios	118
Signo	118
Síntoma	118
Síndrome	119
Entidad nosológica	119
El dominio de aplicación de INTERMED	119
La distribución del conocimiento y razonamiento en INTERMED	120
La arquitectura y el funcionamiento de INTERMED	120
El pizarrón del dominio	122
Las fuentes de conocimiento	125
Las fuentes de conocimiento dependientes del dominio	125
El papel de las fuentes de conocimiento independientes del dominio	133
CONSIDERACIONES FINALES	140
Los Sistemas Pizarrón vistos como una extensión de los Sistemas Expertos basados en reglas	140
Acercas de la arquitectura de pizarrón propuesta	141
Sobre el modelo de consulta e interconsulta médica implementado	144
Diagnóstico médico natural vs diagnóstico médico en INTERMED	145
El proceso diagnóstico que efectúa el médico durante la consulta médica	145
El proceso diagnóstico que efectúa el sistema INTERMED	147

Un comentario final acerca de la implementación y uso de
REDSIEX 149

REFERENCIAS 151

ANEXOS 158

Introducción

Los Sistemas Expertos han llegado a constituir hoy en día una de las áreas más exitosas y útiles de la Inteligencia Artificial. Desde el surgimiento de los Sistemas Expertos, muchas han sido las aplicaciones desarrolladas en el campo de la medicina. En especial, aquellas que brindan ayuda al médico en el diagnóstico y tratamiento de enfermedades.

El diagnóstico médico, como un proceso de solución de problemas, puede ser abordado como un proceso incremental de generación de elementos solución a través de varios niveles de abstracción. Partiendo desde un nivel inicial, el cual contiene información primaria correspondiente a signos y síntomas, recorriendo otros niveles intermedios en los que se arriban a determinados síndromes o se infieren particulares entidades etiológicas y fisiopatológicas, hasta llegar a un nivel superior en el que se concluye cuál es la entidad nosológica presente y cuál debe ser el tratamiento adecuado.

En este sentido incremental, la arquitectura de pizarrón [ENG88a, NI89a] proporciona un escenario ideal para modelar, de una forma coherente, el conocimiento y la experiencia médica relevantes en la solución de problemas de diagnóstico.

El objetivo de esta tesis es la construcción de un sistema experto, que basado en la arquitectura de pizarrón permita la consulta e interconsulta médica en un medio ambiente distribuido.

Como se verá en los comentarios finales de la tesis, no se trata de un simple desarrollo de lo ya existente, se trata de una investigación que toca problemas fundamentales para la Inteligencia Artificial, tales como la surgencia de pericia cooperativa entre agentes autónomos.

El material correspondiente a la tesis ha sido dividido en tres partes, las cuales abarcan aspectos de interés relacionados con los Sistemas Expertos, las Arquitecturas de Pizarrón y la aplicación desarrollada: Una Red de Sistemas Expertos para la Consulta e Interconsulta Médica.

La primera parte de este material ha sido dedicada a un recorrido introductorio al área de los Sistemas Expertos. Está dividida en tres capítulos. En el capítulo 1 se discuten las principales características estructurales y funcionales de los Sistemas Expertos, haciendo un particular énfasis sobre los Sistemas Expertos Basados en Reglas. En este

capítulo también se presentan los métodos de razonamiento característicos de los Sistemas Expertos Basados en Reglas, así como una aproximación a la representación y procesamiento de la incertidumbre, cuando el conocimiento ha sido representado utilizando reglas de producción. El capítulo 2 expone una clasificación o tipología de los Sistemas Expertos, atendiendo al tipo de tarea genérica que son capaces de ejecutar. El capítulo 3 está dedicado a examinar los principales dominios de interés que han servido de escenario para el desarrollo y aplicación de los Sistemas Expertos. El campo de la medicina constituye el foco de atención de este capítulo. En este sentido, se describe una gran variedad de aplicaciones en el diagnóstico y tratamiento de enfermedades, las cuales ilustran el rico quehacer de los Sistemas Expertos en Medicina.

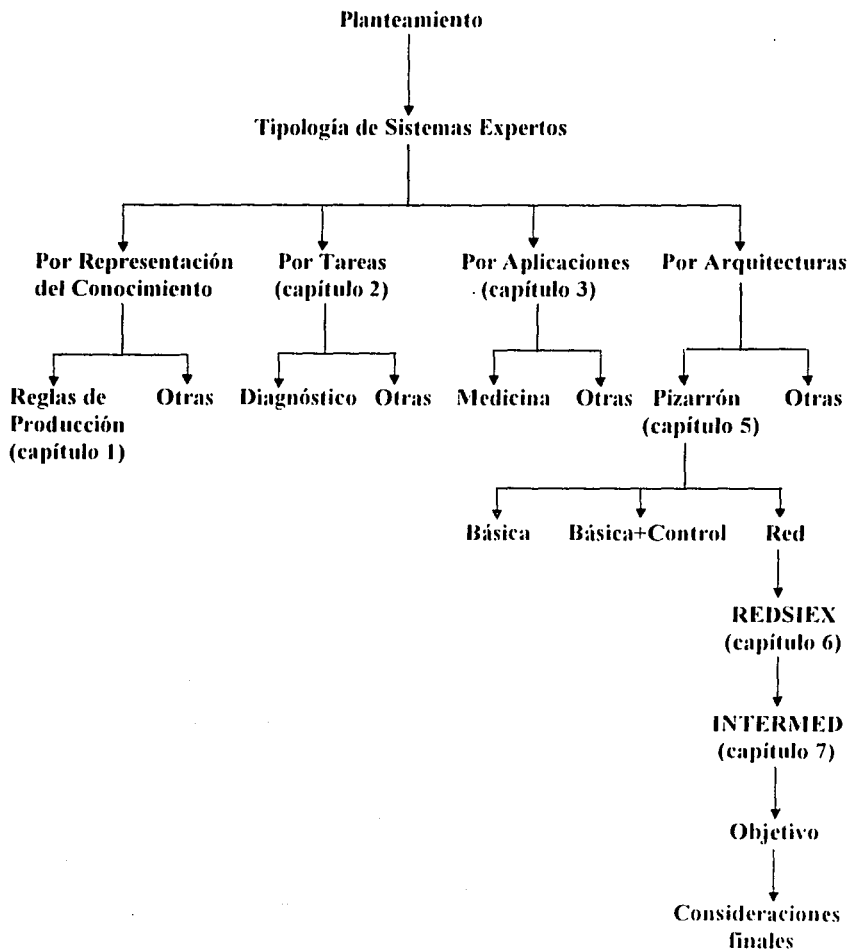
La segunda parte cubre los tópicos más importantes de los Sistemas Pizarrón. Esta parte contiene dos capítulos. El capítulo 4 discute aspectos relacionados al modelo y estructura pizarrón. El capítulo 5 presenta un tratamiento comprensivo de la arquitectura de pizarrón. En este capítulo se discuten tres Sistemas Pizarrón que representan diferentes y ricos enfoques a esta arquitectura : la arquitectura de pizarrón centralizado en HEARSAY II [ENG88b, ERM88], la arquitectura de pizarrón distribuido en DVMT [LES88b] y la arquitectura de pizarrón para el control de BB1 [HAY85]. También se describe brevemente una gran variedad de aplicaciones específicas, generalizaciones, extensiones y refinamientos desarrollados a partir de la arquitectura de pizarrón tradicional del sistema HEARSAY II.

La tercera parte constituye la parte experimental de la tesis. Aquí se describe un prototipo de sistema experto distribuido con arquitectura de pizarrón, para la solución cooperativa de problemas en medicina. Esta parte consiste de dos capítulos. En el capítulo 6 se presenta el sistema prototipo REDSIEX, una Red de Sistemas Expertos con arquitecturas de pizarrón, para el desarrollo de aplicaciones distribuidas en la toma de decisiones médicas. En este capítulo se describen exhaustivamente las características de la arquitectura y funcionamiento de REDSIEX, incluyendo aspectos relacionados con las estrategias de cooperación, tratamiento de la incertidumbre y revisión de creencias en este sistema. El capítulo 7 discute los detalles de INTERMED, una Red de Sistemas Expertos para la Consulta e Interconsulta Médica desarrollada sobre REDSIEX.

Ha sido nuestra intención extendernos un poco en el contenido correspondiente a las dos primeras partes de este material. El primer motivo que justifica este hecho, es que hemos intentado proporcionar al lector todos aquellos aspectos relacionados con los Sistemas Expertos basados en reglas y los Sistemas Pizarrón, los cuales resultarán de gran utilidad para una mejor comprensión de las aplicaciones discutidas en la tercera parte de este material. El segundo motivo constituye la respuesta a la siguiente interrogante : ¿para quienes ha sido escrito este material? Pensamos que no sólo para personas relacionadas con el campo de la Inteligencia Artificial u otras disciplinas de las ciencias de la computación, sino también para personas del campo de la medicina y del área de las investigaciones biomédicas. En este sentido, el material ha sido enriquecido con muchos ejemplos enmarcados en el contexto del diagnóstico médico y con una

amplia gama de aplicaciones de Inteligencia Artificial desarrolladas en el dominio de la medicina. Por último, uno de nuestros objetivos ha sido concluir un material dedicado a los Sistemas Expertos basados en reglas y a los Sistemas Pizarrón, el cual pueda servir como medio de estudio o consulta al interesado en estas temáticas.

La capitulación de la tesis obedece a la serie de decisiones tomadas que se resumen en el siguiente mapa :





PRIMERA PARTE

**INTRODUCCIÓN
A LOS
SISTEMAS EXPERTOS**



Sistemas Expertos

La Inteligencia Artificial (IA) es un campo relativamente nuevo de las ciencias de la computación, concierne con el diseño de sistemas computarizados que exhiben alguna forma de inteligencia. Esto es, sistemas que poseen algunas de las características que nosotros asociamos con inteligencia en el comportamiento humano. [BAR89a, BAR89b, NEG90, NEG91b, PAT90, RIC91]

Al igual que otras disciplinas científicas, la IA contiene varias áreas de investigación. Estas incluyen : solución de problemas, razonamiento lógico, comprensión del lenguaje, programación automática, aprendizaje, sistemas expertos, robótica y visión

Recientemente el área de experticia o Sistemas Expertos ha resultado ser una de las más exitosas y útiles de la IA. Una de las principales razones que ha garantizado este éxito es el hecho de que de que el área de Sistemas Expertos no es un área restringida y cerrada de la IA como muchos han pensado. Muy lejos de esto, los Sistemas Expertos extienden y elaboran los principios y procedimientos básicos que caracterizan a la gran mayoría de las restantes áreas de la IA.

¿Qué son los Sistemas Expertos?

Los Sistemas Expertos fueron desarrollados por primera vez durante las décadas del 60 y 70, y en muy pocos años lograron traspasar la frontera de los laboratorios y la experimentación para demostrar su efectividad en un gran número de dominios de problemas.

Cualquier dominio de aplicación que requiera de la experticia humana para la solución de problemas se convierte de hecho en un escenario probable para la aplicación exitosa de los Sistemas Expertos. Entre estos dominios de aplicación se incluyen la medicina, la biología, la química, la geología, la geofísica, la meteorología, la ingeniería, las operaciones bancarias y financieras, la aeronáutica, las operaciones militares y otros muchos más.

Diferentes definiciones de Sistemas Expertos han sido propuestas hasta la fecha en la literatura de IA [BAR89a, BAR89c, BRA83, BUC89, FOR89, JAC90, MER89, PAT90, RIC91]. Sin embargo, a pesar de los diferentes enfoques y caracterizaciones utilizadas en éstas, una base o fundamento común puede ser extraído de todas ellas : un Sistema Experto es un sistema con experticia en la solución de problemas. Esto es, un sistema que posee conocimientos acerca de un dominio particular, comprensión de problemas del dominio y métodos de inferencia o razonamiento para manipular este conocimiento y resolver los problemas de la misma forma en que lo haría un experto humano.

Principales características de los Sistemas Expertos

Los Sistemas Expertos difieren de los sistemas de computación convencionales en varios aspectos importantes [BAR89c, HAY83] :

- ◆ Para llevar a cabo los procesos de solución de problemas, los Sistemas Expertos utilizan conocimientos específicos del dominio y métodos heurísticos (plausibles) más que tablas y algoritmos.
- ◆ Los Sistemas Expertos utilizan representaciones simbólicas del conocimiento (reglas de inferencia, redes semánticas, marcos, guiones, objetos) y ejecutan los procesos de inferencia a través de computaciones simbólicas (excepto los sistemas expertos conexionistas o basados en redes neurales).
- ◆ Los Sistemas Expertos poseen métodos para explicar cómo una solución a un problema particular fue alcanzada y por qué determinada información ha sido requerida durante el proceso de solución.
- ◆ Durante los procesos de razonamiento los Sistemas Expertos pueden utilizar conocimiento acerca de ellos mismos (metaconocimiento). Esto es, conocimiento sobre sus capacidades y sus limitaciones para enfrentar la solución de un problema en particular.

- ◆ Con excepción, los Sistemas Expertos aprenden de su propia experiencia acumulada sobre la solución de problemas en el pasado.

Representación del Conocimiento

Lo que confiere a los sistemas inteligentes una alta capacidad y elevada ejecución en la solución de problemas del mundo real es la cantidad y calidad del conocimiento contenido en sus bases de conocimientos, más que los métodos de razonamiento que éstos utilizan para llevar a cabo los procesos de inferencia. Si deseamos construir un verdadero sistema inteligente, capaz de resolver exitosamente problemas de la vida real, entonces debemos proporcionar a éste una gran cantidad de conocimientos y de la mejor calidad.

A pesar de que no existe la definición única de conocimiento, en términos generales podríamos decir que el conocimiento consiste de hechos, conceptos, procedimientos, ideas, abstracciones, reglas y asociaciones (en unión a una gran habilidad que permite utilizar estas nociones en la modelación de diferentes aspectos del mundo real). El conocimiento puede ser representado mediante escenas o imágenes mentales, mediante palabras escritas o habladas en algún lenguaje, mediante imágenes gráficas o pictóricas y otras formas de expresión encontradas por el hombre.

Desde el punto de vista de la IA, podríamos interpretar el conocimiento como una combinación de esquemas o estructuras de datos y de procedimientos interpretativos, los cuales al ser utilizados correctamente por un programa le confieren a éste un comportamiento inteligente. Debido a que el conocimiento es importante y de carácter esencial para el comportamiento inteligente, la Representación del Conocimiento ha llegado a convertirse en una de las investigaciones prioritarias en la IA. La representación del conocimiento propone diferentes clases de esquemas y estructuras para la organización y almacenamiento del conocimiento en un sistema inteligente artificial. La elección de un esquema particular dependerá del tipo de problema a ser resuelto y de los métodos de inferencia a utilizar.

Las principales representaciones del conocimiento incluyen: los predicados de primer orden, las reglas de producción, los marcos (*frames*), los guiones (*scripts*), las redes asociativas (también conocidas como redes semánticas), los objetos, y los procedimientos [BUC'85, JAC'90, PAT'90, RIC'91]

De todas estas representaciones, las reglas de producción han sido las más comúnmente utilizadas por los Sistemas Expertos y otros sistemas basados en el conocimiento. Los

sistemas que utilizan este esquema para representar su conocimiento son conocidos como sistemas basados en reglas.

La representación del conocimiento con reglas de producción

Las reglas de producción pueden ser consideradas como un conjunto de cláusulas de la lógica de predicados. Estas son probablemente la forma más popular de representación del conocimiento utilizada en los Sistemas Expertos. Esto se debe a la cómoda estructura que poseen y a la forma tan natural con que pueden ser expresados los conocimientos. Una regla de producción es una estructura de la forma :

IF <antecedente> THEN <consecuente>

donde la parte izquierda es conocida como antecedente y se refiere a las condiciones o premisas correspondientes a una situación dada, mientras que la parte derecha conocida como consecuente contiene la conclusión, acción o consecuencia dada en caso de que las premisas o condiciones hayan sido satisfechas.

El Antecedente

En una regla de producción el antecedente está compuesto por una o varias proposiciones, las cuales se combinan mediante conectivos lógicos para formar una condición o premisa más compleja.

Los conectivos lógicos utilizados son los clásicos de la lógica del cálculo proposicional :

- ◆ Conjunción (AND)
- ◆ Disyunción inclusiva (OR)
- ◆ Disyunción exclusiva (XOR)
- ◆ Negación (NOT)

La conjunción (AND) representa la condición del cumplimiento conjunto de dos proposiciones. Es decir, una conjunción se cumple sólo cuando las dos proposiciones componentes son verdaderas.

La disyunción inclusiva (OR) representa el caso de una alternativa entre dos proposiciones. Para que una disyunción inclusiva se cumpla basta que sea verdadera al menos una de las dos proposiciones componentes.

La disyunción exclusiva (XOR) también representa el caso de una alternativa entre dos proposiciones, pero a diferencia de la disyunción inclusiva, ésta se cumple cuando una y sólo una de las proposiciones componentes es verdadera.

La negación (NOT) significa la satisfacción o cumplimiento de lo contrario que haya sido expresado en la proposición primaria. Esto es, si la proposición primaria es verdadera entonces la negación es falsa y viceversa. La negación puede ser aplicada tanto a proposiciones simples como a proposiciones compuestas.

Ejemplo

Para ejemplificar el uso de los conectivos lógicos en los antecedentes de las reglas de producción, consideremos las siguientes proposiciones simples :

- S1 : El paciente tiene palidez cutáneo mucosa
- S2 : El paciente tiene apetito
- S3 : El paciente tiene somnolencia
- S4 : El paciente tiene fatiga
- S5 : El paciente tiene disnea
- E1 : El paciente tiene sangramiento digestivo mantenido
- E2 : El paciente presenta absorción defectuosa del hierro

Entonces las siguientes proposiciones compuestas o antecedentes poseen un claro significado:

- ANT1 = S1 AND S3 AND S4 AND S5
- ANT2 = S1 AND NOT(S2) AND S3 AND S4 AND S5
- ANT3 = E1 OR E2

El consecuente

De forma general, el consecuente en una regla de producción está compuesto por una proposición simple. Si se quiere representar la incertidumbre en la regla de inferencia, entonces a la proposición conclusión se le puede asociar un factor, el cual expresa una medida de la confianza o seguridad del Sistema Experto acerca de la conclusión alcanzada cuando el antecedente ha sido satisfecho

Ejemplo

Para ejemplificar el papel del consecuente dentro de una regla de producción, consideremos las siguientes proposiciones :

D1 = El paciente tiene anemia

D2 = El paciente tiene anemia por déficit de hierro

entonces las siguientes reglas de producción expresan un claro conocimiento del dominio en cuestión :

```
REGLA 01 : IF S1 AND
            NOT(S2) AND
            S3 AND
            S4 AND
            S5
            THEN D1
```

```
REGLA 02 : IF S1 AND
            S3 AND
            S4 AND
            S5 AND
            E2
            THEN D2
```

Métodos de Inferencia

Como ya es conocido, toda aplicación en IA requiere de conocimientos y métodos de inferencia para manipular este conocimiento en la solución de problemas del dominio. Los esquemas de representación del conocimiento permiten dotar a los sistemas expertos de conocimientos, mientras que los métodos de inferencia permiten a éstos manipular de forma apropiada y eficiente este conocimiento para alcanzar las metas propuestas.

En los sistemas expertos basados en reglas resulta muy común el uso de métodos o paradigmas de inferencia basados en el encadenamiento de reglas. Entre estos métodos, los más usados han sido el razonamiento por encadenamiento de reglas hacia adelante (*forward*), el razonamiento por encadenamiento hacia atrás (*backward*) y el razonamiento oportunístico o combinado [BAR89c, PAT90, RIC91].

Razonamiento por encadenamiento de reglas "hacia adelante"

El razonamiento por encadenamiento de reglas hacia adelante es un proceso de búsqueda direccionado por los datos de entrada y requiere que todos los datos relevantes estén disponibles en el momento de iniciar las inferencias. En este tipo de razonamiento se parte de las evidencias primarias y posteriormente se usa el conocimiento contenido en las reglas para conocer hacia qué conclusión esas evidencias apuntan de forma más plausible.

Un sistema con encadenamiento hacia adelante trabaja de la siguiente forma : partiendo de una colección inicial de hechos (datos de entrada), el sistema intenta hallar todas las reglas cuyos antecedentes aparean con configuraciones formadas a partir de estos hechos, las conclusiones de las reglas apareadas son utilizadas para generar nuevos hechos en la colección. Para esta nueva colección el proceso se repite, se hallan todas las reglas cuyas condiciones aparean con configuraciones de hechos formadas y se utilizan sus conclusiones para generar nuevos hechos. El proceso continúa hasta que una primera meta plausible es alcanzada o hasta que ya no pueden ser generadas nuevas metas.

Razonamiento por encadenamiento de reglas "hacia atrás"

El razonamiento por encadenamiento de reglas hacia atrás es un proceso de búsqueda direccionado por las metas y no requiere que toda la información relevante esté disponible en el momento de iniciar las inferencias, ya que los datos que no hayan sido introducidos inicialmente (o inferidos por el Sistema Experto) serán preguntados al usuario en el momento en que sean necesarios. En este tipo de razonamiento son las metas o hipótesis las que dirigen el proceso de razonamiento. Una vez fijada la meta, es el conocimiento encerrado en las reglas el que indica qué hechos o evidencias deben buscarse. La búsqueda de las evidencias cesa cuando se ha probado la hipótesis o no existan más hechos que investigar.

Un sistema con encadenamiento hacia atrás trabaja de la siguiente forma : partiendo de la meta que se desea alcanzar, el sistema intenta hallar todas las reglas cuyas conclusiones aparean con esta meta. Algunos de los hechos que forman los antecedentes de las reglas apareadas pudieran ser conocidos, debido a que fueron introducidos como datos iniciales mientras que otros pudieran conocerse una vez que sean preguntados al usuario. Sin embargo, otros hechos pudieran no ser estados iniciales y aparean con la conclusión de alguna otra regla, constituyendo una nueva submeta y por lo tanto, obligando que continúe el proceso de encadenamiento hacia

atrás. El proceso finaliza cuando una primera meta es alcanzada como verdadera o después que todas las metas relevantes han sido exploradas.

Razonamiento oportunístico

En ocasiones ciertos aspectos de un problema son mejor tratados por una vía de encadenamiento hacia adelante y otros aspectos por una vía de encadenamiento hacia atrás. Cuando este es el caso, el razonamiento oportunístico resulta muy útil.

El razonamiento oportunístico es una combinación de los razonamientos por encadenamiento de reglas hacia adelante (direccionado por los datos) y hacia atrás (direccionado por las metas). En este tipo de razonamiento el procesamiento general es hacia atrás, pero una vez obtenidas las evidencias éstas son utilizadas en un procesamiento hacia adelante. Esta combinación permite unir las ventajas de ambos tipos de razonamiento.

Un tipo de razonamiento oportunístico resulta muy beneficioso cuando son muchas las inferencias posibles que el sistema puede realizar y cuando es muy probable que más de una vía de razonamiento pueda establecerse. Éste habilita a un sistema para realizar nuevas inferencias cuando nuevos datos de entrada han sido proporcionados y para que también efectúe preguntas acerca de datos relevantes cuando nuevas metas han sido direccionadas.

Representación y procesamiento de la incertidumbre

Existen numerosas aproximaciones al problema de la representación y tratamiento de la incertidumbre en Sistemas Expertos. De forma muy general, estas aproximaciones pueden ser agrupadas en dos grandes clases: la clase de las aproximaciones basadas en métodos simbólicos (tendencia cualitativa) y la clase de las aproximaciones basadas en métodos numéricos (tendencia cuantitativa).

En la clase de las aproximaciones basadas en métodos simbólicos, los mecanismos de inferencia y/o representaciones son ampliados para hacer posible el razonamiento con información incompleta. La incertidumbre es representada cualitativamente como un determinado criterio (verdadero, falso, no verdadero, no falso) y el razonamiento es llevado a cabo a partir de un conjunto de criterios o avales asociados a los conocimientos y a los hechos [KON87, RICH91, SHOS7]

En la clase de las aproximaciones basadas en métodos numéricos, la representación del conocimiento es ampliada para permitir la representación de la incertidumbre mediante una cierta característica numérica, la cual es asociada con cada hecho y regla. Esta medida numérica de la incertidumbre es utilizada en diferentes operaciones, a partir de las cuales se lleva a cabo el proceso de inferencias de nuevos hechos, en base a las evidencias acumuladas y a los conocimientos contenidos en la base.

Son muchos los esquemas numéricos existentes para el problema de la incertidumbre en Sistemas Expertos, encontrándose entre los más conocidos : la Teoría de Bayes [CHE85], las Funciones de Combinación para Factores de Certidumbre [BAR85, BUC84, GOR85, HAJ85, SHO75, SHO76, SHO85], la Teoría de Dempster-Shafer [DEM68, SHA76] y la Lógica Fuzzy [ZAD79]. Aquí solo nos dedicaremos a examinar el esquema de las funciones de combinación para factores de certidumbre, por ser esta aproximación uno de los esquemas más comúnmente utilizados para la representación y tratamiento de la incertidumbre en Sistemas Expertos basados en reglas.

Funciones de combinación para factores de certidumbre

Esta aproximación a la representación y tratamiento de la incertidumbre fue inicialmente experimentada por el Sistema Experto MYCIN [BUC85, SHO76, SHO85] y utilizada posteriormente por una gran variedad de Sistemas Expertos basados en reglas [BUC85, PAT90, RIC91]. MYCIN fue el primer Sistema Experto creado en medicina. Quizá la aportación principal de MYCIN halla sido la representación del conocimiento médico mediante factores de certidumbre

Cuando esta aproximación es utilizada, la incertidumbre es entonces representada mediante un valor numérico conocido como factor o grado de certidumbre. A cada regla de la base de conocimientos se le asocia un factor de certidumbre, el cual es una medida de la magnitud en la cual las evidencias o hechos que forman el antecedente de la regla apoyan la hipótesis o conclusión dada en el consecuente de la regla. De igual forma, la credibilidad o seguridad en cada hecho es expresada mediante su factor de certidumbre asociado. Por ejemplo, en el caso de MYCIN, la presencia de un signo o síntoma puede aumentar o disminuir la creencia de que el paciente tenga una enfermedad dada. Estos factores de certidumbre deben poseer ciertas propiedades dentro de la base de conocimientos que permitan, después, combinarlos al establecer un diagnóstico. Para combinar las estimaciones de certidumbre presentes en cada una de las reglas y producir una estimación final de la certidumbre de las conclusiones, son utilizadas las funciones de combinación. Procederemos en primer lugar, a dejar claro el significado de factor de certidumbre para después ver cuáles son estas funciones de combinación y cómo son utilizadas en el proceso de razonamiento bajo incertidumbre

El factor de certidumbre

El factor de certidumbre ($FC[h,e]$) se define a partir de la siguiente expresión :

$$FC[h,e] = MC[h,e] - MD[h,e]$$

donde :

$MC[h,e]$ es la medida de creencia de la hipótesis h dada la evidencia e . MC es un valor entero entre 0 y 1, éste mide la magnitud en la cual la evidencia e apoya la hipótesis h . Si MC es cero, entonces la evidencia e falla para apoyar la hipótesis h .

$MD[h,e]$ es la medida de no creencia (descreencia) en la hipótesis h dada la evidencia e . MD es un valor entre 0 y 1, éste mide la magnitud en la cual la evidencia e apoya la negación de la hipótesis h . Si MD es cero, entonces la evidencia e apoya la hipótesis h .

Una "pieza" de evidencia sólo puede apoyar o negar una hipótesis, pero no ambas cosas. Luego, si $MC = 0$ entonces $MD = 1$ y si $MD = 0$ entonces $MC = 1$. Teniendo en cuenta este hecho y considerando que el máximo valor que pueden tomar MC o MD es 1, queda claro que FC será una magnitud definida en el intervalo real $[-1, 1]$.

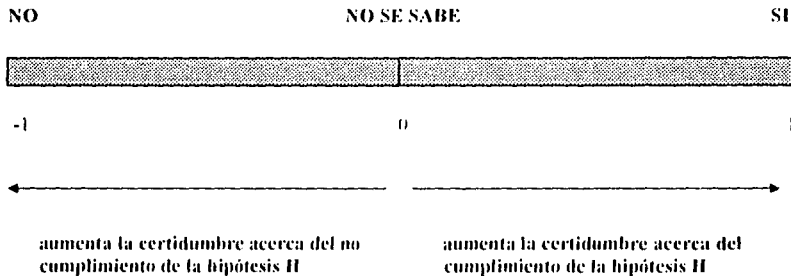


Figura 1.1 Significado de los valores de certidumbre sobre el intervalo real $[-1, 1]$

De esta forma, el valor 1 es equivalente al SI (certeza absoluta de que el hecho se cumple), el valor -1 al NO (certeza absoluta de que el hecho no se cumple) y el valor 0 al NO SE SABE. Puesto que FC puede tomar cualquier valor perteneciente al intervalo real $[-1, 1]$, la parte positiva del intervalo corresponde al caso en que se afirma la presencia de una evidencia o la validez de esta para apoyar determinada hipótesis y la

parte negativa al caso en que se niega. El grado o la fuerza con que se afirma o se niega algo, lo da el valor absoluto del FC . La figura 1.1 muestra una representación del intervalo real $[-1, 1]$.

Los FC 's de las reglas que forman la base de conocimientos son proporcionados por los expertos del dominio al escribir cada una de éstas, mientras que los FC 's asociados a los hechos o evidencias que forman el antecedente de las reglas, son aportados por el usuario, como datos iniciales si se trata de un razonamiento *forward* o como respuesta a las preguntas formuladas por el sistema cuando se trata de un razonamiento *backward*, o son obtenidos a partir de las conclusiones alcanzadas por otras reglas.

Ejemplo

1. "El paciente tiene dolor en epigastrio" (0.9)

Esto significa, en primer lugar, que se acepta o se cumple el hecho planteado, puesto que FC es mayor que cero. Además, dado que la magnitud de FC está muy cercana a 1 (que representa la certeza absoluta de que el hecho se cumple), se está indicando que hay una seguridad muy grande acerca del cumplimiento del hecho.

2. "El paciente presenta insuficiencia respiratoria" (0.6)

En este caso podemos decir que se está afirmando dicho planteamiento, pero con una seguridad menor que en el caso anterior.

3. "El paciente tiene dolor en fosa ilíaca derecha" (-1.0)

Esto significa que no se acepta o no se cumple el hecho planteado. Además, existe una seguridad absoluta acerca del no cumplimiento del hecho.

4. IF dolor en hipocondrio derecho AND
37.5 <= temperatura <= 40.0 AND
náuseas AND
vómito frecuente AND
bilirrubina > 1
THEN colecistitis aguda (0.95)

El significado de esta regla es el siguiente : si poseemos una certeza absoluta ($FC=1$) acerca del cumplimiento de las evidencias establecidas en el antecedente de la regla, entonces existe una seguridad muy grande (0.95) acerca del cumplimiento de la hipótesis colecistitis aguda. Es decir, las evidencias apoyan fuertemente la hipótesis.

El proceso de inferencias bajo incertidumbre requiere que estos FC 's sean combinados para reflejar la operación de todas estas piezas de evidencias y reglas aplicadas a la solución de un problema. Las combinaciones que se necesitan considerar responden a las siguientes situaciones :

1. El antecedente de una regla puede contener más de una proposición o hecho. Cuando este es el caso, es necesario determinar la certidumbre de una colección de hechos tomados conjuntamente (cálculo de la certidumbre de proposiciones compuestas).

2. Cuando el antecedente de una regla ha sido satisfecho con un valor de certidumbre diferente a 1 (certeza absoluta), entonces es necesario determinar el nuevo valor de certidumbre con el cual se cumple la conclusión de la regla (cálculo de la contribución de la regla).

3. Cuando más de una regla proporciona evidencias acerca de una misma hipótesis, entonces hay que calcular el FC de dicha hipótesis teniendo en cuenta todas estas piezas de evidencia (cálculo del efecto global de varias reglas).

Con el interés de reducir la cantidad de expresiones para el cálculo de las diferentes funciones de combinación, en adelante no haremos referencia explícita a las medidas de creencia (AC) y no creencia (AD) que componen el factor de certidumbre (FC), sólo nos referiremos a este último. Queda claro que si FC es negativo, entonces la evidencia niega la hipótesis, mientras que si FC es positivo, entonces la evidencia confirma la hipótesis.

Además, en muchas ocasiones nos referiremos al factor de certidumbre como el grado de certidumbre, el valor de certidumbre, el nivel de certidumbre o simplemente certidumbre. En todos los casos estaremos hablando de la certeza que poseemos acerca del cumplimiento de un hecho dado.

Cálculo de la certidumbre de proposiciones compuestas

Hasta aquí ha quedado claro que todos los hechos almacenados en la memoria de trabajo, así como las reglas que forman la base de conocimientos, poseen un factor de certidumbre asociado. Debido a que en lo común las reglas expresan relaciones entre proposiciones compuestas y proposiciones simples, más que entre proposiciones simples entre sí, es necesario conocer cómo determinar la certidumbre de una proposición compuesta a partir de las certidumbres de los hechos que la integran y los conectivos que vinculan a éstos.

Cálculo de la certidumbre de una negación

La forma más simple de una proposición compuesta es la negación de una proposición. Consideremos la siguiente proposición:

P : "El paciente tiene un síndrome inflamatorio esofágico" (0.8)

Luego, ¿cual será el FC asociado a la negación de P?

Sea **-P** : "El paciente no tiene un síndrome inflamatorio esofágico" la negación de P.

Es lógico pensar que el FC asociado a **-P** debe ser negativo, puesto que si creemos en alguna medida que P es cierta, entonces **-P** debe ser falsa. La medida en que **-P** es falsa es igual en valor absoluto a la medida en que P es verdadera, ya que si afirmamos una proposición con un determinado grado de certidumbre, entonces su negación debe ser postulada con ese mismo grado de certidumbre, pero con signo contrario.

De esta forma, si el grado de certidumbre de una proposición P es c, entonces la certidumbre de la negación de P es -c. Esto es :

$$NEG(c) = -c$$

Cálculo de la certidumbre de una conjunción

Consideremos las siguientes proposiciones :

P1 : "Paciente con dolor en epigastrio" (0.7)

P2 : "Paciente que presenta vómito frecuente" (0.5)

Luego, como P1 y P2 son ciertas (aunque con diferentes grados de certidumbre), entonces la conjunción P1 AND P2 también lo será, el problema está en determinar con qué valor de certidumbre se podrá asegurar el cumplimiento de dicha conjunción.

Una forma muy clara de solucionar este problema es analizando el significado de la conjunción desde el punto de vista geométrico. Para esto es suficiente considerar gráficamente el intervalo real $\{-1,1\}$ y representar sobre este la credibilidad asociada a cada proposición mediante un segmento que vaya desde el mínimo valor de certidumbre (-1) hasta el valor de certidumbre correspondiente a cada proposición, tal como se muestra en la figura 1.2.

Teniendo en cuenta que la conjunción de dos o más proposiciones viene dada por el cumplimiento conjunto de éstas, entonces es necesario asociar a la conjunción de P1 y P2 el segmento de certidumbre común a ambas, el cual siempre será conservadoramente mayor o igual que el correspondiente al mínimo valor de las certidumbres. Por lo tanto, la certidumbre correspondiente a la conjunción de las proposiciones P1 y P2 podría ser el mínimo de las certidumbres de éstas. En el ejemplo, igual a 0.5.

De esta forma, generalizando, aceptamos que la certidumbre de la conjunción de n proposiciones: P₁, P₂, ..., P_n con grados de certidumbre asociados c₁, c₂, ..., c_n respectivamente, es conservadoramente el mínimo de estas certidumbres. Esto es :

$$CONJ(c_1, c_2, \dots, c_n) = MIN(c_1, c_2, \dots, c_n)$$

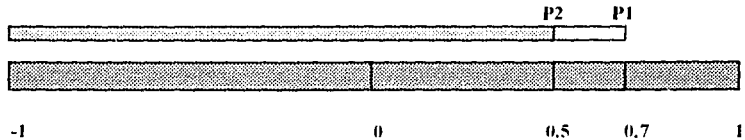


Figura 1.2 Representación de los valores de certidumbre asociados a las proposiciones P1 y P2

Cálculo de la certidumbre de una disyunción

Consideremos nuevamente las proposiciones P1 y P2 con sus respectivos fC 's asociados. Luego, como P1 y P2 son ciertas, entonces la disyunción P1 OR P2 también lo será, ya que para esto es suficiente que al menos una de las proposiciones sea cierta. El problema ahora está en determinar el valor de certidumbre con el cual se cumplirá esta disyunción.

En este caso también podemos utilizar el razonamiento geométrico empleado en el caso de la conjunción, pero ahora considerando la disyunción de las proposiciones P1 y P2 (figura 1.2) Teniendo en cuenta que la disyunción de dos o más proposiciones viene dada por el cumplimiento de al menos una de las proposiciones componentes, entonces es razonable asociar a la disyunción de éstas el segmento que represente a la mayor de las certidumbres de estas proposiciones. Luego, la certidumbre con la cual se cumple la disyunción P1 OR P2 es el máximo de las certidumbres de estas proposiciones, siendo en el ejemplo igual a 0.7.

De esta forma, generalizando, la certidumbre de la disyunción de n proposiciones o elementos P_1, P_2, \dots, P_n con grados de certidumbres asociados c_1, c_2, \dots, c_n respectivamente, es el máximo de estas certidumbres. Esto es :

$$DISY(c_1, c_2, \dots, c_n) = MAX(c_1, c_2, \dots, c_n)$$

Cálculo de la contribución de la regla

Consideremos la siguiente regla de producción :

```

IF palidez cutáneo mucosa AND
    somnolencia AND
    anorexia AND
    fatiga AND
    disnea al esfuerzo AND
    astenia
THEN anemia (0.8)
  
```

La interpretación de esta regla es la siguiente : de ser completamente cierto el antecedente de la regla (esto es, si se sabe con absoluta seguridad que los hechos que integran el antecedente son ciertos), entonces se puede asegurar el cumplimiento de la conclusión de la regla con una certidumbre igual a 0.8. Es decir, si los signos y síntomas establecidos en el antecedente de la regla se cumplen con certidumbre máxima igual a 1, entonces la certeza de que una anemia esté presente es de 0.8.

Esta interpretación nada nos dice acerca de en cuánto contribuye dicha regla a la conclusión en caso de que el antecedente no sea completamente cierto. Es decir, ¿en cuánto contribuirá la regla al consecuente cuando se ha asignado una medida de incertidumbre a los hechos que forman el antecedente? En este caso, la certidumbre de la hipótesis o conclusión de la regla debe tener en cuenta dos elementos : la fuerza con la cual el antecedente de la regla sugiere la hipótesis y el nivel de certidumbre con el que se cumple este antecedente. Luego, es necesario definir una función de combinación que permita combinar estos dos elementos para obtener el nuevo valor de certidumbre correspondiente a la conclusión de la regla.

Sea c el valor de certidumbre asociado a la conclusión de una regla, dado que su antecedente es completamente cierto. Sea a el valor de certidumbre del antecedente de la regla, obtenido a partir de las evidencias aportadas acerca de los hechos que lo integran. Entonces, una forma de calcular la contribución de la regla es la siguiente :

$$CONT(a,c) = c * \max(0,a)$$

Esta función de combinación es conocida como función de contribución proporcional, puesto que ésta asigna al consecuente de la regla un valor de certidumbre proporcional al valor actual de la certidumbre del antecedente

Otra función de combinación comúnmente utilizada es :

$$\min(a,c) \text{ para } a > 0 \text{ y } c > 0$$

$$CONT(a,c) = \begin{cases} \min(a,c) & \text{para } a > 0 \text{ y } c > 0 \\ 0 & \text{para } a < 0 \end{cases}$$

$$0 \text{ para } a < 0$$

la cual es conocida como función de contribución pura.

Toda función de combinación para el cálculo de la contribución debe cumplir las siguientes propiedades

- ◆ Si $a = 0$ y $c = 0$ entonces $CONT(a,c) = 0$
- ◆ Si $a = 0$ y $c = c$ entonces $CONT(a,c) = 0$
- ◆ Si $a = a$ y $c = 0$ entonces $CONT(a,c) = 0$
- ◆ Si $a = a$ y $c = c$ entonces $CONT(a,c) = c$

donde :

α : certidumbre del antecedente de la regla

c : certidumbre con la cual se cumple la conclusión de la regla
dado que el antecedente es absolutamente cierto

Ejemplo

Para ilustrar el cálculo de la contribución de una regla utilizando la función de contribución proporcional, supongamos que para la regla considerada inicialmente se han obtenido las siguientes evidencias :

palidez cutáneo mucosa (0.8)
somnolencia (0.6)
anorexia (0.7)
fatiga (0.9)
disnea al esfuerzo (0.6)
astenia (0.8)

El primer paso consiste en calcular la certidumbre del antecedente de la regla (a), el cual se trata de una conjunción de proposiciones simples :

$$a = \min (0.8, 0.6, 0.7, 0.9, 0.6, 0.8) = 0.6$$

$$a = 0.6$$

y dado que $c = 0.8$, entonces :

$$\begin{aligned} \text{CONT}(a,c) &= \text{CONT}(0.6, 0.8) = 0.8 * \max(0, 0.6) = \\ &= 0.8 * 0.6 = \\ &= 0.48 \end{aligned}$$

Por lo tanto, a partir de las evidencias existentes se puede afirmar la presencia de una anemia con un grado de certidumbre de 0.48.

Cálculo del efecto global de varias reglas

Consideremos las siguientes reglas de producción :

IF palidez cutáneo mucosa AND
somnolencia AND
anorexia AND
fatiga AND
disnea al esfuerzo AND
astenia
THEN anemia (0.8)

**IF síndrome de mala absorción OR
sangramiento digestivo mantenido OR
parasitismo intestinal severo
THEN anemia (0.9)**

La primera de estas reglas ya nos resulta conocida, ésta fue utilizada para ejemplificar el cálculo de la función de contribución proporcional. Al analizar estas reglas, rápidamente podemos apreciar que ambas conducen a la misma conclusión. Es decir, a partir de diferentes colecciones de hechos o evidencias se arriba a una misma conclusión (la presencia de anemia).

Este hecho es muy común, en la vida cotidiana usualmente ocurre que diferentes fuentes aportan criterios o elementos acerca de una misma conclusión. Traduciendo esto a nuestro dominio, es muy común (y de hecho necesario) que en una base de conocimientos diferentes reglas conduzcan a un mismo consecuente. Cuando este es el caso, lo más razonable es tomar en cuenta las contribuciones de todas estas reglas, para arribar con una mayor seguridad a la conclusión común a la que éstas conducen.

