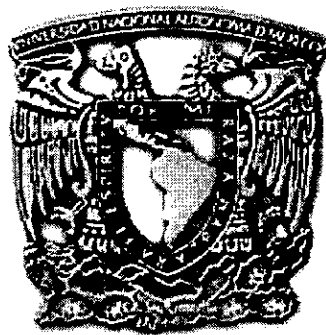


03081
5



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

REDES DE CONDUCTAS INTERNAS COMO NODOS
PIZARRÓN: SELECCIÓN DE ACCIONES
Y APRENDIZAJE EN UN
ROBOT REACTIVO

T E S I S

PARA OBTENER EL GRADO DE:
DOCTOR EN INVESTIGACIÓN BIOMÉDICA BÁSICA

PRESENTA:

M. en IBB. PEDRO PABLO GONZÁLEZ PÉREZ

ASESOR:

DR. JOSÉ NEGRETE MARTÍNEZ

México, 2000

273407



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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 la memoria de mi padre

Agradecimientos

Debo a mis padres, más que a nadie, el haber logrado alcanzar esta nueva meta. Ellos siempre me guiaron, me apoyaron, me alentaron, me respaldaron, y ante todo, me comprendieron.

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 formación como investigador el haber trabajado bajo su dirección.

Agradezco también al Dr. Pablo Pacheco Cabrera y al Dr. Enrique Ruiz Velasco Sánchez, también tutores de este proyecto, por sus valiosas contribuciones para el desarrollo del mismo.

Tengo que expresar mi reconocimiento a la Dra. Hanna Oktaba, al Dr. Jaime Lagunez Otero, al Dr. Francisco Cervantes Pérez y al Dr. Felipe Lara Lozano, por sus valiosas sugerencias, comentarios y críticas durante la revisión de este material.

Me es grato reconocer la valiosa ayuda del Dr. Ariel Barreiro García, por todos estos años de colaboración, y por las invaluable ideas aportadas para el desarrollo de mi tesis doctoral.

Un agradecimiento especial para mi alumno Carlos Gershenson García, por las tantas horas y horas que dedicó a trabajar en este proyecto, por hacer éste suyo, por su contagiosa motivación, y por la disciplina y persistencia que siempre caracterizaron su trabajo. Gracias Carlos.

Le doy las gracias a todos aquellos amigos y compañeros que de muchas formas me brindaron su apoyo, el cual me permitió llevar adelante y concluir esta tarea.

Mi último agradecimiento para la siempre querida Universidad Nacional Autónoma de México, y para México, mi segunda patria, por haberme brindado la oportunidad de alcanzar esta meta tan importante en mi vida.

CONTENIDO

RESUMEN	1
INTRODUCCIÓN	2
1. MECANISMOS PARA LA SELECCIÓN DE ACCIONES	8
1.1. Sistemas basados en el comportamiento	8
1.2. Mecanismos para la selección de acciones	9
1.2.1. El mecanismo de Tinbergen	10
1.2.2. El modelo de Lorenz	12
1.2.3. El modelo de Baerends	14
1.2.4. La arquitectura embebida de Brooks	15
1.2.5. El modelo de Rosenblatt y Payton	17
1.2.6. El mecanismo <i>bottom-up</i> de Maes	19
1.2.7. El modelo neuronal de Beer	21
1.2.8. La red neuroconectora de Halperin	23
1.2.9. El modelo neuro-humoral de Negrete	25
1.2.10. La red recurrente de atractores de Goetz	26
1.3. Otros trabajos desarrollados	27
1.3.1. La aproximación "Sociedad de la Mente" a la selección de acciones	27
1.3.2. Los modelos de objetivos múltiples y de fusión de comportamientos usando votación de Prijanian	28
1.3.3. Aprendizaje de nuevos comportamientos en un red de comportamientos	28
1.4. La combinación de estímulos internos y externos en los mecanismos revisados	28
2. LA ARQUITECTURA DE PIZARRÓN Y LA SELECCIÓN DE ACCIONES	31
2.1. Arquitecturas de mecanismos de selección de acciones	31
2.1.1. Redes jerárquicas	32
2.1.2. Redes distribuidas	32
2.1.3. Redes conexionistas	33
2.2. La arquitectura de pizarrón	33
2.2.1. Las fuentes de conocimiento	35
2.2.2. El pizarrón	35
2.2.3. El sistema de control	36
2.3. La arquitectura de REDSIEX	37
2.3.1. La solución cooperativa de problemas en REDSIEX	37
2.3.2. Distribución de la información, procesamiento y control en REDSIEX	38
2.3.3. La arquitectura de los nodos en REDSIEX	38
2.3.4. Características de la cooperación en REDSEIX	44
2.4. La arquitectura de ECN-MAES	45
2.4.1. La arquitectura de los nodos en ECN-MAES	45
2.4.2. El mecanismo de selección de comportamientos	49

2.4.3. Propiedades de la selección de comportamientos en un nodo ECN-MAES	49
2.4.4. Características de la cooperación inter-nodos en ECN-MAES	50
2.5. Desde REDSIEX y ECN-MAES a RCI	50
3. LA RED DE CONDUCTAS INTERNAS	52
3.1. Los componentes de un nodo-pizarrón	53
3.1.1. El pizarrón	53
3.1.2. Las conductas internas	54
3.1.3. Los registros de los estados de actividad	55
3.1.4. Los mecanismos de interfaz/comunicación	55
3.1.5. Los mecanismos de competencia	55
3.2. El nodo cognitivo	56
3.2.1. El pizarrón del dominio	56
3.2.2. Los exteroceptores	58
3.2.3. La conducta interna acciones reflejas	59
3.2.4. La conducta interna persistencia perceptual	61
3.2.5. La conducta interna atención a preferencias	64
3.2.6. La conducta interna inhibición de respuestas reflejas	65
3.2.7. La conducta interna selección de conductas externas	68
3.2.8. Los actuadores	70
3.2.9. El mecanismo receptor	71
3.2.10. El mecanismo transmisor	72
3.3. El nodo motivacional	73
3.3.1. El pizarrón del dominio	73
3.3.2. Los propioceptores	75
3.3.3. La conducta interna congruencia propio/extero/drive	75
3.3.4. La conducta interna selector de preferencias consumatorias	77
3.3.5. El mecanismo receptor	79
3.3.6. El mecanismo transmisor	80
3.4. Consideraciones finales	81
4. PROPIEDADES DE LA SELECCIÓN DE ACCIONES EN LA RED DE CONDUCTAS INTERNAS	82
4.1. Comportamiento motivado	82
4.2. Competencia a nivel motivacional de conductas externas incompatibles	83
4.3. Comportamiento reactivo	84
4.4. Direccionamiento por las metas	85
4.5. No indecisión en la selección de acciones	85
4.6. Persistencia en la ejecución de una acción externa	85
4.7. Espontaneidad regulada en la red de conductas externas	86
4.8. Saciedad	86
4.9. Cambios en el interés	86
4.10. Variación de la atención	87
4.11. Preactivación de conductas internas	87
4.12. Existencia de una conducta externa "por defecto" orientada a la búsqueda de una señal específica	88
4.13. Aprendizaje asociativo	90