La integración de la certidumbre global sobre una conclusión común es un proceso acumulativo. Es decir, cada contribución aporta determinado valor de certidumbre, el cual contribuye al crecimiento de la certidumbre global parcial, si las evidencias apoyan la hipótesis, o a su decrecimiento si las evidencias niegan la hipótesis. Cuando una regla permite concluir una hipótesis con una certeza absoluta (valor 1), entonces esta conclusión no debe verse modificada.

Una función de suma o globalización muy utilizada, es aquella que considera el grado de certidumbre de una hipótesis, dado el cumplimiento de dos reglas que conducen a ésta, como el grado de creencia aportado a la hipótesis por una de las reglas más un cierto incremento tomado del grado de creencia aportado por la otra regla.

Dicho incremento es calculado de la siguiente forma : se halla la diferencia entre el valor de certidumbre absoluto (valor 1) y el grado de certidumbre de la hipótesis dado el cumplimiento de la primera regla. La diferencia obtenida es entonces escalada por el grado de creencia de la hipótesis dado el cumplimiento de la segunda regla. Esto significa que del valor de certidumbre de la hipótesis dado por la segunda regla, sólo una proporción equivalente a la diferencia calculada será aportado a la globalización.

Sea c_1 el grado de certidumbre de la hipótesis dado el cumplimiento de la primera regla. Sea c_2 el grado de certidumbre de la hipótesis dado el cumplimiento de la segunda regla. Entonces, la función de combinación para el cálculo del efecto global viene dada por la siguiente expresión

$$c_1 + c_2 * (1 - c_1) \quad \text{si } c_1, c_2 \geq 0$$

$$EGLOB(c_1, c_2) = \{ c_1 + c_2 * (1 + c_1) \quad \text{si } c_1, c_2 \leq 0$$

$$(c_1, c_2) / (1 - \min(\text{abs}(c_1), \text{abs}(c_2))) \quad \text{si } c_1 * c_2 < 0$$

Otra interesante función para el cálculo del efecto global es la utilizada por el sistema PROSPECTOR [BUC85, RIC'91]:

$$EGLOB(c_1, c_2) = (c_1 + c_2) / (1 + c_1 * c_2)$$

donde: $-1 \leq c_1, c_2 \leq 1$

Ejemplo

Para ilustrar el cálculo del efecto global, retomemos las dos reglas de producción consideradas al inicio. La contribución para la primera de estas reglas ya fue obtenida al ejemplificar el cálculo de la función de contribución, proporcionando esta regla un valor de certidumbre igual a 0.48 para la hipótesis anemia ($c_1 = 0.48$). Ahora es necesario calcular la contribución de la segunda regla y para esto vamos a suponer que las siguientes evidencias han sido proporcionadas:

síndrome de mala absorción (1.0)
sangramiento digestivo mantenido (0.7)
parasitismo intestinal (-1.0)

Como el antecedente de la regla es una disyunción, entonces:

$$a = \max(1.0, 0.7, -1.0) = 1.0$$

Luego, como existe una certeza absoluta acerca del cumplimiento del antecedente, no es necesario utilizar la fórmula para el cálculo de la contribución, ya que conocemos que en este caso la conclusión de la regla se cumplirá con el mismo grado de certidumbre proporcionado por el experto. Esto es, $c_2 = 0.9$.

Una vez obtenidas ambas contribuciones, es posible calcular el efecto global, teniendo en cuenta que c_1 y c_2 son mayores que cero:

$$\begin{aligned} EGLOB(0.48, 0.9) &= 0.48 + 0.9 * (1 - 0.48) = \\ &= 0.48 + 0.468 = \\ &= 0.94 \end{aligned}$$

Por lo tanto, el efecto global para la hipótesis anemia a partir de las contribuciones aportadas por ambas reglas es de 0.94. Esto significa que hay una certeza muy elevada acerca del cumplimiento de la hipótesis anemia.

Toda función de combinación para el cálculo del efecto global debe cumplir las siguientes propiedades :

- ◆ $-1 \leq EGLOB(c_1, c_2) \leq 1$
- ◆ *EGLOB es conmutativa* : $EGLOB(c_1, c_2) = EGLOB(c_2, c_1)$
- ◆ *EGLOB es asociativa* : $EGLOB(c_1, EGLOB(c_2, c_3)) = EGLOB(EGLOB(c_1, c_2), c_3)$
- ◆ $EGLOB(c_1, 1) = 1$ para $c_1 \in [-1, 1]$
- ◆ $EGLOB(c_1, -1) = -1$ para $c_1 \in [-1, 1]$
- ◆ $EGLOB(c_1, 0) = c_1$

donde : $-1 \leq c_1, c_2, c_3 \leq 1$

Lo analizado hasta aquí puede ser resumido de la siguiente forma :

1. La certidumbre de una proposición compuesta es una función de las proposiciones componentes y los conectivos lógicos utilizados.
2. El grado de certidumbre con el cual una regla contribuye a una hipótesis o conclusión es una función de la certidumbre del antecedente de la regla y de la certidumbre original de la regla aportada por el experto.
3. Cuando dos o más reglas proporcionan evidencias que relacionan a una misma hipótesis, entonces la certidumbre de esta hipótesis es una función de la contribución de cada una de estas reglas.

Cuando las funciones de combinación NEG, CONJ, DISY, CONT y EGLOB han sido definidas en el intervalo real $[-1, 1]$ y éstas cumplen con las propiedades antes descritas, entonces decimos que estas funciones forman una estructura de combinación.

Cuando en un Sistema Experto basado en reglas se representa y procesa la incertidumbre utilizando factores de certidumbre y funciones de combinación, entonces una estructura de combinación particular debe haber sido definida.

Componentes de un Sistema Experto

Los componentes de un típico Sistema Experto basado en reglas son mostrados en la figura 1.3. La base de conocimientos, el mecanismo de inferencia, la memoria de trabajo y la interfaz de entrada/salida son componentes que se encuentran en todo Sistema Experto de este tipo, mientras que el módulo de explicaciones pudiera no estar presente

en alguno de ellos y el módulo de aprendizaje sólo está presente en unos pocos. [BUC89, HAY83, JAC90, MER89, FOR84, PAT90]

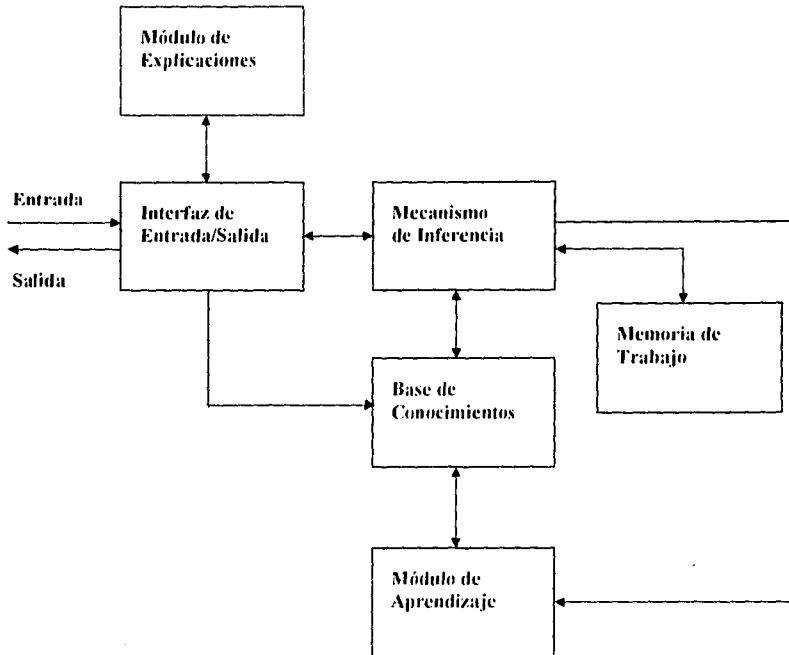


Figura 1.3 Componentes de un Sistema Experto

La base de conocimientos

La base de conocimientos contiene reglas, hechos e información acerca de un dominio especializado de conocimientos. Este conocimiento es utilizado por el mecanismo de inferencia para formular hipótesis

Muchos autores identifican a la base de conocimientos como la base estática, debido a que el conocimiento contenido en ésta no se modifica ni se actualiza durante los procesos de solución de problemas, a menos que el Sistema Experto posea un módulo de aprendizaje incorporado.

La cantidad y calidad de los conocimientos contenidos en la base de conocimientos determinan en primer lugar la bondad del Sistema Experto en la solución de problemas del dominio. Una buena base de conocimientos debe ser exhaustiva y consistente. La exhaustividad significa completitud. Es decir, que todas las reglas y hechos necesarios para solucionar cualquier problema del dominio se encuentren incorporados en la base de conocimientos. La consistencia se refiere a que la base de conocimientos esté libre de reglas contradictorias, redundantes o innecesarias.

El mecanismo de inferencia

Como se explicó en el epígrafe anterior, los procesos de inferencia en los Sistemas Expertos basados en reglas se llevan a cabo a través de encadenamientos de reglas, en dirección *forward*, *backward* o como una combinación de éstos.

Para realizar las inferencias, el mecanismo de inferencia utiliza información dinámica y conocimiento estático. La información dinámica corresponde a los datos de entrada que fueron inicialmente aportados por el usuario y a las respuestas que éste proporciona a preguntas formuladas por el sistema. El conocimiento estático es el contenido en la base de conocimientos, éste es utilizado por el mecanismo de inferencia para formular hipótesis o arribar a conclusiones acerca del caso o situación actual.

Toda la información aportada por el usuario es almacenada como hechos en la memoria de trabajo. Durante el proceso de inferencia las reglas contenidas en la base de conocimiento son comparadas con el contenido de la memoria de trabajo. Cuando más de una regla ha logrado satisfacer sus condiciones, entonces se realiza un proceso de selección (generalmente utilizando criterios heurísticos) mediante el cual se determina la regla que será ejecutada. Una vez ejecutada esta regla, su acción o consecuente pasa a formar parte de los hechos almacenados en la memoria de trabajo. Esto origina la formación de nuevas configuraciones de hechos, para los cuales otras reglas de la base de conocimientos pudieran satisfacer sus condiciones. De esta forma, el proceso continúa recursivamente hasta que ya no sea posible encontrar nuevas reglas que se disparen.

La memoria de trabajo

La memoria de trabajo, también conocida como base dinámica, es un "almacén" temporal de información dinámica. En ésta es almacenada, en forma de hechos, toda la información brindada por el usuario al sistema (datos iniciales y respuestas a preguntas formuladas), así como las conclusiones de todas las reglas "disparadas" en el transcurso del proceso de inferencias.

Cuando el proceso de solución de un problema particular ha concluido, el contenido de la memoria de trabajo es removido o eliminado, de forma tal que esta memoria quede "limpia" antes de iniciar la solución de otro problema.

La interfaz de entrada/salida

La interfaz de entrada/salida permite la comunicación entre el usuario y el sistema. A través de ésta el usuario ofrece datos iniciales al sistema o responde preguntas formuladas por éste.

La gran mayoría de las interfaces de comunicación conocidas establecen la comunicación usuario-sistema mediante simples menús de selección o utilizando lenguajes restringidos, los cuales son aproximaciones al lenguaje natural.

El módulo de explicaciones

El módulo de explicaciones proporciona al usuario explicaciones acerca del proceso de inferencias, cuando éstas son solicitadas. Las explicaciones ofrecidas al usuario responden a la formulación de las siguientes preguntas :

- ◆ ¿Cómo una hipótesis o conclusión fue alcanzada?
- ◆ ¿Por qué cierta información es requerida?

Para responder a la primera de estas preguntas, el módulo de explicaciones exhibe al usuario, en una forma que resulte de fácil comprensión, la cadena de reglas disparada durante el proceso de inferencias. De esta manera, el usuario puede apreciar la línea de razonamiento seguida por el sistema para arribar a la conclusión.

La respuesta del módulo de explicaciones a la pregunta ¿por qué?, consiste en explicar al usuario el papel que juega la información solicitada al dar algún paso necesario dentro del proceso de razonamiento.

De esta forma, a través del módulo de explicaciones el usuario puede seguir de muy cerca el proceso de inferencias llevado a cabo durante la solución de un problema.

El módulo de aprendizaje

La función del módulo de aprendizaje es la de contribuir en la construcción y refinamiento de la base de conocimientos. Es decir, si un Sistema Experto posee un

módulo de aprendizaje, entonces éste posee facultades para adquirir nuevos conocimientos y para refinar el conocimiento existente en su base.

El módulo de aprendizaje no es un componente común de los Sistemas Expertos. De hecho, son muy pocos los Sistemas Expertos conocidos que poseen esta habilidad.

Tipología de Sistemas Expertos

Una de las mejores formas de conocer el desempeño e importancia de los Sistemas Expertos es a través de las diferentes tareas genéricas que son capaces de ejecutar.

La amplia gama o tipología de Sistemas Expertos conocida hoy en día es consecuencia de la gran variedad de dominios de problemas explorados. Existen Sistemas Expertos para la interpretación, la predicción, el diagnóstico, el diseño, la planeación, el monitoreo, la depuración, la reparación, la instrucción, el control y otros tipos de tareas. [BAR89b, HAY83, JAC'90]

Sistemas Expertos para la interpretación

La interpretación es el análisis de un conjunto de datos para determinar el significado de los mismos. El principal requerimiento de estos sistemas es que las interpretaciones realizadas sean correctas y consistentes.

El gran problema que se presenta en la interpretación es que con frecuencia los datos pueden contener ruidos y errores. Esto es, los valores de los datos pueden estar omitidos, ser erróneos o extraños. Esto significa que en muchas ocasiones la interpretación tiene que llevarse a cabo sobre información parcial. También pudiera ocurrir que los datos fueran contradictorios, en este caso el sistema de interpretación debe estar habilitado para determinar cuáles datos son creíbles y cuáles no. Cuando los datos no son confiables, la interpretación tampoco será confiable. Un sistema de interpretación debe ser capaz de identificar dónde la información es incierta o incompleta o dónde han sido hechas ciertas suposiciones.

Las categorías de Sistemas Expertos para la interpretación incluyen dominios de sistemas para la vigilancia, la comprensión del lenguaje, análisis de imágenes, investigación de estructuras químicas, interpretación de señales, entre otros.

Los Sistemas Expertos basados en reglas con encadenamiento hacia adelante han sido los más beneficiosos para el desarrollo de sistemas de interpretación.

Sistemas Expertos para la predicción

La predicción significa pronosticar los eventos que ocurrirán en el futuro a partir de un modelo de la ocurrencia de estos eventos en el pasado y en el presente.

A la predicción le es inherente un razonamiento basado en el tiempo. Un sistema de predicción debe poseer habilidades para tratar con eventos que cambian en el tiempo y con eventos que poseen un orden en el tiempo. Además de contar con modelos adecuados de la forma en que determinadas acciones influyen en el estado del medio ambiente modelado sobre el tiempo, los sistemas para la predicción requieren de la integración de información incompleta. Estos Sistemas Expertos deben ser capaces de explicar todos los futuros posibles mediante razonamiento hipotético, así como indicar la sensibilidad de las predicciones a variaciones en los datos de entrada.

Es muy común en los Sistemas Expertos para la predicción el uso de un modelo paramétrico dinámico, el cual ajusta los valores de los parámetros para cada situación dada. Las consecuencias que se infieren a partir de este modelo forman las bases para la predicción.

El dominio de Sistemas Expertos para la predicción incluye sistemas para el pronóstico del tiempo, predicciones demográficas, predicciones de tráfico, estimaciones de cosechas, pronósticos militares, entre otros.

Los sistemas orientados a objetos son usualmente los más aceptables para esta clase de problemas, aunque los sistemas orientados a procedimientos y los sistemas basados en reglas con encadenamiento hacia adelante también resultan útiles.

Sistemas Expertos para el diagnóstico

El diagnóstico es el proceso mediante el cual se buscan las fallas o desperfectos en un sistema. Un Sistema Experto para el diagnóstico es un sistema que infiere las fallas o mal funcionamiento de un sistema a partir de la interpretación de datos observados, los cuales son potencialmente ruidosos, inseguros o incompletos.

El dominio de Sistemas Expertos para el diagnóstico incluye sistemas para el diagnóstico médico, electrónico, mecánico, de *software*, entre otros diagnósticos.

Los sistemas basados en reglas con encadenamiento hacia atrás han resultado ser fuertes candidatos para problemas del tipo diagnóstico. Por otra parte, los sistemas basados en redes neurales resultan muy útiles para problemas de diagnóstico, sobre todo cuando el diagnóstico es fuertemente dependiente del reconocimiento de patrones. La combinación de sistemas basados en reglas y en redes neurales también ha resultado ser muy útil para este tipo de problemas.

Sistemas Expertos para el diseño

El diseño es la descripción de objetos a partir de ciertas restricciones o requerimientos establecidos. Un Sistema Experto para el diseño es aquel sistema que desarrolla configuraciones de objetos que satisfacen las restricciones del problema a diseñar. Estos sistemas construyen descripciones de objetos, los cuales se pueden encontrar en relación con otros objetos, y verifican que estas configuraciones respondan a las restricciones establecidas. Muchos Sistemas Expertos para el diseño intentan minimizar una función objetivo que mida el costo u otras propiedades no deseables para el diseño potencial.

El dominio de los Sistemas Expertos para el diseño incluye sistemas para el diseño de circuitos, diseño de construcciones, diseño de presupuesto, entre otros.

Los Sistemas Expertos basados en reglas con encadenamiento hacia adelante han resultado ser muy útiles para aplicaciones de diseño. Los sistemas basados en marcos (*frames*), los sistemas orientados a procedimientos y los sistemas basados en lógica son también útiles para sistemas de diseño.

Sistemas Expertos para la planeación

La planeación es la preparación de un programa o secuencia de acciones, las cuales al ser llevadas a cabo permiten alcanzar una meta deseada. Cuando un Planeador construye un plan de acciones, éste vigila que no se consuman excesivos recursos ni sean violadas restricciones establecidas. Si algunas de las metas entran en conflicto, entonces el Planeador debe establecer prioridades entre éstas. Un Planeador debe ser flexible y oportunístico, teniendo en cuenta que los requerimientos de planeación pudieran no ser completamente conocidos o pudieran éstos cambiar en el tiempo.

El dominio de los Sistemas Expertos para la planeación incluye sistemas de programación automática, robótica, proyectos, rutas, comunicaciones, experimentos, planeación militar, entre otros tipos de planeación

Los Sistemas Expertos basados en reglas con encadenamiento hacia adelante, los basados en marcos, los orientados a objetos y los orientados a procedimientos, son

generalmente muy útiles para el desarrollo de sistemas expertos dedicados a la planeación.

Sistemas Expertos para el monitoreo

El monitoreo significa la interpretación continua de señales de entrada y el control y activación de alarmas cuando éstas son requeridas. Un Sistema Experto para el monitoreo es un sistema que realiza diagnósticos parciales a partir de las interpretaciones continuas. El requerimiento de alarmas es en tiempo real. Las condiciones de alarmas son dependientes del contexto en el cual evoluciona el sistema.

El dominio de Sistemas Expertos para el monitoreo incluye sistemas para el monitoreo de plantas nucleares, procesos industriales de alto riesgo, monitoreo de tráfico aéreo, monitoreo de pacientes en salas de cuidados intensivos, entre otros.

Los Sistemas Expertos basados en reglas con encadenamiento hacia atrás son generalmente muy útiles para este tipo de aplicación. Los sistemas basados en lógica y las combinaciones de sistemas basados en reglas con encadenamiento hacia atrás y marcos son también beneficiosos para este tipo de aplicación.

Sistemas Expertos para la depuración

Los Sistemas Expertos para la depuración prescriben remedios para las fallas o mal funcionamiento diagnosticado en un sistema. Este tipo de sistemas posee habilidades para la planeación, diseño y predicción, las cuales son utilizadas para la formulación de recomendaciones para corregir un problema diagnosticado.

Los sistemas para la depuración existen en la forma de bases de conocimientos inteligentes o como editores de texto, pero no poseen cualidades como las de un Sistema Experto.

Para este tipo de aplicación resultan muy útiles los sistemas basados en reglas con encadenamiento hacia atrás. También los sistemas basados en la lógica y la combinación de sistemas basados en reglas con encadenamiento hacia atrás y sistemas basados en marcos son beneficiosos para este tipo de aplicación.

Sistemas Expertos para la reparación

Los Sistemas Expertos para la reparación desarrollan y ejecutan planes para la administración de remedios cuando algún problema ha sido diagnosticado. Estos sistemas exhiben capacidades de depuración y planeación

El dominio de estos Sistemas Expertos incluye sistemas para el mantenimiento de computadoras, mantenimiento de redes, entre otros.

Los Sistemas Expertos basados en reglas con encadenamiento hacia atrás resultan ser muy útiles para este tipo de aplicación. También los sistemas basados en la lógica y una combinación de sistemas basados en reglas con encadenamiento hacia atrás y sistemas basados en marcos resultan beneficiosos para el desarrollo de sistemas de este tipo.

Sistemas Expertos para la instrucción

Los Sistemas Expertos para la instrucción diagnostican y depuran el comportamiento de un estudiante en la solución de problemas correspondientes a un dominio específico. Estos sistemas poseen capacidades de diagnóstico y depuración, las cuales son utilizadas en dependencia de los intereses del sistema. De forma general, los sistemas para la instrucción comienzan construyendo una descripción hipotética del conocimiento del estudiante (modelo del estudiante), a partir de la cual interpretan el comportamiento de éste en la solución de problemas del dominio. El sistema diagnostica las debilidades o fallas en el conocimiento del estudiante, identifica el remedio apropiado y finalmente planea una interacción tutorial intentando comunicar el conocimiento remediable al estudiante.

El dominio de Sistemas Expertos para la instrucción (también conocidos como sistemas tutoriales) incluye sistemas de instrucción en dominios como matemáticas, física, química, ingeniería y medicina, entre otros.

Los Sistemas Expertos basados en reglas con encadenamiento hacia adelante y hacia atrás son los mejores para este tipo de aplicaciones.

Sistemas Expertos para el control

Los Sistemas Expertos para el control modifican de forma adaptativa el comportamiento total de un sistema. Para lograr esto, el sistema experto debe interpretar continuamente la situación actual, pronosticar la situación futura, diagnosticar por anticipado las causas de problemas, formular un plan "remedial" y monitorear su ejecución para garantizar el éxito en el comportamiento del sistema, entre otras tareas. Un sistema para el control posee capacidades de interpretación, diagnóstico, pronóstico, monitoreo, depuración y reparación.

El dominio de Sistemas Expertos para el control incluye sistemas para el control de tráfico aéreo, control de líneas de producción automatizadas, control de equipos de vida artificial en salas de terapia intensiva, entre otros.

Los sistemas orientados a procedimientos, sistemas orientados a objetos y sistemas basados en reglas con encadenamiento hacia adelante son considerados los mejores para aplicaciones de control.

Principales áreas de aplicación de los Sistemas Expertos

Han sido muchas las áreas o dominios de interés que han servido de escenario para el desarrollo y aplicación de los Sistemas Expertos. Dentro de estas áreas de aplicación se destacan las siguientes : administración de información, agricultura, ciencias militares, control de procesos, derecho, electrónica, física, geología, ingeniería, matemáticas, medicina, meteorología, química/bioquímica y sistemas de computación. [BAR89a, BAR89b, BUC85, FOR84, HAY83, JAC90, PAT90]

Administración de información

- ◆ administración de bases de datos (demanda, actualización y recuperación de información)
- ◆ desarrollo de interfaces de lenguaje natural para servicios de recuperación de información en bases de datos en línea (servicios en bibliotecas y centros de información de dominio público)

Agricultura

- ◆ predicción de cosechas
- ◆ control y diagnóstico de plagas y enfermedades de las plantas
- ◆ mejoramiento de cultivos
- ◆ diagnóstico y tratamiento de suelos
- ◆ monitoreo de invernaderos y cultivos en condiciones especiales

Ciencias militares

- ◆ trabajos de cartografía
- ◆ interpretaciones tácticas de reportes de sensores inteligentes
- ◆ identificación de aviones sobre las bases de las características visualmente observadas
- ◆ asistencia a oficiales en operaciones aéreas con lanzamiento y recuperación de aviones sobre un portaaviones
- ◆ asistencia en el análisis y valoración de la situación en el campo de batalla
- ◆ ayuda en la identificación del tipo de radar que ha generado una señal interceptada
- ◆ detección y clasificación de objetivos a partir de imágenes de sensores
- ◆ recomendaciones para la asignación de recursos en situaciones de combate
- ◆ asistencia a pilotos de aviones de guerra en el control de procedimientos de emergencia en vuelo
- ◆ monitoreo de sensores de navegación sobre aviones de táctica avanzada
- ◆ ayuda en el diagnóstico de problemas o fallas en los aviones
- ◆ clasificación de buques a partir de la interpretación de las imágenes del radar
- ◆ detección e identificación de varios tipos de buques a partir del uso de datos acústicos digitalizados
- ◆ simulación interactiva de combates aéreos y terrestres, así como medioambientes en los cuales desarrollar y perfeccionar tácticas y estrategias militares

Control de procesos

- ◆ control y monitoreo de procesos en líneas de producción automatizadas en fábricas e industrias
- ◆ control y monitoreo de procesos en centrales nucleares
- ◆ control y monitoreo de procesos químicos
- ◆ diagnóstico de fallas en procesos automatizados a partir de la interpretación de datos obtenidos desde sensores acoplados al proceso

Derecho

- ◆ asistencia en la investigación de razonamiento legal y argumentación legal en diferentes dominios de aplicación de las leyes legales
- ◆ asistencia en el análisis de decisiones gobernadas por normas "discrecionales"
- ◆ ayuda en la recuperación de información acerca de decisiones y legislaciones de tribunal en diferentes dominios de aplicación de las leyes
- ◆ asistencia en el razonamiento acerca de casos de leyes civiles

- ◆ asistencia en la evaluación de casos legales

Electrónica

- ◆ identificación y localización de problemas en redes telefónicas con la recomendación de la reparación apropiada y mantenimiento rehabilitativo
- ◆ identificación y localización de problemas en redes eléctricas, recomendando la reparación apropiada
- ◆ asistencia en el diseño y prueba de circuitos electrónicos
- ◆ asistencia en el diagnóstico de fallas de componentes eléctricos y electrónicos
- ◆ asistencia en el diseño y rediseño de circuitos digitales para encontrar especificaciones funcionales alteradas
- ◆ asistencia en la síntesis de circuitos eléctricos

Física

- ◆ asistencia en la solución de problemas en física mecánica
- ◆ asistencia en la solución de problemas en termodinámica
- ◆ asistencia en la solución de problemas en física nuclear
- ◆ instrucción inteligente en la solución de problemas de física

Geología

- ◆ inferencia de estructuras geológicas a partir de la interpretación de datos relevantes relacionados
- ◆ asistencia en la detección de yacimientos de minerales importantes
- ◆ asistencia en la interpretación de datos desde troncos de pozos de petróleo

Ingeniería

- ◆ asistencia en el diseño de construcciones
- ◆ asistencia en la determinación de estrategias de análisis para problemas particulares de análisis estructural
- ◆ evaluación de la seguridad y perjuicios de una estructura existente
- ◆ diagnóstico y tratamiento de reactores nucleares a partir de las lecturas de instrumentos de monitoreo
- ◆ determinación de las causas de eventos anormales en plantas nucleares
- ◆ diagnóstico y corrección de fallas en máquinas eléctricas

- ◆ instrucción inteligente en la solución de problemas de ingeniería

Matemáticas

- ◆ instrucción inteligente en la solución de problemas de cálculo, álgebra, ecuaciones diferenciales y otros dominios de las matemáticas
- ◆ asistencia en la manipulación simbólica de expresiones algebraicas encontradas en los problemas de análisis

Meteorología

- ◆ pronóstico del tiempo
- ◆ predicción de tormentas severas

Química/Bioquímica

- ◆ inferencia de la estructura molecular de compuestos desconocidos
- ◆ inferencia de la estructura tridimensional de proteínas
- ◆ análisis de la estructura del DNA
- ◆ asistencia en la síntesis de moléculas orgánicas complejas
- ◆ asistencia en la ejecución de varios tipos de análisis de secuencias de nucleótidos
- ◆ asistencia en la planeación de experimentos de laboratorio complejos
- ◆ síntesis de moléculas orgánicas complejas sin requerir la interacción del usuario

Sistemas de computación

- ◆ diagnóstico y depuración de fallas en sistemas de *software* y *hardware*
- ◆ control y monitoreo de sistemas operativos
- ◆ instrucción inteligente de lenguajes y sistemas de computación

"ZOOM" sobre un dominio de aplicación

Desde el surgimiento de los Sistemas Expertos y hasta nuestros días, la medicina ha sido uno de los dominios en los que más Sistemas Expertos se han desarrollado. La creación del Sistema Experto MYCIN [BUC85, SHO76, SHO85] dio inicio a una prolifera etapa para la creación, desarrollo y evolución de Sistemas Expertos en medicina. No todos los Sistemas Expertos desarrollados en este dominio han alcanzado la etapa de sistemas de aplicación, muchos de éstos sólo han llegado a ser prototipos de investigación o prototipos de demostración. [BAR89b, BUC85, NEG91a]

El desarrollo de los Sistemas Expertos en medicina ha abarcado una gran cantidad de subdominios de interés, entre los que se destacan :

- diagnóstico y tratamiento de enfermedades
- monitoreo de pacientes en unidades de cuidados intensivos
- interpretación de exámenes de laboratorio
- asistencia en la selección de terapias apropiadas
- asistencia en la administración de medicamentos
- asistencia en cirugía
- interpretación de imágenes
- instrucción inteligente en medicina

Han sido tantos los Sistemas Expertos desarrollados para el dominio de la medicina, que no sería posible mencionarlos todos aquí. Es por ello que sólo nos referiremos a un reducido grupo de Sistemas Expertos, a partir de los cuales se pueda ilustrar el papel de esta rama de la inteligencia artificial en los diferentes subdominios.

Los Sistemas Expertos a los cuales haremos referencia aparecen con su nombre original, para facilitar su identificación. En cada caso se describe en forma muy breve el subdominio de aplicación, las metas a alcanzar, el esquema empleado para la representación del conocimiento y aspectos relacionados con el razonamiento diagnóstico del Sistema Experto.

ABEL

El Sistema Experto ABEL fue desarrollado en MIT. ABEL diagnostica desórdenes electrolíticos y ácido-básico en pacientes. Este posee conocimientos acerca de enfermedades del dominio, así como de los síntomas que éstas producen. ABEL utiliza un modelo causal acerca de las posibles enfermedades presentes en el paciente para controlar la interacción con el médico y orientar el proceso de razonamiento diagnóstico. En ABEL el conocimiento es representado con una red semántica.

mediante la cual se especifican las relaciones causa-efecto entre enfermedades y hallazgos. [PAT81]

AI/COAG

El sistema AI/COAG fue desarrollado en la Escuela de Medicina de la Universidad de Missouri. AI/COAG brinda ayuda al médico en el diagnóstico de enfermedades de hemostasis, a partir del análisis e interpretación de pruebas de laboratorio clínico para la coagulación de la sangre. AI/COAG evalúa la historia clínica de un paciente con una enfermedad de hemostasis para confirmar el diagnóstico sugerido por las pruebas de laboratorio realizadas.

AI/RHEUM

El sistema AI/RHEUM fue desarrollado en la Escuela de Medicina de la Universidad de Missouri. AI/RHEUM brinda ayuda al médico en el diagnóstico de enfermedades del tejido conectivo, en el dominio de reumatología clínica. El sistema utiliza los síntomas y los resultados de laboratorios obtenidos del paciente para proporcionar asistencia en el diagnóstico de enfermedades del dominio (en total siete enfermedades). El conocimiento contenido en AI/RHEUM es obtenido desde expertos médicos en reumatología, éste es representado utilizando reglas de producción y la inferencia es alcanzada mediante un encadenamiento *forward*. [LIN80]

ANGY

El sistema ANGY fue desarrollado en la Universidad de Pennsylvania. ANGY brinda ayuda al médico en el diagnóstico del estrechamiento de los vasos coronarios, a partir de la identificación y aislamiento de vasos coronarios en el angiograma. ANGY inicia procesamiento angiogramas digitales de vasos coronarios con el objetivo de hacer una extracción de rasgos iniciales (líneas y regiones). Esta información constituye la entrada a un subsistema experto basado en el conocimiento que contiene ANGY. Este subsistema ejecuta un primer procesamiento de la imagen a bajo nivel, en el cual las líneas y regiones iniciales son transformadas en objetos más significativos. El segundo procesamiento de la imagen es ejecutado a elevado nivel, utilizando conocimientos de fisiología y anatomía del sistema cardiovascular para interpretar los resultados, reconocimiento de estructuras relevantes y la eliminación de estructuras irrelevantes o artefactos causados por ruido. El conocimiento médico en ANGY es representado con reglas de producción.

ARAMIS

El sistema ARAMIS fue desarrollado en la Universidad de Stanford. ARAMIS asiste al médico en la valoración de nuevos pacientes con enfermedades reumáticas. El conocimiento en ARAMIS consiste de una colección de métodos de análisis estadístico y un conjunto de bases de datos, las cuales contienen registros detallados de los

pacientes. El sistema recupera datos de pacientes con enfermedades reumáticas, los cuales han sido previamente almacenados en estas bases de datos y ejecuta análisis estadístico sobre estos datos. Esta actividad le permite a ARAMIS ofrecer análisis y pronósticos acerca de una gran variedad de parámetros, recomendar terapias, así como emitir un análisis del caso por escrito. [SHO79]

BABY

El sistema BABY fue desarrollado en la Universidad de Illinois, Champaign-Urbana. BABY es un sistema experto para el monitoreo de pacientes en una unidad de cuidados intensivos de recién nacidos. El sistema monitorea todos los datos en línea en la unidad de cuidados intensivos, mantiene un seguimiento del estado clínico de los pacientes y responde preguntas acerca de los pacientes. BABY contiene conocimiento médico de neonatología, el cual utiliza para la interpretación de datos clínicos y demográficos. El conocimiento en BABY es representado utilizando reglas de producción, con encadenamiento hacia adelante. El tratamiento de la incertidumbre es llevado a cabo usando un método probabilístico Bayesiano, similar al usado en el sistema PROSPECTOR.

BLUE BOX

El sistema BLUE BOX fue desarrollado en la Universidad de Stanford. BLUE BOX asesora al médico acerca de la selección de la terapia apropiada para pacientes en estados depresivos. El sistema posee conocimientos acerca de los síntomas presentes en el paciente, la trayectoria médica y psiquiátrica seguida por el paciente, medicamentos administrados y problemas o situación familiar, para diagnosticar el tipo de depresión y sugerir un plan terapéutico para su control. El conocimiento en BLUE BOX es representado utilizando reglas de producción. [MUL84]

CANCER-STAGE

El Sistema Experto CANCER-STAGE fue desarrollado en el Departamento de Patología del Centro Médico Cedar Sinai, Los Ángeles. CANCER-STAGE actúa como un consultante clínico en el dominio de patología. CANCER-STAGE determina la etapa a la que pertenecen pacientes con cáncer y recupera información terapéutica y de pronóstico. CANCER-STAGE es un sistema experto basado en reglas. Su base de conocimientos contiene más de 350 reglas que poseen información terapéutica y referencias. El sistema proporciona al usuario la etapa TNM de un paciente particular con cáncer y un resumen de opciones terapéuticas. [MAR93]

CASNET/GLAUCOMA

El Sistema Experto CASNET/GLAUCOMA fue desarrollado en la Universidad de Rutgers. El sistema diagnostica enfermedades relacionadas con glaucoma y recomienda planes terapéuticos para el tratamiento de éstas. El conocimiento experto del sistema

contiene relaciones entre síntomas del paciente, resultados de pruebas realizadas, condiciones anormales internas, enfermedades y tratamientos. Para representar todas estas relaciones el sistema utiliza una red causal-asociación, la cual es un tipo particular de una red semántica. [WEI78]

CENTAUR

El Sistema Experto CENTAUR fue desarrollado en la Universidad de Stanford. CENTAUR brinda ayuda a los fisiólogos del sistema respiratorio en la interpretación diagnóstica de pruebas del funcionamiento respiratorio. El conocimiento experto de CENTAUR incluye los resultados de las pruebas respiratorias y la fisiología del sistema respiratorio para cada enfermedad respiratoria. El conocimiento es representado utilizando una combinación de *frames* y reglas, incluyendo factores de certidumbre para representar la incertidumbre presente en parte del conocimiento. CENTAUR posee capacidad de explicación, la cual le permite proporcionar justificaciones acerca de los resultados o conclusiones alcanzadas. [AIK80]

CLOT

El Sistema Experto CLOT fue desarrollado en la Universidad de Stanford. CLOT proporciona ayuda al médico en la evaluación de evidencias para diagnosticar desórdenes en el sistema de coagulación sanguínea. El sistema diagnostica desórdenes en la coagulación sanguínea, identificando cual de los dos subsistemas de coagulación pudiera presentar un mal funcionamiento. En CLOT el conocimiento experto es representado utilizando reglas de producción con encadenamiento hacia atrás, incluyendo factores de certidumbre para representar la incertidumbre presente en el conocimiento experto. CLOT cuenta con la capacidad de explicación, lo cual le permite proporcionar justificaciones acerca de las conclusiones alcanzadas. [BEN80]

DAFODILL

El sistema DAFODILL fue desarrollado en la Escuela de Medicina de la Universidad de Stanford. DAFODILL brinda ayuda a los radiólogos en el diagnóstico de lesiones hepáticas. El sistema interpreta imágenes abdominales obtenidas por resonancia magnética (IRM) para diagnosticar lesiones en el hígado. DAFODILL utiliza una base de conocimientos de hallazgos obtenidos desde IRM y un mecanismo de inferencia basado en una red de creencias, para generar un diagnóstico diferencial probabilístico de las lesiones hepáticas más comunes encontradas. DAFODILL identifica rasgos clínicamente relevantes, mediante la ejecución de un procesamiento de imágenes limitado. Los rasgos encontrados son presentados al usuario para su confirmación, antes de que éstos sean procesados por la red. [TOM93]

DIAGNOSER

El sistema DIAGNOSER fue desarrollado en la Universidad de Minnesota. DIAGNOSER brinda ayuda al médico en la identificación de enfermedades congénitas del corazón. El conocimiento experto de DIAGNOSER incluye conocimiento acerca del razonamiento diagnóstico, así como conocimiento acerca de la fisiología, anatomía y fisiopatología cardíaca de las estructuras subordinadas a las enfermedades congénitas del corazón. En DIAGNOSER el conocimiento experto es representado utilizando *frames* y reglas. El conocimiento acerca de las enfermedades es representado utilizando *frames*, mientras que el conocimiento del razonamiento diagnóstico es representado con reglas, las cuales están contenidas en los *frames*.

DIALYSIS THERAPY ADVISOR

El sistema DIALYSIS THERAPY ADVISOR (DTA) fue desarrollado en la Universidad de Vanderbilt. DTA proporciona ayuda al médico para la selección de un régimen inicial de diálisis para pacientes que han de comenzar un tratamiento de mantenimiento hemodialísis. El sistema recibe como entradas datos personales (sexo, peso y estatura) y clínicos (volumen de orina, concentración de urea y nitrógeno, entre otros) del paciente y brinda como salida la relación de las terapias adecuadas. El conocimiento experto en DTA es representado utilizando reglas con encadenamiento hacia adelante y hacia atrás. Las reglas especifican el razonamiento de expertos médicos en la especificación de prescripciones iniciales de hemodiálisis.

DRUG INTERACTION CRITIC

El sistema DRUG INTERACTION CRITIC (DIC) fue desarrollado en la Universidad Estatal & Instituto Politécnico de Virginia. DIC proporciona ayuda al médico en la decisión de cómo administrar medicamentos a un paciente ante la presencia de otros medicamentos. DIC identifica las interacciones beneficiosas y perjudiciales entre medicamentos, explica la naturaleza de la interacción y sugiere medidas a tomar ante la ocurrencia de efectos perjudiciales. El conocimiento experto en DIC incluye conocimientos sobre tipos de medicamentos y conocimientos acerca del mecanismo de interacción de los medicamentos. El conocimiento sobre los tipos de medicamentos es representado utilizando una jerarquía de *frames*, incluyendo información acerca de medicamentos específicos (afinidades del medicamento, sitios de almacenamiento, características de la interacción, etc.) El conocimiento acerca de los mecanismos de interacción de los medicamentos también es representado utilizando *frames*, definiendo un *frame* para cada tipo de mecanismo (químico-físico, fármaco-dinámico, fármaco-cinético y fisiológico).

EXAMINER

El sistema EXAMINER fue desarrollado en la Universidad de Pittsburgh. EXAMINER analiza el comportamiento diagnóstico del médico sobre casos de medicina interna.

EXAMINER presenta un caso hipotético (grupo de manifestaciones clínicas) al médico, para que identifique la enfermedad más probable que pudiera ser la causa de dichas manifestaciones, así como indicar otras enfermedades o problemas que pudieran estar presentes. El conocimiento experto en EXAMINER incluye conocimiento heurístico acerca de cómo relacionar enfermedades y grupos de manifestaciones, y conocimiento médico obtenido desde el sistema INTERNIST. Todo el conocimiento es representado utilizando procedimientos que abarcan los principios de medicina interna.

GAIT-ER-AID

El sistema GAIT-ER-AID fue desarrollado en la Universidad del Sur de California. GAIT es un sistema experto para el diagnóstico en la manera de andar de las personas. El conocimiento experto en GAIT ha sido representado utilizando *frames*. Para conducir su razonamiento diagnóstico GAIT sigue la siguiente secuencia de pasos :

1. Clasificación de síntomas. Los síntomas son clasificados a partir de un examen en los datos de movimiento obtenidos del *frame* con la información del paciente.
2. Generación de diagnósticos hipótesis. Las causas probables de estos síntomas son inferidas mediante el apareo de patrones, utilizando los *frames* de referencia diagnóstica.
3. Validación. Los diagnósticos más probables (candidatos) son validados y refinados a partir del examen de datos relevantes del EMG para el movimiento. {BEK92}

GALEN

El sistema GALEN fue desarrollado en la Universidad de Minessota. GALEN diagnostica enfermedades congénitas del corazón en niños. Para identificar la presencia de una enfermedad, GALEN aplica su conocimiento experto (el cual está basado sobre un modelo de enfermedades pediátricas de cardiología) sobre la información obtenida desde el paciente (datos registrados en la historia clínica, resultados de exámenes físicos, rayos-X, electrocardiograma). El conocimiento experto es representado utilizando reglas y *frames*. Las reglas son utilizadas para representar el razonamiento diagnóstico (considerar, aceptar o rechazar una hipótesis), mientras que los *frames* son utilizados para representar toda la información relevante para una enfermedad en particular.