4.13.1. Condicionamiento clásico primario	90
4.13.2. Condicionamiento clásico secundario	97
4.13.3. Principales propiedades de los procesos de condicionamiento clásico primario y secundario en la red de conductas internas	98
4.14. Inhibición de respuestas reflejas	100
4.15. Consideraciones finales	102
5. LA SIMULACIÓN	104
5.1. El robot simulado	104
5.1.1. El sistema perceptual	105
5.1.2. El sistema motor	107
5.1.3. El medio interno	108
5.2. El medio ambiente simulado	109
5.3. El repertorio de comportamientos	111
5.4. Ajuste de parámetros	113
5.4.1. Inicialización de las fuerzas de apareo	115
5.5. Consideraciones finales	116
6. PRUEBAS Y EXPERIMENTOS	117
6.1. La conducta externa observada es influenciada por los estados internos de la entidad	117
6.2. Conductas incompatibles motivacionalmente compiten entre sí, y la competencia se efectúa a nivel motivacional	119
6.3. La conducta exploratoria orientada a la búsqueda de una señal específica logra tiempos de reacción del animal mucho menores	121
6.4. Estabilidad en la selección y persistencia en la ejecución de conductas externas	124
6.5. Discriminación entre diferentes estímulos tomando en cuenta la calidad de los mismo	127
6.6. Evitar estímulos aversivos	129
6.7. La no persistencia en la ejecución de una acción consumatoria ante la percepción de un estímulo aversivo	131
6.8. Condicionamiento clásico primario	134
6.8.1. Asociación de un estímulo neutro con una fuente de comida	134
6.8.2. Asociación de un estímulo neutro con un estímulo aversivo	137
6.9. Condicionamiento secundario	138
6.10. Propiedades del condicionamiento clásico en la red de conductas internas	140
6.10.1. Bloqueo	140
6.10.2. Caída de la actividad del estímulo en el tiempo	141
6.10.3. Dominancia	143
6.10.4. Interrupción del condicionamiento	144
6.10.5. Pérdida del condicionamiento	145
6.11. Acerca de los experimentos realizados	147
CONCLUSIONES	148
• Acerca de la arquitectura de la red de conductas internas	148

- Sobre el modelo de combinación de estímulos internos y externos propuesto 149
- Propiedades emergentes de la red de conductas internas 149
- Mejorando la selección de acciones con el aprendizaje asociativo 150
- La simulación desarrollada: una aproximación a un "laboratorio virtual de conductas" 150
- Aplicaciones 151
- Trabajo futuro: extensiones y refinamientos 151

REFERENCIAS 153

APENDICES 156

- REDSIEX: A cooperative network of expert systems with blackboard architectures
- Net of multi-agent expert systems with emergent control
- A model for Combination of External and Internal Stimuli in the Action Selection of an Autonomous Agent (abstract)
- Action Selection Properties in a Software Simulated Agent (abstract)

RESUMEN

La investigación de los sistemas basados en el comportamiento constituye un nuevo enfoque para la construcción de agentes autónomos. Esta reciente línea de investigación se inspiró principalmente en la etología, la parte de la biología dedicada al estudio del comportamiento de los animales en sus ambientes naturales. Para satisfacer sus metas, un agente autónomo debe seleccionar, en cada momento, la acción más apropiada entre todas las que éste pudiera ejecutar. Un mecanismo de selección de acciones (MSA) es un mecanismo computacional, el cual especifica cómo estas acciones son seleccionadas. La propuesta de la presente tesis doctoral es un mecanismo de selección de acciones construido sobre una arquitectura de pizarrón, la cual facilita la estructuración del mecanismo y la incorporación en un estilo de diseño "bottom-up" de nuevos comportamientos y propiedades. Al mecanismo de selección de acciones aquí propuesto lo hemos nombrado Red de Conductas Internas (RCI). Entre las principales propiedades que hacen de la RCI una aproximación novedosa al problema de la selección de acciones se encuentran: (1) el modelo para la combinación de estímulos externos e internos utilizado por ésta, que retoma y combina los elementos más valiosos de los modelos aditivos y multiplicativos puros; y (2) el rol de los procesos de aprendizaje asociativo en la mejora de la selección de acciones en la RCI, al hacer a ésta más adaptativa. A partir de (1) y de los diferentes procesos competitivos presentes en la RCI, emergen nuevas propiedades "no alambradas" en ésta. La selección de acciones en la RCI se caracteriza por una amplia gama de propiedades y principios presentes en el comportamiento animal. Estas propiedades son comprobadas y discutidas a partir de un conjunto de experimentos realizados sobre una simulación desarrollada para tal efecto.

INTRODUCCIÓN

Uno de los principios importantes observados en la inteligencia artificial (IA) a finales de la década de los 60s, fue aquel que estableció la importancia del conocimiento en los sistemas para la solución de problemas de propósito general. Un sistema para la solución de problemas que usaba una cantidad limitada de conocimiento (reglas, hechos, leyes, axiomas, etc.) resultaba ser muy poco efectivo en la solución de problemas que exhibieran algún tipo de complejidad. Este nuevo enfoque permitió eventualmente el diseño y construcción de lo que hoy en día se conocen como sistemas basados en el conocimiento, es decir, sistemas que dependen de una rica base de conocimiento para la ejecución de tareas complejas.

De esta forma, a finales de la década de los 60s e inicios de la década de los 70s surgió un nuevo período de investigación y desarrollo en IA, en el cual la gran mayoría de los trabajos realizados estaban estrechamente relacionados con la concepción de sistemas basados en el conocimiento (SBC), incluyendo solución general de problemas, comprensión del lenguaje natural, aprendizaje y visión. En 1977 Feigenbaum (Feigenbaum, 1985) enfatizó en esta nueva forma de pensar, al establecer que el poder real de un sistema experto está dado por la cantidad y calidad del conocimiento que éste posee, más que por los mecanismos de inferencia particulares que el mismo utiliza para manipular su conocimiento.

El poder de un SBC radica fundamentalmente en el conocimiento experto que ha sido codificado en hechos, reglas, heurísticas y procedimientos, y almacenado en su base de conocimiento. Sin embargo, este tipo de sistema no interactúa directamente con el dominio del problema del que posee conocimiento y para el que resuelve problemas. Esta interacción se efectúa de forma indirecta a través de un usuario externo (comúnmente, una persona), quien a la vez controla dicha interacción. El usuario presenta al sistema una descripción del problema del dominio a solucionar en un lenguaje simbólico comprensible por el sistema, el cual entonces aplica su conocimiento experto en la solución del problema, y devuelve al usuario una descripción simbólica de la respuesta o conclusión a la que ha logrado arribar. Para un sistema basado en el conocimiento, el dominio del problema no cambia durante el tiempo en que éste se encuentra ejecutando los procesos de solución del problema. De aquí que los sistemas basados en el conocimiento no tengan que caracterizarse por ser sistemas adaptativos.