GUIDON

El sistema GUIDON fue desarrollado en la Universidad de Stanford. GUIDON es un sistema experto para la instrucción. Este instruye a los estudiantes en la selección de terapias adecuadas para pacientes con infecciones bacterianas. Para llevar a cabo el proceso de instrucción, el sistema selecciona un caso y lo resuelve, entonces presenta este caso al estudiante para que éste lo resuelva, analizando las respuestas y preguntas del estudiante durante el proceso de solución. Teniendo en cuenta estos elementos, el sistema compara el procedimiento diagnóstico seguido por el estudiante con el

procedimiento diagnóstico ejecutado por éste en la solución del caso. Las diferencias encontradas son utilizadas por GUIDON para guiar su comportamiento tutorial y sus mecanismos de explicación. El conocimiento experto que utiliza GUIDON para solucionar los casos es proporcionado por el sistema MYCIN. En GUIDON el conocimiento para la instrucción es representado utilizando reglas. [CLA79a, CLA79b]

HEADMED

El sistema HEADMED fue desarrollado en la Universidad de California. HEADMED brinda asesoramiento al médico en el diagnóstico y tratamiento de desórdenes psiquiátricos. Este posee las capacidades de sistema tutorial y de apoyo a la consulta. El conocimiento en HEADMED incluye el diagnóstico diferencial de los principales desórdenes afectivos, esquizofrenia y otros desórdenes orgánicos del cerebro, así como conocimientos acerca de desórdenes en el comportamiento, neurosis e ingestión abusiva de sustancias. En HEADMED el conocimiento es representado utilizando reglas, incluyendo factores de certidumbre para representar la incertidumbre presente en el conocimiento. HEADMED fue escrito en EMYCIN. [HEI78]

HEME

El Sistema Experto HEME fue desarrollado en la Universidad de Cornell. HEME proporciona ayuda al médico en el diagnóstico de enfermedades hematológicas. Para llevar a cabo su razonamiento diagnóstico HEME utiliza el teorema de Bayes para calcular la probabilidad de ocurrencia de cada una de las enfermedades contenidas en su base de conocimientos, dadas las manifestaciones clínicas (signos o síntomas) presentes en el paciente. De esta forma, HEME ejecuta un diagnóstico diferencial, asociando una probabilidad de ocurrencia a cada una de las enfermedades. El conocimiento experto en HEME incluye estimaciones de la frecuencia de ocurrencia de una enfermedad dada, la probabilidad de que un paciente con una enfermedad particular tenga determinada manifestación clínica y la probabilidad de que un paciente que no posee dicha enfermedad tenga determinada manifestación clínica.

HEPAXPERT I

El sistema HEPAXPERT I fue desarrollado en la Escuela de Medicina de la Universidad de Stanford. HEPAXPERT I es un sistema para el análisis interpretativo de la serología de la hepatitis A y B. HEPAXPERT I es un sistema basado en reglas. Su base de conocimientos contiene 13 reglas para la serología de la hepatitis A y 106 reglas para la serología de la hepatitis B. [ADL93]

INTERNIST/CADUCEUS

El sistema CADUCEUS, originalmente llamado INTERNIST fue desarrollado en la Universidad de Carnegie-Mellon a principios de los 70s. CADUCEUS es un sistema consultante médico que realiza diagnósticos en el dominio de la medicina interna. En

CADUCEUS el conocimiento experto está representado por una gran red semántica de relaciones entre síntomas y enfermedades. La base de conocimientos de CADUCEUS ha sido considerada como una de las más grandes entre todos los sistemas expertos conocidos. En 1982 su conocimiento llegaba a representar el 85 % de todo el conocimiento relevante en el dominio de la medicina interna. Muchos especialistas han llegado a considerar que CADUCEUS posee más conocimientos de medicina interna que cualquier experto humano en el dominio, pudiendo diagnosticar correctamente una gran variedad de casos por muy complejos que éstos resulten. [POP77]

IRIS

El sistema IRIS fue desarrollado en la Universidad de Rutgers. IRIS brinda ayuda al médico en el diagnóstico y tratamiento de enfermedades. El sistema proporciona el conjunto de los diagnósticos posibles, a partir de las manifestaciones clínicas aportadas por el médico. Finalmente, selecciona entre los posibles diagnósticos el diagnóstico y tratamiento adecuados para el cuadro clínico referido por el médico. El conocimiento experto en IRIS es representado utilizando una red semántica, la cual define las relaciones entre síntomas, enfermedades y tratamientos. La inferencia es controlada por tablas de decisión asociadas a los nodos de la red semántica y son utilizados factores de certidumbre para representar la incertidumbre presente en el conocimiento experto. [TRI77, TRI78]

MDX

El sistema MDX fue desarrollado en la Universidad Estatal de Ohio. El sistema diagnostica la existencia y las causas de un síndrome del hígado (colestasis). MDX es un sistema experto distribuido, donde cada experto posee conocimientos y experticia correspondiente a una determinada especialidad o subdominio de conocimientos médicos. Los expertos se comunican entre ellos para resolver problemas que requieren de la experticia y conocimientos especiales de otros expertos, estableciéndose de esta forma lazos de cooperación entre éstos. La comunicación entre los expertos es llevada a cabo a través de un pizarrón (base de datos compartida). El conocimiento experto en MDX incluye heurísticas para el diagnóstico y un modelo jerárquico de la estructura conceptual de la colestasis. En MDX el conocimiento es representado utilizando *frames* y reglas.

MECS-AI

El sistema MECS-AI fue desarrollado en la Universidad del Hospital de Tokio. MECS-AI proporciona ayuda al médico en el diagnóstico y tratamiento de enfermedades cardiovasculares y de la tiroides. El conocimiento experto en MECS-AI ha sido obtenido desde expertos médicos en el dominio de enfermedades cardiovasculares y de la tiroides y éste es representado utilizando reglas con encadenamiento hacia atrás.

MEDICO

El sistema MEDICO fue desarrollado en la Universidad de Illinois. MEDICO proporciona consejos al médico oftalmólogo acerca del manejo de enfermedades coriorretinales. La experticia de MEDICO incluye conocimiento clínico general y una base de datos que contiene hechos acerca de los pacientes y eventos. El conocimiento clínico es representado utilizando reglas con encadenamiento hacia adelante. El tratamiento de la incertidumbre es llevado a cabo asociando probabilidades a las reglas, las cuales son proporcionadas por el experto del dominio.

MEDI

El sistema MEDI fue desarrollado en la Universidad de Kaiserslautern. MEDI brinda ayuda al médico en el diagnóstico de enfermedades asociadas al sintoma dolor en el pecho. Para determinar qué enfermedad o enfermedades deben ser consideradas, MEDI se basa en la frecuencia de latidos del corazón y en la presión sanguínea del paciente. El sistema lleva a cabo su razonamiento diagnóstico utilizando encadenamiento de reglas hacia adelante y hacia atrás. El encadenamiento de reglas hacia adelante es utilizado para sugerir enfermedades hipótesis, éste es efectuado a través de reglas que aportan evidencias subjetivas. El encadenamiento de reglas hacia atrás es utilizado para evaluar las reglas que conducen a las hipótesis previamente consideradas, con el fin de confirmar alguna enfermedad o sugerir otras alternativas. Para representar la incertidumbre presente en su conocimiento experto, MEDI utiliza un modelo inspirado en el esquema de factores de certidumbre del sistema MYCIN.

MENINGE

El Sistema Experto MENINGE fue desarrollado en la Facultad de Medicina de la Universidad de Grenoble, Francia. MENINGE es un sistema experto para el diagnóstico y tratamiento de la meningitis en niños. MENINGE discrimina entre meningitis bacterial y viral utilizando una puntuación basada sobre el análisis estadístico de casos clínicos. El tratamiento propuesto por el sistema es sensible y eficiente en más del 90 % de los casos. El conocimiento experto en MENINGE es representado utilizando reglas que no incluyen pesos (factores de certidumbre). MENINGE incluye un modelo lineal, el cual es considerado un subsistema del sistema experto [FRA92]

M-HIP

El sistema M-HIP fue desarrollado en la Universidad de Pavia, Italia. M-HIP es un sistema para el monitoreo de pacientes con trasplante del corazón. M-HIP contiene dos componentes principales: un sistema para el manejo de bases de datos, diseñado para el manejo de datos clínicos del paciente y un sistema basado en el conocimiento (SBC), el cual es capaz de razonar acerca de una gran cantidad de datos. El SBC recupera hallazgos almacenados en la base de datos y crea una taxonomía compleja de objetos.

representando una red temporal de los eventos y episodios importantes registrados en la historia clínica del paciente. A partir de esta representación temporal, el SBC lleva a cabo su razonamiento basado sobre su conocimiento médico. En M-HIP el conocimiento experto es representado utilizando *frames* y reglas. [LAR92]

MI

El sistema MI fue desarrollado en la Universidad de Rutgers. MI proporciona ayuda al médico en el diagnóstico de infarto al miocardio a partir del análisis de la actividad enzimática. Para diagnosticar daños en el corazón, MI se basa en los niveles elevados de ciertas enzimas en la sangre durante un período de varios días. El conocimiento experto en MI ha sido representado utilizando reglas con encadenamiento hacia adelante. Las reglas y hechos son dependientes del tiempo.

MODIS

El sistema MODIS fue desarrollado en Tbilisi, Georgia. MODIS brinda ayuda al médico en el diagnóstico de varios tipos de hipertensión arterial. Para llevar a cabo su razonamiento diagnóstico, el sistema primero obtiene toda la información relevante acerca del estado del paciente (signos, síntomas, resultados de laboratorio) y posteriormente determina todas las enfermedades que se corresponden con las manifestaciones clínicas obtenidas. Cuando son muchas las enfermedades consideradas, el sistema selecciona aquellas que según la experiencia médica resultan ser las más probables. El conocimiento experto en MODIS es obtenido desde médicos expertos en el dominio, éste es representado utilizando una combinación de red semántica, *frames* y reglas. La red semántica contiene *frames* y los *frames* contienen reglas y datos.

MYCIN

El Sistema Experto MYCIN fue desarrollado en la Universidad de Stanford. MYCIN brinda ayuda al médico en el diagnóstico y tratamiento de enfermedades infecciosas. MYCIN diagnostica la causa de la infección (identifica al organismo que produce la infección) y selecciona una terapia antimicrobial apropiada para pacientes con alguna enfermedad infecciosa (bacteremia, meningitis y cistitis infecciosa).

Para diagnosticar la causa de la infección, MYCIN utiliza conocimientos relacionados con los organismos que provocan las infecciones, conjuntamente con datos de interés registrados en la historia clínica del paciente, síntomas y resultados de laboratorio. Para sugerir terapias, MYCIN utiliza los procedimientos seguidos por médicos experimentados en la terapia de enfermedades infecciosas

El conocimiento experto en MYCIN ha sido representado utilizando reglas, mientras que el razonamiento diagnóstico es llevado a cabo a través de un encadenamiento de reglas hacia atrás. MYCIN utiliza factores de certidumbre para representar la

incertidumbre presente en su conocimiento experto y funciones de combinación para llevar a cabo el razonamiento inexacto. [BUC85, SHO76, SHO85]

NEOMYCIN

El sistema NEOMYCIN fue desarrollado en la Universidad de Stanford. NEOMYCIN brinda ayuda al médico en el diagnóstico y tratamiento de meningitis y otras enfermedades similares. El conocimiento experto en NEOMYCIN es obtenido desde el sistema MYCIN. El conocimiento experto es representado utilizando reglas con encadenamiento hacia adelante, de forma tal que el proceso diagnóstico se ejecuta como un razonamiento *forward*. El sistema proporciona explicaciones acerca de su razonamiento diagnóstico. La principal diferencia entre MYCIN y NEOMYCIN es la separación del procedimiento diagnóstico del conocimiento de la enfermedad. [CLAS1]

NEUREX

El sistema NEUREX fue desarrollado en la Universidad de Maryland. NEUREX proporciona ayuda al médico en el diagnóstico de enfermedades del sistema nervioso. A partir de los resultados de exámenes neurológicos realizados a pacientes inconscientes, el sistema localiza el daño del sistema nervioso y clasifica a los pacientes de acuerdo al área o zona dañada. El conocimiento experto en NEUREX es representado utilizando reglas de producción con encadenamiento *forward* y *backward*, para la representación y procesamiento de la incertidumbre es utilizado un mecanismo de factores de certidumbre similar al empleado en MYCIN.

NEUROLOGIST-I

El sistema NEUROLOGIST-I fue desarrollado en la Universidad Estatal de New York, Buffalo. El sistema proporciona ayuda al médico en el diagnóstico de desórdenes neurológicos a partir de la localización de lesiones producidas en el sistema nervioso central. NEUROLOGIST-I primero analiza la información obtenida del paciente (síntomas neurológicos y resultados de exámenes físicos) y posteriormente determina cuales zonas del sistema nervioso pudieran encontrarse alteradas o funcionando mal, a partir de un mapeo entre los síntomas y el estado de la zona afectada. El conocimiento fisiológico en NEUROLOGIST-I es representado mediante un modelo analógico/geométrico del sistema nervioso central.

ONCOCIN

El sistema ONCOCIN fue desarrollado en la Universidad de Stanford. ONCOCIN proporciona ayuda al médico en el tratamiento y manejo de pacientes cancerosos, los cuales se encuentran sometidos a prácticas de quimioterapia, conocidas como protocolos. Para seleccionar una determinada terapia, ONCOCIN toma en cuenta toda la información relacionada con el paciente, incluyendo diagnóstico, tratamientos previos

y resultados de laboratorio. ONCOCIN contiene conocimiento experto acerca de un numeroso grupo de enfermedades de Hodgkin y protocolos de linfoma. El conocimiento en ONCOCIN es representado utilizando reglas con encadenamiento *forward* y *backward*. [SH081]

PUFF

El sistema PUFF fue desarrollado en la Universidad de Stanford. PUFF diagnostica la presencia y gravedad de enfermedades de los pulmones en pacientes, a partir de los resultados obtenidos en pruebas del funcionamiento respiratorio. Los datos interpretados incluyen la capacidad total del pulmón y volumen residual, así como la historia clínica del paciente. PUFF basa sus decisiones sobre el conocimiento que éste posee acerca de los tipos de resultados producidos en las pruebas de funcionamiento respiratorio por diferentes enfermedades o desórdenes respiratorios. El conocimiento experto en PUFF es representado utilizando reglas con encadenamiento hacia atrás.[KUN78]

SYSTEM D

SYSTEM D fue desarrollado en la Universidad de Maryland. SYSTEM D brinda ayuda al médico en el diagnóstico de las causas probables de mareo o vértigos en un paciente. El sistema utiliza la información obtenida acerca del estado del paciente (síntomas referidos, medicamentos empleados) para producir diagnósticos alternativos. El conocimiento experto en SYSTEM D incluye información de casos específicos y experticia en el diagnóstico de causas de vértigos. En SYSTEM D el conocimiento experto es representado utilizando *frames* y la inferencia es llevada a cabo mediante un mecanismo de generación-y-prueba.

THYROID MODEL

THYROID MODEL fue desarrollado en la Universidad de Rutgers. THYROID MODEL brinda ayuda al médico en el diagnóstico de desórdenes de la tiroides (por ejemplo, hipotiroidismo). La experticia del sistema incluye conocimientos para el razonamiento diagnóstico y una estructura taxonómica para la fisiología y patología de la tiroides. El conocimiento experto es representado utilizando reglas con factores de certidumbre y el razonamiento diagnóstico es llevado a cabo a partir de un encadenamiento de reglas hacia adelante. THYROID MODEL posee capacidades de explicación (del por qué determinadas hipótesis son confirmadas o cierta información es requerida).

VM

El sistema VM fue desarrollado en la Universidad de Stanford. VM proporciona ayuda en el diagnóstico y terapia de pacientes recién intervenidos quirúrgicamente, los cuales

se encuentran en una unidad quirúrgica de cuidados intensivos. VM identifica condiciones de alarma, reconoce datos erróneos, caracteriza el estado del paciente y sugiere las terapias adecuadas. VM interpreta continuamente las mediciones efectuadas por un sistema de monitoreo ubicado en la unidad de cuidados intensivos (ritmo del corazón y presión sanguínea), así como los datos correspondientes a equipos de vida artificial. El conocimiento experto en VM ha sido representado utilizando reglas. [FAG79, FAG85]

WHEEZE

El sistema WHEEZE fue desarrollado en la Universidad de Stanford. WHEEZE diagnostica la presencia y gravedad de enfermedades respiratorias a partir de los resultados obtenidos en las pruebas de funcionamiento respiratorio y datos de interés contenidos en la historia clínica del paciente. El conocimiento experto en WHEEZE ha sido obtenido del sistema PUFF y la representación del conocimiento ha sido cambiada de reglas a *frames*. Los *frames* contienen factores de certidumbres para reflejar la incertidumbre presente en el conocimiento. [SMI85]



SEGUNDA PARTE

**SISTEMAS
PIZARRÓN**

El Modelo y la Estructura de Pizarrón para la solución de problemas

El Modelo de Pizarrón

El modelo de pizarrón [ENG88a, NI189a] es un modelo para la solución de problemas, el cual describe la organización del conocimiento del dominio y el comportamiento en la solución de problemas, dentro de una estructura global. Este es un modelo que trata la solución de problemas como un proceso oportunístico e incremental de ensamblaje de una configuración satisfactoria de elementos solución.

En el modelo de pizarrón, el espacio de solución (todas las posibles soluciones parciales y globales del problema) está organizado en una o más jerarquías (dependientes de la aplicación). La información en cada nivel, dentro de una jerarquía, representa soluciones parciales al problema, existiendo en cada nivel un único vocabulario que describe la información. El conocimiento del dominio se encuentra particionado en módulos de conocimiento independientes, los cuales transforman la información de un nivel de jerarquía en información en el mismo nivel o en otros niveles. Los módulos de conocimiento ejecutan estas transformaciones usando procedimientos algorítmicos o reglas heurísticas (IF condición THEN acción)

El razonamiento oportunístico que caracteriza al modelo de pizarrón se manifiesta en los siguientes términos : en cada paso de la solución del problema, se determina dinámicamente qué módulo de conocimiento aplicar, resultando con ello una generación incremental de soluciones parciales. La selección de un módulo de conocimiento

específico se basa sobre el estado de la solución sobre el pizarrón y sobre la existencia de módulos de conocimiento capaces de mejorar el estado actual de la solución.

En cada paso de la solución del problema pueden ser aplicados métodos de razonamiento hacia adelante o hacia atrás, existiendo, además, otros métodos de razonamiento tales como : direccionamiento por los eventos, direccionamiento por las metas, direccionamiento por los planes, direccionamiento por las expectativas, etc.

El modelo de pizarrón consiste de dos componentes principales : las fuentes de conocimiento y la estructura de datos, el pizarrón, tal como se muestra en la figura 4.1.

1. Las fuentes de conocimiento.

El conocimiento necesario para resolver el problema es particionado en fuentes de conocimiento, éstas constituyen módulos separados e independientes.

2. La estructura de datos pizarrón.

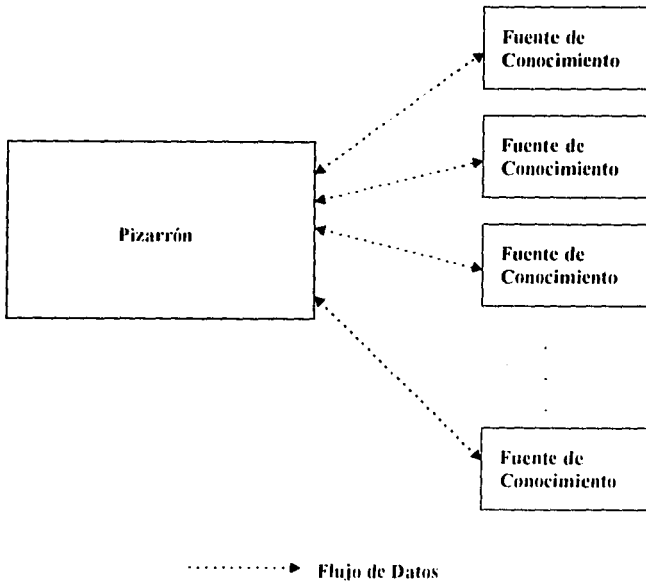


Figura 4.1 El Modelo de Pizarrón.

La información o datos correspondientes al estado de la solución del problema se encuentran almacenados conjuntamente en una base de datos global, el pizarrón.

Las fuentes de conocimiento producen cambios sobre la información almacenada sobre el pizarrón, lo cual conduce, incrementalmente, a la formación de una solución para el problema. El pizarrón es el medio a través del cual se comunican las fuentes de conocimiento.

El modelo de pizarrón no especifica la presencia de un elemento de control. Este sólo describe un comportamiento general en la solución de problemas. El control pudiera encontrarse ubicado en las fuentes de conocimiento, en el pizarrón, o constituir un módulo o elemento separado.

Esta descripción del modelo de pizarrón sólo esboza los principios generales de organización de dicho modelo, sin especificar cómo pudiera ser implementado como una entidad computacional : el modelo de pizarrón constituye una entidad conceptual, no una especificación computacional. Para diseñar y construir un sistema pizarrón, es necesaria una descripción más detallada del modelo de pizarrón, la cual ha de añadir muchos más detalles a los tres componentes principales que debe poseer un sistema pizarrón: fuentes de conocimientos, pizarrón y control, en términos de sus estructuras, funciones y comportamiento.

La Estructura de Pizarrón

La descripción de la estructura de pizarrón [ENG88a, NI189a] es mucho más detallada que la descripción del modelo de pizarrón, aunque menos detallada que la especificación de una aplicación pizarrón. La estructura de pizarrón contiene las descripciones de los componentes del sistema pizarrón que son fundamentales en la construcción de alguna aplicación computacional. El objetivo de la estructura de pizarrón es proporcionar una guía de diseño apropiada para la construcción de sistemas pizarrón en un medio ambiente computacional. La estructura de pizarrón prescribe qué es lo que debe estar presente en un sistema pizarrón. Diferentes aplicaciones pizarrón pudieran requerir diferentes extensiones o modificaciones a la estructura de pizarrón.

La estructura de pizarrón está compuesta por los siguientes elementos básicos :

1. Un conjunto de módulos independientes, llamados fuentes de conocimiento, que contienen conocimiento específico acerca del dominio del sistema.
2. Un pizarrón (o estructura de datos compartida), a través del cual las fuentes de conocimiento se comunican entre sí.

3. Un sistema de control, que determina el orden en que las fuentes de conocimiento operarán sobre las entradas en el pizarrón.

La figura 4.2 muestra la estructura de pizarrón con sus tres elementos básicos.

Las fuentes de conocimiento

Las fuentes de conocimiento son representadas como procedimientos, conjuntos de reglas o aserciones lógicas. Estas poseen un formato condición-acción. La condición describe situaciones en las cuales la fuente de conocimiento puede contribuir a los procesos de solución del problema. Para que la condición de una fuente de conocimiento sea satisfecha, se requiere de una configuración particular de elementos sobre el pizarrón. La acción se vincula a la creación o modificación de elementos solución sobre el pizarrón.

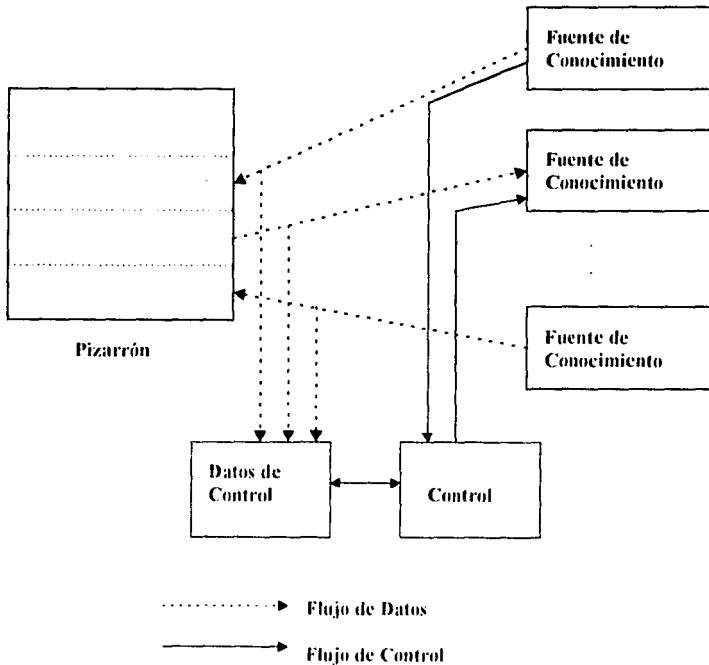


Figura 4.2 La Estructura de Pizarrón.

De forma general, la actividad de las fuentes de conocimiento es direccionada por los eventos. Cada cambio sobre el pizarrón constituye un evento que, en presencia de otra información específica sobre el pizarrón, puede disparar (satisfacer la condición de) una o más fuentes de conocimiento. Cada disparo se inscribe en un único registro de activación de las fuentes de conocimiento (RAFC). Sólo cuando un registro del RAFC es seleccionado por el mecanismo de control, la acción de la fuente de conocimiento asociada a éste es ejecutada, produciendo típicamente nuevos eventos (elementos solución) sobre el pizarrón.

La estructura de pizarrón produce independencia y cooperación simultánea entre las fuentes de conocimiento. Éstas son independientes en el sentido en que ellas mismas no se invocan unas a otras, y comúnmente no tienen conocimiento de la experticia, conocimiento o existencia de las otras fuentes de conocimiento. Son cooperativas en la medida que contribuyen elementos solución al problema compartido.

El pizarrón

Los datos correspondientes al estado de la solución del problema son colocados en una "base de datos global" denominada pizarrón. Las fuentes de conocimiento producen cambios sobre el pizarrón, los cuales conducen incrementalmente a la formación de una solución o conjunto aceptable de soluciones para el problema que se intenta resolver. La interacción entre las fuentes de conocimiento es indirecta y ésta sólo toma lugar a través de los cambios producidos sobre el pizarrón.

El pizarrón consiste de elementos solución pertenecientes al espacio solución del problema. Estos elementos solución pueden ser datos de entrada, soluciones parciales, soluciones alternativas o soluciones finales.

Los elementos solución se encuentran organizados sobre el pizarrón en varios niveles de análisis o abstracción. Los diferentes niveles de abstracción representan la solución del problema en diferente cantidad de detalle.

El sistema de control

Debido a que las fuentes de conocimiento responden oportunísticamente a cambios sobre el pizarrón, es necesario un mecanismo de control que monitoree estos cambios y decida, en cada momento, cuáles acciones se deben tomar. El mecanismo de control utiliza varios tipos de información de gran utilidad, que se puede encontrar sobre el pizarrón o fuera de éste, para así determinar el Foco de Atención.

El Foco de Atención indica que será lo próximo a ser procesado. Éste puede contener información referente a una fuente de conocimiento (por ejemplo, la próxima fuente de conocimiento a activar); a un elemento solución sobre el pizarrón (por ejemplo, cuál elemento solución separar para dedicarse a éste próximamente) o a una combinación de ambos (cuáles fuentes de conocimiento aplicar a cuáles elementos solución). La gran mayoría de los sistemas utilizan sólo una de estas tres posibilidades.

Durante el proceso de solución del problema, las fuentes de conocimiento y el mecanismo de control ejecutan sus actividades en la siguiente secuencia iterativa :

1. Ocurrencia de un evento sobre el pizarrón : una fuente de conocimiento ha generado o modificado un elemento solución sobre el pizarrón. Esta acción también ha producido un registro en una estructura de datos con información de control.
2. Cada una de las fuentes de conocimiento informa la contribución que puede hacer al nuevo estado de solución
3. Teniendo en cuenta la información contenida en 1 y 2, el mecanismo de control selecciona un Foco de Atención.
4. El Foco de Atención es preparado para su ejecución teniendo en cuenta lo siguiente :
 - ◆ Si el Foco de Atención es una fuente de conocimiento : un elemento solución del pizarrón (o un conjunto de elementos solución) es seleccionado para servir de contexto a su invocación.
 - ◆ Si el Foco de Atención es un elemento solución del pizarrón : es seleccionada una fuente de conocimiento que sea capaz de producir este elemento solución.
 - ◆ Si el Foco de Atención es una fuente de conocimiento y un elemento solución : la fuente de conocimiento se encuentra lista para su ejecución, siendo ésta ejecutada conjuntamente con el elemento solución de contexto.

Comúnmente, una fuente de conocimiento es la encargada de indicar cuándo los procesos de solución del problema han terminado. Esto pudiera ocurrir debido a que una solución aceptable ha sido hallada o debido a la falta de información o conocimientos que impide que el sistema continúe su desempeño.

El comportamiento en la solución de problemas de un sistema pizarrón está determinado por el desempeño del sistema de control. La selección de una región particular del pizarrón y de una fuente de conocimiento particular para operar sobre esta región determinan el comportamiento del sistema en la solución de problemas.

La Arquitectura de Pizarrón

La arquitectura de pizarrón [ENG88a, HAY85, NI189a] es un término genérico el cual se refiere al diseño de un sistema pizarrón o a una parte de un sistema. La arquitectura de pizarrón fue desarrollada por primera vez por el sistema para la comprensión del lenguaje HEARSAY II, usada en una gran variedad de otros dominios de problemas y resumida en muchos medios ambientes para la construcción de sistemas.

Dentro de la gran variedad de arquitecturas de pizarrón propuestas y desarrolladas hasta la fecha [ENG88a, NI189a], se encuentran tres arquitecturas que reflejan enfoques muy diferentes dentro del paradigma de los sistemas pizarrón :

- ◆ Pizarrón Centralizado : Arquitectura de pizarrón del sistema HEARSAY II.
- ◆ Pizarrón Distribuido : Arquitectura de pizarrón del sistema DVMT.
- ◆ Pizarrón del Dominio y Pizarrón del Control : Arquitectura de pizarrón para el control.

Las ideas básicas correspondientes a estas tres arquitecturas han servido de escenario para el desarrollo de una gran variedad de sistemas pizarrón y son tomadas como marco contextual para el desarrollo de las aplicaciones pizarrón que se presentan en la tercera parte de este material.

La arquitectura de pizarrón del sistema HEARSAY II

El dominio de HEARSAY II

El sistema para la comprensión del lenguaje Hearsay II [ENG88b, ERM80, ERM88, LES88a, NI189a] fue desarrollado en la Universidad de Carnegie-Mellon en 1976. La

La Arquitectura de Pizarrón

La arquitectura de pizarrón [ENG88a, HAY85, NII89a] es un término genérico el cual se refiere al diseño de un sistema pizarrón o a una parte de un sistema. La arquitectura de pizarrón fue desarrollada por primera vez por el sistema para la comprensión del lenguaje HEARSAY II, usada en una gran variedad de otros dominios de problemas y resumida en muchos medios ambientes para la construcción de sistemas.

Dentro de la gran variedad de arquitecturas de pizarrón propuestas y desarrolladas hasta la fecha [ENG88a, NII89a], se encuentran tres arquitecturas que reflejan enfoques muy diferentes dentro del paradigma de los sistemas pizarrón :

- ◆ Pizarrón Centralizado : Arquitectura de pizarrón del sistema HEARSAY II.
- ◆ Pizarrón Distribuido : Arquitectura de pizarrón del sistema DVMT.
- ◆ Pizarrón del Dominio y Pizarrón del Control : Arquitectura de pizarrón para el control.

Las ideas básicas correspondientes a estas tres arquitecturas han servido de escenario para el desarrollo de una gran variedad de sistemas pizarrón y son tomadas como marco contextual para el desarrollo de las aplicaciones pizarrón que se presentan en la tercera parte de este material.

La arquitectura de pizarrón del sistema HEARSAY II

El dominio de HEARSAY II

El sistema para la comprensión del lenguaje Hearsay II [ENG88b, ERM80, ERM88, LES88a, NII89a] fue desarrollado en la Universidad de Carnegie-Mellon en 1976. La

meta de este proyecto fue el reconocimiento de pronunciaciones (voz) desde un vocabulario limitado en tiempo casi real. Hearsay II era capaz de reconocer lenguaje correspondiente a un vocabulario de 1000 palabras y lograba interpretaciones correctas para un 90% de las oraciones de prueba.

Para lograr el desempeño eficiente de Hearsay II en un dominio tan complejo como la comprensión del lenguaje, fue necesario vincular técnicas generales de inteligencia artificial con conocimientos específicos de acústica y lingüística. En Hearsay II, el razonamiento simbólico constituyó la base para el procesamiento de señales.

Debido al elevado grado de complejidad y dificultad que caracteriza el problema de la comprensión del lenguaje, un resolvidor de problemas en este dominio requiere de varios tipos de conocimiento para poder desempeñar un trabajo efectivo. El sistema debe recolectar datos, analizar datos, proponer metas para guiar los procesos de búsqueda, producir inferencias que permitan la creación de nuevas hipótesis, y decidir cuándo detener la búsqueda (si una buena solución ha sido alcanzada o los recursos se han agotado).

El problema de la comprensión del lenguaje en Hearsay II ha sido estructurado como un espacio de búsqueda en el cual los resolvidores de problemas efectúan la búsqueda de una posible solución. El espacio de búsqueda está formado por el conjunto de todas las interpretaciones parciales y completas de una señal acústica de entrada. La meta del sistema es encontrar la mejor interpretación completa, la cual debe maximizar alguna función de evaluación basada sobre conocimientos acerca de acústica, fonética, vocabulario, gramática, semántica y discurso.

El modelo HEARSAY II

La meta principal de un sistema de interpretación es la construcción de la interpretación completa más confiable a partir de los datos de entrada. En el sistema Hearsay II, una interpretación es construida a partir de la combinación de interpretaciones parciales producidas por las diferentes fuentes de conocimiento. Cada fuente de conocimiento posee una estructura condición-acción. La condición prescribe las situaciones bajo las cuales la fuente de conocimiento pudiera contribuir en los procesos de solución del problema, mientras que la acción estipula en qué consiste esta contribución y cómo es integrable al estado actual de la solución del problema.

Las fuentes de conocimiento han sido concebidas para ejecutar una variedad de funciones tales como extracción de parámetros acústicos, clasificación de segmentos acústicos en clases fonéticas, reconocimiento de palabras, creación de frases y generación y evaluación de predicciones para palabras o sílabas no detectadas.

La interacción de las fuentes de conocimiento se lleva a cabo de una forma iterativa direccionada por los datos, donde cada iteración abarca la creación de una hipótesis,

una interpretación posible de alguna parte de la solución y la prueba de su plausibilidad. Para ejecutar estas acciones, las fuentes de conocimiento utilizan su conocimiento "a priori" acerca del problema y lo aplican a hipótesis que han sido previamente creadas sobre el pizarrón por otras fuentes de conocimiento para generar nuevas hipótesis. De esta forma, las fuentes de conocimiento extienden interpretaciones parciales existentes con más información, reduciendo así la incertidumbre de la interpretación. El proceso concluye cuando se ha generado una hipótesis consistente que satisface los requerimientos de la solución global.

Existen dos factores principales que pudieran hacer que una fuente de conocimiento creara hipótesis incorrectas. El primero de ellos es el conocimiento que posee la fuente de conocimiento acerca del dominio del problema o de la parte del dominio del problema que es de su competencia. El segundo factor se refiere a la información de entrada que utiliza la fuente de conocimiento para aplicar su conocimiento, esta información pudiera incluir hipótesis previamente generadas que contienen errores o son incompletas. Para evitar lo anterior, una fuente de conocimiento genera varias hipótesis alternativas para cada parte específica del problema y asocia a cada hipótesis creada un grado de creencia o razón de credibilidad, el cual es su estimado de la probabilidad de que la hipótesis sea correcta.

El conjunto de todas las interpretaciones parciales creadas por las fuentes de conocimiento define el espacio de búsqueda para la solución del problema. Debido a que cada interpretación parcial pudiera dar lugar a múltiples extensiones, una explosión combinatoria pudiera ocurrir. Para evitar esto, en cada paso de la búsqueda, un subconjunto de interpretaciones parciales existentes es seleccionado para su extensión. Las interpretaciones parciales que resulten extendidas compiten con otras interpretaciones parciales previamente generadas para ser seleccionadas como buenas.

La arquitectura de HEARSAY II

La figura 5.1 muestra un esquema simplificado de la arquitectura del sistema Hearsay II. Los elementos que pueden apreciarse en esta arquitectura de pizarrón son los siguientes:

- ◆ la base de datos compartida (El pizarrón)
- ◆ las fuentes de conocimientos
- ◆ el sistema de control, formado por :
 - . el despachador
 - . la cola de despacho
 - . la base de datos Foco de Control
 - . el monitor del pizarrón

El pizarrón

El pizarrón en Hearsay II se encuentra subdividido en diferentes niveles de información, cada uno de los cuales corresponde a un tipo diferente de representación del espacio del problema (sílabas, palabras, frases, etc.), tal como se muestra en la figura 5.2. Los principales elementos sobre el pizarrón son las hipótesis, cada hipótesis corresponde a un determinado nivel del pizarrón y éstas contienen información adicional, incluyendo el grado de creencia o razón de credibilidad asociado a ellas.

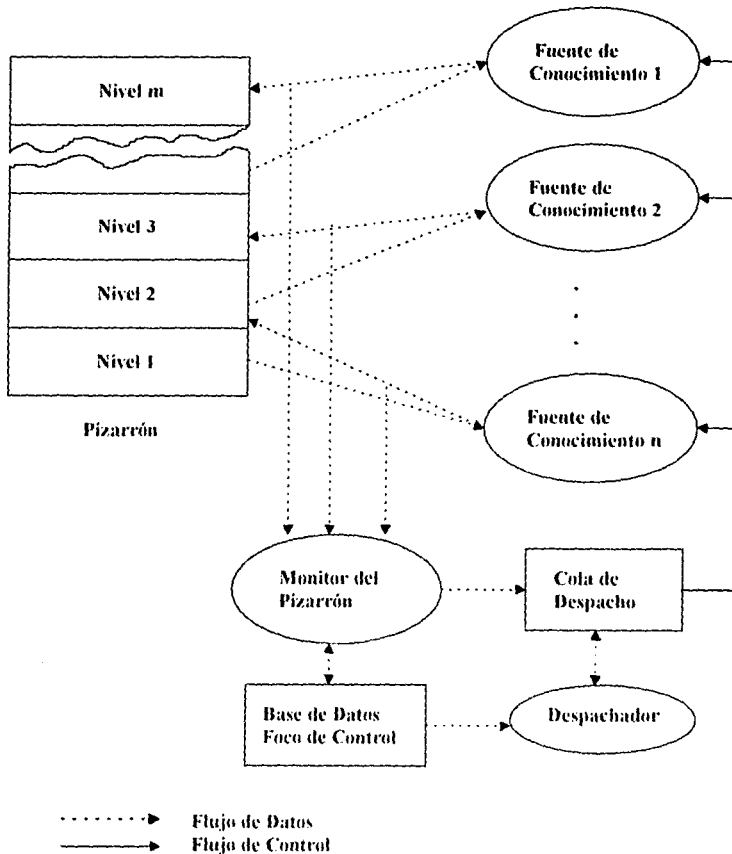


Figura 5.1 Arquitectura del sistema HEARSAY II.

La secuencia de niveles sobre el pizarrón forman una estructura jerárquica, donde las hipótesis correspondientes a cada nivel pueden ser descritas como abstracciones de las hipótesis correspondientes al nivel inmediato inferior o a otros niveles inferiores. Todas las hipótesis pertenecientes a un determinado nivel forman el espacio de búsqueda para las fuentes de conocimiento que operan en ese nivel.

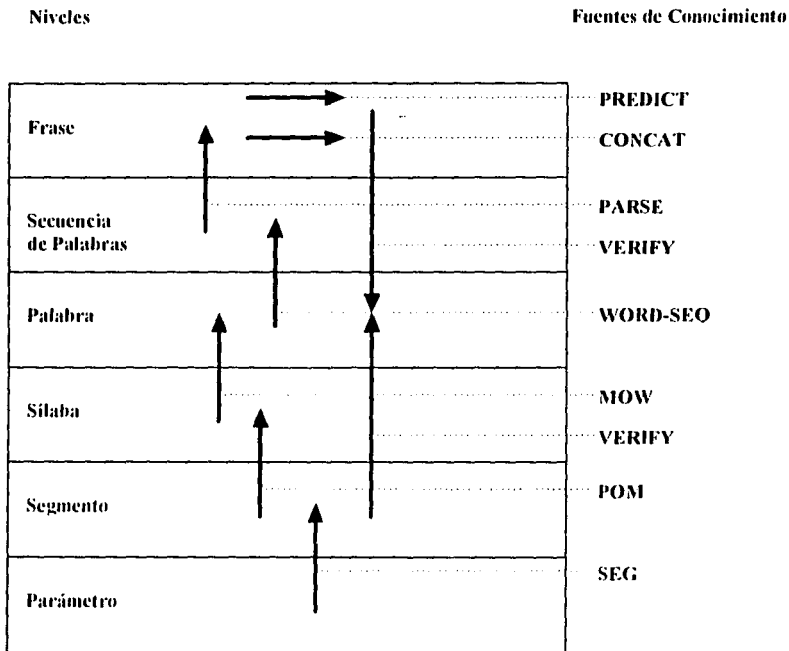


Figura 5.2 Niveles del Pizarrón y Fuentes de Conocimiento en HEARSAY II.

Las fuentes de conocimiento

En Hearsay II la estructura de las fuentes de conocimiento responden a un formato condición-acción, esto permite implementar (para las fuentes) una activación direccionada por los datos. Cuando la condición de una fuente de conocimiento es apareada por determinadas hipótesis en un nivel del pizarrón, entonces es creado un

registro de activación de la fuente de conocimiento (RAFC). La acción de la fuente de conocimiento es ejecutada cuando el RAFC creado por ésta es seleccionado por el despachador. Las fuentes de conocimiento de HEARSAY II pueden ser apreciadas en la figura 5.2, mientras que la tabla 5.1 brinda su descripción funcional.

Descripción funcional de las fuentes de conocimiento en HEARSAY II

SEG : digitaliza la señal de entrada y genera segmentos etiquetados.
POM : crea hipótesis clase-silaba a partir de los segmentos
MOW : crea hipótesis palabras desde las clases de sílabas
WORD-SEQ : crea hipótesis secuencia-palabras que representan frases potenciales desde hipótesis palabras
PARSE : crea una hipótesis frase desde una secuencia de palabras
VERIFY : valora la consistencia entre hipótesis segmento y un par de palabra-frase contiguas
PREDICT : predice todas las posibles palabras que pudieran preceder o suceder sintácticamente a una frase dada
CONCAT : crea una hipótesis frase desde un par de palabra-frase contiguas verificadas

Tabla 5.1

El sistema de control

La base de datos Foco-de-Control. La base de datos foco-de-control contiene meta-información acerca del estado de la actividad del sistema en la solución de problemas. La meta-información incluye las mejores hipótesis que se encuentran actualmente sobre el pizarrón, el tiempo que ha transcurrido desde que estas hipótesis fueron creadas, así como otra información de control relevante. La meta-información es utilizada para calcular el impacto que tendrá la nueva información a generar (RAFCs) sobre el estado actual de la solución del problema, tomando como base la credibilidad, alcance y diagnosticidad de esta nueva información

La cola de despacho Contiene activaciones pendientes de las fuentes de conocimiento (RAFCs). Un RAFC posee además información relevante acerca de la hipótesis a generar (por ejemplo, credibilidad, alcance, diagnosticidad)

El despachador El despachador calcula una prioridad para cada una de las activaciones pendientes en la cola de despacho y selecciona para su ejecución la activación con la mayor prioridad. El cálculo de la prioridad es una medida para estimar el impacto de la nueva información a ser generada sobre el estado actual de la solución del problema. El impacto de la información constituye una medida del grado de

reducción de la incertidumbre acerca de la interpretación o de reducción del número de interpretaciones que se encuentran compitiendo.

El monitor del pizarrón. Cuando una fuente de conocimiento es activada (su condición es apareada por hipótesis en el pizarrón), el monitor del pizarrón crea un RAFC y almacena éste en la cola de despacho. El monitor del pizarrón es también utilizado para implementar la activación direccionada por los datos de las fuentes de conocimiento.

La arquitectura de pizarrón del sistema DVMT

Un Medio Ambiente Distribuido para el Monitoreo de Vehículos (DVMT)

DVMT [LES88b] es un medio ambiente de investigaciones, construido para la evaluación empírica de diseños alternativos en redes cooperativas para la solución de problemas distribuidos. DVMT simula una red de nodos resolvedores de problemas, cada nodo en la red aplica su conocimiento para el procesamiento de señales a partir de datos sentidos acústicamente, con el objetivo de identificar, localizar y rastrear patrones de vehículos en movimiento a través de un espacio bidimensional.

El diseño parametrizado de DVMT hace posible modelar una amplia variedad de situaciones cooperativas en la solución de problemas distribuidos, a partir de la variación de los siguientes parámetros :

- ◆ la cantidad de fuentes de conocimiento disponibles en cada nodo.
- ◆ la exactitud o precisión de las fuentes de conocimiento individuales.
- ◆ las características de los vehículos y sensores
- ◆ la configuración de los nodos y las características de los canales de comunicación
- ◆ las responsabilidades en la solución de problemas y comunicación de cada nodo.
- ◆ las relaciones de autoridad entre los nodos.

La arquitectura de los nodos en DVMT

DVMT simula una red de nodos Hearsay II trabajando conjuntamente sobre las tareas de monitoreo de vehículos. En DVMT, cada nodo es arquitectónicamente un sistema Hearsay II, con pizarrón, fuentes de conocimiento y control apropiados para el monitoreo de vehículos.

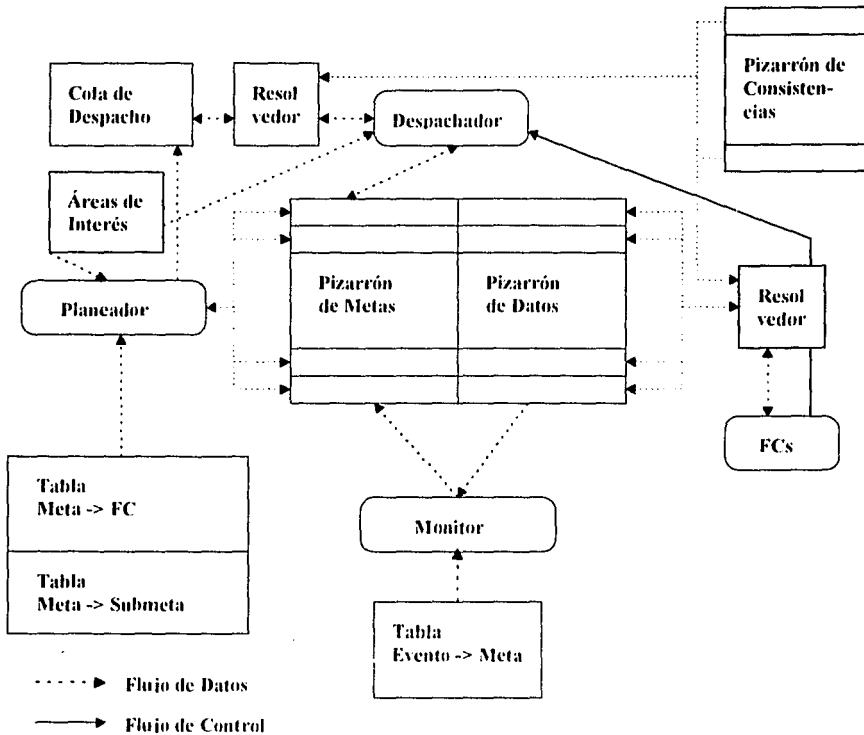


Figura 5.3 Arquitectura de un nodo DVMT

La arquitectura Hearsay II de los nodos ha sido ampliada para incluir fuentes de conocimientos para la comunicación de hipótesis y metas entre los nodos, un módulo de planeación, un pizarrón de metas y un pizarrón de control meta-nivel. Además, a cada nodo se ha adicionado un pizarrón de consistencias para medir la ejecución de éstos y la ejecución global de la red, y resolutores para modificar la inteligencia de las fuentes de

conocimiento y el despachador. La figura 5.3 muestra la arquitectura de un nodo en DVMT.

El pizarrón de datos

En DVMT, las hipótesis son organizadas sobre un pizarrón dividido en cuatro niveles de abstracción : señal, grupo, vehículo y patrón, tal como se puede apreciar en la figura 5.4.

Nivel Señal. Es el más bajo nivel de abstracción, las hipótesis pertenecientes a este nivel constituyen las entradas básicas del nodo y corresponden a la salida del análisis a bajo nivel de los datos sensados. Cada señal incluye la frecuencia, la posición aproximada, el intervalo de tiempo de detección y el grado de creencia de la señal acústica, así como la identidad del sensor que detectó la señal.

Nivel Grupo. En este nivel se encuentran los grupos de señales. Un grupo es una colección de señales relacionadas armónicamente. Un grupo incluye la frecuencia fundamental de las señales relacionadas, la posición aproximada, el intervalo de tiempo y el grado de creencia asociado al grupo, el cual es una función de los grados de creencias de las señales relacionadas en el grupo.

Nivel Vehículo. Las hipótesis en este nivel representan vehículos. Un vehículo consiste de una colección de grupos asociados con un vehículo particular. Un vehículo incluye la identidad del vehículo, el intervalo de tiempo, la posición aproximada y el grado de creencia asociado.

Nivel Patrón. Las hipótesis en este nivel representan patrones de vehículos. Un patrón es una colección de tipos de vehículos particulares con una relación espacial particular entre ellos. Un patrón incluye la identidad del patrón, su intervalo de tiempo, la posición aproximada y el grado de creencia asociado. Un patrón puede estar constituido por un único vehículo.

Cada uno de estos cuatro niveles de abstracción es dividido en dos niveles : un nivel para las hipótesis de localización y un nivel para las hipótesis de ruta. Una hipótesis de localización representa un único evento sobre un intervalo particular de tiempo. Una hipótesis de ruta representa una secuencia de eventos sobre un número de intervalos de tiempo contiguos.

Las fuentes de conocimiento

DVMT presenta dos clases de fuentes de conocimiento : las fuentes de conocimiento del dominio o de procesamiento y las fuentes de conocimiento de comunicación.

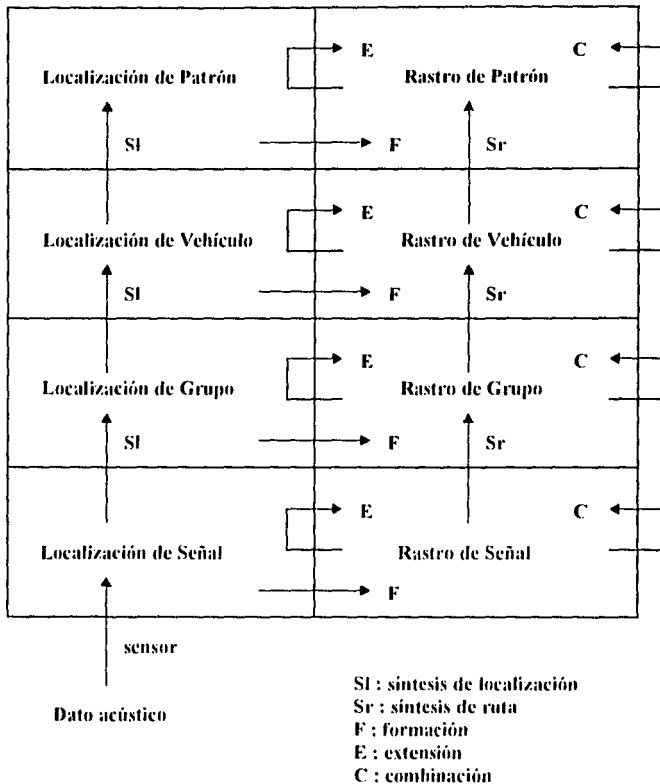


Figura 5.4 Niveles de abstracción y fuentes de conocimiento

Las fuentes de conocimiento del dominio. Al igual que en el sistema Hearsay II, las fuentes de conocimiento en DVMT responden a un formato condición-acción. Cuando la condición de una fuente de conocimiento es apareada por hipótesis específicas en algún nivel del pizarrón, un registro de activación es creado para esta fuente de conocimiento. Cada nodo mantiene una cola de registros de activación de las fuentes de conocimiento pendientes y en algún momento dado, debe valorar los registros para decidir cuál será la próxima fuente de conocimiento invocada

En DVMT, las fuentes de conocimiento del dominio ejecutan las tareas básicas de extensión y refinamiento de hipótesis (soluciones parciales) correspondientes a la

solución del problema. En cada pizarrón local de DVMT operan fuentes de conocimiento que ejecutan las siguientes tareas :

- ◆ Síntesis de localización : Es la generación de nuevas hipótesis de localización en un nivel del pizarrón a partir de hipótesis de localización correspondientes al nivel inferior.
- ◆ Síntesis de rutas : Es la generación de nuevas hipótesis de rutas en un nivel del pizarrón a partir de hipótesis de rutas correspondientes al nivel inferior.
- ◆ Formación de rutas : Es la formación de una hipótesis ruta a partir de la combinación de dos hipótesis de localización combinables correspondientes a instantes de tiempos adyacentes.
- ◆ Extensión de rutas : Es la extensión de una hipótesis ruta a partir de la combinación de ésta con una hipótesis de localización compatible.
- ◆ Unión localización-ruta : Es la combinación de una hipótesis de localización con una hipótesis ruta compatible que comience o termine en un instante de tiempo adyacente.
- ◆ Combinación de Rutas : Es la combinación de dos hipótesis rutas superpuestas o contiguas en una sola hipótesis ruta en el mismo nivel de abstracción.
- ◆ Sensado : Es la creación de hipótesis de localización de señales a partir de los datos percibidos.

Las fuentes de conocimiento correspondientes a cada una de estas tareas pueden ser apreciadas en la figura 5.4.

Las fuentes de conocimiento de comunicación. Las fuentes de conocimiento de comunicación permiten el intercambio de hipótesis y metas entre los nodos. Al igual que las fuentes de conocimiento del dominio, éstas trabajan de forma independiente y asincrónica. En cada nodo de DVMT existen fuentes de conocimiento que ejecutan las siguientes actividades de comunicación :

- ◆ Enviar hipótesis : Es la transmisión de hipótesis creadas sobre el pizarrón de datos del nodo a otros nodos, sobre la base de las características de la hipótesis (nivel, instante de tiempo, localización y creencia).
- ◆ Recibir hipótesis : Es la ubicación de nuevas hipótesis sobre el pizarrón de datos del nodo, las cuales han sido creadas por otros nodos y transmitidas a éste. Cada hipótesis recibida en el nodo es filtrada en base a sus características (nivel, instante de tiempo, localización y creencia) para determinar el interés del nodo acerca de la hipótesis.
- ◆ Enviar metas : Es la transmisión de metas creadas sobre el pizarrón de metas del nodo a otros nodos, sobre la base de las características de la meta (nivel, instante de tiempo, región y categoría).
- ◆ Solicitud de ayuda para la generación de metas : Es la transmisión de metas que no pueden ser satisfechas localmente por el nodo hacia otros nodos.

- ◆ **Recibir metas :** Es la ubicación de nuevas metas sobre el pizarrón de metas del nodo, las cuales han sido creadas por otros nodos y transmitidas a éste. Las metas recibidas en el nodo son filtradas en base a sus características (nivel, instante de tiempo, región y categoría) para determinar el interés del nodo acerca de las mismas.
- ◆ **Requerimientos de información acerca de metas :** Es la transmisión de hipótesis creadas sobre el pizarrón, en respuesta a información recibida desde el nodo para la solicitud de metas.

El pizarrón de metas

El pizarrón de metas posee una estructura similar al pizarrón de datos, sólo que en lugar de hipótesis, las unidades básicas sobre el pizarrón son metas. Cada meta sobre el pizarrón representa la posibilidad de crear o extender una hipótesis con determinados atributos sobre el pizarrón de datos.

El monitor del pizarrón

El monitor del pizarrón es el encargado de crear las metas sobre el pizarrón de metas. Estas metas representan la intención del nodo de resumir o extender hipótesis particulares. Sobre el pizarrón de metas también pueden ser ubicadas metas recibidas desde otros nodos.

El planeador

Un plan representa la intención de alcanzar una meta de elevado nivel a partir de la ejecución de una secuencia de actividades. En DVMT, el planeador desarrolla planes para alcanzar metas de alto nivel, las cuales son insertadas por éste en el pizarrón de metas, e identifica cuáles fuentes de conocimiento llevarán a cabo estos planes.

La cola de despacho

Contiene activaciones pendientes de las fuentes de conocimiento que han logrado satisfacer su condición sobre hipótesis del pizarrón. La cola de despacho posee información relevante acerca de cada nueva hipótesis a generar, esta información es utilizada por el despachador para decidir que fuente de conocimiento será ejecutada.

El despachador

El despachador decide cómo deben ser distribuidos los recursos de comunicación y procesamiento del nodo, a partir de la relación entre los registros de activación de las fuentes de conocimiento y las metas localizadas sobre el pizarrón de metas

El pizarrón de consistencias

El pizarrón de consistencias mide la eficiencia del nodo y la eficiencia global de la red. Este permite medir los estados intermedios de procesamiento del nodo. La exactitud de las hipótesis es obtenida desde el pizarrón de consistencias, siendo precomputada a partir de la simulación de datos de entrada. El pizarrón de consistencias determina las interpretaciones que deberían estar en cada nivel de información del pizarrón de datos, de haber trabajado el sistema con conocimientos correctos. Esto se logra a partir de la simulación de datos de entrada.

Los resolvedores

En DVMT, la modificación del poder de una fuente de conocimiento se logra mediante la separación de cada fuente de conocimiento en dos etapas : un generador de candidatos y un resolvidor. La etapa generador de candidatos produce hipótesis plausibles para la salida de las fuentes de conocimiento y asigna a cada hipótesis un valor de creencia tentativo. La etapa resolvidor usa información proporcionada por el pizarrón de consistencias para modificar mínimamente los valores de creencias iniciales asociados a estas hipótesis plausibles, con el objetivo de alcanzar una fuente de conocimiento con un poder deseado. Las hipótesis con los valores de creencias alterados más elevados, son usadas en la etapa resolvidor como las hipótesis correspondientes a la salida actual de las fuentes de conocimiento.

Las áreas de interés

Las áreas de interés son estructuras de datos que residen en el pizarrón de control meta-nivel y son utilizadas para implementar configuraciones particulares de la red y políticas de coordinación. Un área de interés es una lista de regiones del pizarrón de datos o de metas. Para cada área de interés de procesamiento local existe un único parámetro asociado : un valor de peso que especifica la importancia de la ejecución del procesamiento local dentro del área de interés.

Para cada nodo en DVMT existen las siguientes áreas de interés :

- ◆ Área de interés de procesamiento local.
- ◆ Área de interés de transmisión de hipótesis.
- ◆ Área de interés de recepción de hipótesis.
- ◆ Área de interés de transmisión de metas.
- ◆ Área de interés de solicitud de ayuda para la generación de metas.
- ◆ Área de interés de recepción de metas

La actividad de los nodos en la solución de problemas

A partir de un medio ambiente particular para la solución de problemas, la actividad de los nodos en DVMT puede ser descrita en los siguientes términos : cada nodo comienza transformando los datos sensados en un conjunto de hipótesis de localización de señal, usando para esto la fuente de conocimiento sensor. Para cada nueva hipótesis de localización de señal, el nodo genera metas que permitan refinar esta hipótesis y crea registros de activación para las fuentes de conocimiento que puedan alcanzar estas metas. Cuando todas las hipótesis de localización de señal han sido creadas, el nodo entonces selecciona un registro de activación para que sea invocado. La ejecución de un registro de activación pudiera causar la creación de nuevas hipótesis sobre el pizarrón, esto implicaría la generación de nuevas metas, las cuales a su vez causarían la creación de nuevos registros de activación. Nuevamente el nodo selecciona un registro de activación para su invocación y el ciclo se repite.

Los nodos ejecutan actividades de bajo nivel en la solución de problemas, creando hipótesis para la localización de grupos e hipótesis para la localización de vehículos, mientras que las actividades de alto nivel ejecutadas se corresponden a la formación, extensión y fusión de hipótesis de rutas de vehículos e hipótesis de rutas de patrones. No todos los nodos en la red participan en la generación de una solución, debido a que éstos tienen diferentes conocimientos y responsabilidades. Cuando un nodo encuentra la solución, éste debe transmitir sus resultados a los otros nodos de la red antes de concluir sus actividades. Una vez que esta información ha sido recibida por los restantes nodos, éstos deben concluir sus actividades en la solución del problema. Las actividades de la red concluyen en un intervalo de tiempo posterior al momento en que un nodo encontró la solución del problema.

La arquitectura de Pizarrón para el Control

El control

Cuando un sistema inteligente se enfrenta ante la solución de un problema de su dominio, ejecuta una serie de acciones orientadas a encontrar la solución del problema. En cada punto en el proceso de solución del problema, varias acciones pudieran resultar posibles. El problema del control determina, entre todas las acciones posibles, cuál debe ser ejecutada por el sistema en cada punto en el proceso de solución del problema.

Cuando un sistema inteligente soluciona el problema del control, éste decide qué problemas intentará resolver, qué conocimiento utilizar, qué métodos y estrategias de la solución de problemas aplicar, cómo evaluar soluciones alternativas al problema, cómo saber que un problema particular ha sido resuelto y cuándo abandonar un problema para seleccionar otro.

La arquitectura de pizarrón para el control [HAY85, HAY88] extiende y elabora la arquitectura de pizarrón del sistema Hearsay II con el objetivo de solucionar el problema del control. En la solución del problema del control, la arquitectura de pizarrón determina su comportamiento cognitivo sobre la base de las siguientes metas :

- ◆ Hacer explícitas las decisiones de control que resuelven el problema de control.
- ◆ Decidir qué acciones ejecutar, teniendo en cuenta las acciones deseables y las acciones posibles.
- ◆ Adoptar heurísticas de control apropiadas para cada situación en la solución del problema.
- ◆ Adoptar heurísticas de control que focalicen sobre aquellas acciones que resulten beneficiosas en la situación actual de la solución del problema.
- ◆ Adoptar, retener y descartar heurísticas individuales de control en respuesta a situaciones dinámicas en la solución del problema.
- ◆ Decidir cómo integrar múltiples heurísticas de control de importancia variable.
- ◆ Planear dinámicamente secuencias estratégicas de acciones.
- ◆ Razonar acerca de prioridades relativas de las acciones del dominio y del control.

Componentes de la arquitectura

La arquitectura de pizarrón para el control define los siguientes componentes :

- ◆ Un pizarrón explícito para el dominio
- ◆ Un pizarrón explícito para el control
- ◆ Fuentes de conocimientos explícitas para el dominio
- ◆ Fuentes de conocimientos explícitas para el control
- ◆ Un mecanismo de despacho adaptativo para controlar los registros de activación de las fuentes de conocimientos del dominio y del control

El pizarrón del dominio

Al igual que en Hearsay II, el pizarrón del dominio registra elementos solución para el problema actual del dominio. Los niveles de abstracción, intervalos de solución y

vocabulario del pizarrón del dominio son específicos del dominio y determinados por el diseñador del sistema o de la aplicación.

El pizarrón del control

El pizarrón de control registra elementos solución para el problema del control. Estos elementos solución son decisiones acerca del comportamiento actual, factible y deseable del propio sistema. Cada decisión de control es representada como una estructura de datos con los atributos definidos en la tabla 5.2.

Atributos básicos de las decisiones de control
Nivel y número de identificación
Acción prescrita
Condición de expiración
Importancia de la meta (un valor de 0 a 1)
Motivo para la meta
El RAFC que creó la decisión
Decisión de disparo
Papel en el plan de control
Función en el plan de control
Primer ciclo operativo
Ultimo ciclo operativo

Tabla 5.2

Un conjunto de decisiones relacionadas sobre el pizarrón de control constituyen un plan de control parcial.

Los niveles de abstracción, intervalos de solución y vocabulario del pizarrón de control son independientes del dominio y definidos por la propia arquitectura.

El pizarrón de control define los siguientes niveles de abstracción :

- Nivel Problema : El problema que el sistema ha decidido resolver.
- Nivel Estrategia : Plan general de acciones encaminadas a la solución del problema.
- Nivel Foco : Metas locales para la solución del problema.
- Nivel Política : Criterio global de despacho.
- Nivel Registros-Pendientes : Conjuntos de RAFCs pendientes.
- Nivel Acción-Seleccionada : RAFC seleccionado para su ejecución

Los niveles de abstracción del pizarrón de control representan diferentes categorías de las decisiones de control. Las decisiones que se localizan en los niveles Problema, Estrategia, Foco y Política describen acciones deseables. Estas acciones determinan cuáles de las heurísticas de control del sistema deben operar durante intervalos de

tiempo particulares en la solución del problema. Las acciones ubicadas en el nivel Registros-Pendientes describen acciones posibles, mediante la identificación de todos los RAFCs que pueden ser seleccionados en cada ciclo de la solución del problema. Las acciones localizadas en el nivel Acción-Seleccionada describen las acciones que han sido seleccionadas para su ejecución sobre cada ciclo en la solución del problema.

NIVELES

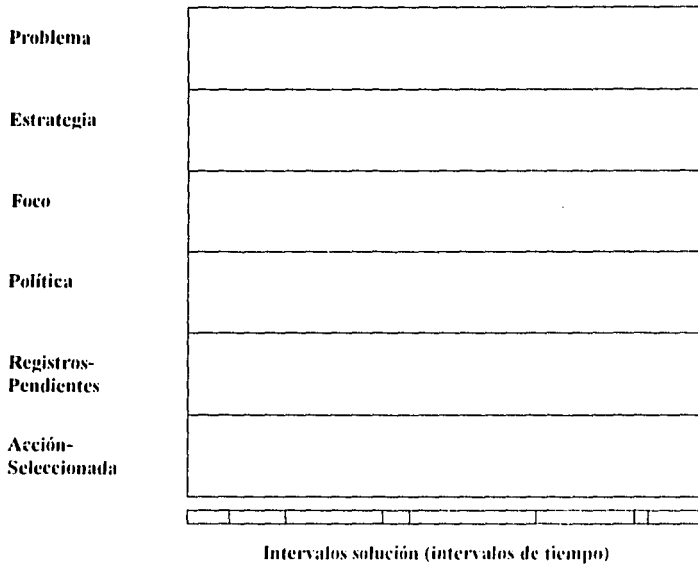


Figura 5.5 Niveles de abstracción e intervalos solución

Los intervalos de solución sobre el pizarrón del control representan intervalos de tiempo en la solución del problema, estos son definidos mediante ciclos en la solución del problema. Las decisiones de control ubicadas hacia la parte izquierda del pizarrón de control corresponden a las acciones planeadas para ciclos iniciales en la solución del problema, mientras que las decisiones ubicadas hacia la parte derecha del pizarrón de control corresponden a las acciones planeadas para los últimos ciclos en la solución del problema. La figura 5.5 muestra una representación del pizarrón del control, sobre la base de sus niveles de abstracción e intervalos solución

Las fuentes de conocimiento del dominio

Sobre el pizarrón del dominio operan las fuentes de conocimiento del dominio. La actividad de éstas consiste en la creación o modificación de elementos solución sobre el pizarrón del dominio. Las fuentes de conocimiento del dominio son específicas del dominio y determinadas por el diseñador del sistema o aplicación.

Las fuentes de conocimiento del dominio son representadas como estructuras de datos con los atributos definidos en la tabla 5.3.

Las fuentes de conocimiento del control

Sobre el pizarrón del control operan las fuentes de conocimiento del control. La actividad de éstas consiste en la articulación, interpretación y modificación de representaciones del conocimiento y comportamiento del sistema. Ellas generan y modifican elementos solución sobre el pizarrón del control bajo el control del mecanismo de despacho y forman dinámicamente un plan de control *prescriptivo*.

Algunas fuentes de conocimiento del control son específicas del dominio, mientras que otras son definidas por la arquitectura. Las fuentes de conocimiento del control son representadas a partir de la misma estructura de datos que utilizan para su representación las fuentes de conocimiento del dominio (tabla 5.3).

Atributos de las fuentes de conocimiento
Identificación
Dominio de aplicación
Comportamiento característico
Condición de disparo (situación de interés sobre el pizarrón)
Especificación de las variables usadas en la condición
Especificación de las variables usadas por el mecanismo de despacho
Acción (creación o modificación de elementos solución sobre el pizarrón)

Tabla 5.3

El mecanismo de despacho

La arquitectura de pizarrón para el control define un mecanismo de despacho adaptativo para controlar los registros de activación de las fuentes de conocimiento (RAFC's) del dominio y del control. El mecanismo de despacho define tres fuentes de conocimiento de control básico, las cuales iteran en un ciclo de tres pasos para llevar a cabo las siguientes acciones :

- ◆ Enumerar RAFCs pendientes
- ◆ Seleccionar un RAFC
- ◆ Ejecutar la acción del RAFC seleccionado

Estas fuentes de conocimientos operan en los niveles Registros-Pendientes y Acción-Seleccionada del pizarrón del control y determinan todas las acciones actuales del sistema que resultan posibles para la solución del problema.

Una fuente de conocimiento enumera RAFCs pendientes. Esta fuente de conocimiento es invocada cuando el RAFC previamente seleccionado para ejecutarse ha sido ejecutado. Su acción crea un nuevo conjunto de RAFCs pendientes en el nivel Registros-Pendientes del pizarrón del control. Esta también calcula y registra el valor de cada RAFC, comparando éstos con las decisiones localizadas en los niveles Foco y Política del pizarrón del control. Para computar el valor de un RAFC, la fuente de conocimiento evalúa cada decisión meta correspondiente a los niveles Foco y Política contra las variables de despacho del RAFC. La prioridad del RAFC es calculada teniendo en cuenta el valor u oferta del mismo. Cada valor y prioridad calculada para un RAFC son registrados sobre el pizarrón del control.

Una fuente de conocimiento selecciona un RAFC. Esta fuente de conocimiento se dispara cuando un nuevo conjunto de RAFCs pendientes es creado en el nivel Registros-Pendientes del pizarrón del control. Su acción consiste en la selección del RAFC que debe ser ejecutado, tomando en cuenta los valores y prioridades previamente calculados.

Una fuente de conocimiento ejecuta la acción del RAFC seleccionado. Esta fuente de conocimiento se dispara cuando un nuevo RAFC, que ha sido seleccionado para ser ejecutado, es registrado en el nivel Acción-Seleccionada del pizarrón del control. La acción de esta fuente de conocimiento interpreta y ejecuta la acción del RAFC registrado y registra los eventos resultantes sobre el pizarrón.

Otras Aplicaciones Pizarrón

Han sido muchas las aplicaciones, generalizaciones, extensiones y refinamientos desarrollados a partir de la arquitectura de pizarrón clásica surgida con el sistema HEARSAY II. Algunos de estos sistemas son descritos, en forma breve, a continuación.

AGE

El sistema AGE [NI188b] proporciona al usuario un paquete de herramientas para realizar ingeniería del conocimiento, al permitir definir y almacenar herramientas de ingeniería del conocimiento y reglas que guían al usuario en el uso de estas herramientas. AGE es un sistema basado en el conocimiento que permite la construcción de sistemas basados en el conocimiento.

AGE proporciona al usuario un modelo de pizarrón para la formación incremental de hipótesis. En AGE, el objetivo del programa basado en la arquitectura de pizarrón es permitir una mayor flexibilidad en la representación y aplicación de otros métodos de solución de problemas dentro de este marco.

Al igual que en HEARSAY II, el programa pizarrón de AGE consiste de tres componentes básicos : el pizarrón, las fuentes de conocimiento y el control.

El pizarrón contiene hipótesis, las cuales se encuentran ubicadas en los diferentes niveles de análisis de la tarea. Estos niveles de análisis de la tarea constituyen los niveles de abstracción del pizarrón. Existen dos formas mediante las cuales una hipótesis puede ser generada sobre el pizarrón : deductivamente, usando apoyo desde niveles superiores del pizarrón e inductivamente, utilizando evidencias ubicadas en niveles inferiores del pizarrón. Las hipótesis son generadas por reglas de inferencia contenidas en las fuentes de conocimiento.

Las fuentes de conocimiento contienen conocimiento acerca del dominio de la tarea. Estas son representadas como conjuntos de reglas de producción. Las fuentes de conocimientos utilizan las reglas de producción para generar o modificar elementos hipótesis sobre el pizarrón o para generar o modificar relaciones entre estas hipótesis.

El sistema de control está compuesto por varios componentes, los cuales pueden ser especificados o programados por el usuario. Estos subsistemas son los siguientes : formato de los datos de entrada, función de inicialización, método de procesamiento que será aplicado en cada paso de inferencia, reglas para seleccionar el próximo paso que será procesado, reglas para seleccionar la fuente de conocimiento relevante para procesar el paso seleccionado, condición de finalización y función de posprocesamiento. Algunos de estos componentes son proporcionados en forma preprogramada por AGE.

BB1

BB1 [HAY85, HAY88] es una implementación de la arquitectura de pizarrón para el control para el desarrollo de aplicaciones pizarrón

Un típico sistema BBI incluye los siguientes tipos de pizarrones : pizarrones problema, pizarrones solución, pizarrones de conocimiento, pizarrón del control, pizarrón del plan de control y pizarrón de datos de control.

Los Pizarrones Problema registran la información asociada con problemas particulares. Estos pizarrones son específicos de la aplicación y definidos por el diseñador del sistema.

Los Pizarrones Solución registran elementos solución (hipótesis) para problemas particulares generados por las fuentes de conocimiento. Los pizarrones solución son específicos de la aplicación y definidos por el diseñador del sistema.

Los Pizarrones de Conocimiento, registran información relativamente estable, la cual se encuentra disponible al sistema para la solución de problemas. En BBI, todas las fuentes de conocimiento se encuentran registradas sobre los pizarrones de conocimiento. Si un sistema posee más de un pizarrón de conocimiento, esto significa que se ha encontrado una forma útil de particionar su base de conocimiento.

El Pizarrón del Control registra información acerca de las propias acciones del sistema : acciones deseables, acciones factibles y acciones ejecutadas. La estructura del Pizarrón del Control es independiente de la aplicación particular, éste es definido por la propia arquitectura.

El Pizarrón de Planes de Control registra el plan dinámico de acciones deseables construido por las fuentes de conocimiento de control durante la solución del problema.

El Pizarrón de Datos de Control registra la ejecución de la agenda de acciones ejecutables y algunas decisiones tomadas por el despachador.

Las fuentes de conocimiento de BBI son representadas mediante estructuras de datos (como fueron definidas en la arquitectura de pizarrón para el control). BBI define tres tipos de fuentes de conocimientos : fuentes de conocimientos de tarea, las cuales describen acciones que resuelven un tipo de problema particular; fuentes de conocimiento de control, para describir acciones que resuelven el problema del control y fuentes de conocimiento de aprendizaje, para modificar el conocimiento del sistema.

En BBI las fuentes de conocimiento del control ejecutan un ciclo básico de tres pasos :

1. Ejecutar un RAFC despachado.
2. Actualizar la agenda para incluir nuevos RAFC's disparados por recientes cambios sobre el pizarrón.
3. Despachar el próximo RAFC ejecutable que mejor se adapta al actual plan de control.

BLOBS

BLOBS [ZAN88] es una estructura de sistema pizarrón orientado a objetos para llevar a cabo procesos de razonamiento en tiempo real. **BLOBS** consiste de una colección de objetos con arquitectura de pizarrón, los cuales se comunican entre sí a través del paso de mensajes.

Cada objeto-pizarrón consiste de un conjunto de declaraciones y definiciones. Entre estas definiciones se encuentran : un conjunto de declaraciones de variables locales, las cuales definen el estado del objeto y pueden ser declaradas como variables públicas o variables locales; un conjunto de definiciones de pseudo-variables, las cuales proporcionan operadores que al ser evaluados retornan un resultado; un conjunto de definiciones de procedimientos, que son locales a la definición del objeto-pizarrón y ayudan a mejorar la readaptabilidad del código; un conjunto de comportamientos, donde cada comportamiento es similar a un bloque de procedimientos y constituye la unidad de procesamiento principal del objeto-pizarrón; una lista de herencia, la cual amplía el conjunto de definiciones y declaraciones del objeto-pizarrón.

Para establecer la comunicación entre los objetos-pizarrón, **BLOBS** define dos grupos de métodos de comunicación : un primer grupo de métodos que contienen acciones imperativas ejecutadas dentro del cuerpo de un comportamiento y un segundo grupo de métodos que causan la activación de un comportamiento.

El primer grupo de métodos define acciones para : interrogar las variables públicas de otros objetos, invocar operadores públicos y creación de una nueva instancia de un objeto-pizarrón con un estado inicial dado.

Las acciones contenidas en el segundo grupo incluyen : envío de mensajes a instancias específicas de un objeto-pizarrón, actualización de los valores de alguna de las variables públicas de un objeto, disparo de demonios que monitorean la creación de nuevas instancias de objetos y el disparo de demonios que monitorean la eliminación de instancias de objetos.

Para ejecutar los comportamientos de los objetos-pizarrón la arquitectura de **BLOBS** define un despachador basado en la prioridad. El objeto-pizarrón con una mayor prioridad y con un comportamiento listo para su ejecución será el primero en ser ejecutado. Un objeto-pizarrón puede modificar su prioridad, permitiendo con esto que las actividades sean ejecutadas de acuerdo a su importancia.

CAGE

CAGE [NI88a, NI89b] es un sistema pizarrón concurrente basado sobre la arquitectura del sistema pizarrón AGE. **CAGE** constituye un marco para la solución de

problemas concurrentes. La arquitectura de un sistema construido en CAGE es un multiprocesador de memoria compartida.

Los componentes básicos de un sistema construido con CAGE son los siguientes :

- ◆ Un pizarrón sobre el cual las soluciones emergentes son colocadas. Los elementos sobre el pizarrón son organizados en niveles jerárquicos.
- ◆ Listas globalmente accesibles sobre las cuales es colocada la información de control.
- ◆ Fuentes de conocimiento, cada una de las cuales consiste de un conjunto de reglas.
- ◆ Información de control, la cual ayuda a determinar los elementos del pizarrón que van a constituir el foco de atención y cuáles fuentes de conocimiento serán utilizadas en algún momento dado en los procesos de solución del problema.
- ◆ Declaraciones para especificar cuáles componentes (fuentes de conocimiento, reglas) van a ser ejecutadas en paralelo y en qué momento esto ocurrirá.

CAGE puede ejecutar en serie (exhibiendo un comportamiento idéntico al del sistema AGE) o puede ejecutar con uno o más de sus componentes corriendo concurrentemente.

Cuando CAGE ejecuta en forma serial, entonces :

1. La fuente de conocimiento de control selecciona un evento y ejecuta la fuente de conocimiento asociada a éste.
2. La fuente de conocimiento asociada al evento seleccionado realiza las siguientes actividades en forma serial :
 - a. Ejecuta una evaluación del contexto.
 - b. Evalúa la condición de una regla y si ésta ha sido apareada entonces ejecuta su acción, o
 - c. Evalúa las condiciones de todas las reglas y posteriormente ejecuta aquellas reglas cuyas condiciones han sido apareadas
3. Cada clausula correspondiente a la condición de una regla es evaluada y posteriormente ejecutadas las acciones descritas en la conclusión de la regla.

Cuando CAGE ejecuta en forma concurrente, entonces :

1. La fuente de conocimiento de control ejecuta las siguientes actividades :
 - a. Ejecuta en paralelo las fuentes de conocimiento asociadas a cada evento seleccionado, o

b. Espera que concluya la ejecución de todas las fuentes de conocimiento activas y ejecuta concurrentemente las fuentes de conocimiento asociadas a los nuevos eventos generados, o

c. Espera que varios eventos hayan sido generados y selecciona un subconjunto de éstos, ejecutando las fuentes de conocimientos asociadas a todos los eventos del subconjunto seleccionado en paralelo.

2. Cada fuente de conocimiento asociada a un evento generado ejecuta las siguientes actividades :

a. Ejecuta una evaluación del contexto en paralelo.

b. Evalúa todas las condiciones de sus reglas en paralelo y entonces una vez que esta evaluación haya sido concluida, selecciona una acción para su ejecución; o ejecuta en serie las acciones de las condiciones apareadas; o ejecuta en paralelo las acciones de las condiciones apareadas.

3. Todas las cláusulas correspondientes a la condición de una regla son evaluadas en paralelo y entonces las acciones descritas en la conclusión son ejecutadas en paralelo.

GBB

GBB [COR88] es un sistema para el desarrollo de pizarrones genéricos. GBB fue diseñado para implementar eficientemente grandes sistemas pizarrón, los cuales permiten la creación y recuperación de miles de objetos pizarrón (hipótesis). La mayor contribución de GBB resultó ser la unificación de las tecnologías de pizarrón existentes en un sistema de desarrollo para aplicaciones de gran funcionamiento.

En GBB, el pizarrón es una estructura jerárquica compuesta de piezas pizarrón llamadas espacios (niveles). Para implementar una aplicación en GBB es necesario especificar la estructura del pizarrón (espacios y jerarquía) y de las unidades (objetos o hipótesis) que van a residir sobre éste. Los espacios, jerarquía y unidades del pizarrón son definidos utilizando las siguientes funciones definidas en COMMON LISP :

define-spaces : define los espacios (niveles del pizarrón)

define-blackboards : define la jerarquía del pizarrón

define-unit : define las unidades (hipótesis) sobre el pizarrón

Una vez definida la estructura del pizarrón y de las unidades que residirán sobre éste, es necesario especificar implementaciones particulares de bases de datos pizarrón, los mecanismos para inicializar éste, así como la creación y recuperación de unidades sobre el pizarrón. Para llevar a cabo estas operaciones, GBB dispone de las siguientes funciones, también escritas en COMMON LISP :

define-unit-mapping : especifica el mecanismo que almacena unidades sobre los espacios del pizarrón

instantiate-bb-database : inicializa la base de datos pizarrón

make-unit-type : crea y almacena una unidad sobre un espacio del pizarrón

find-units : recupera unidades desde espacios del pizarrón

GUARDIAN

La tarea del sistema GUARDIAN [HAY89] es el monitoreo de pacientes en una unidad quirúrgica de cuidados intensivos (UQCI). GUARDIAN modela el trabajo cooperativo de un equipo médico en una UQCI, garantizando una disponibilidad continua de todos los conocimientos y habilidades relevantes, que pudieran requerirse en un momento dado para la tarea de monitoreo.

El sistema GUARDIAN es una aplicación en IAD, basada sobre la arquitectura de pizarrón para el control. La arquitectura del sistema GUARDIAN define los siguientes tipos de agentes :

- ◆ Un único agente de control a nivel superior, con funciones de supervisión e integración.
- ◆ Agentes de razonamiento para interpretar la información percibida, resolver problemas del dominio, tomar decisiones y planear acciones.
- ◆ Agentes de percepción/acción para adquirir información desde el medio ambiente e influir sobre entidades en el medio ambiente respectivamente.

El agente de control de GUARDIAN está estructurado dentro de la arquitectura de BBI y contiene tres agentes componentes : un administrador de agenda que identifica las posibles operaciones de razonamiento disparadas por eventos ocurridos recientemente; un despachador que selecciona la operación que mejor se corresponda al plan de control actual; un ejecutor que ejecuta la operación seleccionada.

Cada agente de razonamiento constituye un conjunto de operaciones de razonamiento componentes, las cuales pueden ser apareadas a un contexto particular y organizadas heurísticamente para contribuir a la solución de un problema o alcanzar una determinada meta. Cada agente de razonamiento ha sido implementado como un sistema pizarrón dentro de la arquitectura BBI.

Los agentes de percepción/acción se encuentran jerárquicamente organizados para manejar interacciones entre el sistema y su medio ambiente. La percepción es controlada por "preprocesadores", los cuales monitorean sensores periféricos, trasladan y filtran los datos sensados y envían los resultados al agente de control para su incorporación en la valoración de la situación global. La acción es controlada por "manejadores", los cuales reciben descripciones de acciones y restricciones para su ejecución desde el agente de control a nivel superior y manejan la ejecución de las acciones a través de efectores periféricos

HASP/SIAP

La tarea del sistema HASP/SIAP [NII82, NII89a, WIL84] consiste en el desarrollo y mantenimiento de un "tablero de situación" que refleje las actividades de buques portaaviones y submarinos en una región bajo vigilancia. El papel del "tablero de situación" es la interpretación continua de cadenas de señales acústicas producidas por objetos en la región bajo vigilancia y la integración de reportes inteligentes producidos con los resultados de la interpretación.