Es partir de mediados de la década de los 80s cuando las investigaciones en IA dejan de centrarse en el estudio de los sistemas basados en el conocimiento para tomar una nueva dirección: el estudio de los sistemas basados en el comportamiento (Brooks, 1986). Esta nueva línea de investigación, también conocida como “agentes autónomos” se inspiró principalmente en la etología, la parte de la biología dedicada al estudio del comportamiento de los animales en sus ambientes naturales.

A diferencia de los sistemas basados en el conocimiento, los sistemas basados en el comportamiento sí interactúan directamente con su dominio del problema. Un agente autónomo percibe su dominio del problema a través de sus sensores y actúa sobre éste a través de sus actuadores. El dominio del problema de un agente autónomo es comúnmente un medio ambiente dinámico, complejo e impredecible, en el cual éste trata de satisfacer un conjunto de metas o motivaciones, las cuales pudieran variar en el tiempo. Un agente autónomo decide por sí mismo cómo relacionar sus entradas externas e internas con sus acciones motoras de forma tal que sus metas puedan ser satisfechas (Maes, 1994). La adaptación es una de las características deseables en los agentes autónomos. Un agente autónomo es adaptativo si éste posee habilidades que le permitan mejorar su desempeño en el tiempo.

Para satisfacer sus metas, un agente autónomo debe seleccionar, en cada momento en el tiempo, la acción más apropiada entre todas las acciones posibles que éste pudiera ejecutar. Esto es precisamente lo que en el contexto de los sistemas basados en el comportamiento se conoce como el problema de la selección de acciones. Mientras que el problema de la selección de acciones (PSA) se refiere a cual acción el agente (robot, *animat*, o criatura artificial) debe seleccionar en cada momento en el tiempo, un mecanismo para la selección de acciones (MSA) especifica cómo estas acciones son seleccionadas. Un MSA es un mecanismo computacional, el cual debe producir como salida una acción seleccionada cuando diferentes estímulos externos y/o internos han sido proporcionados como entradas. Visto de esta forma, un PSA indica *cuál*, mientras que un MSA especifica *cómo*.

Entre los principales MSAs y trabajos relacionados con la selección de acciones se encuentran la red jerárquica de centros de Tinbergen (Tinbergen, 1950 y Tinbergen, 1951), el modelo psicohidráulico de Lorenz (Lorenz, 1950 y Lorenz, 1981), la red jerárquica de nodos de Baerends (Baerends, 1976), la arquitectura embebida de Brooks (Brooks, 1986 y Brooks, 1989), el modelo conexionista de Rosenblatt y Payton (Rosenblatt y Payton, 1989), el mecanismo *bottom-up* de Maes (Maes, 1990 y Maes, 1991), el modelo neuronal de Beer (Beer, 1990 y Beer, Chiel y Sterling, 1990), la red neuroconectora de Halperin (Hallam, Halperin y Hallam, 1994), el modelo neuro-humoral de Negrete (Negrete y Martínez, 1996), y el modelo de redes de comportamiento recurrentes de Goetz (Goetz y Walters, 1997). Estos mecanismos han sido inspirados en modelos pertenecientes a disciplinas tales como la etología, psicología, ciencias cognitivas, robótica, ingeniería, redes neuronales artificiales e inteligencia artificial, entre otras. Algunos de estos mecanismos abordan completamente el problema de la selección de acciones, mientras que otros sólo tratan parte del problema.

La diferencia esencial entre los diversos MSAs propuestos va más allá del tipo de arquitectura en la que éstos han sido estructurados, o del modelo en el cual éstos han sido inspirados. Nosotros postulamos que la diferencia esencial entre estos mecanismos radica en la forma en que los mismos combinan los estímulos externos e internos para seleccionar una acción determinada, y por lo tanto, en el diverso repertorio de propiedades que logra emerger producto del tipo de combinación; así como en el rol que juegan los procesos de aprendizaje en el mejoramiento de la selección de acciones, al hacerla más adaptativa.

La presente tesis es una investigación que abarca los mecanismos computacionales que proveen solución al problema de la selección de acciones, y los tipos de arquitecturas que proporcionan escenarios adecuados para la implementación de estos mecanismos. El problema que aquí se

aborda consiste en proponer un mecanismo de selección de acciones para un agente autónomo, que exhiba algunas de las propiedades no satisfechas o explicadas por los MSAs que aquí serán presentados y discutidos; y que incorpore nuevas propiedades derivadas de los principios que caracterizan el comportamiento animal, los cuales deben enriquecer la selección de acciones haciéndola más adaptativa.

Para lograr lo anterior, nos hemos basado en una arquitectura de pizarrón distribuida, la cual, dada su gran capacidad para la coordinación e integración de varias tareas en tiempo real y su extrema flexibilidad para la incorporación de nueva funcionalidad, facilita la implementación del modelo, así como la incorporación incremental de nuevas propiedades y procesos de aprendizaje, encaminados a enriquecer la selección de acciones. De esta forma, la implementación del mecanismo de selección de acciones aquí propuesto constituye en sí un escenario ideal para la modelación y prueba de nuevos comportamientos y propiedades deseadas en la selección de acciones de un agente autónomo (un tipo de “laboratorio virtual de conductas”), provenientes de áreas tales como la robótica reactiva y la etología.

Algunas de las propiedades observadas en el mecanismo de selección de acciones aquí propuesto emergen directamente de la forma en la cual se combinan las entradas externas e internas. Es decir, la forma de combinación de estímulos utilizada intenta incorporar en un único modelo los elementos más relevantes de los modelos de combinación aditivos y multiplicativos puros. Ejemplos de estas propiedades son: la fuerte dependencia de la conducta externa observada de los estados internos de la entidad (comportamiento motivado), y la selección de acciones dirigida tanto hacia el medio externo como hacia el medio interno de la entidad.

La tesis ha sido estructurada de la siguiente forma. El capítulo 1 introduce el concepto de sistema basado en el comportamiento y el problema de la selección de acciones. Posteriormente se presenta un panorama de los principales mecanismos de selección de acciones (MSA) propuestos en la literatura. La descripción de estos MSAs es realizada tomando en cuenta características tales como: ¿qué parte de la selección de acciones resuelve?, ¿cuál es la naturaleza del mecanismo? (disciplina a la que pertenece el modelo en el cual ha sido inspirado el mecanismo), tipo de arquitectura en la que ha sido estructurado el mecanismo, nivel de detalle y/o explicación de los procesos que tienen lugar, procesos de aprendizaje logrados con el mecanismo, y un especial énfasis en la forma en que se combinan las entradas externas e internas para la selección de acciones, y en las nuevas propiedades que pueden emerger a partir de las características de esta combinación. El capítulo finaliza con una discusión sobre estos mecanismos, relacionando las principales ventajas y limitaciones de cada uno de ellos.