En HASP/SIAP, el pizarrón ha sido dividido en seis niveles de abstracción : segmentos, líneas, conjuntos armónicos, fuentes, plataformas y flota. Las señales de entrada arriban sobre el nivel segmento, mientras que los datos correspondientes a los reportes producidos arriban a los niveles plataformas o flota, en dependencia del contenido del reporte. El estado reflejado sobre el pizarrón en un momento dado, refleja la mejor comprensión de la situación para ese momento. La respuesta del sistema no viene dada por las hipótesis generadas en los niveles plataforma o flota, sino que ésta viene dada por toda la red de soluciones parciales generada sobre todo el pizarrón.

Todos los cambios efectuados sobre el pizarrón, conjuntamente con los tipos de estos cambios, son colocados en una lista de eventos. Cuando un evento es seleccionado de la lista de eventos por el mecanismo de control, éste pasa a constituir el foco de atención. Otras listas definidas en HASP/SIAP son la lista de expectativas, la cual contiene eventos que se esperan que ocurran en el futuro; la lista de problemas, conteniendo descripciones de varios tipos de problemas y fuentes de conocimiento; la lista de evento-reloj, la cual consiste de un tiempo dado y reglas asociadas a éste

En HASP/SIAP, una fuente de conocimiento consiste de una parte condición y una parte acción, ambas contenidas en un mismo módulo. La parte condición consiste de una lista de pares de elementos, donde un elemento del par constituye el nombre del tipo de evento y el otro elemento su modificador, este último indica el estatus del elemento hipótesis. La parte acción está formada por un conjunto de reglas

El módulo de control en HASP/SIAP fue estructurado de la misma forma que las fuentes de conocimiento del dominio. El módulo de control consiste de una fuente de conocimiento de estrategia y de un grupo de manejadores de eventos, uno para cada tipo de evento definido (evento cambio sobre el pizarrón, evento-reloj, expectación y problema)

MUSE

El sistema MUSE [REY88] es un juego de herramientas que brinda ayuda en el desarrollo de prototipos de aplicaciones experimentales en tiempo real en Inteligencia Artificial.

La estructura básica de una aplicación en MUSE es un conjunto de módulos de razonamiento separados, los cuales se comunican mediante el acceso compartido a bases de datos particulares. La gran mayoría de las aplicaciones en MUSE utilizan una aproximación estilo pizarrón, donde las diferentes bases de datos corresponden a diferentes niveles de un pizarrón multinivel y los módulos de razonamiento corresponden a los paquetes de reglas que ejecutan entre los niveles del pizarrón. En MUSE los módulos de razonamiento son conocidos como fuentes de conocimiento.

MUSE es un sistema orientado a objetos, donde cada fuente de conocimiento, conjunto de reglas, regla y base de datos es un objeto.

En MUSE, la estructura de una fuente de conocimiento estándar consiste de una única base de datos de la fuente de conocimiento y uno o más sistemas de reglas, cuyos componentes son un conjunto de reglas compiladas y una base de datos local para el conjunto de reglas.

La base de datos local actúa como una memoria de trabajo rápida durante los procesos de razonamiento y sólo es accesible por su conjunto de reglas. La base de datos de la fuente de conocimiento es accesible por todos los conjuntos de reglas dentro de la fuente de conocimiento. Sobre esta base de datos son almacenados objetos que pudieran constituir datos de entrada para alguno de los sistemas de reglas o una especificación de meta. Las conclusiones de las reglas disparadas son colocadas sobre esta base de datos, las cuales en turno pudieran disparar algún otro sistema de reglas de la fuente de conocimiento o de otra fuente de conocimiento.

Otro componente de la estructura de MUSE es un tipo especial de fuente de conocimiento que sólo contiene una base de datos. Este tipo de fuente de conocimiento es llamada tablón de anuncios y es utilizada como un canal de comunicación. Cada tablón de anuncios actúa como un nivel del pizarrón.

El control en MUSE es llevado a cabo a través de una agenda de despacho, la cual se basa sobre un simple ordenamiento de prioridades. Esta agenda es utilizada para controlar la ejecución de las fuentes de conocimiento en competencia.

Los procesos de razonamiento en MUSE son llevados a cabo a través de encadenamientos de reglas hacia adelante (direccionamiento por los eventos) y hacia atrás (direccionamiento por las metas). El encadenamiento de reglas hacia adelante juega el papel principal en la arquitectura global de las fuentes de conocimiento por su gran habilidad para monitorear múltiples bases de datos. El encadenamiento de reglas hacia atrás sigue el principio de la unificación de cláusulas más que el apareo de éstas y es controlado por una búsqueda con encadenamiento hacia atrás en profundidad.

MXA

MXA [LAK84, TAI88] es un *shell* para la construcción de sistemas expertos con arquitectura de pizarrón, los cuales requieren de la capacidad de ejecutar procesos de razonamiento en tiempo real, como es el caso de la interpretación de señales.

Al igual que otros muchos sistemas pizarrón, la estructura de MXA abarca tres elementos principales : el pizarrón, las fuentes de conocimiento y el control.

En MXA el pizarrón registra dos tipos de elementos : las hipótesis y las conexiones entre las hipótesis. A cada hipótesis sobre el pizarrón le es asociada una medida de probabilidad, la cual indica la fuerza de convicción del sistema acerca de la hipótesis. Existen varias formas en las que una hipótesis puede ser creada sobre el pizarrón. La forma más elemental es como resultado de una entrada, donde los detalles de cada entrada recibida (por ejemplo, desde los sensores) son registrados sobre el pizarrón como una hipótesis.

Otra forma en que una hipótesis puede ser creada sobre el pizarrón es como resultado de las conclusiones alcanzadas por las reglas que forman las fuentes de conocimiento y que han logrado satisfacer sus condiciones sobre otras hipótesis previamente generadas sobre el pizarrón. También las expectativas generadas por la reglas pueden ser registradas sobre el pizarrón en la forma de hipótesis, las cuales en turno son utilizadas por otras reglas para establecer correlaciones con hechos reales.

El otro elemento registrado sobre el pizarrón son las conexiones entre las hipótesis. Una conexión evidencial desde una primera hipótesis hasta una segunda hipótesis indica que la primera hipótesis apoya la segunda.

En MXA una fuente de conocimiento es definida como un grupo de reglas con las siguientes características

- ◆ la fuente de conocimiento puede ser activada o desactivada, esto permite focalizar los recursos del sistema para alcanzar la mejor meta o seleccionar una fuente de conocimiento particular para la solución de un problema particular
- ◆ la fuente de conocimiento tiene una prioridad establecida por meta-reglas
- ◆ todas las reglas dentro de una fuente de conocimiento que se puedan disparar, serán disparadas
- ◆ una fuente de conocimiento sólo debe contener reglas de un mismo tipo o dedicadas a un propósito particular

MXA opera en un ciclo continuo conocido como Ciclo Sistema. Durante este ciclo son seleccionadas las fuentes de conocimiento que pueden disparar y posteriormente se decide cuál de ellas será ejecutada. La selección, ejecución, activación y desactivación

de las fuentes de conocimiento las realiza un sistema de despacho formado por fuentes de meta-conocimiento.

Otra característica relevante de MXA es su capacidad para generar explicaciones acerca de las acciones tomadas.

POLIGON

Al igual que CAGE, POLIGON [NH88a, NH89b] es un marco para la solución de problemas concurrentes. La arquitectura de un sistema construido en POLIGON es un multiprocesador de memoria distribuida. POLIGON fue diseñado para correr sobre un *hardware* que proporcione un mecanismo básico (primitivas de paso de mensajes) para la comunicación entre los elementos de procesamiento.

La estructura de POLIGON abarca un conjunto de nodos pizarrón independientes y en paralelo, los cuales se comunican entre sí a través del paso de mensajes. El despachador centralizado, presente en la gran mayoría de los sistemas pizarrón, fue eliminado de la estructura de POLIGON y ésta no prevé ningún tipo de sincronización global. De esta forma, los diferentes componentes de POLIGON pueden ejecutar a diferentes velocidades y poseer diferentes puntos de vista acerca del progreso en los procesos de solución del problema. En POLIGON, cada nodo pizarrón es visto como un proceso responsable de su propio control y del procesamiento de mensajes.

En POLIGON, las fuentes de conocimiento no son unidades de despacho, sólo son colecciones de conocimiento. Una fuente de conocimiento es una colección de reglas. Todas las reglas dentro de una fuente de conocimiento pueden, en principio, ser ejecutadas en paralelo. Las reglas son disparadas como demonios por actualizaciones hechas sobre el nivel del pizarrón que representa el nodo. Después de la ocurrencia de un evento sobre el pizarrón, todas las reglas que logren satisfacer sus condiciones son invocadas concurrentemente.

Las dos principales fuentes de concurrencia en POLIGON son :

1. Cada nodo pizarrón puede estar activo simultáneamente para reflejar el paralelismo de datos.
2. Las reglas pertenecientes a un nodo pueden estarse ejecutando sobre varios elementos de procesamiento simultáneamente

POLIGON introduce un mecanismo de control jerárquico para explotar la estructura de los datos pizarrón. Los niveles del pizarrón están organizados como una clase jerárquica. Cada nivel es una clase y un nodo pizarrón es una copia de esta clase. Los nodos clases contienen información acerca de sus copias y las fuentes de conocimiento pueden estar contenidas en los nodos clases para controlar sus nodos copias. El nodo

clase es conocido como controlador de nivel. Un nodo super-controlador es aquel que puede controlar los nodos clases.

En POLIGON, el potencial para el control se encuentra localizado en tres lugares diferentes : dentro de cada nodo, donde las acciones de las reglas pueden ser ejecutadas en serie; en el controlador de nivel, el cual puede ser usado para monitorear las actividades de sus nodos; y en el super-controlador, para la creación de controladores de niveles y sus actividades.

PROTEAN

La tarea del sistema PROTEAN [HAY87, NII89a] es determinar la posible estructura tridimensional de una molécula de proteína dada. PROTEAN construye la estructura tridimensional a partir de los datos de resonancia magnética nuclear obtenidos desde la proteína en solución.

PROTEAN es una aplicación con arquitectura de pizarrón para el control escrita en BBI. La arquitectura de PROTEAN define tres estructuras pizarrón : un pizarrón para el control de datos, uno para el control de planes y uno solución.

El pizarrón para el control de datos tiene tres niveles los cuales son especificados por BBI : los niveles evento, registro de activación de las fuentes de conocimiento (RAFC) y agenda. Los objetos sobre el nivel evento corresponden a cambios sobre el pizarrón y éstos contienen información acerca del nodo que va a ser creado o modificado. Los RAFCs en el nivel RAFC contienen información de la fuente de conocimiento a la cual están asociados, información acerca del nodo evento que disparó a la fuente de conocimiento e información acerca del nodo que va a ser creado o modificado. El nivel agenda contiene cuatro nodos, cada uno de los cuales representa un estado diferente en el cual la fuente de conocimiento apareada pudiera encontrarse.

El pizarrón para el control de planes contiene cuatro niveles especificados en BBI : los niveles estrategia, foco, heurística y despacho, los cuales ya fueron descritos en la arquitectura de pizarrón para el control

El pizarrón solución contiene dos niveles : el nivel estructura secundaria y el nivel arreglos parciales. Un arreglo parcial es un grupo de estructuras secundarias. La información en estos niveles corresponde a la estructura de la proteína en análisis.

TRICERO

La tarea del sistema TRICERO [NII89a] es el monitoreo de una región del espacio aéreo, en la cual desarrollan sus actividades los aviones. TRICERO es un sistema pizarrón distribuido, donde cada sistema posee un conjunto diferente de fuentes de conocimiento y una diferente organización del pizarrón.

La arquitectura de TRICERO comprende tres subsistemas pizarrón organizados jerárquicamente en dos niveles. Ubicados en el nivel inferior se encuentran los subsistemas ELINT y COMINT, los cuales ejecutan tareas diferentes de interpretación de señales. En el nivel superior se encuentra el subsistema CORRELATION, el cual integra los reportes procedentes de ELINT y COMINT conjuntamente con otros datos provenientes del propio nivel superior. TRICERO analiza dos tipos de datos diferentes, utilizando los subsistemas ELINT y COMINT y posteriormente correlaciona los datos analizados, utilizando el subsistema CORRELATION.

Los subsistemas ELINT y COMINT definen diferentes organizaciones de datos para el pizarrón y diferentes fuentes de conocimiento. El pizarrón de ELINT contiene tres niveles : nivel observación, donde son recibidos los datos de entrada; nivel emisor, donde se ubican los nodos que contienen una historia de detecciones desde un sitio que tiene la misma etiqueta de identificación y nivel grupo, donde se localizan las emisiones detectadas por el radar en diferentes sitios, combinadas sobre en una plataforma hipotética. De forma similar fueron estructurados los pizarrones de los subsistemas COMINT y CORRELATION, definiendo para éstos niveles de abstracción apropiados para la interpretación de sus datos. Las fuentes de conocimiento y el control han sido definidos teniendo en cuenta los requerimientos del sistema AGE.



TERCERA PARTE

**INTERMED: UNA RED DE
SISTEMAS EXPERTOS
PARA LA CONSULTA E
INTERCONSULTA
MEDICA**



REDSIEX : Una Red Cooperativa de Sistemas Pizarrón

REDSIEX y la solución cooperativa de problemas distribuidos

Los sistemas expertos cooperativos abarcan el desarrollo de mecanismos de interacción cooperativa, que permitan a múltiples sistemas expertos trabajar conjuntamente en la solución de un problema común. El carácter heterogéneo de los sistemas expertos cooperativos significa que cada experto pudiera tener diferentes metas, diferentes aproximaciones a la solución del problema, diferente evaluación de criterios para la solución y diferentes representaciones internas de soluciones y problemas. La cooperación en este marco es algo mucho más complejo que proporcionar a estos expertos un lenguaje de comunicación común. Para que estos expertos puedan cooperar, cada uno de ellos necesita tener modelos bien detallados que abarquen conocimientos, metas, planes y preferencias de los restantes expertos. De esta forma, puede emerger cooperativamente una nueva solución global del problema.

REDSIEX (RED de Sistemas EXpertos) es un sistema prototipo para el desarrollo de aplicaciones basadas en la solución cooperativa de problemas distribuidos. REDSIEX es una red de nodos resolutores de problemas, donde cada nodo puede trabajar de forma independiente en la solución de un problema, pero cuando éstos se enfrentan a un problema global, no puede ser resuelto sin la cooperación; necesaria cuando un único nodo no posee los recursos, el conocimiento ni la experticia suficiente para resolver un problema.

Diferentes nodos pudieran tener experticia para resolver diferentes partes de un problema, diferentes recursos disponibles para abordar la solución de un problema y poseer conocimientos o puntos de vistas diferentes acerca de un problema.

Para lograr que los nodos efectivamente trabajen de forma cooperativa, es necesario combatir su racionalidad limitada en conocimiento, procesamiento e interacción. En REDSIEX, estas limitaciones son superadas a partir de la introducción de las siguientes estrategias de cooperación :

- ◆ Comunicación
- ◆ Organización
- ◆ Planeación
- ◆ Modelado

Las estrategias de comunicación, planeación y modelado son abordadas en este mismo capítulo en el epígrafe "Los componentes programas del nodo y su funcionamiento", donde se describe la actividad de las fuentes de conocimiento de comunicación y el planeador local.

En una red cooperativa para la solución de problemas, la organización especifica cuáles son las responsabilidades y los patrones de interacción para cada uno de los nodos en la red. La organización puede llegar a mejorar la ejecución de la red al reducir las responsabilidades de cada nodo en la solución de problemas.

En REDSIEX, la organización se especifica a partir de los siguientes criterios :

1. Asignando roles en la solución de problemas a los nodos, lo cual hace que cada nodo sea un especialista en un tipo diferente de problema o subproblema.
2. Indicando a los nodos las conexiones para el flujo de información entre nodos, de forma tal que éstos puedan transferir problemas o subproblemas, que requieren ser resueltos, a otros nodos que poseen mejores habilidades para resolverlos.

Con el uso de estas estrategias los agentes logran realmente coordinar sus actividades y trabajar de forma cooperativa, con el fin de alcanzar la meta global a la cual aspira el sistema en su conjunto.

En REDSIEX se alcanza cooperación intranodo y cooperación internodo. La cooperación intranodo se manifiesta a través del trabajo conjunto de todas las fuentes de conocimiento locales del nodo. Este se caracteriza por ser un proceso oportunístico e incremental de creación y modificación de los elementos solución más prometedores sobre el pizarrón del dominio, con el objetivo de alcanzar las metas locales propuestas. La cooperación internodo se alcanza a partir de las estrategias de cooperación de REDSIEX, mediante las cuales los nodos pueden intercambiar hipótesis y resultados parciales de gran utilidad para los procesos de solución del problema

La cooperación internodo puede llevarse a cabo en tiempo real o diferida, en dependencia de las disponibilidades de tiempo y recursos de los nodos a los que se les ha solicitado el compromiso. La cooperación en tiempo real tiene lugar cuando la respuesta a un compromiso solicitado por un nodo es recibida por éste dentro del intervalo de tiempo que él ha dedicado a los procesos de solución del problema, para el cual fue solicitado el compromiso. La cooperación diferida tiene lugar cuando el problema para el cual el compromiso fue solicitado ha sido abandonado temporalmente, sin que se haya recibido una respuesta, pero este problema puede ser retomado una vez que el nodo reciba respuesta a la solicitud de compromiso. Estas características de la cooperación en REDSIEX son abordadas con mayor detalle en el epígrafe "Los componentes programas del nodo y su funcionamiento".

Distribución de la información, procesamiento y control en REDSIEX

Existen varias dimensiones para la descomposición de un sistema pizarrón centralizado en un medio ambiente distribuido y varias opciones dentro de cada una de ellas. [LES88a]. La estructura de REDSIEX tiene en cuenta las siguientes dimensiones y opciones para cada dimensión :

1. La información.

- a. Distribución del pizarrón : el alcance del pizarrón local de un nodo define su área de interés.
- b. Transmisión de hipótesis : un nodo transmite y recibe hipótesis hacia y desde un subconjunto local de nodos.

2. El procesamiento.

- a. Distribución de las fuentes de conocimiento : cada nodo tiene un subconjunto de fuentes de conocimiento.
- b. Acceso al pizarrón por las fuentes de conocimiento : la activación de una fuente de conocimiento puede acceder sólo el pizarrón en su nodo local.

3. El control.

- a. Distribución de la activación de las fuentes de conocimiento : un evento sobre el pizarrón activa sólo a fuentes de conocimiento dentro del nodo local.
- b. Distribución del mecanismo de despacho, de la base de datos de control y del planeador : cada nodo posee su propio mecanismo de despacho, y realiza el despacho basado sobre información local. Cada nodo posee un pizarrón de control y un planeador para ejecutar tareas de control locales al nodo.

La arquitectura de los nodos en REDSIEX

Como se muestra en la figura 6.1, cada nodo en REDSIEX es arquitectónicamente un sistema pizarrón con elementos heredados desde la arquitectura de Hearsay II, la arquitectura del sistema DVMT, la arquitectura de Pizarrón para el Control, así como nuevos elementos definidos por la propia arquitectura de REDSIEX.

La figura 6.2 muestra los detalles de la arquitectura que exhiben los nodos en REDSIEX. La estructura de cada nodo en la red define dos tipos de componentes fundamentales : componentes-base de datos y componentes-programa.

Los componentes-base de datos del nodo

Los componentes bases de datos del nodo son estructuras de datos sobre las cuales los componentes programas del nodo ejecutan acciones de búsqueda, creación y modificación de elementos pertenecientes a dichas estructuras. Cada nodo en la red posee los siguientes componentes bases de datos .

- ◆ Pizarrón del Dominio
- ◆ Base de Datos de Control
- ◆ Lista de Registros de Activación de las Fuentes de Conocimiento (LRAFC), la cual incluye
 - Registros de Activación de las Fuentes de Conocimiento de Razonamiento (RAFRZ)
 - Registros de Activación de la Fuente de Conocimiento Receptora (RAFRE)
 - Registros de Activación de la Fuente de Conocimiento Transmisora (RAFTR)

Registros de Activación de la Fuente de Conocimiento de Procesamiento
"Backward" (RAFPB)

- ◆ Buzón de Mensajes de Entrada (BME)
- ◆ Árbol de Búsqueda

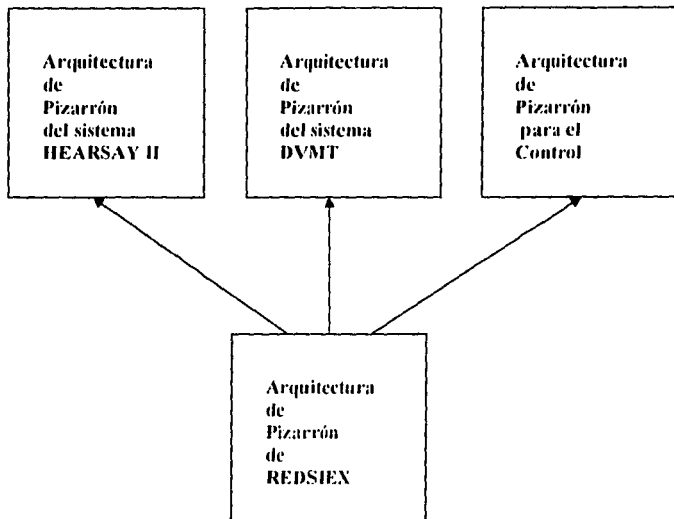


Figura 6.1 La arquitectura de REDSIEX hereda elementos estructurales desde las arquitecturas de HEARSAY II, DVMT y Pizarrón para el Control

El Pizarrón del Dominio

El pizarrón del dominio es una estructura de datos compartida a través de la cual las fuentes de conocimiento del dominio (fuentes de conocimiento de razonamiento) se comunican entre sí. El pizarrón del dominio organiza los elementos solución, también llamados hipótesis, en diferentes niveles de abstracción. Estos niveles de abstracción representan la solución del problema en diferente grado de detalle.

A cada hipótesis sobre el pizarrón del dominio se asocia la siguiente información :

- ◆ Identificador o número de la hipótesis
- ◆ Certidumbre o grado de creencia asociado a la hipótesis

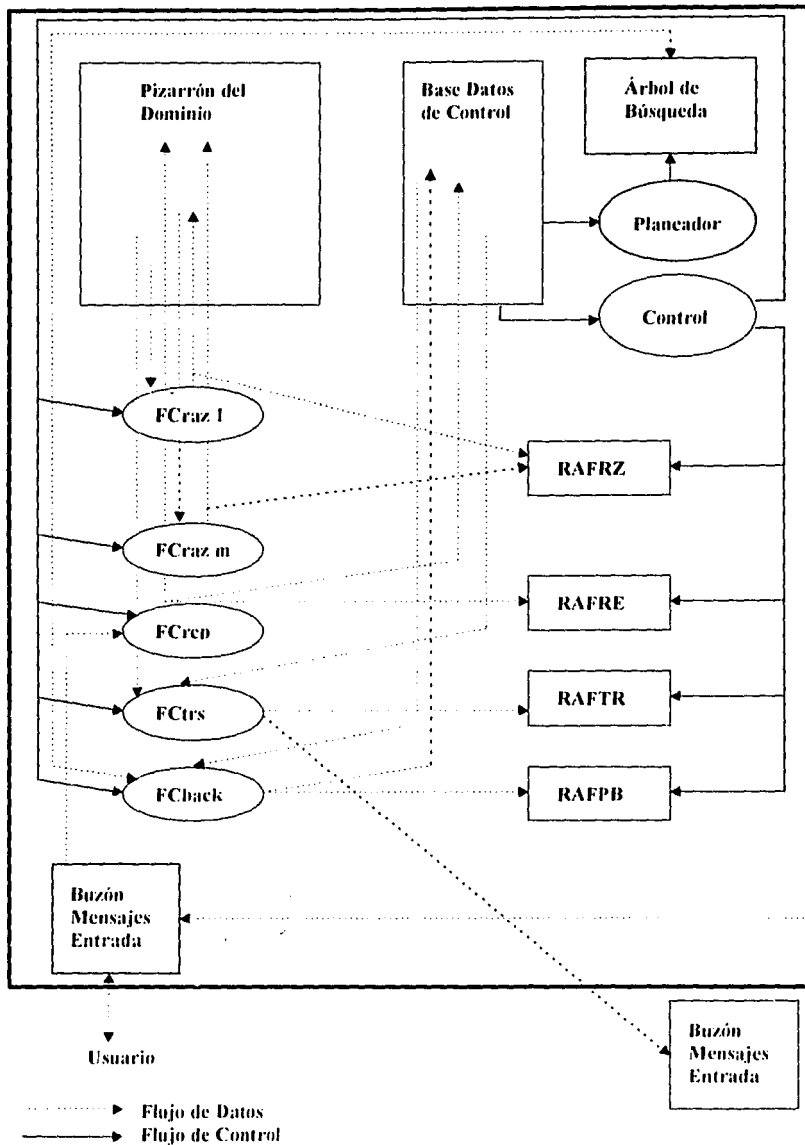


Figura 6.2 Arquitectura de un nodo en REDSIEX

La Base de Datos de Control

La base de datos de control (o base de metaconocimiento) es una estructura de datos definida por la arquitectura de REDSIEX. Los elementos solución sobre la base de datos de control son registros cuyos atributos encierran información o decisiones relevantes para el problema del control : elementos solución para la revisión de creencias y elementos solución para la investigación y comunicación de hipótesis.

Elementos solución para la revisión de creencias

Para cada hipótesis sobre el pizarrón del dominio se asocia un elemento solución sobre la base de datos de control, con información correspondiente a la certidumbre o grado de creencia de la hipótesis. Esta información es utilizada en el proceso de revisión de creencias de las hipótesis, el cual se describe más adelante.

Cada elemento solución para la revisión de creencias es un registro con los siguientes atributos :

- ◆ Lista de los identificadores de las reglas locales que han aportado evidencias para la demostración de la hipótesis.
- ◆ Lista de las certidumbres o valores de creencias con los cuales las reglas en cuestión han contribuido a la demostración de la hipótesis.
- ◆ Lista de los identificadores de los nodos que han contribuido al valor de creencia de la hipótesis.
- ◆ Lista de las certidumbres o valores de creencia con los cuales los nodos en cuestión han contribuido a la demostración de la hipótesis.

Elementos solución para la investigación y comunicación de hipótesis

Estos elementos solución constituyen parte del foco de control del nodo local y son decisiones de control acerca de qué elementos solución sobre el pizarrón del dominio deben ser actualmente investigados, a partir de los propios recursos del nodo, qué hipótesis deben ser transmitidas como resultados parciales a otros nodos o qué preguntas deben ser formuladas a otros nodos de la red

Los elementos solución para la investigación y comunicación de hipótesis incluyen los siguientes tipos de registros :

Investigar-Hipótesis : Identificadores de las hipótesis que deben ser actualmente investigadas por el nodo

Transmitir-Hipótesis : Identificadores de las hipótesis que deben ser transmitidas a otros nodos.

Preguntar-Hipótesis : Identificadores de las hipótesis que deben ser preguntadas a otros nodos.

Acceso Denegado-Hipótesis : Identificadores de las hipótesis para las cuales se ha prohibido la modificación de sus certidumbres.

Sobre la base de datos de control operan las fuentes de conocimiento receptora, transmisora, de procesamiento "*backward*" y el planeador local, que también exhibe una estructura condición-acción. Estas fuentes de conocimiento se comunican entre sí a través de la creación de elementos solución sobre la base de datos de control.

La lista de los Registros de Activación de las Fuentes de Conocimiento

Cuando las condiciones o premisas de una fuente de conocimiento son satisfechas, ésta se dispara creando un registro de activación de la fuente de conocimiento (RAFC). Un RAFC es una estructura de datos que describe las características de la acción disparada por la fuente de conocimiento. Cuando un RAFC es seleccionado por el mecanismo de control (discutido más adelante) la acción de la fuente de conocimiento asociada al RAFC es ejecutada en el contexto de su información de disparo y produce típicamente nuevos eventos sobre el pizarrón del dominio o la base de datos de control, o acciones de comunicación con otros nodos de la red, en dependencia de la naturaleza de la fuente de conocimiento que lo originó.

No todos los RAFCs poseen la misma estructura de datos, para cada tipo de fuente de conocimiento es definida una estructura de datos apropiada para sus RAFCs. Los tipos de RAFCs definidos para cada nodo son los siguientes :

- ◆ Registros de Activación de las Fuentes de Conocimiento de Razonamiento (RAFRZ.s).
- ◆ Registros de Activación de la Fuente de Conocimiento Receptora (RAFREs).
- ◆ Registros de Activación de la Fuente de Conocimiento Transmisora (RAFTRs).
- ◆ Registros de Activación de la Fuente de Conocimiento de Procesamiento "*Backward*" (RAFPBs).

Las estructuras de datos correspondientes a cada uno de los tipos de RAFCs son mostradas en las tablas 6.1, 6.2, 6.3 y 6.4.

Buzón de Mensajes de Entrada

Cada mensaje enviado al nodo por cualquiera de los restantes nodos de la red (incluyendo al usuario del nodo) es recibido en el buzón de mensajes de entrada y se crea un registro para el mensaje de entrada (RME). Un RME es una estructura de datos con los atributos definidos en la tabla 6.5.

Estructura de datos de un RAFRZ

Ciclo de activación : Número del ciclo de ejecución en el cual fue creado el registro de activación
Identificador de la fuente : Número o identificador de la fuente de razonamiento que creó el registro
Identificador de la regla : Número o identificación de la regla de inferencia disparada
Evento acción : Hipótesis del pizarrón que va a ser creada o modificada
Nivel de Apareo : Nivel del pizarrón en el cual fue apareada la condición de la regla disparada
Nivel de Síntesis : Nivel del pizarrón en el que se ejecutará la acción de la regla disparada
Fuerza de apareo de la regla : Certidumbre o valor de creencia con el que fue satisfecha la condición de la regla
Importancia de la regla : Certidumbre o valor de creencia de la regla propuesta por el experto
Eficiencia de la regla : Certidumbre calculada para la regla en base a su fuerza de apareo e importancia
Creencia final del evento : Certidumbre o valor de creencia final calculado para el evento creado
Oferta : Valor calculado por el mecanismo de control en base a los restantes parámetros

Tabla 6.1

Estructura de datos de un RAFRE

Ciclo de activación : Número del ciclo de ejecución en el cual fue creado el registro de activación
Evento asociado al mensaje : Hipótesis del pizarrón para la cual se brinda o solicita información
Nivel de síntesis : Nivel del pizarrón al que pertenece la hipótesis asociada al mensaje
Valor de creencia asociado al mensaje : Certidumbre o valor de creencia de la hipótesis asociada al mensaje
Creencia final del evento : Certidumbre o valor de creencia final calculado para el evento creado
Colaborador del mensaje : Número o identificador del nodo que envía el mensaje
Clave del mensaje : Especifica el tipo de información contenida en el mensaje
Oferta : Valor calculado por el mecanismo de control en base a los restantes parámetros

Tabla 6.2

Estructura de datos de un RAFTR

Ciclo de activación : Número del ciclo de activación en el cual fue creado el registro de activación
Evento asociado al mensaje : Hipótesis del pizarrón para la cual se brinda o solicita información
Valor de creencia asociado al mensaje : Certidumbre o valor de creencia de la hipótesis asociada al mensaje
Clave del mensaje : Especifica el tipo de información contenida en el mensaje
Oferta : Valor calculado por el mecanismo de control en base a los restantes parámetros

Tabla 6.3

Estructura de datos de un RAFTP

Ciclo de activación : Número del ciclo de activación en el cual fue creado el registro de activación
Evento acción : Hipótesis del pizarrón del dominio para la cual se requiere información
Nivel de síntesis : Nivel del pizarrón al que pertenece la hipótesis requerida
Oferta : Valor calculado por el mecanismo de control en base a los restantes parámetros

Tabla 6.4

Estructura de datos de un RME

Evento asociado al mensaje : Número o identificador de la hipótesis para la cual se propone o solicita información
Valor de creencia asociado al mensaje : Certidumbre o valor de creencia de la hipótesis asociada al mensaje
Colaborador del mensaje : Número o identificador del nodo que envía el mensaje
Clave del mensaje : Especifica el tipo de información contenida en el mensaje (oferta, respuesta o solicitud de compromiso)

Tabla 6.5

Árbol de Búsqueda

El árbol de búsqueda es una estructura de datos que permite la construcción de todas las posibles vías de inferencias que conducen a cualquiera de las hipótesis localizadas en el nivel meta del pizarrón del dominio. Cada elemento en el árbol de búsqueda tiene una parte condición y una parte acción. La parte condición describe una configuración

particular de hipótesis sobre el pizarrón. La parte acción especifica la hipótesis que será creada o modificada sobre el pizarrón una vez que la parte condición ha sido satisfecha. Una vía de inferencia es construida a partir de la concatenación de elementos del árbol de búsqueda. El elemento final de la cadena formada es la hipótesis meta que se desea alcanzar. El planeador local del nodo, a través de la fuente de conocimiento de procesamiento "*backward*", utiliza el árbol de búsqueda estructurado para la construcción de planes locales que indiquen al nodo que acciones deben ser ejecutadas para alcanzar determinadas metas.

Los componentes-programa del nodo y su funcionamiento

Los componentes-programa del nodo ejecutan los procesos de búsqueda, creación y modificación de elementos solución sobre las estructuras de datos antes descritas y llevan a cabo los procesos relacionados con el control local del nodo. Cada nodo en la red posee los siguientes componentes programas :

- ◆ Fuentes de Conocimiento de Razonamiento
- ◆ Fuente de Conocimiento Receptora
- ◆ Fuente de Conocimiento Transmisora
- ◆ Fuente de Conocimiento de Procesamiento *Backward*
- ◆ Fuente de Conocimiento Interfaz
- ◆ Fuente de Conocimiento Parada
- ◆ Planeador local
- ◆ Mecanismo de Control, el cual incluye :
 - . Fuente de Conocimiento de Control I
 - . Fuente de Conocimiento de Control II
 - . Fuente de Conocimiento de Control III

Las Fuentes de Conocimiento de Razonamiento

Las fuentes de conocimiento de razonamiento (FCraz) contienen conocimiento específico acerca del dominio del sistema. Este conocimiento lo proporcionan las reglas de inferencia que forman cada FCraz. Una FCraz está formada por una parte de identificación y una parte inferencial. La parte de identificación describe elementos de identificación de la FCraz, tales como: identificador, nivel de apareo y nivel de síntesis. La parte inferencial especifica las características de las reglas de inferencia definidas por la FCraz. Cada regla de inferencia tiene un formato condición-acción. Una FCraz es representada como una estructura de datos con los atributos definidos en la tabla 6.6.

Estructura de datos de las FCraz

Parte Identificación

Identificador de la FCraz : Número o identificador de la FCraz

Nivel de apareo : Nivel del pizarrón sobre el cual son apareadas las condiciones de la FCraz

Nivel de síntesis : Nivel del pizarrón sobre el cual es ejecutada la acción de la FCraz

Parte inferencial

Identificador de la Regla : Número o identificador de la regla

Acción : Evento sobre el pizarrón que ejecuta la regla

Condición de disparo : Configuración de elementos solución requerida para el disparo de la regla

Importancia : Certidumbre o grado de creencia asociado a la regla por el experto del dominio

Fuerza de apareo : Certidumbre o grado de creencia con el que se satisface la condición de la regla

Eficiencia : Certidumbre o grado de creencia con el que se cumple la acción de la regla

Identificador de la Regla

Acción

Condición de disparo

Importancia

Fuerza de apareo

Eficiencia

Tabla 6.6

La condición de cada regla de inferencia describe la configuración de elementos solución sobre el pizarrón necesaria para que la FCraz pueda contribuir a los procesos de solución del problema. La forma en que una FCraz contribuye a la solución del problema es especificada en la acción de cada regla de inferencia. La acción de una regla de inferencia está vinculada a la creación o modificación de elementos solución sobre el pizarrón.

La actividad de las FCraz es direccionada por los eventos. O sea, cada cambio que se ejecute sobre el pizarrón constituye un evento que, en presencia de otra información específica sobre el pizarrón, puede disparar una o más reglas de inferencia dentro de una FCraz. Cada disparo produce un único registro de activación de la FCraz (RAFRZ).

Cada vez que las condiciones de una FCraz son satisfechas, ésta ejecuta dos acciones. La primera acción consiste en la inscripción del RAFRZ en la lista de RAFCs. La segunda acción es la que la FCraz lleva a cabo una vez que el mecanismo de control ha seleccionado en la lista de RAFCs el RAFRZ asociado a la FCraz. Esta última acción consiste en la creación o modificación directa de algún elemento solución sobre el pizarrón del dominio.

Las fuentes de conocimiento de comunicación

Cada nodo en la red recibe mensajes desde otros nodos y selecciona hipótesis sobre su pizarrón del dominio local para enviarlas hacia otros nodos. Para llevar a cabo estas actividades de comunicación, la arquitectura define para cada nodo de la red dos fuentes de conocimiento de comunicación : la fuente de conocimiento receptora (FCrep) y la fuente de conocimiento transmisora (FCtrs).

La Fuente de Conocimiento Receptora (FCrep)

La FCrep decide cuáles mensajes, de los que hasta el momento han sido recibidos en el buzón de mensajes de entrada (BME), resultan de interés para la solución del problema actual y los pone a la consideración del mecanismo de control.

La FCrep posee un formato condición-acción. La condición es satisfecha cuando un mensaje de interés para la solución del problema actual es encontrado en el BME. Cada vez que la condición de la FCrep ha sido satisfecha ésta ejecuta dos acciones. La primera acción consiste en la inscripción de un RAFRE en la lista de RAFCs. La segunda acción es llevada a cabo cuando el mecanismo de control selecciona el RAFRE inscrito para su ejecución. La ejecución de un RAFRE depende del tipo de mensaje que contiene. Si el mensaje es del tipo **oferta** o **respuesta-a-compromiso**, entonces la ejecución del RAFRE está asociada a la creación o modificación de elementos solución sobre el pizarrón del dominio. Si el mensaje es del tipo **solicitud-de-compromiso**, entonces la ejecución del RAFRE está asociada a la creación de un registro del tipo **investigar-hipótesis** sobre la base de datos de control.

La Fuente de Conocimiento Transmisora (FCtrs)

La FCtrs transmite mensajes tipo **oferta**, **respuesta-a-compromiso** o **solicitud-de-compromiso** hacia otros nodos de la red. Los mensajes tipo **oferta** o **respuesta-a-compromiso** son hipótesis seleccionadas sobre el pizarrón del dominio, para cada una de las cuales la FCtrs encuentra sobre el pizarrón del control un registro del tipo **transmitir-hipótesis**. Los mensajes del tipo **solicitud-de-compromiso** están asociados a registros del tipo **preguntar-hipótesis**, los cuales han sido previamente creados sobre la base de datos de control.

Al igual que la FCrep, la FCtrs también posee un formato condición-acción. La condición de la FCtrs es satisfecha cuando algún registro del tipo **transmitir-hipótesis** o **preguntar-hipótesis** es hallado sobre la base de datos de control. Cuando la FCtrs logra satisfacer su condición, ésta se dispara creando inicialmente un RAFTR en la lista de RAFCs. Una vez que dicho RAFTR es seleccionado por el mecanismo de control, la FCtrs ejecuta su acción final, la cual consiste en la transmisión del mensaje asociado al RAFTR hacia algún nodo o subconjunto de nodos de la red.

Cuando un RAFTR ha sido seleccionado para su ejecución por el mecanismo de control, la FCtrs decide a qué nodo o nodos de la red debe enviar el mensaje asociado al RAFTR, mediante consulta realizada al modelo del mundo externo que posee el nodo. El modelo del mundo externo es una estructura de datos definida por el nodo local, a la cual sólo la FCtrs tiene acceso. El modelo del mundo externo describe las principales características, habilidades, conocimientos y recursos de los restantes nodos de la red.

Cuando un registro del tipo **transmitir-hipótesis** ha sido creado sobre la base de datos de control, la FCtrs investiga la naturaleza del mensaje asociado al registro. Esto es, si el mensaje es tipo oferta o respuesta a compromiso. Si el mensaje es del tipo respuesta a compromiso, entonces la FCtrs conoce a qué nodo de la red debe enviar este mensaje, no obstante, la FCtrs consulta el modelo del mundo externo con el objetivo de conocer si para algún otro nodo de la red pudiera resultar valiosa esta información. Si el mensaje es del tipo oferta, entonces es alguna hipótesis de interés que ha sido seleccionada desde el pizarrón del dominio. En este caso la FCtrs consulta el modelo del mundo externo para conocer a qué nodos de la red podría resultar de interés la información obtenida, y transmitirla a los nodos. Cuando un registro del tipo **preguntar-hipótesis** ha sido creado sobre el pizarrón del control, entonces el mensaje asociado a éste es del tipo solicitud de compromiso. En este caso, la FCtrs consulta el modelo del mundo externo para determinar cuál es el nodo que posee las mejores habilidades y recursos para elaborar una respuesta a dicho compromiso y envía a este nodo el mensaje en cuestión.

La fuente de conocimiento de procesamiento *backward*

Como ya hemos mencionado, la actividad de las fuentes de conocimiento de razonamiento (FCraz) es direccionada por los eventos que ocurren sobre el pizarrón del dominio. Este tipo de actividad constituye un proceso de encadenamiento de reglas hacia adelante ("*forward*") encaminado a la creación de elementos solución sobre el pizarrón del dominio, para el problema que se intenta resolver. Al igual que las FCrazs, la fuente de conocimiento de procesamiento *backward* (FCback) también contribuye a la creación de elementos solución sobre el pizarrón del dominio, aunque a diferencia de las primeras, la actividad de FCback es direccionada por las metas y planes, y no por los eventos que ocurren sobre el pizarrón.

La actividad de la FCback constituye un proceso de encadenamiento de reglas hacia atrás ("*backward*") encaminado hacia la construcción de un árbol de secuencias para generar los elementos solución. Las secuencias son iniciadas con la configuración o estado meta en la raíz del árbol. El próximo nivel del árbol es generado encontrando todas las reglas cuyos consecuentes o acciones aparecen con el nodo raíz. De poder aplicar sólo estas reglas entonces se podría generar el estado meta deseado. Los antecedentes o condiciones de estas reglas son utilizados para generar los nodos en este segundo nivel del árbol. El próximo nivel del árbol es generado tomando cada nodo en el nivel previo y encontrando todas las reglas cuyos consecuentes o acciones corresponden con éste, entonces se usan los antecedentes de las reglas correspondientes para generar los nuevos nodos en el próximo nivel del árbol. El proceso continúa hasta

generar una configuración de elementos solución que sólo correspondan con estados iniciales.

Para construir el árbol de secuencias, la FCback utiliza la información registrada sobre el árbol de búsqueda estructurado del nodo, seleccionando dentro de todas las vías de inferencia que conducen al nivel meta del pizarrón sólo aquellas que generan los elementos solución necesarios para alcanzar la hipótesis meta deseada.

La FCback posee un formato condición-acción. La condición se satisface cuando algún registro del tipo **investigar-hipótesis** es creado sobre la base de datos de control. La creación de un registro del tipo **investigar-hipótesis** está asociada a la ocurrencia de alguna de las siguientes situaciones en el nodo local :

1. No existe suficiente información (elementos solución) en los niveles inferiores del pizarrón del dominio, que garanticen el trabajo continuo de las FCraz en la creación o modificación de elementos solución en niveles superiores del pizarrón. Por lo tanto, los procesos encaminados hacia la búsqueda de la solución del problema han sido detenidos.
2. El nodo local ha recibido alguna solicitud de compromiso de alguno de los restantes nodos de la red, acerca de algún elemento solución para el cual aún no posee información, pero sí posee las vías de inferencia que permiten la creación de dicho elemento.

Ante la ocurrencia de alguna de estas situaciones, la FCback se dispara, e inicia la construcción de un árbol de secuencias de generación de elementos solución, asignando a la raíz del árbol el elemento meta asociado al registro **investigar-hipótesis** que provocó el disparo de la FCback. Una vez que la FCback ha concluido la construcción del árbol de secuencias, el planeador local del nodo evalúa un grupo de criterios heurísticos para determinar en qué orden generar los nodos del árbol de secuencias (formación de un plan). Como veremos más adelante, el trabajo del planeador local está estrechamente vinculado al trabajo de la FCback

Cuando un nodo del árbol de secuencias ha sido seleccionado para completar su resolución, la FCback ejecuta una primera acción la cual consiste en inscribir un registro de activación (RAFPB) en la lista de RAFCs, que indica cuál elemento solución debe ser preguntado al usuario del nodo o a otros nodos de la red, según sea el caso. Cuando el RAFPB inscrito es seleccionado por el mecanismo de control, la FCback ejecuta su acción final, la cual consiste en la creación de un registro del tipo **preguntar-hipótesis** sobre la base de datos de control. La creación de este último registro dispara a la FCtrs, la cual se encarga de transmitir la pregunta al nodo correspondiente de la red. La posterior creación sobre el pizarrón del dominio de este elemento solución contribuirá a aumentar o a completar la medida de resolución del nodo seleccionado dentro del árbol de secuencias. La generación de nodos no resueltos del árbol significa la creación de elementos solución deseables sobre el pizarrón del dominio

La fuente de conocimiento "Interfaz"

La fuente de conocimientos interfaz (FCitfz) establece protocolos de comunicación entre el nodo local y el usuario del nodo. FCitfz posee un formato condición-acción. La condición de FCitfz es satisfecha cuando alguna de las siguientes situaciones ha ocurrido :

1. El usuario ha solicitado al nodo introducir conocimientos relevantes acerca del problema que intenta resolver. De forma general, estos conocimientos se traducen a la creación de elementos solución en los niveles inferiores del pizarrón.
2. El usuario ha solicitado al nodo la investigación de alguna meta particular.
3. El usuario ha solicitado al nodo información acerca del estado de la solución del problema.
4. El nodo requiere enviar al usuario un mensaje tipo solicitud de compromiso.
5. El nodo oferta al usuario información acerca del estado de la solución del problema.

Cuando se ha generado alguna de las situaciones 1, 2 ó 3, FCitfz se dispara y establece los protocolos de comunicación necesarios para que el usuario pueda introducir la información deseada. Ante la ocurrencia de las situaciones 1 ó 2, la acción final de FCitfz consiste en registrar en el buzón de mensajes de entrada la información proveniente del usuario, para su posterior procesamiento por parte de FCrep. La situación 3 es ejecutada totalmente por la propia FCitfz, y el resultado final de ésta es informar al usuario acerca del estado actual de la solución del problema. Por otra parte, ante la generación de las situaciones 4 ó 5 FCitfz se dispara y establece los protocolos de comunicación necesarios para que el nodo pueda transmitir sus mensajes al usuario. Si ha ocurrido la situación 4, la acción final de FCitfz consiste en registrar en el buzón de mensajes de entrada la respuesta del usuario acerca del elemento en cuestión. La situación 5 es sólo una oferta de información al usuario y no requiere de una respuesta.

FCitfz sólo establece interfaz de comunicación entre el nodo y el usuario del nodo, ya que cuando la comunicación se lleva a cabo entre dos nodos cualesquiera de la red, la interfaz de comunicación es establecida por las fuentes de conocimiento de comunicación (FCrep y FCirs). Éstas se encargan de establecer los protocolos necesarios que garanticen una comunicación eficiente entre el nodo emisor y el nodo receptor.

La fuente de conocimiento "Parada"

La fuente de conocimiento parada (FCpar) es invocada cuando alguna de las siguientes situaciones ha ocurrido en el nodo local :

1. Los procesos encaminados hacia la búsqueda de la solución del problema se han detenido, al no existir suficiente información (elementos solución) en los niveles inferiores del pizarrón. Por lo tanto, la creación o modificación de elementos solución en niveles superiores del pizarrón no puede ser llevada a cabo.
2. Se han agotado totalmente los recursos para la solución del problema y/o se ha excedido el tiempo de espera para recibir respuesta acerca de algún compromiso solicitado a otro nodo de la red, sin que se haya logrado alcanzar una solución al problema.
3. Una solución confiable para el problema actual fue alcanzada.

FCpar también posee un formato condición-acción. La condición es satisfecha cuando alguna de las situaciones antes descritas ha tenido lugar. El contenido de la acción depende de la naturaleza de la situación que ha sido responsable del disparo de FCpar.

Si el disparo de FCpar ha sido provocado por la situación 1, entonces la acción de ésta consiste en la creación de un registro del tipo **investigar-hipótesis** sobre la base de datos de control, el cual a su vez constituirá una condición de disparo para el planeador local y FCback. La acción conjunta de estos últimos consiste en seleccionar, entre todas las metas del nivel meta del pizarrón del dominio, aquella que posea una mayor medida de resolución y la posterior generación de un plan encaminado a la creación de los elementos solución necesarios para completar la resolución de la meta seleccionada.

Si ha ocurrido la situación 2, esto significa que los recursos para la solución del problema actual se han agotado de forma temporal o permanente, sin que se haya alcanzado una solución aceptable para el problema en cuestión. Este es el caso en el cual la información existente sobre el pizarrón es insuficiente para continuar la creación de elementos solución que permitan al nodo, por sí solo, alcanzar la solución del problema. Por lo tanto, el nodo envía mensajes tipo solicitud de compromisos hacia otros nodos apropiados y espera durante un intervalo de tiempo dado la llegada de alguna respuesta que le permita continuar con la creación o modificación de elementos solución. Si el tiempo de espera ha concluido y el nodo no ha recibido ninguna respuesta a los compromisos solicitados ni otra información de oferta que le permita continuar con los procesos de solución del problema, entonces se forma un contexto de disparo para FCpar. Ante esta situación, la FCpar se dispara ejecutando las siguientes acciones :

- ◆ Almacenar el estado actual de la solución del problema (configuración actual del pizarrón del dominio y base de datos de control) Esto permite la

posterior recuperación del problema ante la existencia de condiciones que permitan avanzar en la solución.

- ◆ Retomar un problema pendiente o iniciar un nuevo problema. Si en el buzón de mensajes de entrada (RME) existen mensajes registrados para un problema previamente almacenado, entonces este problema (última configuración del pizarrón y base de datos de control) es recuperado y se reinician los procesos de solución. Si en el RME no existen mensajes para alguno de los problemas que han sido previamente almacenados, pero existen mensajes para algún nuevo problema, entonces se selecciona el nuevo problema para el cual exista una mayor cantidad de mensajes y se inician los procesos de solución.

Cuando ha ocurrido la situación 3, significa que una solución confiable para el problema ha sido alcanzada. Ante esta situación la FCpar se dispara almacenando opcionalmente el estado final de la consulta, una vez que FCitfz ha informado al usuario del nodo los resultados alcanzados. Si existen en el RME mensajes registrados para algún problema previamente almacenado, entonces es recuperado y se reinician los procesos para su solución. Si no existen mensajes que permitan la recuperación de algún problema previamente almacenado, pero existen mensajes para alguno nuevo, entonces se selecciona un nuevo problema y se inician los procesos para su solución.

El planeador local

La función del planeador es la generación de planes locales orientados a la obtención de metas particulares. Un plan es una secuencia de actividades que deben llevar a cabo determinadas fuentes de conocimientos del nodo, con la finalidad de crear sobre el pizarrón los elementos solución necesarios que permitan al nodo alcanzar la meta deseada. Para iniciar la creación de un plan local, el planeador utiliza a FCback para crear el árbol de secuencias de generación de elementos solución. Los nodos ubicados en los diferentes niveles del árbol de búsqueda corresponden a elementos solución particulares, algunos de los cuales pudieran existir ya sobre el pizarrón del dominio y otros que deben crearse. La creación de cada elemento solución es una actividad que requiere del trabajo de una o varias fuentes de conocimiento específicas. Una vez creado el árbol de secuencias, el planeador evalúa criterios heurísticos para decidir en qué orden ejecutar estas actividades. De esta forma, concluye la generación del plan local.

Al igual que las fuentes de conocimiento, el planeador local posee una estructura condición-acción. La condición es satisfecha cuando alguna de las siguientes situaciones ha ocurrido :

1 Los procesos encaminados hacia la búsqueda de la solución del problema se han detenido al no existir suficiente información sobre el pizarrón del dominio. Esto significa que el trabajo de las fuentes de conocimiento de razonamiento, direccionado

por los eventos, no puede ser llevado a cabo, siendo necesario un direccionamiento por las metas que permita la creación de nuevos elementos solución sobre el pizarrón del dominio.

2. Sobre la base de datos de control ha sido creado un registro del tipo **investigar-hipótesis**. La creación de este registro ha sido causada por alguno de los siguientes hechos :

- ◆ El nodo se ha comprometido con algún otro nodo de la red a la investigación de alguna meta o submeta determinada.
- ◆ El nodo ha recibido de su usuario una solicitud para entrar a la investigación de alguna meta o submeta determinada.

Cuando la situación 1 ha tenido lugar, el planeador, haciendo uso de sus criterios heurísticos, determina cuál meta debe entrar a investigar y crea entonces un registro del tipo **investigar-hipótesis** sobre la base de datos de control.

Una vez que la condición del planeador local ha sido satisfecha, la acción se manifiesta al crear un plan apropiado que concibe una relación de actividades y ejecutores para estas actividades.

El primer criterio heurístico que evalúa el planeador local es el cálculo de una medida de resolución para cada nodo no resuelto en el árbol de secuencias. Como su nombre lo indica, la medida de resolución refleja el grado en que un nodo ha sido resuelto, ésta es un valor entre 0 y 1, y su cálculo viene dado por la siguiente expresión :

$$R_{ij} = \frac{CNR_{j,i}}{CTN_{j,i}}$$

donde :

R_{ij} = medida de resolución del nodo i perteneciente al nivel j del árbol de secuencias

$CNR_{j,i}$ = cantidad de nodos resueltos en el nivel $j-1$ correspondientes al antecedente de alguna regla cuya acción es la creación del nodo i en el nivel j

$CTN_{j,i}$ = cantidad total de nodos en el nivel $j-1$ correspondientes al antecedente de alguna regla cuya acción es la creación del nodo i en el nivel j

Podría existir más de una vía de inferencia (regla) que permitiera la creación de algún nodo i en el nivel j . Esto a su vez implicaría la existencia de más de una medida de resolución para el nodo i . Cuando este sea el caso, la medida de resolución del nodo i se tomará como el máximo entre las medidas de resolución aportadas por el grupo de reglas. Una vez calculada la medida de resolución para todos los nodos inferibles

pertenecientes a un nivel j del árbol, entonces se procede a la aplicación de alguno de los restantes criterios heurísticos aportados por el planeador local. Estos criterios heurísticos tienen como objetivo común la selección de aquellas reglas de inferencia que aportarán mayor cantidad de información para resolver el problema, y por lo tanto, contribuirán a alcanzar la solución del problema de una forma que se acerca al óptimo.

Los restantes criterios heurísticos aportados por el planeador local son los siguientes :

- ◆ Heurística Global
- ◆ Heurística Diferencial
- ◆ Heurística de Aporte General
- ◆ Heurística del Valor Propio
- ◆ Heurística de Estimación de Umbral

Mientras que la Medida de Resolución centra su atención sobre qué elemento solución debe ser creado sobre el pizarrón del dominio, los restantes criterios heurísticos centran su atención sobre las reglas, de las que son capaces de crear el elemento solución deseado, que resultan ser las más factibles para su selección a partir de toda la información existente sobre el pizarrón del dominio, exceptuando el criterio de Heurística Global, el cual se basa en la selección de la meta o submeta más apropiada dentro de cada nivel del árbol de búsqueda.

Heurística Global

Este criterio heurístico consiste en la selección constante de la meta o submeta que se entrará a analizar, a partir de la evaluación de toda la información disponible sobre el pizarrón del dominio (elementos solución). El criterio se basa en la determinación de cuál de todas las metas (submetas) para las cuales se ha generado información sobre el pizarrón, posee un mayor valor de certidumbre asociado, esto posibilita focalizar la búsqueda de la solución del problema en función de dicha meta (submeta). El éxito de este criterio heurístico depende fuertemente de la magnitud de la información aportada de forma espontánea por el usuario del nodo o por otros nodos de la red a los procesos de solución del problema (creación de elementos solución sobre el pizarrón), mientras mayor sea esta magnitud, mayor será la probabilidad de éxito del criterio de heurística global.

Heurística Diferencial

Este criterio de búsqueda consiste en priorizar la investigación de aquellas reglas de inferencia, que al dispararse, serían capaces de establecer una mayor diferenciación entre las metas (submetas) de la base de conocimientos. Esto es, la aplicación del criterio de heurística diferencial focaliza la selección de aquellas reglas de inferencia capaces de aumentar el nivel de certidumbre asociado a la meta (submeta) en análisis y que a la vez son capaces de disminuir las evidencias acerca del cumplimiento de otras

metas (submetas) en la base de conocimientos, e incluso contribuir a descartar el cumplimiento de algunas de estas.

Heurística de Aporte General

Este criterio de búsqueda se basa en priorizar la investigación de aquellas reglas de producción que serían capaces, una vez que se han disparado, de aportar información a mayor cantidad de metas (submetas) en la base de conocimientos. El criterio consiste en determinar cuál regla, de aquellas que apuntan con certidumbre mayor que cero a la meta en análisis, es capaz de aportar evidencias a una mayor cantidad de metas de la base de conocimientos, sin tener en cuenta si las evidencias contribuyen a demostrar el cumplimiento o no cumplimiento de estas metas. Este criterio permite una evaluación paralela de aquellas metas a las que apunta la regla seleccionada.

Heurística del Valor Propio

La búsqueda en función del valor propio significa la selección de aquellas reglas "más valiosas" para lograr la demostración de la meta en análisis. Este criterio de búsqueda se fundamenta en el análisis de dos atributos fundamentales de la regla : certidumbre máxima de contribución y frecuencia de cumplimiento, y un atributo de la meta (submeta) : umbral estimado. Una vez calculada la diferencia entre el umbral estimado y el valor actual de la meta, se procede a la selección de aquellas reglas de inferencia que serían capaces, cuando se han disparado, de provocar que la meta (submeta) alcance su umbral estimado. De todas las reglas que han logrado satisfacer esta condición, se selecciona para su investigación aquella que posea una mayor frecuencia de cumplimiento. Esto posibilitará una reducción del tiempo en la demostración de la meta de interés. Este criterio de búsqueda resulta especialmente beneficioso en la demostración de submetas.

Heurística de Estimación de Umbral

La búsqueda heurística en función del umbral estimado se manifiesta a partir de la cantidad de certeza con la que se desea llegar a demostrar la meta (submeta) en análisis. El umbral estimado se obtiene a partir de los valores de certidumbre de las reglas que conducen a la meta en cuestión y de un valor o constante de aceptación "C".

Existen varios criterios para determinar el umbral estimado :

1. Criterio en función de la certidumbre. Consiste en determinar el máximo valor de certidumbre a alcanzar por la meta en análisis y posteriormente calcular el "C%" de este valor.
2. Criterio en función de la frecuencia. Consiste en determinar el "C%" de las reglas de inferencia que conducen a la meta en análisis, una vez que éstas han sido

ordenadas ascendentemente, y posteriormente determinar el máximo valor de certidumbre posible a alcanzar por tales reglas.

3. Criterio en función de la certidumbre y la frecuencia. Primero se aplica el criterio en función de la frecuencia y posteriormente se determina el "C%" del valor obtenido.

Como se comprenderá, el uso de estos criterios heurísticos hacen más eficientes los procesos de solución del problema, ya que permiten focalizar la información más valiosa y capaz de acortar la búsqueda en el espacio de soluciones, en ocasiones logrando descartar grandes porciones de la base de conocimientos, otras veces aportando información para gran cantidad de metas, o seleccionando las reglas de inferencia imprescindibles y que más frecuentemente se cumplen para contribuir a la creación de elementos solución.

El mecanismo de control

Cuando una fuente de conocimiento se dispara, crea un registro de activación (RAFRZ, RAFRE, RAFTR o RAFPB) que describe las características asociadas a la acción que la fuente de conocimientos pretende ejecutar. En cada ciclo de ejecución sólo un RAFC es ejecutado. Para determinar el orden en el cual las fuentes de conocimiento ejecutarán sus acciones finales, el sistema de control establece una competencia entre los RAFCs inscritos, a partir del cálculo de una oferta para cada RAFC. La fuente de conocimiento asociada al RAFC de mayor oferta será la seleccionada para ejecutar su acción final en ese ciclo de ejecución.

El sistema de control está compuesto por tres fuentes de conocimientos de control básico [HAY85] : fuente de conocimiento de control I, fuente de conocimiento de control II y fuente de conocimiento de control III, las cuales iteran en un ciclo de tres pasos para ejecutar respectivamente las siguientes actividades :

1. Cálculo de la oferta para cada RAFC inscrito.
2. Selección del RAFC con mayor oferta.
3. Invocación de la fuente de conocimientos asociada al RAFC seleccionado.

Debido a que existen varios tipos de registros de activación (RAFRZ, RAFRE, RAFTR y RAFPB), pudiera ocurrir que en más de uno de estos bloques estructurales existan registros de activación compitiendo por ejecutar su acción. Dada esta situación, una selección jerárquica ha sido establecida para seleccionar el bloque de RAFC dentro del cual debe ser seleccionado el registro con mayor oferta. La jerarquía establecida ha sido la siguiente :

RAFRZ : prioridad 1 (mayor jerarquía)

RAFRE : prioridad 2

RAFTR : prioridad 3

RAFPB : prioridad 4 (menor jerarquía)

De modo que, sólo se pasará a ejecutar la acción de fuentes de conocimiento que hayan inscrito registros de más baja prioridad, siempre que no existan registros de mayor prioridad inscritos.

El orden o jerarquía que se ha establecido, responde al interés de perfeccionar al máximo la información interna contenida en cada nodo de la red, antes de establecer lazos de colaboración. Esto posibilita una mayor coherencia y robustez en la cooperación, se evita el envío de mensajes con información incompleta e insegura y la formulación de preguntas innecesarias.

El cálculo de la oferta para los registros de activación

Este proceso se realiza una vez que se ha decidido según la jerarquía establecida el bloque de RAFCs dentro del cual se ha de seleccionar el registro que ejecutará su acción. El cálculo de oferta es llevado a cabo por la fuente de conocimiento de control I. La condición de esta fuente de conocimiento es satisfecha cuando existen registros inscritos dentro de algún bloque, para los cuales es necesario calcular o recalcular su oferta. La acción de ésta consiste en el cálculo o recálculo de la oferta para todos los registros del bloque seleccionado.

La oferta es una función de los parámetros que componen el registro, por lo tanto, la forma de calcular la oferta dependerá del tipo de registro seleccionado.

Cálculo de la oferta para los RAFRZs. Los parámetros utilizados para calcular la oferta de los RAFRZ son los siguientes :

Certidumbre o creencia final de la hipótesis

Certidumbre o creencia actual de la hipótesis sobre el pizarrón

Fuerza de apareo de la regla

Importancia de la regla

Se procede a calcular una medida de la cantidad de información neta que se aportaría a la hipótesis sobre el pizarrón, en caso de ser permitida la acción de la fuente de conocimiento asociada al registro. A esta medida se le denomina Aporte Neto y su expresión es la siguiente :

$$\text{Aporte Neto} = \text{Certidumbre final de la hipótesis} - \text{Certidumbre actual de la hipótesis}$$

y el cálculo de la oferta se realiza según la siguiente expresión :

$$\text{Oferta} = \text{Aporte Neto} + r1.(\text{Fuerza de aparcamiento de la regla}) + r2.(\text{Importancia de la regla})$$

donde : $0 \leq r1, r2 \leq 1$

Nótese que el Aporte Neto brinda una medida de que tan "interesante" es la información que se aporta, mientras que el resto de los parámetros brinda una medida de que tan "completa" es esta información.

Cálculo de la oferta para los RAFREs. El cálculo de la oferta para los RAFREs sigue el principio de maximizar la cantidad de información interna del nodo. Por lo tanto, en el caso de que existan registros correspondientes a mensajes con clave tipo **Oferta** o **Respuesta-a-Compromiso**, éstos tendrán mayor prioridad que los mensajes registrados con clave tipo **Pregunta** o **Solicitud-de-Compromiso**.

La expresión del cálculo de la oferta para los RAFREs depende del tipo de mensaje registrado. Si el RAFRE es del tipo **Oferta** o **Respuesta-a-Compromiso**, entonces los parámetros del registro utilizados para calcular la oferta son los siguientes :

- Certidumbre o creencia final de la hipótesis
- Certidumbre o creencia actual de la hipótesis sobre el pizarrón
- Ciclo de activación

y el cálculo de la oferta se realiza según la siguiente expresión :

$$\text{Oferta} = \text{Aporte Neto} + r1. \text{Retardo}$$

donde :

$$\text{Aporte Neto} = \text{Certidumbre final de la hipótesis} - \text{Certidumbre actual de la hipótesis}$$

$$\text{Retardo} = \text{Ciclo Actual} - \text{Ciclo de Activación}$$

$$0 \leq r1 \leq 1$$

En este caso, la oferta brinda una medida de que tan "interesante" es la información que se le ofrece al nodo más una ponderación del tiempo que ha transcurrido desde que el registro fue creado. No se posee una medida de que tan completa es esta información, este conocimiento sólo lo posee el nodo que envía el mensaje. Sin embargo, esta completitud la intenta garantizar el nodo que envía la información, a partir de los criterios seguidos por el mecanismo de control, el cual prioriza que la información esté lo más completa posible antes de enviarla a otros nodos.

Si el RAFRE es del tipo **Pregunta** o **Solicitud-de-Compromiso**, entonces los parámetros del registro utilizados para calcular la oferta son los siguientes :

Ciclo de activación
Ciclo actual

y la oferta es calculada de acuerdo a la siguiente expresión :

$$\text{Oferta} = r1 \cdot \text{Retardo}$$

donde :

$$\text{Retardo} = \text{Ciclo actual} - \text{Ciclo de activación}$$

$$0 \leq r1 \leq 1$$

La oferta calculada para este tipo de registros es una función del retardo. Por lo tanto, registros inscritos en los primeros ciclos de ejecución tendrán una mayor prioridad que los registros inscritos en ciclos posteriores.

Cálculo de la oferta para los RAFTRs. En el cálculo de la oferta para los RAFTRs se sigue el principio de maximizar la cantidad de información disponible del sistema en su conjunto, evitando la formulación de preguntas o la solicitud de compromisos mientras que se pueda aportar información de interés a otros nodos (ofertas o respuestas a compromisos). Por lo tanto, existen prioridades entre los diferentes RAFTRs en dependencia del valor de la clave de la información que se desea transmitir (pregunta o solicitud- de-compromiso y oferta o respuesta-a-compromiso).

El orden de prioridades es el siguiente :

1. Envío de información tipo Oferta o Respuesta a Compromiso
2. Envío de información tipo Pregunta o Solicitud de Compromiso a otros nodos
3. Envío de información tipo Pregunta o Solicitud-de-Compromiso al usuario del nodo

El cálculo de la oferta para los RAFTRs está asociado a dos expresiones, en dependencia del valor de la clave del mensaje.

Si el RAFTR tiene clave tipo **Oferta** o **Respuesta-a-Compromiso**, entonces los parámetros utilizados para calcular la oferta son los siguientes.

Ciclo de activación
Ciclo actual
Certidumbre o valor de creencia asociado al mensaje

calculándose la oferta de acuerdo a la siguiente expresión :

$$\text{Oferta} = \text{Certidumbre asociada al mensaje} + rI.\text{Retardo}$$

donde :

$$\text{Retardo} = \text{Ciclo actual} - \text{Ciclo de activación}$$

$$0 \leq rI \leq 1$$

Para valores de r grandes (cercaos a 1) el parámetro Retardo toma un peso predominante en la expresión de la oferta, de forma tal que mensajes inscritos en ciclos iniciales de ejecución tengan una mayor probabilidad de ser seleccionados para su envío y evitar, de esta forma, la demora en la transmisión de información a otros nodos de la red. El parámetro Certidumbre asociada al mensaje es una medida de la eficiencia (calidad) del mensaje a transmitir.

Si el RAFTR tiene clave tipo Pregunta o Solicitud de Compromiso, entonces la oferta es calculada como una función del retardo :

$$\text{Oferta} = rI.\text{Retardo}$$

donde :

$$\text{Retardo} = \text{Ciclo actual} - \text{Ciclo de activación}$$

$$0 \leq rI \leq 1$$

Cálculo de la oferta para los RAFPBs. La oferta calculada para los RAFPBs es una función del retardo, su expresión es la siguiente :

$$\text{Oferta} = rI.\text{Retardo}$$

donde :

$$\text{Retardo} = \text{Ciclo actual} - \text{Ciclo de activación}$$

$$0 \leq rI \leq 1$$

La oferta calculada para este tipo de registros sólo considera el retardo (cantidad de ciclos que han transcurrido desde que el registro fue inscrito), ya que la importancia y factibilidad del registro son previamente determinadas por el planeador local del nodo antes de su inscripción

Los métodos antes propuestos para el cálculo de la oferta en los diferentes RAFCs no son métodos completos y exhaustivos, todavía pueden llegar a ser mucho más robustos

al considerar dentro de sus parámetros otros elementos de interés capaces de aportar mayor información en el cálculo de la oferta. No obstante, el trabajo experimental en la solución de problemas ha demostrado que el uso de estos métodos garantiza una mayor coherencia en los procesos cooperativos de la red.

Selección del RAFC a ejecutar

Una vez calculada la oferta para todos los registros correspondientes al bloque seleccionado (según el criterio jerárquico descrito), corresponde a la fuente de conocimiento de control II llevar a cabo la selección del registro con mayor oferta para su ejecución.

Ejecución del RAFC seleccionado

Cuando el registro de mayor oferta ha sido seleccionado para su ejecución, la fuente de conocimiento de control III invoca a la fuente de conocimiento responsable de la creación de dicho registro para que ésta ejecute su acción final : la creación o modificación del elemento solución correspondiente sobre el pizarrón del dominio.

De esta forma, las tres fuentes de conocimiento de control básico iteran en un ciclo de tres pasos, determinando en cada ciclo de ejecución qué registro de activación ha de ser ejecutado y llevando a cabo su ejecución.

Representación y Procesamiento de la Incertidumbre en REDSIEX

En REDSIEX, la incertidumbre es representada utilizando factores de certidumbre. [BUC85, SHO76, SHO85] A cada regla contenida dentro de cada fuente de conocimiento de razonamiento (FCraz) se le asocia un factor de certidumbre, una medida de la magnitud en la cual el antecedente de la regla soporta al consecuente. De igual forma, la certidumbre asociada a cada hecho o hipótesis sobre el pizarrón del dominio se expresa mediante un factor de certidumbre, que denotaremos como valor de creencia asociado a la hipótesis.

Para combinar las estimaciones de certidumbre presentes en las reglas y producir una estimación final de la certidumbre asociada a cada hipótesis sobre el pizarrón, REDSIEX define para cada uno de sus nodos la siguiente estructura de combinación :

Funciones para el cálculo de la certidumbre del antecedente de la regla:

$$NEG(c) = -c$$

$$CONJ(c_1, c_2, \dots, c_n) = MIN(c_1, c_2, \dots, c_n)$$

$$DISY(c_1, c_2, \dots, c_n) = MAX(c_1, c_2, \dots, c_n)$$

Función para el cálculo de la contribución de la regla :

$$CONT(a,c) = c * max(0,a)$$

Función para el cálculo del efecto global de varias reglas :

$$c_1 + c_2 * (1 - c_1) \quad \text{si } c_1, c_2 \geq 0$$

$$EGLOB(c_1, c_2) = \begin{cases} c_1 + c_2 * (1 + c_1) & \text{si } c_1, c_2 \leq 0 \end{cases}$$

$$(c_1 + c_2) / (1 - \min(\text{abs}(c_1), \text{abs}(c_2))) \quad \text{si } c_1 * c_2 \leq 0$$

donde :

$c_1, c_2, \dots, c_n, c, a$ son valores de certidumbre definidos en el intervalo real $[-1, 1]$.

Cada vez que una FCraz ha logrado satisfacer alguna de sus condiciones sobre determinada configuración de hipótesis sobre el pizarrón del dominio, la estructura de combinación es invocada. Esto trae como resultado que sean combinadas las estimaciones de certidumbre presentes en las hipótesis que lograron satisfacer el antecedente de la regla que provocó el disparo de la FCraz y el factor de certidumbre asociado a la conclusión de la regla, produciéndose las siguientes estimaciones parciales y finales de certidumbres .

- ◆ fuerza de apareo de la regla
- ◆ eficiencia de la regla
- ◆ creencia final asociada a la hipótesis

La fuerza de apareo de la regla es una medida de la fuerza con que ha sido satisfecho el antecedente de la regla. Esta es obtenida a partir del cálculo de la certidumbre del antecedente de la regla, utilizando para esto las funciones NEG, CONJ y DISY, según la expresión del antecedente

La eficiencia de la regla es la magnitud en que la regla contribuye al valor de certidumbre o creencia de la hipótesis sobre el pizarrón. Esta medida de eficiencia es obtenida a partir de la función de contribución CONT

La creencia final asociada a la hipótesis es obtenida al calcular el efecto global (EGLOB) entre el valor de certidumbre con el que contribuye la regla a la hipótesis y el

valor actual de certidumbre de la hipótesis (aportado por otras reglas internas o nodos de la red).

Estas tres estimaciones de certidumbre son inscritas como parte de los campos del RAFC creado por la FCraz, una vez que ésta se ha disparado y posteriormente utilizadas por el mecanismo de control en el cálculo de la oferta. Cuando el RAFC creado es seleccionado por el mecanismo de control, la FCraz asociada al mismo crea o modifica una hipótesis sobre el pizarrón del dominio, asignándole a esta hipótesis el valor de certidumbre contenido en el campo "creencia final asociada a la hipótesis".

La revisión de creencias

Una de las principales características que hacen de REDSIEX un sistema consistente en la solución de problemas es su proceso de revisión de creencias.

La revisión de creencias es el proceso mediante el cual son ajustados los valores de creencias de determinadas hipótesis, creadas sobre el pizarrón del dominio, en base a modificaciones efectuadas en el valor de creencia de una o varias hipótesis, que han aportado evidencias para la creación de las primeras. En otras palabras, si el valor de creencia de una hipótesis sobre el pizarrón es modificado, entonces los valores de creencia de las hipótesis encontradas en todas las vías de inferencia con encadenamiento "*forward*" a partir de esta hipótesis, también serán modificados en base a este nuevo valor.

Pasemos ahora a ver el proceso de revisión de creencias en REDSIEX en un mayor nivel de detalle.

Cuando la modificación del valor de creencia de una hipótesis sobre el pizarrón ha tenido lugar, el sistema consulta si esta hipótesis ha contribuido a la creación de alguna otra hipótesis en ese mismo nivel o en niveles superiores del pizarrón. Si este es el caso, entonces un procesamiento "*forward*" es ejecutado, mediante el cual vuelven a dispararse las reglas cuyas condiciones ya habían sido satisfechas, dado el valor de creencia anterior de la hipótesis.

El proceso de revisión de creencias requiere de la consulta de los elementos tipo **revisión de creencias**, registrados sobre la base de datos de control. Estos elementos son:

LIR : Lista de los identificadores de las reglas internas que han contribuido a la demostración de la hipótesis.

LCR : Lista de los valores de certidumbre con los cuales las reglas registradas en LIR han contribuido a la demostración de la hipótesis.

LIN : Lista de los identificadores de los nodos que han contribuido a la demostración de la hipótesis.

LCN : Lista de los valores de certidumbre con los cuales los nodos registrados en LIN han contribuido a la demostración de la hipótesis.

Para cada regla que vuelve a dispararse, las siguientes acciones son ejecutadas :

1. Calcular la nueva fuerza de apareo de la regla (certidumbre del antecedente).
2. Calcular la nueva eficiencia de la regla (contribución de la regla).
3. Revisar en la lista LIR, asociada a la hipótesis conclusión de la regla, si ha sido inscrito al menos el identificador de una regla diferente al de la regla en análisis.

- Si este es el caso, calcular nuevamente el efecto global entre la nueva eficiencia de la regla y los valores de certidumbre correspondiente a los identificadores inscritos en LIR. Estos valores de certidumbre se encuentran registrados en la lista LCR.

4. Revisar en la lista LIN, asociada a la hipótesis conclusión de la regla, si ha sido inscrito al menos un identificador correspondiente a algún otro nodo de la red.

- Si este es el caso, calcular nuevamente el efecto global entre el último valor de certidumbre calculado para la hipótesis conclusión y los valores de certidumbre correspondientes a los identificadores inscritos en LIN. Estos valores de certidumbre se encuentran registrados en la lista LCN.

El proceso de revisión de creencias continúa hasta que los valores de creencia correspondientes a todas las hipótesis a modificar en las vías de inferencia hayan sido modificados.

INTERMED : Una Red de Sistemas Expertos para la Consulta e Interconsulta Médica

La cooperación en el proceso de diagnóstico médico : la interconsulta médica

El diagnóstico médico

El diagnóstico médico es el proceso encaminado a la identificación o reconocimiento de una enfermedad sobre la base de los signos y síntomas presentes, con el apoyo de los estudios de laboratorio y gabinete. El diagnóstico médico indica todo el proceso de investigación del paciente, a partir de las observaciones y razonamientos del médico para determinar la enfermedad.

Diagnóstico diferencial

El diagnóstico diferencial está íntimamente relacionado con la precisión diagnóstica y por lo tanto, con la capacidad para poder agrupar enfermedades relacionadas ya sea como entidades nosológicas o síndromes. Este entra dentro del curso natural para lograr un diagnóstico final lo más depurado posible. La capacidad diagnóstica de un buen médico está relacionada directamente con su capacidad para poder ver todas las

posibilidades diagnósticas para un cierto grupo de manifestaciones clínicas (signos, síntomas, laboratorio y gabinete) y su capacidad para discernir, sobre la base de su experiencia, cuál de todos estos padecimientos es con mayor certeza la causa del proceso mórbido. La meta del diagnóstico diferencial es "arribar" a un diagnóstico específico, sobre la base de la exclusión razonada de otras enfermedades que tienen en común gran parte de los síntomas.

La interconsulta médica

La interconsulta médica es un proceso médico-administrativo mediante el cual dos o más especialistas, en diferentes dominios del conocimiento médico, cooperan con el objetivo de esclarecer un diagnóstico probable, inconcluso o dudoso. La meta final de la interconsulta es arribar a un diagnóstico preciso que permita aplicar el tratamiento adecuado a un paciente en particular.

La interconsulta médica es capaz de lograr una mayor eficiencia en la atención médica cuando los siguientes objetivos son observados :

- ◆ Mejorar el proceso de diagnóstico y/o tratamiento médico.
- ◆ Complementar el proceso diagnóstico y/o terapéutico sobre una enfermedad específica.
- ◆ Efectuar el diagnóstico y/o tratamiento sobre un padecimiento que desconoce parcial o totalmente el médico que solicita la interconsulta.

Cuando la interconsulta se efectúa con prontitud, eficiencia y plena participación profesional de los especialistas interrelacionados, se convierte en un método diagnóstico y terapéutico muy eficiente, pues logra unir en un momento dado la experticia y el conocimiento que difícilmente pueda poseer un especialista. En ocasiones debemos entender por interconsulta médica no sólo la interconsulta a un especialista, sino la interconsulta entre varios especialistas para lograr el diagnóstico y/o tratamientos adecuados.

Los siguientes hechos se encuentran asociados al proceso de interconsulta :

1. Implica la interrelación de más de un médico en el proceso directo de diagnóstico y/o tratamiento de un estado morbido en un paciente, lo que conlleva la interrelación humana, con todos sus problemas de comunicación y disponibilidad.

2. Jerarquía durante la interconsulta - Es de suponer que en el momento de la interconsulta, el médico interconsultante es jerárquicamente superior al médico que solicitó la interconsulta, pues se considera que el primero posee mayor conocimiento y experiencia acerca del dominio en cuestión y contribuirá a mejorar el proceso diagnóstico y terapéutico sobre el paciente-problema.

3. Accesibilidad del interconsultante : El médico interconsultante debe tratar de ser lo más claro y explícito en sus diagnósticos y terapéuticas, con el fin de lograr una mayor comprensión por parte del médico que solicita la interconsulta.

4. Disponibilidad del interconsultante : En muchas ocasiones no está disponible el médico interconsultante y esto contribuye a retrasar el proceso diagnóstico/terapéutico.

5. A nivel institucional, el hecho de solicitar una interconsulta significa sobrecarga de trabajo para el interconsultante, ya que lo habitual es que su programación diaria de trabajo esté saturada.

Algunos conceptos y términos médicos necesarios

Para lograr una mayor comprensión del dominio de aplicación y funcionamiento del sistema INTERMED, es necesario introducir los siguientes conceptos y términos médicos relacionados: signo, síntoma, síndrome y entidad nosológica.

Signo

Evidencia objetiva de una enfermedad, en especial cuando ésta es observada e interpretada más por el médico que por el paciente. Un signo físico es una indicación de la condición corporal que puede ser percibida directamente (por ejemplo, a través de la auscultación) cuando el médico examina al paciente. El signo es parte principal del proceso de diagnóstico médico, ya que su principal virtud es el ser un dato objetivo. Como ejemplo de signos podemos citar los signos asociados al diagnóstico de apendicitis : signo de McBurney, signo de Von Blumberg, signo del obturador; todos ellos se obtienen durante el proceso de exploración física y por lo tanto sólo son detectados por el médico que realiza la exploración.

Síntoma (del latín *symptoma*, del griego *symptōma*)

Es cualquier prueba subjetiva de enfermedad o del estado de un paciente. Se refiere a los datos proporcionados por el paciente, relacionados con alguna enfermedad. Son subjetivos, matizados por el padecimiento y las características psicológicas, sociológicas y biológicas del paciente. Es decir, el síntoma "dolor en la rodilla" referido por un paciente pudiera estar matizado por su capacidad para resistir el dolor, su historia sobre antecedentes de dolores semejantes y su fortaleza para soportarlo.

Durante el proceso de diagnóstico médico, los síntomas ayudan a focalizar al sistema anatómico que pudiera estar comprometido o asociado a la enfermedad del paciente,

como es el caso de los síntomas respiratorios de vías respiratorias altas : dolor de garganta (odinofagia), dificultad para deglutir (disfagia), o síntomas digestivos : pirosis, diarrea, estreñimiento. También existen síntomas totalmente inespecíficos y que pudieran estar, por lo tanto, asociados a muchas enfermedades y aunque no orientan directamente al sistema anatómico afectado, si refieren sobre la intensidad o evolución del padecimiento, por ejemplo, astenia, adinamia, anorexia, los cuales pudieran presentarse desde el inicio o aparecer durante la evolución del padecimiento.

Síndrome (del griego syndromée, concurrencia)

Grupo de signos y síntomas que se manifiestan simultáneamente y que, considerados como un todo, son característicos de cierta enfermedad. Existen procesos mórbidos o grupos de enfermedades que se caracterizan siempre por la presencia de ciertos signos y síntomas, y cuando éstos no están presentes en su totalidad, no es posible arribar a ese diagnóstico sindromático. En el proceso diagnóstico, el síndrome puede ocupar el diagnóstico final o el paso previo a este diagnóstico cuando no se puede encontrar la causa definitiva y única de ese síndrome. Como ejemplo de síndrome podemos citar el síndrome nefrótico, el cual se caracteriza por la presencia de edema generalizado, este síndrome inicialmente es percibido por el paciente (síntoma) y que durante el proceso de exploración lo describe el médico (signo), asociado a la presencia de albuminuria (albúmina en orina), e hipoalbuminemia. Si faltara alguno de estos tres datos, ya no estaríamos en presencia de un síndrome nefrótico

Entidad Nosológica (Enfermedad)

Relacionado a la nosología (del griego nósos = enfermedad y lógos = palabra, razón, tratado). La enfermedad es un dato físico y funcional manifestado por síndromes y comprobado por análisis de laboratorio y estudios de gabinete.

El dominio de aplicación de INTERMED

INTERMED es una red cooperativa de sistemas expertos que ejecuta el trabajo de un grupo de especialistas médicos en la consulta e interconsulta médica. Cada sistema experto en Intermed posee conocimientos, métodos de razonamiento y experticia para ejecutar diagnósticos dentro de un dominio reducido de enfermedades, pertenecientes a una especialidad determinada.

En INTERMED cada sistema experto o especialista es capaz de ejecutar por sí solo una tarea completa de diagnóstico, siempre y cuando posea todo el conocimiento, destreza y recursos necesarios para alcanzar la solución del problema en cuestión. Cuando éste no es el caso, el especialista que lleva a cabo la consulta médica puede solicitar ayuda a los restantes especialistas de la red, para completar los procesos de solución del problema y "arribar" así a un diagnóstico seguro. De esta forma, cada especialista o consultante médico en INTERMED está habilitado para ejecutar tareas de diagnóstico dentro de su dominio o área de interés, a la vez que posee facultades que le permiten establecer procesos de cooperación encaminados a alcanzar la solución de su problema o de problemas competentes a otros especialistas de la red.