El capítulo 2 aborda el tema de las arquitecturas de los MSAs. Inicialmente se identifican las principales clases de arquitecturas propuestas y las características más relevantes de éstas son discutidas. Dentro de estas arquitecturas se encuentran: la red de nodos jerárquica, la red de nodos distribuida y la red conexionista. Posteriormente se describe detalladamente la arquitectura que hemos seleccionado para la implementación del MSA propuesto en el capítulo 3: la arquitectura de pizarrón; así como las dos principales variantes de esta arquitectura desarrolladas por nosotros (González y Negrete, 1997 y Negrete y González, 1998), las cuales proporcionan el sustrato básico para la estructuración del MSA propuesto.

El capítulo 3 presenta el MSA que hemos desarrollado: la red de conductas internas construida con nodos pizarrón. El capítulo inicia definiendo el concepto “conducta interna”, siendo ésta la unidad básica funcional del modelo propuesto. La estructura completa de la red de conductas internas es presentada y entonces las características estructurales y funcionales de cada uno de los nodos son discutidas. Esta discusión abarca el *qué* y el *cómo* del mecanismo de la selección de acciones propuesto, siendo éstos explicados en términos de: (1) la estructura del pizarrón en términos de los diferentes niveles de análisis o abstracción que lo componen, así como las características de los elementos solución creados o modificados en cada uno de estos niveles; (2) la estructura y funcionamiento de las conductas internas que operan sobre el pizarrón del nodo; (3) la estructura y funcionamiento de los mecanismos *exteroceptores*, *actuadores* y de comunicación (receptor y transmisor); (4) el funcionamiento de los mecanismos de selección de conductas internas elementales (procesos competitivos que tienen lugar entre conductas internas de un mismo tipo); (5) el trabajo conjunto de todas las conductas internas y mecanismos de comunicación que operan a nivel del nodo; y (6) los procesos de comunicación que toman lugar entre los nodos.

El capítulo 4 se centra en la discusión de las principales características de la selección de acciones y de los procesos de aprendizaje que tienen lugar en la red de conductas internas. Este capítulo inicia haciendo un análisis detallado de la expresión utilizada en el modelo propuesto para la combinación de entradas externas e internas, derivando desde este análisis gran parte del conjunto de los principios básicos que caracterizan al modelo. En este capítulo también se discute cómo el aprendizaje puede mejorar los procesos de la selección de acciones, haciéndolos más adaptativos. En particular, se discuten las dos formas básicas de aprendizaje que se pueden apreciar en la red de conductas internas: el condicionamiento clásico primario y el secundario, y las propiedades que caracterizan a estos procesos de aprendizaje en el modelo propuesto. Finalmente, se presenta el principio jerárquico implementado en la red de conductas internas, el cual resulta ser muy exclusivo de este modelo.

La simulación desarrollada que implementa la red de conductas internas es presentada en el capítulo 5. Los dos principales componentes de la simulación, el agente simulado (*animat*) y medio ambiente simulado, son presentados aquí. Respecto al *animat*, se describe la representación externa del mismo y los restantes componentes que integran su estructura interna, subsistema perceptual, subsistema motor y medio interno, son explicados. Las características del ambiente simulado son explicadas tomando en cuenta los diversos tipos de estímulos que pueden ser creados en el mismo. Posteriormente, se describe el repertorio de comportamientos ejecutados por el *animat*, y las entradas externas e internas asociadas a cada uno de éstos. El capítulo finaliza explicando cómo se lleva a cabo, a través de la simulación, el ajuste de los diferentes parámetros que caracterizan a la red de conductas internas; y la inicialización de la fuerzas de apareo de las conductas internas.

En el capítulo 6 se presentan y discuten el conjunto de experimentos realizados para comprobar cuándo la red de conductas internas fue capaz de producir por sí misma los efectos aclamados por ésta. Los experimentos realizados fueron encaminados a comprobar: (1) la influencia de los estados internos en la conducta externa observada en el *animat*, (2) el papel de la competencia a nivel motivacional en la selección de la conducta externa a ejecutar, (3) la conducta exploratoria orientada a la búsqueda de una señal específica y la reducción de los tiempos de respuesta del *animat*, (4) la estabilidad en la selección y la persistencia en la ejecución de las conductas

externas, (5) la discriminación entre diferentes estímulos tomando en cuenta la calidad de los mismos, (6) la no persistencia en la ejecución de una acción consumatoria ante la percepción de un estímulo aversivo y (7) los procesos de aprendizaje asociativo en el mejoramiento de la selección de acciones.

La estructuración de la tesis obedece a la serie de decisiones tomadas que se resumen en la figura 0.1.