La distribución del conocimiento y razonamiento en INTERMED

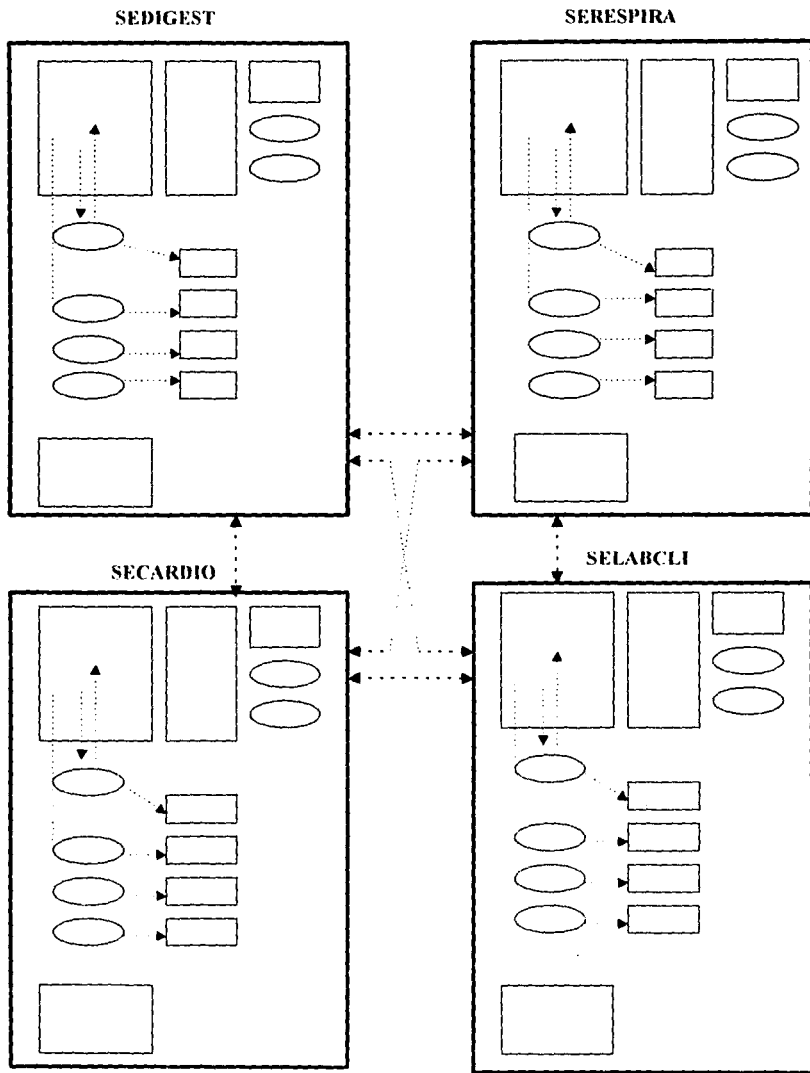
El dominio de conocimientos de INTERMED abarca conocimientos acerca de cuatro especialidades médicas : sistema digestivo, sistema respiratorio, sistema cardiovascular y laboratorio clínico. Para distribuir el conocimiento y el razonamiento correspondiente a cada una de estas especialidades, INTERMED define los siguientes sistemas expertos:

- SEDIGEST : Sistema Experto en el Diagnóstico y Tratamiento de Enfermedades del Sistema Digestivo
- SERESPIRA : Sistema Experto en el Diagnóstico y Tratamiento de Enfermedades del Sistema Respiratorio
- SECARDIO : Sistema Experto en el Diagnóstico y Tratamiento de Enfermedades del Sistema Cardiovascular
- SELABCLI : Sistema Experto para el Análisis e Interpretación de Estudios de Laboratorio Clínico

La relación de entidades fisiopatológicas y nosológicas, para las cuales INTERMED propone diagnósticos y tratamientos son relacionadas en los anexos A, B y C, mientras que el anexo D relaciona las pruebas y diagnósticos de laboratorio clínico a los que el sistema puede arribar.

La arquitectura y el funcionamiento de INTERMED

INTERMED es una aplicación desarrollada sobre REDSIEX, la Red Cooperativa de Sistemas Pizarrón presentada en el capítulo anterior. La estructura de INTERMED



—▶ Flujo de Datos

Figura 7.1 Arquitectura de INTERMED

abarca cuatro nodos expertos, los cuales han sido definidos atendiendo a los subdominios o áreas de interés que constituyen el dominio global del sistema :

SEDIGEST, SERESPIRA, SECARDIO y SELABCLI.. Cada nodo en Intermed es un sistema experto con arquitectura de pizarrón con todas las características estructurales y funcionales definidas para los nodos de REDSIEX. La figura 7.1 muestra la estructura general del sistema INTERMED.

El pizarrón del dominio

El sistema INTERMED define características estructurales y funcionales comunes para el pizarrón del dominio de los nodos SEDIGEST, SERESPIRA Y SECARDIO, mientras que las características del pizarrón del nodo SELABCLI son particulares de éste.

Como se muestra en la figura 7.2, en los nodos SEDIGEST, SERESPIRA Y SECARDIO, el pizarrón del dominio organiza los elementos solución o hipótesis en cinco niveles de abstracción :

- 1er nivel : nivel Signos-y-Sintomas Grano Grueso (SSGG)
- 2do nivel : nivel Signos-y-Sintomas Grano Fino (SSGF)
- 3er nivel : nivel Entidades Fisiopatológicas (EFP)
- 4to nivel : nivel Entidades Nosológicas (EN)
- 5to nivel : nivel Terapéutico (T)

Los elementos solución en el nivel SSGG corresponden de forma general a entradas primarias aportadas por el usuario del nodo. Estos constituyen manifestaciones objetivas o subjetivas de alguna enfermedad en particular y son expresados en muy poco nivel de detalle. Por ejemplo, dolor abdominal, vómito, disnea y fiebre.

Los elementos solución en el nivel SSGF son producto del refinamiento o especificación de los elementos solución creados en el nivel SSGG, aunque algunos de éstos pudieran ser producto de entradas primarias aportadas directamente por el usuario del nodo. Son manifestaciones objetivas o subjetivas de alguna enfermedad en particular y se expresan en el más elevado nivel de detalle. Por ejemplo, dolor en epigastrio, vómito frecuente, disnea de pequeños esfuerzos y fiebre por encima de 38 grados.

Los elementos solución en el nivel EFP representan entidades fisiopatológicas, éstos son creados a partir de configuraciones particulares de signos y síntomas en los niveles SSGG y SSGF.

Los elementos solución en el nivel EN representan enfermedades, éstas son especificaciones de entidades fisiopatológicas (nivel EFP), manifestadas por la

presencia de signos y síntomas (niveles SSGG y/o SSGF) y corroboradas por análisis de laboratorio y estudios de gabinete.

Los elementos solución en el nivel T son los planes terapéuticos o tratamientos adecuados para las entidades nosológicas generadas en el nivel EN.

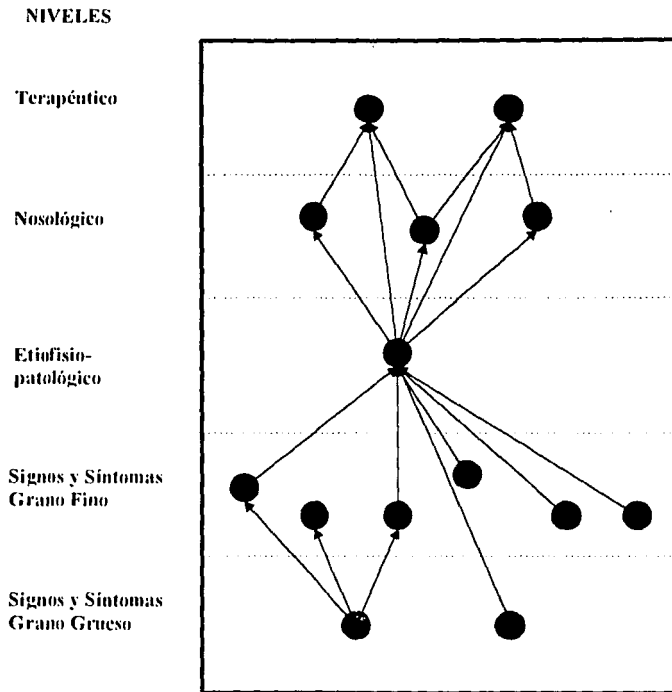


Figura 7.2 Niveles de abstracción en el Pizarrón del Dominio de los nodos SEDIGEST, SERESPIRA y SECARDIO

La figura 7.3 muestra los niveles de abstracción del pizarrón del dominio del nodo SELABCLI. Los elementos solución son organizados en cuatro niveles de abstracción :

- 1er nivel : nivel Pruebas de Laboratorio (PLAB)
- 2do nivel : nivel Datos de Laboratorio (DLAB)
- 3er nivel : nivel Diagnósticos (D)
- 4to nivel : nivel Causas (C)

Los elementos solución en el nivel PLAB corresponden a las identificaciones de las pruebas de laboratorio que son solicitadas al experto en laboratorio clínico por algún otro experto de la red. Por ejemplo, biometría hemática, química sanguínea, coproparasitoscópico, entre otras.

En el nivel DLAB se registran los valores de todos los datos de laboratorio que han sido informados para cada una de las pruebas de laboratorio en el nivel PLAB. En este nivel del pizarrón a los elementos solución no se les asocian grados de certidumbre, sino que éstos son tratados como variables y se le asignan valores. Por ejemplo, si la prueba química sanguínea ha sido solicitada y por lo tanto creada como elemento solución en el nivel PLAB entonces, una vez que se obtenga el informe correspondiente a dicha prueba (valores de urea, creatinina, ácido úrico, glucosa, amilasa, lipasa, fosfata ácida, fosfata alcalina, entre otros), cada dato y su correspondiente valor será creado como un elemento solución en el nivel DLAB.

NIVELES

Causas

Diagnóstico

Datos de
Laboratorio

Pruebas de
Laboratorio

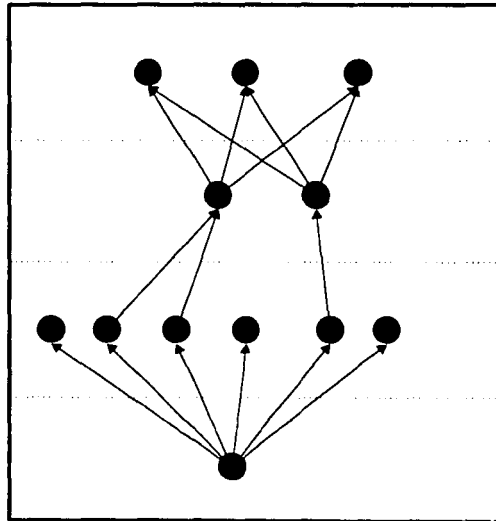


Figura 7.3 Niveles de abstracción en el Pizarrón del Dominio
nodo SELABCLI

Los elementos solución en el nivel D son los diagnósticos que ha realizado el experto en laboratorio clínico a partir de los valores de los datos de laboratorio registrados en el nivel DLAB. La cantidad de diagnósticos efectuados por cada prueba de laboratorio

depende de la magnitud o característica de los valores informados para el conjunto de datos correspondientes a dicha prueba. En el nivel D, cada diagnóstico es creado como un elemento solución con un valor de certidumbre máximo (igual a 1), esto significa que la seguridad o creencia del experto acerca del diagnóstico efectuado es absoluta. Por ejemplo, si consideramos determinados valores de los datos informados al solicitar la prueba de química sanguínea, entonces los siguientes diagnósticos pudieran resultar posibles : fosfatasa alcalina elevada, transaminasas elevadas, hiperfosfatemia, hiperglicemia, entre otros.

En el nivel C se registran las entidades, que según el criterio del experto en laboratorio clínico, podrían ser las causas más probables de los resultados diagnosticados que han sido creados como elementos solución del nivel D, y que, por lo tanto, deben ser consideradas como posibilidades diagnósticas. Los elementos solución creados como "entidades a considerar" no poseen valor de certidumbre. Estos elementos solución constituyen sugerencias hechas al experto que solicitó el estudio, para que éste decida si debe considerarlos entre sus posibilidades diagnósticas.

La arquitectura del nodo experto en laboratorio clínico define además una base de datos para cada uno de los tipos de pruebas de laboratorio que el nodo realiza, informa e interpreta. Estas son utilizadas para el registro, actualización y consulta de los datos de laboratorio informados para cada prueba efectuada.

Las fuentes de conocimiento

Para cada nodo en INTERMED se definen dos tipos de fuentes de conocimiento : fuentes de conocimiento dependientes del dominio y fuentes de conocimiento independientes de éste.

Las fuentes de conocimiento dependientes del dominio (fuentes de conocimiento de razonamiento) poseen conocimiento específico acerca del dominio del sistema y su función consiste en la creación o modificación de elementos solución sobre el pizarrón del dominio. Las fuentes de conocimiento independientes del dominio son definidas por la propia arquitectura de REDSIX.

Las fuentes de conocimiento dependientes del dominio

Para los nodos SEDIGEST, SERESPIRA y SECARDIO, INTERMED define las siguientes fuentes de conocimiento dependientes del dominio : FCrefina, FCfisiopat, FCnosología y FCterapia.

La fuente de conocimiento FCrefina

La función de la fuente de conocimiento FCrefina consiste en refinar o aumentar el nivel de detalle de signos y síntomas creados en el nivel SSGG del pizarrón del dominio.

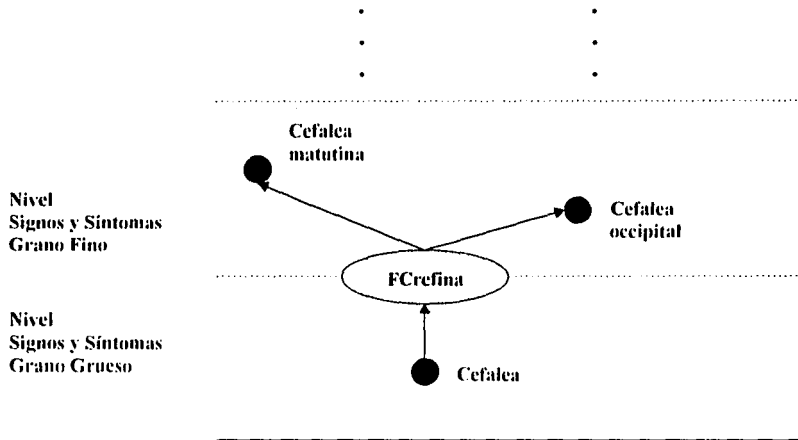


Figura 7.4 Ejemplo que muestra la actividad de la fuente de conocimiento FCrefina en el dominio de Sistema Cardiovascular

Cuando un elemento solución (signo o síntoma) ha sido creado en el nivel SSGG, la fuente de conocimiento FCrefina se dispara ejecutando las siguientes acciones :

- i. Interroga al usuario acerca de las características del signo o síntoma creado, generando de esta forma nuevos hechos asociados al signo o síntoma. Cada nuevo hecho creado expresa alguna característica de interés presente o ausente en el signo o síntoma en cuestión.
- ii. Para cada nuevo hecho creado, inscribe un registro de activación en el bloque de registros de las fuentes de razonamiento
- iii. Cuando el mecanismo de control selecciona para su ejecución alguno de los registros creados por FCrefina, ésta ejecuta su acción final, la cual consiste en la creación del elemento solución asociado a dicho registro en el nivel SSGF del pizarrón del dominio

La figura 7.4 muestra la actividad llevada a cabo por FCrefina. La actividad de esta fuente de conocimiento es determinante para la identificación de entidades fisiopatológicas llevada a cabo por la fuente de conocimiento FCfisiopat y en particular, para el ulterior diagnóstico diferencial que ejecuta la fuente de conocimiento FCback.

La fuente de conocimiento FCfisiopat

La función de esta fuente de conocimiento consiste en identificar las posibles entidades fisiopatológicas presentes, a partir de las configuraciones de signos y síntomas creadas en los niveles SSGG y/o SSGF del pizarrón del dominio.

La creación de un nuevo elemento solución en los niveles SSGG y/o SSGF constituye un evento que en presencia de la restante información contenida en estos niveles, genera nuevas configuraciones de signos y síntomas. Cuando alguna configuración particular logra satisfacer alguna de las condiciones o antecedentes de FCfisiopat, ésta se dispara ejecutando las siguientes acciones :

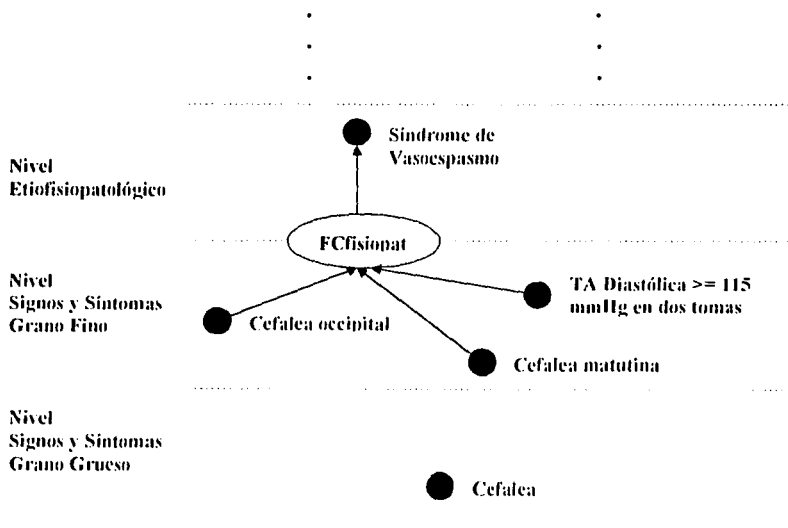


Figura 7.5 Ejemplo que muestra la actividad de la fuente de conocimiento FCfisiopat en el dominio de Sistema Cardiovascular

i. Inscribe en el bloque de registros de las fuentes de razonamiento un nuevo registro de activación con información relevante acerca del elemento solución que intenta crear.

ii. Cuando el mecanismo de control selecciona para su ejecución el registro creado por FCfisiopat, ésta ejecuta su acción final, la cual consiste en crear en el nivel EFP del pizarrón del dominio el elemento solución especificado en el registro de activación.

La figura 7.5 muestra la acción final que ejecuta FCfisiopat, una vez que ha logrado satisfacer alguna de sus condiciones sobre los niveles SSGG y/o SSGF.

La fuente de conocimiento FCnosología

Esta fuente de conocimiento es la encargada de identificar qué elementos solución podrían ser generados en el nivel EN, una vez que ha sido creado un nuevo elemento solución en el nivel EFP.

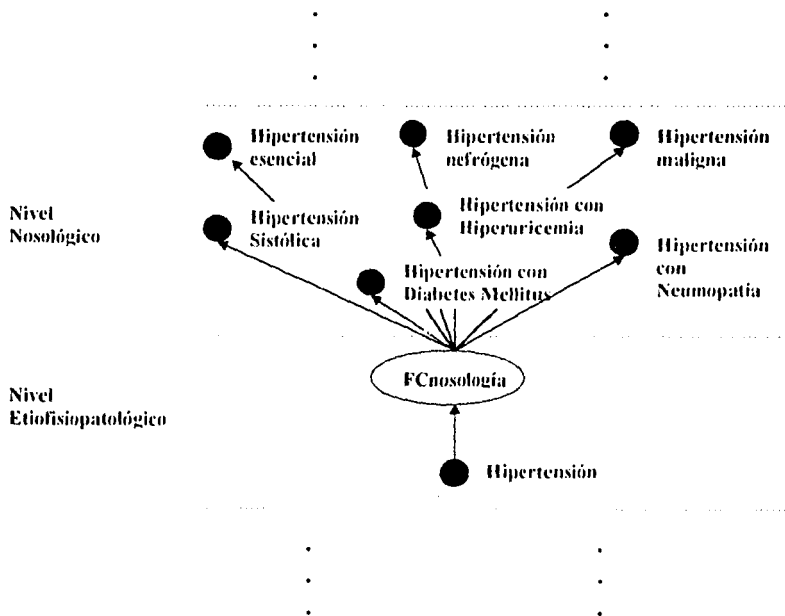


Figura 7.6 Ejemplo que muestra la actividad de la fuente de conocimiento FCnosología en el dominio de Sistema Cardiovascular

O sea, ante la presencia de una nueva entidad etiofisiopatológica en el nivel EFP, FCnosología aplica su conocimiento para identificar las posibles enfermedades que pudieran ser la causa de la manifestación sintomatológica expresada en los niveles SSGG y SSGF.

Cuando un nuevo elemento solución ha sido creado en el nivel EFP, FCnosología se dispara ejecutando las siguientes acciones :

- i. Inscribe en el bloque de registro de las fuentes de razonamiento los registros de activación correspondientes a los elementos solución que intenta crear.
- ii. Cuando el mecanismo de control selecciona para su ejecución alguno de los registros de activación creados por FCnosología, ésta ejecuta su acción final, la creación de un registro del tipo **investigar-hipótesis** sobre la base de datos de control, el cual corresponde al elemento solución que debe ser generado en el nivel EN del pizarrón del dominio.

La figura 7.6 muestra la actividad llevada a cabo por FCnosología.

La fuente de conocimiento FCterapia

La función de esta fuente de conocimiento consiste en proponer los planes terapéuticos más apropiados para las entidades fisiopatológicas metas o entidades nosológicas confirmadas en el proceso de diagnóstico diferencial. Cuando se ha arribado al diagnóstico de una determinada entidad nosológica, FCterapia identifica el conjunto de las medidas terapéuticas posibles para dicha enfermedad y posteriormente, FCback determina entre todas estas medidas terapéuticas posibles cuál resulta ser la más factible para el paciente en análisis

Cuando un nuevo elemento solución ha sido creado en el nivel EN, FCterapia se dispara ejecutando las siguientes acciones :

- i. Inscribe en el bloque de RAFRZs los registros de activación correspondientes a los elementos solución que ésta intenta crear
- ii. Cuando un registro creado por FCterapia es seleccionado por el mecanismo de control para su ejecución, la fuente de conocimiento ejecuta su acción final, la cual consiste en la creación de un registro del tipo **investigar-hipótesis** sobre la base de datos de control, el cual corresponde al elemento solución que debe ser generado en el nivel T del pizarrón del dominio.

La figura 7.7 muestra actividad llevada a cabo por FCterapia

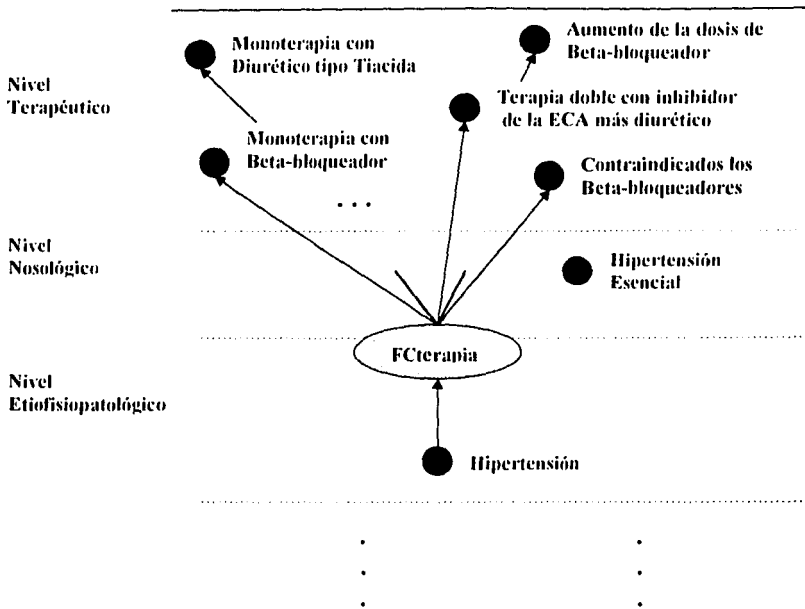


Figura 7.7 Ejemplo que muestra la actividad de la fuente de conocimiento FCterapia en el dominio de Sistema Cardiovascular

La arquitectura de INTERMED define para el nodo experto en laboratorio clínico las siguientes fuentes de conocimiento dependientes del dominio : FCdatlab, FCdiagnost y FCCausas.

La fuente de conocimiento FCdatlab

La función de FCdatlab es obtener el informe de laboratorio para la prueba o estudio solicitado. Cuando un elemento solución (prueba de laboratorio) ha sido creado en el nivel PLAB, la fuente de conocimiento FCdatlab se dispara ejecutando las siguientes acciones .

- i. Consulta la base de datos correspondiente al tipo de prueba de laboratorio solicitada, con la intención de encontrar algún informe de laboratorio registrado, cuya identificación (nombre y apellidos, expediente o número de historia clínica) coincida con la del paciente para el cual fue solicitada la prueba.

ii. Si fue encontrado en la base de datos un informe de laboratorio con estas características, entonces FCdatlab copia toda la información (datos de laboratorio) contenida en éste hacia el nivel DLAB del pizarrón del dominio y genera de esta forma un conjunto de elementos solución para este nivel.

Ejemplo

Para la prueba **química sanguínea** en el nivel PLAB, algunos de los elementos solución generados en el nivel DLAB pudieran ser : **ácido úrico = 7.5, amilasa = 14, lipasa = 2.4**, entre otros.

Si no fue encontrado en la base de datos un informe de laboratorio correspondiente al paciente, entonces FCdatlab solicita la realización de esta prueba a partir de la inscripción sobre la base de datos de control de un registro del tipo **realizar-prueba**[prueba,paciente]. Más adelante discutiremos qué papel juega la fuente de conocimiento FCitfz cuando esta situación ha tenido lugar.

La fuente de conocimiento FCdiagnost

El papel de FCdiagnost es arribar a todos los diagnósticos posibles a partir de los datos de laboratorio registrados en el nivel DLAB del pizarrón del dominio.

Ejemplo

Supongamos que en el nivel DLAB han sido generados, entre otros, los siguientes datos de laboratorio correspondientes a la prueba de **química sanguínea** realizada a un paciente del sexo femenino : **urea = 28, creatinina = 4, ácido úrico = 7.1 y fósforo = 5.2**. Entonces, la regla de inferencia

```
IF sexo = "F" and
  (urea > 25 or
   creatinina > 2 or
   ácido úrico > 7.3 or
   fósforo > 4.7)
THEN Diagnóstico_Laboratorio = Nefropatía
```

logrará satisfacer su condición y como consecuencia de ello FCdiagnost creará en el nivel D el elemento solución **Nefropatía** asociando a éste un valor de certidumbre igual a 1.

Cuando un conjunto de datos de laboratorio ha sido generado en el nivel DLAB, la fuente de conocimiento FCdiagnost se activa y revisa cuales de sus reglas de inferencia han logrado satisfacer sus condiciones. Ante cada regla satisfecha, FCdiagnost se

dispara y genera un elemento solución en el nivel D, correspondiente al diagnóstico indicado en la acción de la regla.

La fuente de conocimiento FCcausas

Esta fuente de conocimiento es la encargada de identificar aquellas enfermedades, patologías o trastornos que pudieran ser las causas más probables de las alteraciones observadas en los datos de laboratorio.

Ante la creación de un elemento solución (diagnóstico de laboratorio) en el nivel D, FCcausas se activa y utiliza el conocimiento contenido en sus reglas de inferencia para inferir las causas que con mayor credibilidad pudieran ser responsables del diagnóstico de laboratorio obtenido. Cuando alguna regla de inferencia logra satisfacer su condición en el nivel D, FCcausas se dispara y crea en el nivel C los elementos solución especificados en la consecuencia de la regla satisfecha.

Ejemplo

Al ser creado el elemento solución **Nefropatía** en el nivel D, la regla

IF Diagnóstico_Laboratorio = Nefropatía
THEN considerar = Obstrucción Urinaria and
considerar = Deshidratación and
considerar = Infección Renal

logrará satisfacer su condición y como consecuencia de ello FCcausas creará en el nivel C los elementos solución **Obstrucción Urinaria**, **Deshidratación** e **Infección Renal**, e identifica a cada uno de éstos como causa probable.

Ejemplo

Por ejemplo, ante la creación del elemento solución **Hipercolesterolemia** en el nivel D, la regla :

IF Diagnóstico_Laboratorio = Hipercolesterolemia
THEN considerar = Hepatopatía Obstructiva and
considerar = Colelitiasis and
considerar = Cirrosis Hepática and
considerar = Hipotiroidismo and
considerar = Diabetes Mellitus

logrará satisfacer su condición y como consecuencia de ello FCcausas creará en el nivel C los elementos solución especificados en la acción de la regla, lo cual indica que éstos deben ser considerados en el proceso diagnóstico.

El papel de las fuentes de conocimiento independientes del dominio

Excepto el nodo SELABCLI, cada nodo experto en INTERMED posee todas las fuentes de conocimiento independientes del dominio definidas por la arquitectura de REDSIEX. Las características estructurales y funcionales de cada una ya fueron descritas en el capítulo anterior. Ahora sólo resta conocer qué papel juegan estas fuentes de conocimiento en los procesos de consulta e interconsulta médica.

La fuente de conocimiento "Interfaz"

La fuente de conocimiento FCitfz es la encargada de establecer las vías de comunicación entre el usuario y el nodo experto durante el proceso de diagnóstico médico. En este sentido, FCitfz define protocolos de comunicación que garantizan el establecimiento de las siguientes acciones comunicativas :

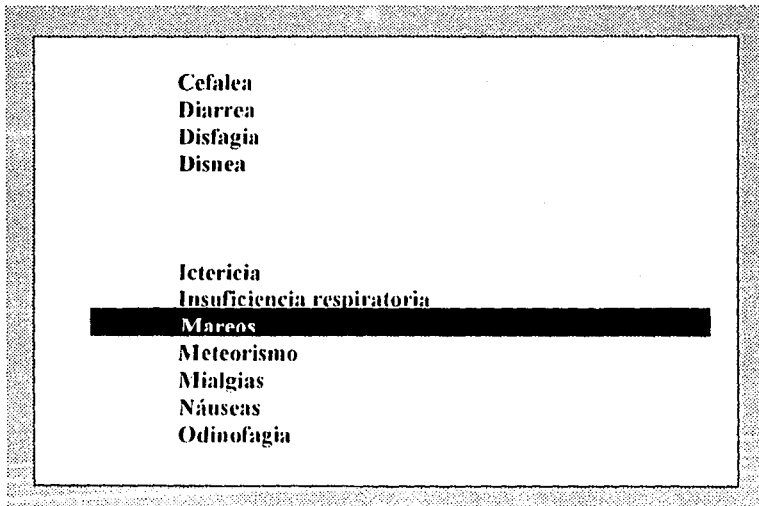


Figura 7.8 Entrada de información clínica en el dominio de Sistema Digestivo

i. Entrada de información para el planteamiento del problema-diagnóstico o para el avance de su solución

Esta información abarca todas las manifestaciones clínicas (signos y síntomas) que posee el usuario para describir el caso que intenta resolver. Durante todo el proceso de diagnóstico médico el usuario tiene la posibilidad de introducir información clínica que considere relevante para la solución del problema diagnóstico. Naturalmente, esto permitirá un mayor avance y focalización en la solución del problema. La figura 7.8 muestra un ejemplo de entrada de información clínica para la solución de un problema diagnóstico en el dominio de Sistema Digestivo.

ii. Solicitud de la investigación de alguna entidad nosológica en particular.

Pudiera ocurrir que el usuario posea sospechas acerca de la presencia de alguna entidad nosológica en particular para un problema-diagnóstico dado. Cuando este es el caso, el usuario puede solicitar al nodo experto que inicie la investigación de dicha entidad nosológica y además, puede brindar toda la información clínica que posea acerca esta enfermedad. La figura 7.8 muestra un ejemplo de solicitud de investigación de enfermedades particulares (esofagitis por reflujo y gastritis crónica) en el dominio del sistema experto SEDIGEST.

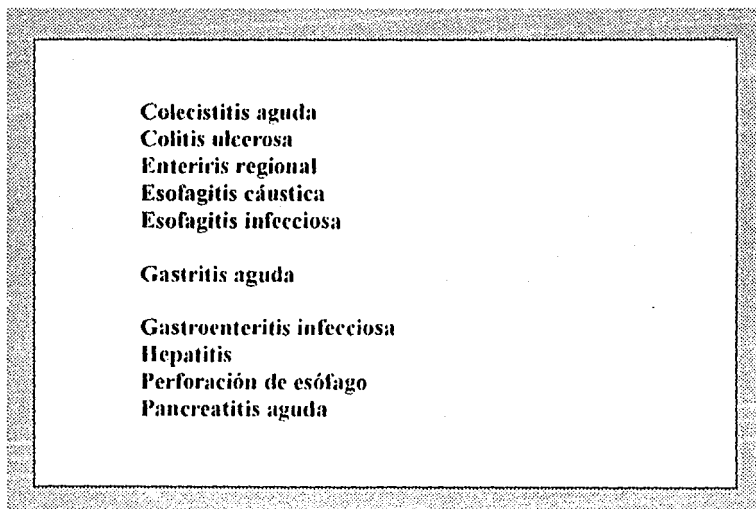


Figura 7.9 Solicitud de investigación de metas particulares en el dominio de Sistema Digestivo

iii. Solicitud y oferta de información acerca del estado del diagnóstico médico.

En cualquier instante, durante el proceso de diagnóstico médico, el usuario puede solicitar al nodo experto información acerca del estado de la consulta. Por otra parte, el nodo experto informa el estado de la consulta al usuario cada vez que un cambio sustancial se haya producido en éste. La información que brinda el nodo experto al usuario, ya sea atendiendo a una solicitud o como una oferta, refleja solamente las entidades fisiopatológicas que se han logrado constatar y los resultados actuales del diagnóstico diferencial : enfermedades presentes y enfermedades no presentes.

iv. El nodo experto envía al usuario un mensaje tipo pregunta.

Cuando el nodo experto formula una pregunta al usuario, es porque está ejecutando una actividad correspondiente a un plan local conjuntamente elaborado por el planeador y FCback. La meta de este plan es la inferencia de alguna entidad fisiopatológica o entidad nosológica específica, para lo cual se requiere de alguna información que sólo el usuario es capaz de brindar.

Las fuentes de conocimiento de comunicación

El principal papel de las fuentes de conocimiento receptora y transmisora en el proceso de diagnóstico médico es el establecimiento de canales de comunicación entre los nodos expertos de la red, con el fin de llevar a cabo los procesos de solicitud y envío de resultados de pruebas de laboratorio, gabinete e interconsulta médica.

Un nodo experto utiliza a la fuente de conocimiento FCtrs para solicitar compromisos a otros nodos de la red, con el objetivo de esclarecer un diagnóstico probable, inconcluso o dudoso; para dar respuesta a compromisos de diagnósticos que le han sido solicitados a éste y para ofertar alguna información relevante (entidades fisiopatológicas o entidades nosológicas presentes) la cual ha alcanzado recientemente en su proceso de diagnóstico médico y conoce que ésta podría ser de gran utilidad para algún otro nodo experto de la red.

Por otra parte, la fuente de conocimiento FCrep es la encargada de recepcionar, clasificar y seleccionar la información que ha llegado al nodo desde otros nodos de la red como respuesta a un compromiso de interconsulta, el cual fue previamente solicitado por el nodo, como solicitud de un compromiso de interconsulta hecho al nodo o como una oferta efectuada a éste por algún otro nodo de la red.

La fuente de conocimiento FCback y el Planeador Local

La fuente de conocimiento FCback y el planeador local ejecutan conjuntamente la etapa final del proceso de diagnóstico médico : el diagnóstico diferencial. La función de éstos es discernir, entre todas las posibilidades diagnósticas sugeridas como probables

causantes de un cierto grupo de manifestaciones clínicas, cuál de éstas es con mayor certeza la causa del proceso mórbido.

Cuando un grupo de registros del tipo **investigar_hipótesis** ha sido creado sobre el pizarrón del control, el planeador y la fuente de conocimientos FCback se disparan para ejecutar las siguientes acciones :

i. FCback inicia la construcción de todos los posibles árboles de secuencias de generación de elementos solución para cada una de las metas (posibilidades diagnósticas o entidades nosológicas) a investigar. Cada una de estas metas ha sido previamente registrada como un elemento solución del tipo **investigar-hipótesis[II]** sobre la base de datos de control (acción final de FCnosología).

El elemento solución en la raíz del árbol es la entidad nosológica que se intenta investigar. Los elementos solución en los restantes niveles del árbol son todos aquellos signos, síntomas, resultados de laboratorio o gabinete cuya afirmación o negación permiten inferir la ocurrencia de la entidad nosológica en cuestión.

Cuando se ha encontrado una entidad nosológica a investigar, para la cual no ha sido posible la construcción de un árbol de secuencias, entonces es necesario recurrir a la interconsulta médica. Esto significa que el nodo experto posee sospechas acerca de la ocurrencia de esta entidad nosológica, pero no posee los conocimientos ni métodos de inferencia que le permitan efectuar un diagnóstico certero.

ii. Para cada árbol de secuencias construido, el planeador aplica sus criterios heurísticos para construir el mejor plan de actividades (posibles y factibles), encaminado a la inferencia de la entidad nosológica que constituye la raíz del árbol.

iii. FCback inscribe un nuevo registro de activación en su bloque de registros, que indica, según el plan previamente construido, qué actividad debe ser ejecutada. Por lo general, esta actividad está asociada a la formulación de preguntas al usuario del nodo (signos o síntomas) o a la solicitud de compromisos a otros nodos de la red (pruebas de laboratorio, gabinete o interconsulta).

iv. Cuando el mecanismo de control selecciona para su ejecución el registro creado por FCback, ésta ejecuta su acción final, que consiste en la creación de un registro del tipo **preguntar_hipótesis[II]**, e indica la pregunta que debe ser formulada o el compromiso que debe ser solicitado.

La fuente de conocimiento "Parada"

La fuente de conocimiento FCpar posee conocimientos para decidir cuando detener o evitar que se detengan los procesos encaminados a la solución del problema-diagnóstico actual.

Una solución confiable para el problema-diagnóstico actual ha sido alcanzada cuando dentro de un grupo de entidades nosológicas, causantes de determinado síndrome o manifestaciones clínicas, al menos una enfermedad ha sido constatada con un elevado valor de certidumbre y las restantes han sido descartadas. Cuando este es el caso, la FCpar se dispara para ejecutar las siguientes acciones :

- i. Los procesos encaminados hacia la solución del problema-diagnóstico actual son detenidos totalmente.
- ii. Un problema-diagnóstico pendiente es retomado o un nuevo problema-diagnóstico es iniciado.

Un problema-diagnóstico pendiente corresponde al estado de una consulta ya iniciada, para la cual se agotaron temporalmente los recursos disponibles sin que se lograra alcanzar una solución al problema.

Un nuevo problema-diagnóstico es el inicio de una nueva consulta, para la cual existe suficiente información registrada en el buzón de mensajes de entrada. Esta información pudiera corresponder a una solicitud de interconsulta efectuada por algún otro experto de la red o simplemente al planteamiento de un nuevo problema introducido por el usuario del nodo experto.

Cuando la información en los niveles inferiores del pizarrón (signos y síntomas) es insuficiente, entonces la generación de nuevos elementos solución en los niveles superiores del pizarrón (síndromes y entidades nosológicas) no puede ser llevada a cabo. Ante esta situación, la FCpar se dispara inscribiendo un registro del tipo **investigar-hipótesis[]** sobre la base de datos de control. La creación de este registro desencadena la actividad de FCback y del planeador, que ejecutan planes locales encaminados a la creación de los elementos solución más prometedores sobre el pizarrón del dominio.

Si el tiempo de espera a un compromiso solicitado (pruebas de laboratorio, gabinete o interconsulta) ha sido excedido y no existen recursos para continuar con los procesos de solución del problema, entonces FCpar decide retomar un problema-diagnóstico pendiente para el cual ya posee información de interés (resultados de pruebas de laboratorio, gabinete o interconsulta) o iniciar un nuevo problema-diagnóstico a partir de un planteamiento dado.

La arquitectura del nodo SELABCLI no abarca todas las fuentes de conocimiento independientes del dominio definidas en REDSIEX. El estilo de trabajo de este nodo en la solución de problemas del dominio no requiere del papel de la fuente de conocimiento FCback ni del planeador local. Las restantes fuentes de conocimiento si son concebidas dentro de la arquitectura del nodo, aunque sus características funcionales responden al tipo de problemas que se solucionan en el dominio de laboratorio clínico. Luego,

resultaría beneficioso describir en forma breve cuál es la función de estas fuentes de conocimiento dentro de los procesos de solución de problemas.

Para establecer las vías de comunicación entre el usuario de laboratorio clínico y el nodo experto en laboratorio clínico, la fuente de conocimiento Interfaz define las siguientes acciones comunicativas :

i. Registro, actualización y consulta de estudios de laboratorio en las correspondientes bases de datos. Este servicio lo brinda FCitfz al usuario del nodo, de forma muy similar al proceso de solicitud y oferta de información que lleva a cabo ésta en los restantes nodos expertos de la red.

ii. Solicitud para la realización y registro de estudios de laboratorio. Este servicio lo brinda FCitfz al nodo experto. La condición que dispara a FCitfz para ejecutar esta acción es que algún registro del tipo **realizar-prueba**[prueba,paciente] haya sido creado sobre la base de datos de control. Ante la ocurrencia de este evento, FCitfz transfiere la prueba solicitada al usuario del nodo, elimina este registro de la base de datos de control e inscribe la prueba en la cola de pruebas pendientes.

La fuente de conocimiento receptora se encarga de la recepción de las solicitudes de pruebas de laboratorio que llegan al nodo experto en laboratorio clínico desde otros nodos de la red. Por su parte, la fuente de conocimiento transmisora se encarga de transmitir los resultados de estas pruebas (datos de laboratorio, diagnóstico y causas probables) hacia los nodos que formularon estas solicitudes.

La fuente de conocimiento Parada indica cuándo los procesos encaminados a la solución de un problema se han completado y cuándo debe ser iniciada la solución de un nuevo problema.

La solución para un problema del dominio ha sido alcanzada cuando para una prueba de laboratorio solicitada las siguientes acciones han tenido lugar:

1. Un informe de laboratorio se ha desplegado en el nivel DLAB.
2. Se han arribado a uno o varios diagnósticos en el nivel D.
3. De existir causas probables asociadas, éstas han sido identificadas y generadas en el nivel C.
4. Todos los resultados obtenidos para la prueba de laboratorio en cuestión han sido transmitidos al nodo experto interesado.

Cuando estas acciones han sido ejecutadas, FCparada consulta la cola de pruebas pendientes para conocer si alguna de estas ya fue realizada e informada. De ser así, selecciona la primera que encuentra y un nuevo proceso encaminado a la solución de la misma es iniciado. Si no existen pruebas pendientes o si aún no ha sido informada ninguna de las existentes, entonces son atendidas las nuevas solicitudes de pruebas que han sido recibidas por FCrep. Otro criterio observado por FCparada para atender otra

solicitud de prueba de laboratorio es el hecho de que una prueba previamente registrada en el nivel PLAB haya sido transferida como prueba pendiente, al no existir información registrada en la base de datos correspondiente.

Consideraciones finales

Finalmente, queremos considerar algunos aspectos de interés relacionados con los Sistemas Expertos, los Sistemas Pizarrón, la arquitectura de pizarrón propuesta y el modelo de consulta e interconsulta médica implementado.

Los Sistemas Pizarrón son una extensión de los Sistemas Expertos basados en reglas

Las dos primeras partes de este material han brindado un amplio panorama acerca de las características de la arquitectura y funcionamiento de los Sistemas Expertos basados en reglas y los Sistemas Pizarrón, respectivamente.

Las características de los Sistemas Expertos basados en reglas se encuentran subyacentes de alguna forma en la arquitectura y funcionamiento de los Sistemas Pizarrón. Estos últimos parecen haber heredado y refinado muchos de los componentes de los Sistemas Expertos basados en reglas.

Según nuestro criterio, los Sistemas Pizarrón constituyen una extensión de los Sistemas Expertos basados en reglas. Determinadas transformaciones sobre los principales componentes de un Sistema Experto basado en reglas (base de conocimientos, memoria de trabajo y mecanismo de inferencia) permiten hacer de éste un sistema pizarrón. Analicemos a continuación algunas de estas transformaciones.

Las reglas contenidas en la base de conocimientos de un sistema basado en reglas son empaquetadas según su tipo (por ejemplo, en dependencia del tipo de conclusión al que éstas son capaces de arribar) en módulos autónomos e independientes conocidos como fuentes de conocimiento. Ya resulta para nosotros conocido que una fuente de

conocimiento es una estructura de la forma condición-acción y que en la gran mayoría de estos casos se define en términos de un conjunto de reglas de producción.

La memoria de trabajo (donde son almacenados los hechos de entrada, así como los nuevos hechos inferidos por las reglas) es ahora estructurada en varios niveles de abstracción, ahora conocida como pizarrón. Los hechos que se almacenan en la base de conocimientos son entonces creados en los diferentes niveles de abstracción según su tipo (o en dependencia del tipo de regla que lo generó). Estos hechos son conocidos como hipótesis o elementos solución.

El mecanismo de inferencia es transformado en un mecanismo de control o despacho, cuya función principal es decidir el orden en el cual las fuentes de conocimiento disparadas van a ejecutar su acción sobre el pizarrón.

Al igual que en un Sistema Experto basado en reglas, en un Sistema Pizarrón los procesos de razonamiento son ejecutados a través de encadenamientos de reglas hacia adelante (direccionamiento por los eventos que ocurren sobre el pizarrón), encadenamientos de reglas hacia atrás (direccionamiento por las metas) o una combinación de ambos métodos.

Según este enfoque, un Sistema Pizarrón hereda y elabora los componentes de un Sistema Experto basado en reglas, para exhibir una nueva arquitectura que constituye la antesala a los Sistemas Expertos Distribuidos : un sistema donde el razonamiento ha sido distribuido en entidades autónomas e independientes, las cuales se comunican entre sí a través de una base de datos compartida y obedecen a un control centralizado.

Acerca de la arquitectura de pizarrón propuesta

En la segunda parte de este material fueron discutidos varios sistemas pizarrón desarrollados en etapas posteriores al surgimiento del sistema HEARSAY II [ENG88b, ERM80, ERM88, LES88a, NII89a]. Algunos de estos sistemas fueron desarrollados para aplicaciones específicas, como es el caso de HEARSAY II, HASP/SLAP [NII82, NII89a, WIL84], DVMT [LES88b], GUARDIAN [HAY89], PROTEAN [HAY87, NII89a] y TRICERO [NII89a]. Otros sistemas constituyen generalizaciones (paquetes de herramientas) para el desarrollo de aplicaciones pizarrón, ejemplos de estos son AGE [NII88b], BBI [HAY85, AHY89], BLOBS [ZAN88], MIX [LAK84, TA88] y MUSE [REY88]. También existen sistemas pizarrón que han surgido como extensiones o refinamientos de generalizaciones o aplicaciones existentes, ejemplos de ellos son los sistemas GBB [COR88], CAGE [NII88a, NII89b] y POLIGON [NII88a, NII89b].

El sistema REDSIEX constituye una generalización orientada al desarrollo de aplicaciones pizarrón para la solución de tareas de diagnóstico. Por su parte, el sistema INTERMED es una aplicación específica para el diagnóstico médico, desarrollada sobre REDSIEX.

REDSIEX pertenece a la clase de los sistemas pizarrón distribuidos. A esta clase también pertenecen los sistemas BLOBS, GUARDIAN, POLIGON, DVMT y TRICERO. De forma general, los sistemas clasificados en esta categoría se caracterizan por estar compuestos por un conjunto de nodos o agentes, donde cada uno de ellos es un sistema pizarrón con sus propias fuentes de conocimiento, pizarrón y elementos de control. En este tipo de sistemas, los nodos se comunican entre sí a través del paso de mensajes.

Varios de los componentes de REDSIEX son heredados de las arquitecturas de HEARSAY II, DVMT y BBI. Esto hace que en algunos aspectos particulares las características y el estilo de funcionamiento de REDSIEX sean similares al de estos sistemas. Sin embargo, los nuevos componentes y estrategias de trabajo incorporadas a la arquitectura de REDSIEX, hacen que su estilo de trabajo global en la solución de problemas sea muy característico y exclusivo.

Algunas de las características más notables que hacen de REDSIEX un sistema pizarrón diferente a los restantes sistemas ubicados en su clase son las siguientes :

1. La cooperación es facultativa.

En REDSIEX, la cooperación es una facultad que posee cada nodo pizarrón en la red, no constituye una necesidad permanente. Esto quiere decir que un nodo-pizarrón puede resolver un problema completo, si éste posee la capacidad y los recursos necesarios para arribar a la solución. Cuando este no es el caso, entonces el nodo responsable del problema puede solicitar la cooperación de otros nodos de la red. La arquitectura de REDSIEX no prevé un nodo administrador con facultades para descomponer una tarea en subtarear y distribuir éstas entre diferentes nodos de la red para su ejecución. En lugar de esto, la tarea es solicitada a un nodo particular de la red (el cual se supone que posee las mejores habilidades, conocimientos y recursos para llevar a cabo ésta), convirtiéndose en responsable de la tarea y decidiendo si en alguna parte del proceso de solución del problema la ayuda de algún otro nodo de la red es requerida. De ahí que el sistema REDSIEX sea visto como una colección de Sistemas Expertos "facultativamente" asociados en red cooperativa

2. La cooperación puede ser ejecutada en tiempo real o puede ser diferida.

En algún momento dado del tiempo cada nodo-pizarrón pudiera encontrarse trabajando en la solución de un problema diferente a los que están siendo resueltos por los otros nodos de la red. Sin embargo, aun bajo estas condiciones es posible que los nodos

establezcan vínculos de cooperación entre ellos. Esto se logra a partir de dos estrategias de cooperación definidas en REDSIEX : la cooperación en tiempo real y la cooperación diferida. Cuando la cooperación se lleva a cabo en tiempo real, el nodo que solicita el compromiso recibe la respuesta en el intervalo de tiempo que el nodo ha dedicado a los procesos de solución del problema. La cooperación diferida ocurre cuando la respuesta al compromiso solicitado no puede ser elaborada y enviada dentro de un intervalo de tiempo dado. Esto podría ocurrir porque el nodo, al cual le fue solicitado el compromiso, se encuentra enfrascado en la solución de otro problema que para él resulta prioritario y decide concluir algún proceso ya iniciado antes de atender la solicitud recibida. Cuando el nodo que solicitó el compromiso recibe la respuesta, posiblemente éste se encuentre trabajando en otro problema y entonces decide entre continuar los procesos encaminados a la solución del problema actual, o retomar el problema inconcluso para el cual había solicitado el compromiso.

3. El planeador local trabaja asociado a una fuente de conocimiento direccionada por las metas.

El planeador local y la fuente de conocimiento FCback trabajan conjuntamente para construir los planes locales más adecuados, encaminados a la creación de elementos solución deseados sobre el pizarrón del dominio. FCback propone todas las secuencias de actividades posibles que conducen a la creación de un determinado elemento solución, mientras que el planeador local se basa en sus criterios heurísticos para decidir entre todas las actividades posibles, cuáles son las actividades más factibles que deben integrar el plan final.

4. Un proceso de revisión de creencias garantiza la consistencia de las hipótesis creadas sobre el pizarrón.

Las nuevas evidencias obtenidas a favor o en contra de una hipótesis ya creada sobre el pizarrón del dominio, modifican el valor de creencia asociado a la hipótesis. Esto a su vez provoca que los valores de creencia de todas las hipótesis creadas directa o indirectamente a partir de la hipótesis modificada, sean también modificados en base al nuevo valor de creencia calculado para la hipótesis.

5. El foco de atención se encuentra distribuido entre la base de datos de control y la estructura de registros de activación de las fuentes de conocimiento

Dos tipos de elementos pueden constituir el foco de atención en REDSIEX : una fuente de conocimiento o un elemento solución del pizarrón. Si el foco de atención es una fuente de conocimiento, éste se encuentra localizado en la estructura de registros de activación de las fuentes de conocimiento. En este caso, el foco de atención indica qué fuente de conocimiento va a crear o modificar un elemento solución sobre el pizarrón. Si el foco de atención es un elemento solución, este se encuentra localizado en la base de datos de control, como un registro del tipo `investigar-hipótesis[i]`. En este caso, el foco de atención indica que elemento solución debe ser creado sobre el pizarrón

Existen otros muchos aspectos tanto de la estructura como del funcionamiento de REDSIEX que resultan ser muy específicos de la arquitectura. Estos ya fueron discutidos en la tercera parte de este material, por lo que no es necesario volver a referirnos a ellos.

En resumen, la arquitectura propuesta en REDSIEX constituye una nueva aproximación a la solución cooperativa de problemas distribuidos. En particular, ésta ha sido concebida como un medio ambiente o escenario de prueba para la solución de problemas de tipo diagnóstico.

Sobre el modelo de consulta e interconsulta médica implementado

Dentro de la amplia gama de Sistemas Expertos desarrollados para tareas de diagnóstico médico, los Sistemas Expertos basados en reglas han resultado ser los candidatos más fuertes para este tipo de aplicación. Los Sistemas Expertos con encadenamiento de reglas hacia adelante, encadenamiento de reglas hacia atrás, oportunistas (combinación de ambos métodos) y otros híbridos obtenidos a partir de éstos (*frame*-reglas, redes neurales-reglas y redes semánticas-reglas) han sido muy útiles para aplicaciones en el diagnóstico médico.

Sin embargo, han sido muy escasas (o al menos no se han encontrado reportadas en la literatura) las aplicaciones de sistemas pizarrón desarrolladas en tareas de diagnóstico médico. El sistema GUARDIAN constituye un ejemplo de aplicaciones de este tipo. Aunque la principal tarea de GUARDIAN es el monitoreo de pacientes en una unidad de cuidados intensivos, este posee agentes de razonamiento que ejecutan tareas de diagnóstico.

El sistema INTERMED es otro ejemplo de una aplicación pizarrón para la solución de tareas de diagnóstico médico. El modelo de consulta e interconsulta médica implementado en INTERMED constituye una nueva aproximación a la solución de problemas de este tipo.

Diagnóstico médico natural vs diagnóstico médico en INTERMED

El proceso diagnóstico que efectúa el médico durante la consulta médica

El médico utiliza el interrogatorio durante la consulta médica, como fase inicial de la recopilación de datos, y los agrupa en conjuntos conocidos (o parcialmente conocidos) la serie de síntomas, antecedentes heredofamiliares y personales patológicos que le son referidos en forma directa o indirecta. Posteriormente, mediante la exploración física orientada y la realización, en muchos casos, de estudios de laboratorio y gabinete también orientados, llega a un diagnóstico final. Este diagnóstico, en la mayoría de los casos, debe referirse a una entidad nosológica específica, pero no siempre es posible lograr una agudeza diagnóstica ideal y sólo se llega al diagnóstico sindromático o fisiopatológico. En ambos casos, el llegar a un determinado diagnóstico permite el inicio de una terapéutica orientada a la entidad diagnosticada. La figura c1 resume esta primera aproximación al proceso de diagnóstico médico.

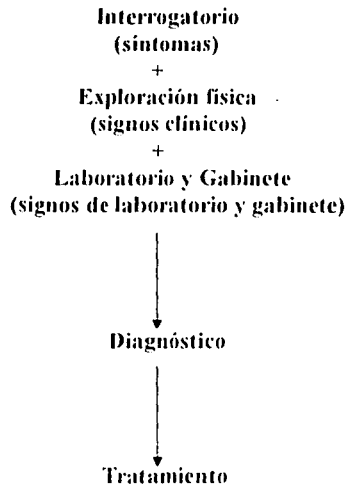


Figura c1 Una primera aproximación al proceso de diagnóstico médico

Concluir un proceso completo de diagnóstico y tratamiento no es una tarea ni clara ni sencilla. Esta labor requiere del médico años de entrenamiento académico y clínico, ya que aunque parezca que el proceso diagnóstico consiste en agrupar A más B para llegar a C, en la realidad no es así. Este proceso se complica al individualizar el diagnóstico, ya que, por ejemplo, la expresión de los signos y síntomas no siempre es la misma en cada individuo. Por otra parte, ocurre con frecuencia que las características de una enfermedad se superponen a las características de otras enfermedades semejantes, lo que determina que una enfermedad o grupo de enfermedades llegue a tener diferentes expresiones clínicas. En este caso, el diagnóstico sólo se logra conjuntando los conocimientos académicos y la experiencia práctica acumulada, proceso que se obtiene con el tiempo y la práctica clínica. Todas estas situaciones inherentes al proceso de diagnóstico médico, conllevan a la ocurrencia de dos problemas básicos :

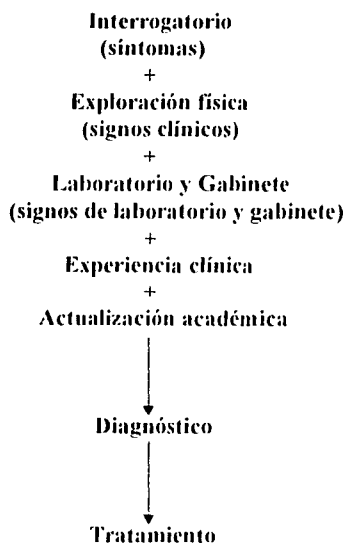


Figura c2 Una segunda aproximación al proceso de diagnóstico médico

1. Los médicos recién egresados, médicos en servicio social, médicos internos de pregrado y médicos generales, que no se actualizan constantemente o carecen de experiencia clínica, al enfrentarse al proceso diagnóstico y terapéutico, independientemente de su preparación académica, cometerán una falla diagnóstica y

terapéutica. La calidad a la que aspira llegar el proceso de diagnóstico médico exige que esta deficiencia normal de diagnóstico y tratamiento sea la menor.

2. Los recursos disponibles durante el proceso de diagnóstico, ya sean de laboratorio clínico o gabinete, están lejos de ser optimizados. Estos recursos pueden ser racionalizados, al efectuarse sólo las pruebas o exámenes necesarios para llegar al diagnóstico médico y no encarecer aún más este proceso. Los recursos de laboratorio y gabinete deben ocupar su valor real dentro del proceso de diagnóstico y tratamiento. Por ejemplo, el electrocardiograma que diagnostica un infarto y permite iniciar el tratamiento adecuado o el coproparasitoscópico que diagnostica una parasitosis y/o valora el resultado de un tratamiento antiparasitario.

Todos estos aspectos conllevan una nueva aproximación al proceso de diagnóstico, la cual requiere de una elevada actualización académica y de una gran experiencia clínica acumulada. La figura c2 resume esta segunda aproximación al proceso de diagnóstico médico.

El proceso diagnóstico que efectúa el sistema INTERMED

En INTERMED, el proceso de diagnóstico médico inicia con la selección de un nodo experto (especialista médico), el cual se supone que posea los mayores conocimientos relacionados con el problema del paciente. El sistema inicia la recolección de los signos y síntomas presentes en el paciente y un posterior refinamiento de éstos, para obtener signos y síntomas en dos niveles de granularidad : signos y síntomas grano grueso y signos y síntomas grano fino (semiología). La colección de signos y síntomas de grano grueso y grano fino formada permite al sistema integrar diagnósticos a diferentes niveles de profundidad, principalmente a nivel etiopatogénico o a nivel nosológico/causal. Sin embargo, como ocurre en la práctica clínica, para poder llegar a este nivel se requiere con frecuencia del apoyo del laboratorio y/o gabinete.

Cuando el problema-diagnóstico lo requiere, los diferentes nodos expertos pueden establecer lazos de cooperación, con el objetivo de lograr un diagnóstico definitivo. Una vez que se llega al diagnóstico, como ocurre en la práctica, se plantea un tratamiento adecuado, que no depende solamente del diagnóstico logrado, sino también de las características del paciente (edad, sexo, etapa biológica, embarazo, senectud, adolescencia, entre otras) y de sus antecedentes heredofamiliares y personales patológicos. Cuando todas estas características han sido evaluadas, se puede proponer un manejo completo de acuerdo al diagnóstico alcanzado para un paciente específico. La figura c3 resume la aproximación al proceso diagnóstico que ejecuta INTERMED.

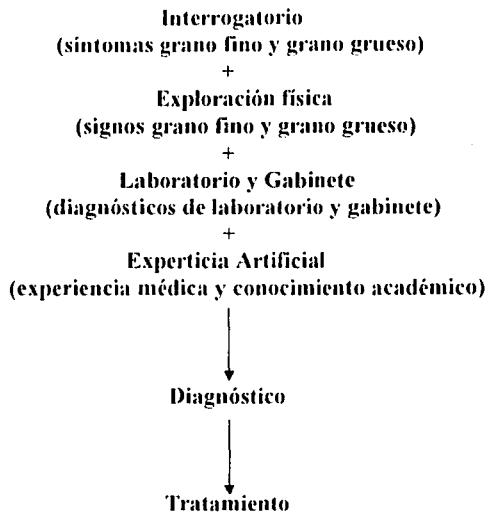


Figura 3c Proceso de Diagnóstico médico en INTERMED

Algunas de las principales ventajas del proceso diagnóstico que lleva a cabo el sistema INTERMED son las siguientes :

1. Estandarización de la calidad de la atención médica y disponibilidad de conocimiento médico especializado.

Cuando el proceso diagnóstico proporcionado por el sistema INTERMED se aplica como un proceso de diagnóstico inicial, se logra que el proceso diagnóstico sea de la misma calidad para todos los pacientes. La capacidad del sistema para intercomunicarse con otros nodos expertos (especialistas médicos) asegura , además, una profundidad adecuada en las posibilidades diagnósticas, la cual rebasa en mucho la capacidad de un médico general, por el conocimiento especializado contenido en cada nodo experto. Resumiendo, el uso de INTERMED permite la unificación de criterios diagnósticos con el fin de alcanzar una alta calidad diagnóstica

2. Optimización de recursos de laboratorio y gabinete

Para llegar a un diagnóstico y tratamiento final, el sistema propone solo aquellos estudios de laboratorio y gabinete necesarios y útiles. Por una parte, este hecho exige del proceso diagnóstico que se efectúen los estudios paraclínicos que se consideren necesarios y adecuados. Por otra parte, se evita el dispendio de recursos, además de evitar la solicitud de estudios incompletos, los cuales la gran mayoría de las veces

incrementan considerablemente el tiempo dedicado al proceso diagnóstico y por supuesto el cargo económico asociado a este proceso. En resumen, el modelo de diagnóstico médico propuesto en INTERMED contribuye a reducir el costo asociado al proceso de diagnóstico médico.

3. Otra ventaja no menos importante, asociada a todo sistema de diagnóstico de este tipo, es el aprendizaje médico inherente al uso del sistema. El usuario puede aprender del sistema las estrategias y métodos de razonamiento desplegados por éste para llegar a la solución de un problema de diagnóstico. A la vez, nuevos métodos de razonamiento y conocimiento pudieran ser incorporados al sistema, para enriquecer su experticia en la solución de problemas.

Las principales desventajas o debilidades presentes en INTERMED son las siguientes :

1. Es imposible representar o "plasmear" todo el conocimiento y experiencia médica en un sistema artificial, aunque la cantidad y calidad del conocimiento contenido en INTERMED es mayor que la de un médico general promedio, ya que el conocimiento ha sido estructurado a nivel de especialidades.

2. INTERMED no posee un módulo de aprendizaje que le permita aprender de su propia experiencia. Esto significa que cada cierto tiempo la experticia médica contenida en INTERMED tenga que estar sujeta a actualización.

3. El sistema depende del usuario para decidir en qué nodo experto iniciar la consulta, dependiendo también de la habilidad y destreza de éste para encontrar signos y síntomas relevantes. Una nueva variante de INTERMED o de algún otro sistema de este tipo debería concebir un nodo con funciones de médico general, el cual recibe toda la información clínica disponible y en base a ella decide a que nodo experto remitir el problema.

Un comentario final acerca de la implementación y uso de REDSIEX

El sistema REDSIEX fue implementado en el lenguaje de programación C++, en un estilo de programación muy cercano a la programación orientada a objetos. En REDSIEX se encuentran presentes muchas de las características deseables de los sistemas orientados a objetos y los sistemas pizarrón, las cuales se unen para dar lugar a un eficiente estilo de trabajo en la solución cooperativa de problemas distribuidos

Cada nodo experto en REDSIEX es un objeto-pizarrón, el cual se comunica con los restantes objetos-pizarrón a través del paso de mensajes. Las fuentes de conocimiento, el pizarrón y otros componentes de la arquitectura también han sido tratados como objetos, definidos y contenidos dentro de cada objeto-pizarrón.

REDSIEX ha sido diseñado e implementado como un medio ambiente ("shell" o paquete de herramientas) para la construcción de aplicaciones distribuidas en la solución cooperativa de problemas de diagnóstico. El sistema proporciona al usuario todas las estructuras y procedimientos necesarios (métodos de razonamiento, protocolos de comunicación, estrategias de cooperación, etc.) para la construcción de una aplicación pizarrón distribuida. Por su parte, el usuario debe proporcionar a REDSIEX el conocimiento del dominio específico para el cual se desea construir la aplicación.

Para construir una aplicación en REDSIEX, el conocimiento debe ser representado utilizando reglas de producción. Las reglas son agrupadas, según su tipo, en diferentes clases o colecciones. Cada colección de reglas es encapsulada dentro de una fuente de conocimiento específica, constituyendo éstas el conocimiento de la fuente.

Algunas de las especificaciones requeridas para construir una aplicación en REDSIEX son las siguientes :

- ◆ Cantidad de objetos-pizarrón en la red
- Para cada objeto-pizarrón :
- ◆ Cantidad de niveles del pizarrón
 - ◆ Número máximo de elementos solución por nivel
 - ◆ Cantidad de fuentes de conocimiento por nivel
 - ◆ Colección de reglas de producción que definen a cada fuente de conocimiento
 - ◆ Modelo del mundo externo del objeto-pizarrón (con qué objetos-pizarrón establecer comunicación y qué comunicar)

Todos los detalles necesarios para construir una aplicación en REDSIEX pueden ser consultados en la documentación contenida en su manual de usuario.

Actualmente REDSIEX corre sobre un único procesador físico, pero éste es flexible en el sentido de que cada nodo de la red puede ser asignado a un procesador independiente

Referencias

ADL93 K. P. Adlassnig, W. Horak. 1993. "The knowledge base of the Hepaxpert I System : automatic interpretation of hepatitis A and B serology". *Leber-Magen-Darm*. 23, 6 : 251-60, 263-4, 267-6.

AIK80 J. Aikins. 1980. "Prototypes and production rules : A knowledge representation for computer consultations". Rep. No. STAN-CSD-80-814, Computer Science Dept., Stanford University.

BAR85 J. Barclay Adams. 1985. "Probabilistic Reasoning and Certainty Factors. Rule-Based Expert Systems". En *The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

BAR89a Avron Barr y Edward A. Feigenbaum. 1989. *The Handbook of Artificial Intelligence. Volumen I*. Addison-Wesley.

BAR89b Avron Barr y Edward A. Feigenbaum. 1989. *The Handbook of Artificial Intelligence. Volumen II*. Addison-Wesley.

BAR89c Avron Barr, Paul R. Cohen, Edward A. Feigenbaum. 1989. *The Handbook of Artificial Intelligence. Volumen IV*. Addison-Wesley.

BEK92 G. A. Bekey, J-W. J. Kim, J. K. Gronley, E. L. Bontrager y J. Perry. 1992. "GAIT-ER-AID : An expert system for diagnosis of human gait" *Artificial Intelligence in Medicine* 4, 4 : 293-308.

BEN80 J. S. Bennett y D. Goldman. 1980. "CLOT : A knowledge-based Consultant for bleeding disorders". Report no. HPP-80-7, Computer Science Department, Stanford University.

BRA83 Ronald J. Brachman, Saul Amarel, Carl Engelman, Robert S. Englemore, Edward A. Feigenbaum y David E. Wilkins 1983. *What are Expert Systems. Building Expert Systems*. Addison-Wesley.

BUC85 Bruce G. Buchanan y Edward H. Shortliffe. 1985. *Rule-Based Expert Systems The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

BUC89 Bruce G. Buchanan y Reid G. Smith. 1989. "Fundamentals of Expert Systems". En The Handbook of Artificial Intelligence. Volumen IV. Addison-Wesley.

CHE85 P. Cheeseman. 1985. "In defense of probability". Proceedings IJCAI-85.

CLA79a W. J. Clancey. 1979. "Dialogue management for rule-based tutorials". IJCAI 6, 155-161.

CLA79b W. J. Clancey. 1979. "Transfer of rule-based expertise through a tutorial dialogue". Rep. No. STAN-CS-769, Computer Science Dept., Stanford University.

CLA81 W. J. Clancey y R. Letsinger. 1981. "NEOMYCIN : Reconfiguring a rule-based expert system for application to teaching". IJCAI 7, 829-836.

COR88 D. D. Corkill, K. Q. Gallagher y K. E. Murray. 1988. "GBB : A Generic Blackboard Development System". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

DEM68 A. P. Dempster. 1968. "A generalization of Bayesian inference". Journal of the Royal Statistical Society, Series B, 30 :205-247.

ENG88a R. Englemore, T. Morgan y H. P. Nii. 1988. Blackboard Systems. Addison-Wesley Publishing Company.

ENG88b R. Englemore, A. J. Morgan y H. P. Nii. 1988. "HEARSAY II". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

ERM80 L. D. Erman, F. Hayes-Roth, V. R. Lesser y D. R. Reddy. 1980. "The HEARSAY II speech understanding system : Integrating knowledge to resolve uncertainty". ACM Computing Survey 12 : 213-253.

ERM88 L. D. Erman, F. Hayes-Roth, V. R. Lesser y D. Raji Reddy. 1988. "The Hearsay II Speech-Understanding System : Integrating knowledge to Resolve Uncertainty". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

FAG79 L. Fagan. 1979. "Knowledge engineering for dynamic clinical settings : Giving advice in the intensive care unit" Doctoral disertation. Computer Science Dept., Stanford University.

FAG85 Lawrence M. Fagan, John C. Kunz, Edward A. Feigenbaum y John J. Osborn. 1985. "Extensions to the Rule-Based Formalism for a Monitoring Task Rule-Based Expert Systems". En The MYCIN experiments of the Stanford Heuristic Programming Project. Addison-Wesley.

FOR84 Richard Forsyth. 1984. Expert Systems Principles and case studies. Chapman and Hall Computing.

FRA92 P. Francois, B. Cremilleux, C. Robert y J. Demongeot. 1992. "MENINGE : A medical consulting system for the child's meningitis". Study on a series of consecutive cases. Artificial Intelligence in Medicine 4, 4 : 281-292.

GOR85 J. Gordon y E. H. Shortliffe. 1985. "The Dempster-Shafer Theory of Evidence. Rule-Based Expert Systems The MYCIN experiments of the Stanford Heuristic Programming Project". Addison-Wesley.

HAJ85 P. Hajek. 1985. "Combining functions for certainty degrees in consulting systems". Int. Jour. Man-Machine Studies 22 : 59-67.

HAY83 Frederick Hayes-Roth, Donald A. Waterman y Douglas B. Lenat. 1983. "On Overview of Expert Systems". En Building Expert Systems. Addison Wesley.

HAY85 B. Hayes-Roth. 1985. "A blackboard architecture for control". Artificial Intelligence, 26 : 251-321.

HAY87 B. Hayes-Roth, B. Buchanan, O. Lichtarge, M. Hewett, R. Altman, J. Brinkley, C. Cornelius, B. Duncan y O. Jardetzky. 1987. "PROTEAN : Deriving protein structure from constraints". Proceedings of the National Conference on Artificial Intelligence, 904-909.

HAY88 B. Hayes-Roth y M. Hewett. 1988. "BB1 : An implementation of the Blackboard Control Architecture" En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

HAY89 B. Hayes-Roth, M. Hewett, R. Washington, R. Hewett y A. Seiver. 1985. "Distributing Intelligence within an Individual". En Distributed Artificial Intelligence. Volumen II. Ed. L. Gasser y M. Huhns. Pitman Publishing.

HE178 J. F. Heiser, R. E. Brooks y J. P. Ballard. 1978. "Progress report : A computerized psychopharmacology advisor". Proceedings of the Eleventh Collegium Internationale Neuro-Psychopharmacologicum, Vienna, Austria.

JAC90 Peter Jackson. 1990. Introduction to Expert Systems. Segunda Edición. Addison-Wesley

KON87 K. Konolige. 1987. On the relations between default theories and auto epistemic logic. Proceedings IJCAI-87.

KUN78 J. Kunz et al. 1978. "A physiological rule-based system for interpreting pulmonary function test results". Heuristic Programming Project Rep. No. HPP-78-19, Computer Science Dept., Stanford University.

LAK84 W. L. Lakin y J. A. Miles. 1984. "A blackboard system for multi-sensor fusion". Technical Report. ASWE. Portsdown, Portsmouth, England.

LAR92 C. Larizza, A. Moglia y M. Stefanelli. 1992. "M-HIP : A system for monitoring heart transplant patients". Artificial Intelligence in Medicine 4, 2 : 111-126.

LES88a V. R. Lesser y L. D. Erman. 1988. "A retrospective View of the Hearsay II Architecture". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

LES88b V. R. Lesser y D. D. Corkill. 1988. "The Distributed Vehicle Monitoring Testbed : A Tool for Investigating Distributed Problem Solving Networks". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

LIN80 D. A. Lindberg, G. C. Sharp, L. C. Kingsland, S. M. Weiss, S. P. Hayes, H. Ueno y S. E. Hazelwood. 1980. "Computer based rheumatology consultant". The syllabus for the tutorial on computers in medicine : Applications of artificial intelligence techniques, Stanford University School of Medicine.

MER89 Dennis Merritt. 1989. Building Expert Systems in Prolog. Springer-Verlag.

MAR93 A. M. Marchevsky, G. Coons. 1993. "Expert Systems as an aid for the pathologist's role of clinical consultant : CANCER-STAGE". Mod-Pathol. May : 6,3 : 265-9.

MUL84 B. Mulsant y D. Serran-Schreider. 1984. "Knowledge engineering: A daily activity on a hospital ward". Computers and Biomedical Research 17 : 71-91

NEG90 José Negrete. 1990. Inteligencia aunque sea Artificial. Editorial Limusa.

NEG91a José Negrete y Gerardo López. 1991. Informática Médica. Editorial Limusa.

NEG91b José Negrete. 1991. Inteligencia Experimental en Computadoras. Editorial Limusa.

NI82 H. P. Nii, E. A. Feigenbaum, J. J. Anton y A. J. Rockmore. 1982. "Signal-to-symbol transformation : HASP/SLAP case study". AI Magazine 3(2) : 23-35.

NI188a H. P. Nii, N. Aiello y J. Rice. 1988. "Frameworks for Concurrent Problem Solving : A Report on CAGE and POLIGON". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

NI188b H. P. Nii y N. Aiello. 1988. "AGE (Attempt to GEneralize) : A knowledge-Based Program for Building Knowledge-Based Programs". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison Wesley Publishing Company.

NI189a H. P. Nii. 1989. "Blackboard Systems". En The Handbook of Artificial Intelligence. Volumen IV. Editores Avron Barr, Paul R. Cohen y Edward A. Feigenbaum. Addison-Wesley.

NI189b H. P. Nii, N. Aiello y J. Rice. 1989. "Experiment on Cage and Poligon : Measuring the performance of parallel blackboard systems". En Distributed Artificial Intelligence. Volumen II. Editores L. Gasser y M. N. Huhns. Morgan Kaufmann.

PAT81 R. S. Patil, P. Szolovits y W. B. Schwartz. 1981. "Causal understanding of patient illness in medical diagnosis". In Proceedings of 7th International Joint Conference on Artificial Intelligence, pp. 893-899.

PAT90 Dan W. Patterson. 1990. Introduction to Artificial Intelligence and Expert Systems. Prentice Hall.

PAU76 S. G. Pauker, G. A. Gorry, J. P. Kassirer y W. B. Schwartz. 1976. "Toward the simulation of clinical cognition : Taking a present illness by computer". American Journal of Medicine 60 : 981-995.

PAU77 S. G. Pauker y P. Szolovits. 1977. "Analyzing and simulating taking the history of the present illness : Context formation". IFIP Conference on Computational Linguistics in Medicine pp 109-118. Amsterdam : North-Holland.

POP77 H. Pople. 1977. "The formation of composite hypotheses in diagnostic problem solving - An exercise in synthetic reasoning". IJCAI 5, 1030-1037.

REY88 D. Reynolds. 1988. "MUSE : A Toolkit for Embedded, Real-Time AI". En Blackboard Systems. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

RIC91 Elaine Rich y Kevin Knight. 1991. Artificial Intelligence. Segunda Edición. McGraw Hill.

SHA76 G. Shafer. 1976. A Mathematical Theory of Evidence. Princeton, NJ : Princeton University Press.

SHO75 E. H. Shortliffè y B. G. Buchanan. 1975. "A model of inexact reasoning in medicine". *Mathematical Biosciences* 23 : 351-379.

SHO76 E. H. Shortliffè. 1976. *Computer-based medical consultations : MYCIN*. New York : American Elsevier.

SHO79 E. H. Shortliffè, B. G. Buchanan y E. A. Feigenbaum. 1979. "Knowledge engineering for medical decision making : A review of computer-based clinical decision aids". *Proceedings of the IEEE* 67 : 1207-1224.

SHO81 E. H. Shortliffè, A. C. Scott, M. B. Bischoff, A. B. Campbell, W. van Melle y C. D. Jacobs. 1981. "ONCOCIN : An expert system for oncology protocol managemet". *IJCAI* 7.

SHO85 E. H. Shortliffè y B. G. Buchanan. 1985. "A model of Inexact Reasoning in Medicine". En *Rule-Based Expert Systems The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

SHO87 Y. Shoham. 1987. "Nonmonotonic logics : Meaning and utility". *Proceedings IJCAI-87*.

SMI85 David E. Smith y Jan E. Clayton. 1985. "Another Look at Frames". En *Rule-Based Expert Systems The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

SWA77a W. Swartout. 1977. "A digitalis therapy advisor with explanations". Rep. No. TR-176, MAC Project, Computer Science Dept., Massachusetts Institute of Technology.

SWA77b W. Swartout. 1977. "A digitalis therapy advisor with explanations". *IJCAI* 5, 819-825.

TAI88 A. Taylor. 1988 "A Blackboard Expert System Shell". En *Blackboard Systems*. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

TOM93 R. Tombropoulos, S. Shiffman, C. Davidson. 1993. "A decision aid for diagnosis of liver lesions on MRI". *Proc- Annu-Symp-Comp-Appl-Med-Care* : 439-43

TR177 M. Trógoboff y C. Kulikwski . 1977. "IRIS : A system for the propagation of inferences in a semantic net". *IJCAI* 5, 274-280.

TR178 M. Trógoboff. 1978. "IRIS : A Framework for the construction of clinical consultation systems". *Doctoral disertations Computer Science Dep., Rutgers University*.

WEI78 S. M. Weiss, C. A. Kulikowski, S. Amarel y A. Safir. 1978. "A model-based method for computer-aided medical decision-making". *Artificial Intelligence* 11 : 145-172.

WIL84 M. Williams, H. Brown y T. Barnes. 1984. "TRICERO Design Description". Technical Report ESL-NS539, ESL, Inc.

ZAD79 L. A. Zadeh. 1979. "A theory of approximate reasoning". *Machine Intelligence* 9 : 149-194. New York : Halstead Press.

ZAN88 R. Zanconato. 1988. "BLOBS. An Object-Oriented Blackboard System Framework for Reasoning in Time". En *Blackboard Systems*. Editado por R. Englemore y T. Morgan. Addison-Wesley Publishing Company.

Anexo A

Relación de entidades etiofisiopatológicas y nosológicas diagnosticadas por el nodo experto SEDIGEST

ENTIDADES ETIOFISIOPATOLÓGICAS

Cálculos biliares
Esofagitis
Hemorroides
Hepatitis
Infecciones gastrointestinales
Insuficiencia hepática
Parasitosis
Sangrado de Tubo Digestivo
Síndrome intestinal funcional
Úlcera péptica

ENTIDADES NOSOLÓGICAS

Colecistitis aguda
Colecistitis crónica
Colelitiasis
Coledocolitiasis
Esofagitis péptica
Esofagitis biliar
Hemorroides grado I
Hemorroides grado II
Hemorroides grado III
Hemorroides grado IV
Hepatitis viral tipo A
Hepatitis viral tipo B
Hepatitis viral no A no B
Hepatitis por medicamentos

Anexo A (continuación)

Hepatitis por alcohol
Hepatitis aguda
Hepatitis fulminante
Hepatitis crónica activa
Hepatitis crónica persistente
Infección gastrointestinal por shigellosis
Infección gastrointestinal por salmonelosis
Infección gastrointestinal por escherichia coli
Insuficiencia hepática con encefalopatía
Insuficiencia hepática con ascitis
Insuficiencia hepática con hipertensión portal
Insuficiencia hepática por alcoholismo crónico
Insuficiencia hepática por hepatitis viral
Insuficiencia hepática por hemocromatosis
Insuficiencia hepática por hepatitis autoinmune
Insuficiencia hepática por colangitis esclerosante
Insuficiencia hepática por enfermedad de Wilson
Insuficiencia hepática por fibrosis quística
Insuficiencia hepática por hepatotoxinas
Parasitosis giardiasis
Parasitosis amibiasis
Parasitosis estroingiloidosis
Parasitosis ascariasis
Parasitosis uncinariasis
Parasitosis tricuriasis
Sangrado de tubo digestivo alto por esofagitis
Sangrado de tubo digestivo alto por varices esofágicas
Sangrado de tubo digestivo alto por Síndrome de Mallory-Weiss
Sangrado de tubo digestivo alto por gastritis
Sangrado de tubo digestivo alto por úlceras de stress
Sangrado de tubo digestivo alto por úlcera péptica
Sangrado de tubo digestivo alto por úlcera duodenal
Sangrado de tubo digestivo bajo por Enfermedad de Crohn
Sangrado de tubo digestivo bajo por divertículo de Meckel
Sangrado de tubo digestivo bajo por divertículos en colon
Sangrado de tubo digestivo bajo por hemorroides
Sangrado de tubo digestivo bajo por fisura anal
Sangrado de tubo digestivo bajo por polipos
Úlcera péptica no complicada
Úlcera péptica complicada con sangrado de tubo digestivo
Úlcera péptica complicada con obstrucción intestinal
Úlcera péptica complicada con perforación
Úlcera péptica complicada con penetración

Anexo B

Relación de entidades etiofisiopatológicas y nosológicas diagnósticas por el nodo experto SECARDIO

ENTIDADES ETIOFISIOPATOLÓGICAS

Hipertensión
Insuficiencia cardíaca
Síndrome de vasoespasmio
Síndrome de enfermedad coronaria cardíaca

ENTIDADES NOSOLÓGICAS

Hipertensión esencial
Hipertensión nefrótica
Hipertensión por otras causas
Hipertensión leve
Hipertensión moderada
Hipertensión severa
Hipertensión inicial
Hipertensión establecida
Hipertensión maligna
Hipertensión con diabetes mellitus
Hipertensión con neumopatía
Hipertensión con hiperuricemia
Hipertensión sistólica
Asma
Enfermedad pulmonar obstructiva crónica
Insuficiencia cardíaca aguda
Insuficiencia cardíaca crónica
Insuficiencia cardíaca izquierda
Insuficiencia cardíaca derecha
Insuficiencia cardíaca global
Insuficiencia cardíaca con hipertensión
Insuficiencia cardíaca con coronariopatía
Insuficiencia cardíaca con hipertensión y coronariopatía
Insuficiencia cardíaca por otras causas
Choque cardiogénico
Angina de pecho estable
Angina de pecho inestable
Angina variante (prinzmetal)
Infarto agudo del miocardio
Infarto agudo del miocardio silencioso

Anexo C

Relación de entidades etiofisiopatológicas y nosológicas diagnosticadas por el nodo experto SERESPIRA

ENTIDADES ETIOFISIOPATOLÓGICAS

Infección de vías respiratorias altas
Infección de vías respiratorias bajas

ENTIDADES NOSOLÓGICAS

Amigdalitis aguda
Amigdalitis crónica
Asma en período interrecurrente
Asma en crisis asmática
Bronquitis aguda
Bronquitis crónica
Enfisema pulmonar con insuficiencia respiratoria aguda
Enfisema pulmonar con insuficiencia respiratoria crónica
Faringitis aguda
Faringitis bacteriana
Faringitis viral
Laringitis aguda
Laringitis crónica
Neumonía neumocócica
Neumonía por klebsiella
Neumonía estafilocócica
Neumonía por micoplasma
Sinusitis aguda
Sinusitis crónica
Tuberculosis

Anexo D

Estudios de laboratorio clínico ejecutados por el nodo experto SELABCLI

Biometría hemática
Coproparasitoscópico
Electrolitos séricos
Examen general de orina
Pruebas de función hepática
Pruebas de coagulación
Pruebas inmunológicas
Química sanguínea