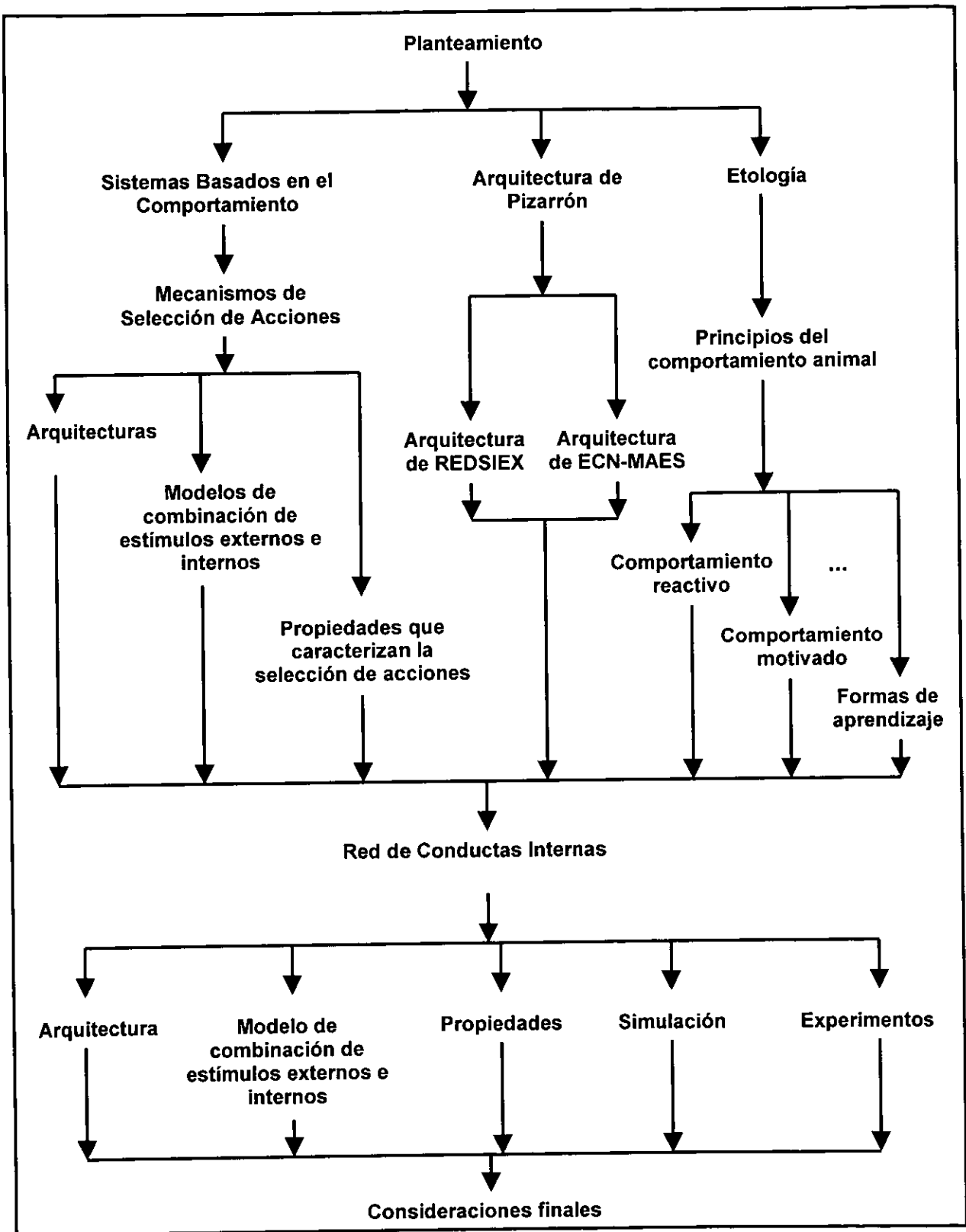


Figura 0.1. Estructuración de la tesis.

CAPÍTULO 1

MECANISMOS PARA LA SELECCIÓN DE ACCIONES

1.1. Sistemas basados en el comportamiento

La teoría de los sistemas basados en el comportamiento (Brooks, 1986, Brooks, 1989 y Brooks, 1991) proporciona una nueva filosofía para la construcción de agentes autónomos, inspirada en el campo de la etología. A diferencia de los sistemas basados en el conocimiento, los sistemas basados en el comportamiento interactúan directamente con su dominio del problema. Un agente autónomo percibe su dominio del problema a través de sus sensores y actúa sobre éste a través de sus actuadores. El dominio del problema de un agente autónomo es comúnmente un medio ambiente dinámico, complejo e impredecible, en el cual éste trata de satisfacer un conjunto de metas o motivaciones, las cuales pudieran variar en el tiempo. Un agente autónomo decide por sí mismo cómo relacionar sus entradas externas e internas con sus acciones motoras de forma tal que sus metas puedan ser satisfechas (Maes, 1994). La adaptación es una de las características deseables en los agentes autónomos. Un agente autónomo es adaptativo si éste posee habilidades que le permitan mejorar su desempeño en el tiempo.

El término “agente” ha sido ampliamente usado en una gran variedad de dominios de investigación, siendo los más comunes aquellos dominios pertenecientes a las áreas de inteligencia artificial y ciencias de la computación. Varias han sido las definiciones e interpretaciones aportadas para el término “agente”, dependiendo éstas del dominio en el cual el agente ha sido utilizado, así como del uso particular que al mismo se le ha dado.

Según Pattie Maes (Maes, 1994) *“un agente es un sistema que trata de satisfacer un conjunto de metas en un ambiente dinámico y complejo. Un agente está situado en el ambiente: éste puede sentir el ambiente a través de sus sensores y actuar sobre el ambiente usando sus actuadores”*.

Un agente puede exhibir diferentes formas, en dependencia del tipo de ambiente que éste habita. Se pueden diferenciar claramente tres tipos de agentes, comúnmente definidos y utilizados en diferentes campos de investigación:

- Agentes que poseen una estructura física y habitan un mundo físico, estos agentes típicamente son robots.

- Agentes simulados en software y que habitan en ambientes también simulados en software, estos agentes comúnmente son robots o “animats” simulados en la pantalla de una computadora.
- Agentes construidos en software y que habitan un ambiente cibernético consistente en computadoras y redes, estos agentes son conocidos como “agentes software” o “agentes interfaz”.

Ejemplos de agentes que poseen una estructura física y habitan un mundo físico son *el robot de seis patas* de Brooks, construido a partir de la arquitectura embebida propuesta por este mismo autor (*The Subsumption Architecture*) (Brooks, 1989); y *Bed Hope*, un robot móvil autónomo basado en transputers, y construido a partir de la red neuroconectora propuesta por Halperin (Hallam, Halperin y Hallam, 1994). Ejemplos de agentes simulados y que habitan en un ambiente también simulado son *el insecto artificial* de Beer, cuyo diseño fue basado en parte sobre comportamientos y circuitos neuronales específicos de varios insectos reales (Beer, Chiel y Sterling, 1990); y *la criatura artificial simulada* de Maes, basada sobre el mecanismo *bottom-up* para la selección de comportamiento propuesto por esta misma autora (Maes, 1990 y Maes, 1991). Finalmente, ejemplos de agentes software vienen dados por toda la categoría de los llamados *agentes interfaz* (Maes, 1994), los cuales son sistemas autónomos inteligentes expresados como entidades de software, que llevan a cabo un conjunto de operaciones solicitadas por un usuario u otro programa con cierto grado de independencia y autonomía, empleando para esto algún conocimiento o representación de las metas o deseos del usuario. Un ejemplo particular de un agente interfaz lo constituye un sistema que navega en redes de computadoras para encontrar un tipo particular de datos.

Un agente posee metas, y en dependencia del dominio en el cual el agente haya sido definido éstas pueden tomar diferentes formas. Para satisfacer sus metas, un agente autónomo debe seleccionar, en cada momento en el tiempo, la acción más apropiada entre todas las acciones posibles que éste pudiera ejecutar. Esto es precisamente lo que en el contexto de los sistemas basados en el comportamiento se conoce como el problema de la selección de acciones.

1.2. Mecanismos para la selección de acciones

En el contexto de los sistemas basados en el comportamiento, el problema de la selección de acciones (PSA) se refiere a cual acción el agente (robot, *animat*, o criatura artificial) debe seleccionar en cada momento en el tiempo, mientras que un mecanismo para la selección de acciones (MSA) especifica cómo estas acciones son seleccionadas. Un MSA es un mecanismo computacional el cual debe producir como salida una acción seleccionada cuando diferentes estímulos externos y/o internos han sido proporcionados como entradas.

Los estímulos externos son percepciones del estado del mundo externo del agente, tales como fuente de comida percibida, fuente de agua percibida, lugar de descanso percibido, etc. Los estímulos internos son percepciones del estado del mundo interno de la entidad, tales como hambre, sed, cansancio, etc. Las acciones son conductas externas ejecutadas por el agente en respuesta a la presencia de determinados estímulos externos y/o internos. Sólo una acción podrá

ser ejecutada en un momento del tiempo. Un MSA combina estímulos externos e internos y selecciona entre todas las posibles acciones potenciales que puede ejecutar la entidad en el momento actual, aquella que resulte ser la más adecuada.

Entre los principales MSA y trabajos relacionados con la selección de acciones se encuentran los presentados por Tinbergen (Tinbergen, 1950 y Tinbergen, 1951), Lorenz (Lorenz, 1950 y Lorenz, 1981), Baerends (Baerends, 1976), Brooks (Brooks, 1986 y Brooks, 1989), Rosenblatt y Payton (Rosenblatt y Payton, 1989), Maes (Maes, 1990 y Maes, 1991), Beer (Beer, 1990 y Beer, Chiel y Sterling, 1990), Halperin (Hallam, Halperin y Hallam, 1994) Negrete (Negrete y Martínez, 1996), y Goetz (Goetz y Walters, 1997). Estos mecanismos han sido inspirados en modelos pertenecientes a disciplinas tales como etología, biología, psicología, ciencias cognitivas, robótica, ingeniería, redes neuronales artificiales e inteligencia artificial, entre otras. Algunos de estos mecanismos abordan completamente el problema de la selección de acciones, mientras que otros solo tratan parte del problema.

1.2.1. El mecanismo de Tinbergen

El mecanismo propuesto por Tinbergen no constituye una especificación detallada de un MSA, éste fue concebido solo como una "hipótesis de trabajo". Esto justifica que aspectos relevantes de la selección de acciones no puedan ser apreciados de una forma clara en el modelo. El mecanismo aborda el PSA completo, con un marcado énfasis en la "etapa reproductiva".

El MSA de Tinbergen es una red jerárquica de nodos o centros. En el nivel jerárquico superior (nivel de mayor instinto) se encuentran los nodos que representan comportamientos más generales, tales como "reproducción" y "alimentación" (nivel sistema). Los restantes niveles sucesivos corresponden a los niveles subsistemas y de conceptos de menor abstracción. En el nivel de menor jerarquía se encuentran las acciones consumatorias, tales como "copular" y "comer". La figura 1.1 muestra una representación esquemática de los niveles de jerarquía en el modelo de Tinbergen. En esta figura, los círculos representan nodos o centros, las flechas continuas verticales e inclinadas representan la entrada de energía, proveniente desde el nodo superordinado o desde estímulos internos o externos motivacionales, las flechas horizontales gruesas se corresponden con los mecanismos de liberación innata, las líneas sólidas horizontales son los bloqueos a la propagación de la activación de los nodos, y las flechas discontinuas son los mecanismos de inhibición mutua intranivel. Cinco parámetros fundamentales caracterizan a los nodos en la red: el nivel de activación, el valor umbral, el mecanismo de liberación innato, los estímulos externos, y el valor crítico.

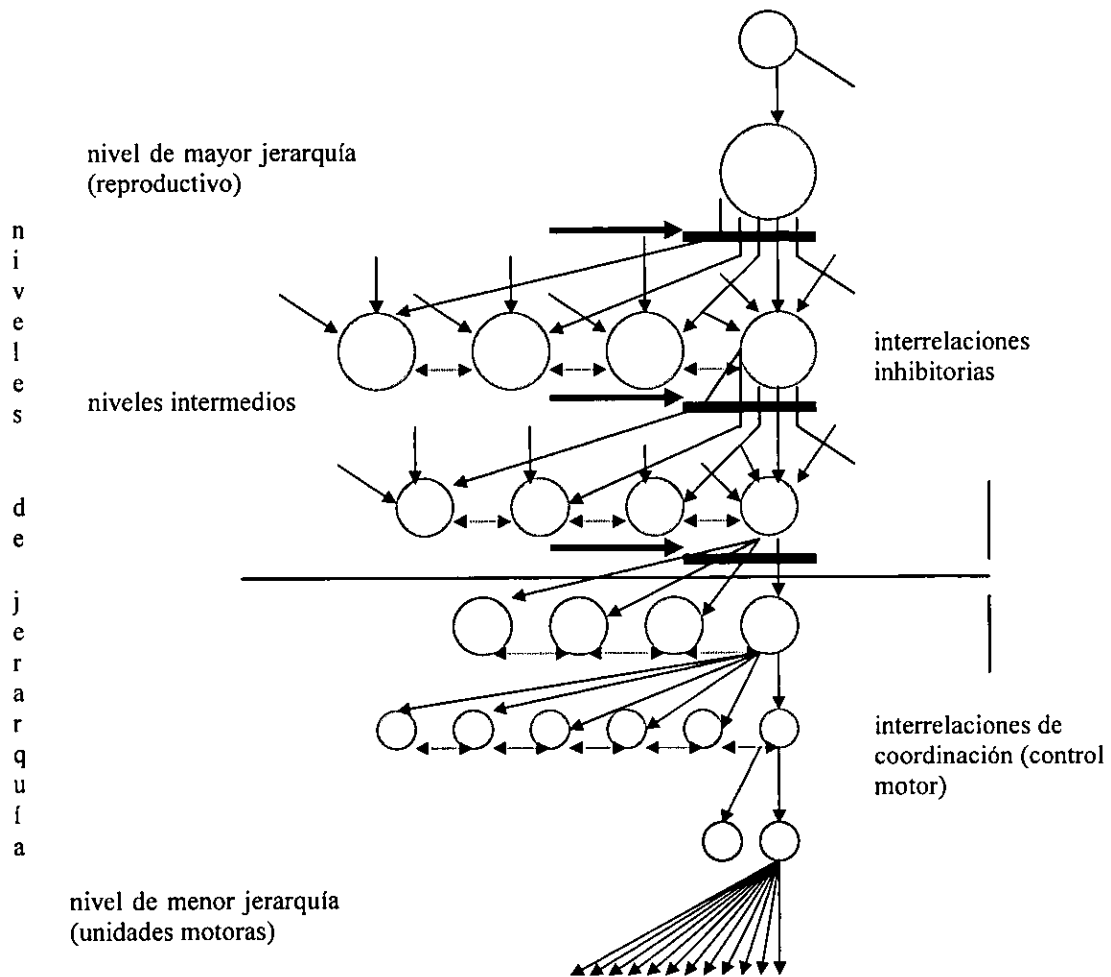


Figura 1.1. Representación esquemática de los niveles de jerarquía para un nodo o centro (niveles de instinto) en el modelo de Tinbergen. Esquema basado en la representación original que aparece en (Tinbergen, 1951).

El funcionamiento del MSA de Tinbergen puede ser explicado en los siguientes términos. Los nodos ubicados en el nivel de mayor jerarquía computan un nivel de activación a partir de la energía recibida desde las motivaciones, la cual usualmente corresponde a estímulos internos. Esta activación es entonces pasada a los nodos subordinados en los niveles inferiores, si el nodo emisor no se encuentra "bloqueado". Un nodo está "bloqueado" cuando su mecanismo de liberación innata está inactivo. La función de estos mecanismos es desbloquear aquellos nodos que son relevantes a la situación actual (o sea, los nodos que pueden contribuir o desencadenar la ejecución de una acción apropiada a las condiciones perceptuales observadas). Los mecanismos de liberación innata están controlados por los estímulos externos.

El impacto de los estímulos externos sobre el mecanismo de liberación innata es determinado a partir de una suma pesada. Cuando el estímulo perceptual excede un valor crítico, entonces el

mecanismo de liberación innata es activado y en turno el nodo es "desbloqueado", si además su nivel de activación excede un determinado valor umbral. Este hecho permite que la activación del nodo pueda ser pasada a los nodos subordinados en los niveles de menor jerarquía. El nivel de activación de un nodo también puede ser decrementado, a partir de las ligas inhibitorias existentes dentro de cada nivel. La figura 1.2 representa un nodo correspondiente a un nivel intermedio de la red jerárquica de Tinbergen.

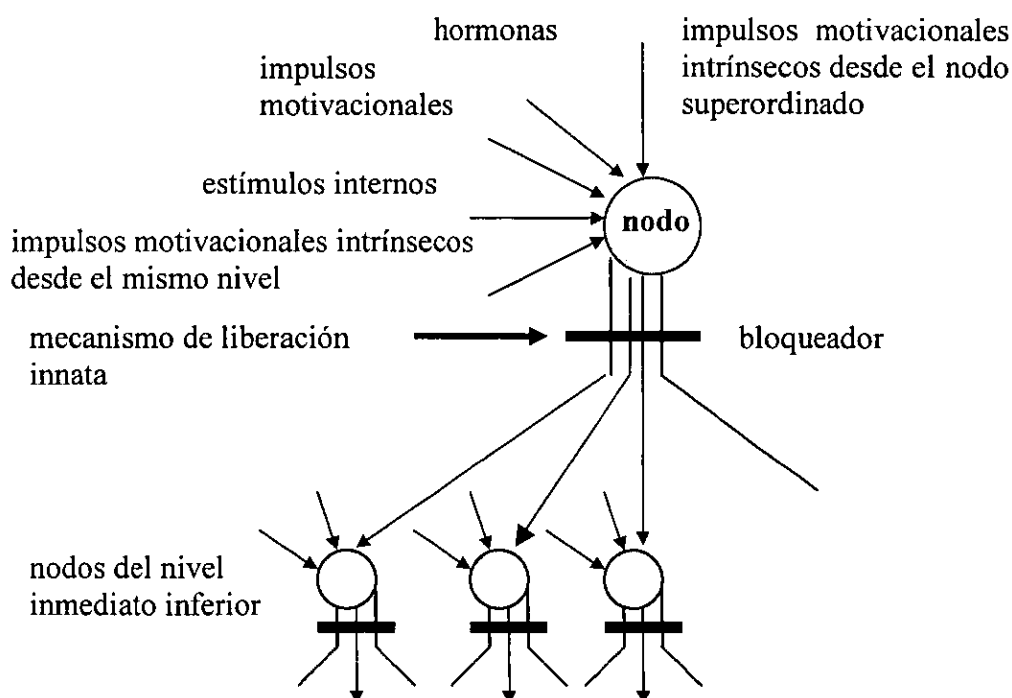


Figura 1.2. Representación de un nodo o centro en el modelo de Tinbergen. Esquema basado en la representación original que aparece en (Tinbergen, 51).

1.2.2. El modelo de Lorenz

El MSA propuesto por Lorenz (Lorenz presentó dos modelos muy similares para tratar el PSA, aquí haremos referencia solo al modelo original) es un modelo psico-hidráulico basado en una analogía entre el potencial de acción específico y la presión ejercida sobre una válvula por una masa de agua contenida en un depósito. El modelo de Lorenz pretende explicar algunos fenómenos etológicos, sin abordar el PSA completo, sino solo una parte de la selección de acciones. Este modelo no constituye una especificación detallada de un MSA, por lo que resulta poco claro y específico en algunos aspectos relativos a la selección de acciones. La figura 1.3 muestra una representación basada en el modelo original psico-hidráulico de Lorenz. En este

modelo la tendencia de que una acción particular sea activada es igual a la fuerza que está siendo ejercida sobre la válvula V. La expresión de la fuerza F es la siguiente:

$$F = S_p + k \cdot R - S$$

donde: $k \cdot R$ es la presión, la cual es proporcional al volumen de agua en el depósito R; S_p es la fuerza ejercida por los estímulos externos; y S es una fuerza inhibitoria proporcionada por el resorte, la cual contrarresta la fuerza total ejercida sobre la válvula V.

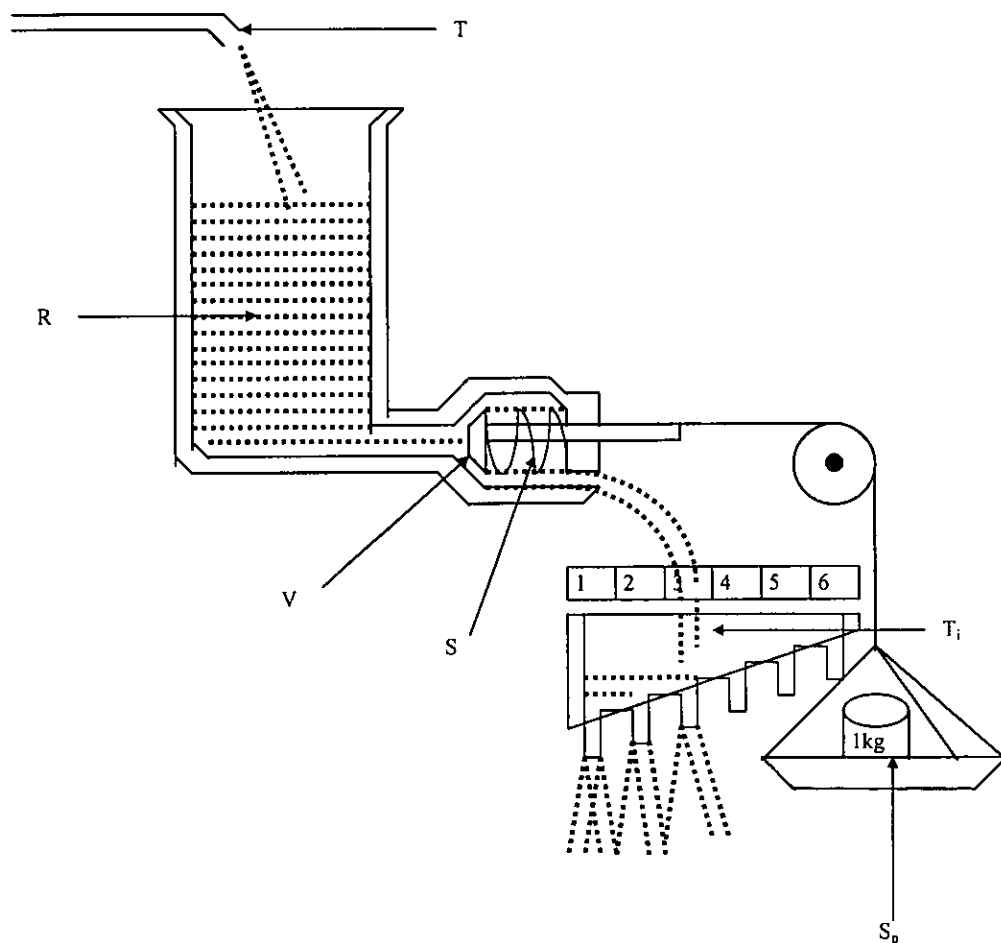


Figura 1.3. Representación del modelo original psico-hidráulico de Lorenz. Esquema basado en la representación original que aparece en (Lorenz 1950).

La selección de una acción en el modelo de Lorenz depende de la magnitud de los estímulos internos y externos presentes. El estímulo interno está dado por el flujo continuo de agua a través del tubo T, mientras que el estímulo externo corresponde a la fuerza externa S_p agua ejercida sobre la válvula V. El estímulo interno es considerado constante, y como resultado la cantidad de

agua que llega a través del tubo T se va acumulando en el depósito R. Esto implica que un tipo de comportamiento particular pudiera ser liberado aún en la ausencia total de estímulo externo, si el estímulo interno es lo suficientemente fuerte. Por otra parte, los estímulos internos y externos son adicionados conjuntamente, dependiendo de la magnitud de esta suma el comportamiento que será seleccionado. T_i es la proporción en la cual el agua llega a la cubeta. El llenado de la cubeta y la cantidad de desagües de ésta a través de los cuales fluye el agua dependen de T_i .

Mientras mayor sea la cantidad de agua contenida en el depósito y mayor sea la fuerza S_p ejercida sobre la válvula V, entonces mayor será el volumen de agua que fluye fuera del depósito por unidad de tiempo y como resultado final el agua saldrá al exterior por una mayor cantidad de desagües en la cubeta. El número de desagües en la cubeta por los cuales fluye el agua al exterior representan la intensidad de la ejecución del tipo de comportamiento seleccionado. Cuando un comportamiento está siendo ejecutado, el nivel de agua en el depósito disminuye, por lo que la presión que ejerce la masa de agua sobre la válvula V también disminuirá, esto a su vez implica que la intensidad con la cual está siendo ejecutada la acción también debe disminuir, hasta que ésta cese por completo. Cuando una acción consumatoria relevante para el sistema es ejecutada, entonces el depósito de agua es vaciado (o sea, el nivel de motivación es inicializado en cero).

1.2.3. El modelo de Baerends

Los mecanismos para la selección de acciones propuestos por Baerends están inspirados en los estudios etológicos realizados por este autor en clases particulares de insectos y aves: la avispa cavadora (*digger wasp*), y la gaviota arenquera (*herring gull*), respectivamente.

La figura 1.4 muestra un ejemplo de la estructura de un MSA similar a los propuestos por Baerends. Estos mecanismos son estructuras jerárquicas de nodos, donde cada nodo en el nivel superior controla la activación de sus correspondientes nodos subordinados en el nivel (o niveles) inmediato inferior. Dentro de cada nivel de jerarquía los nodos mantienen relaciones inhibitorias entre ellos, de forma tal que en un momento dado, sólo un nodo podrá estar activo. El nivel de mayor jerarquía es el nivel sistema (localizado en la parte superior del esquema), el cual ejerce un control directo sobre sus nodos subordinados en el nivel subsistema inmediato inferior, pudiendo existir más de un nivel subsistema. El nivel de menor jerarquía es el nivel acciones o patrones de acción fija (localizado en la parte inferior del esquema).

Para ejemplificar el control jerárquico y procesos inhibitorios que toman lugar interniveles e intraniveles, tomemos en cuenta el modelo de Baerends basado en el comportamiento de incubación de la gaviota arenquera (Baerends 1976). En este modelo el nodo "huir", perteneciente al nivel sistema, ejerce control jerárquico sobre el nodo "locomoción", perteneciente al nivel subsistema, mientras que este último controla la activación de los nodos "elevarse", "caminar", y "volar", pertenecientes al nivel acción. El nodo "incubación", controla los nodos "construir" y "anidar", en el nivel subsistema. Entre los nodos "huir", "incubación" y "limpiar y acomodar las plumas", todos en el nivel sistema, existen relaciones inhibitorias, de forma tal que en un momento dado sólo uno de estos nodos podrá mantenerse activo. La selección de un nodo en un nivel de jerarquía depende de los estímulos internos y externos recibidos. Una vez que un nodo se convierte activo, éste tendrá una tendencia a mantenerse activo durante un intervalo de tiempo.

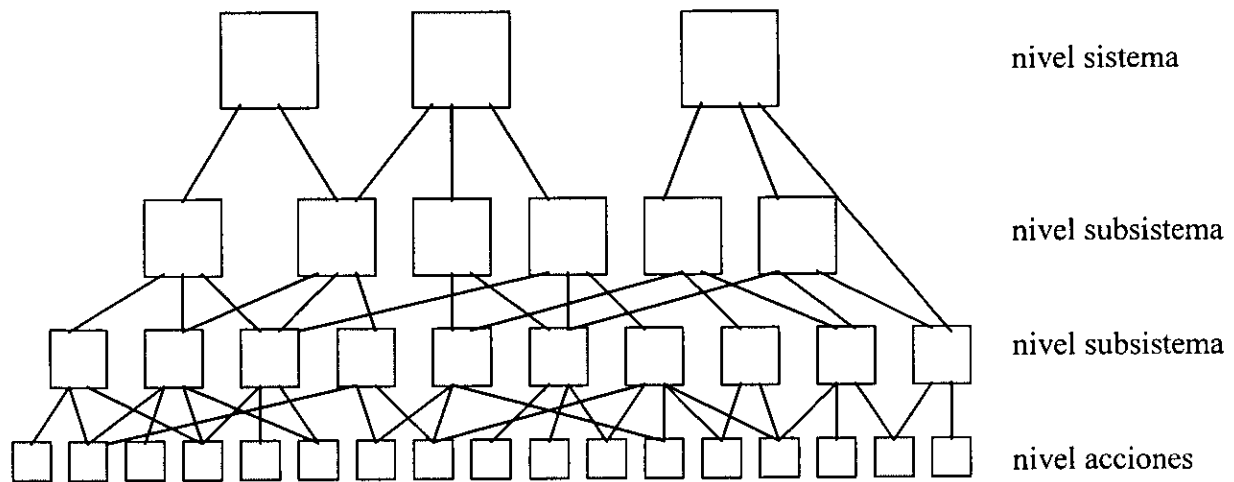


Figura 1.4. Ejemplo de una estructura de un MSA similar a las propuestas en los modelos de Baerends. Esquema basado en la representación original que aparece en (Lorenz, 1981).

1.2.4. La arquitectura "embebida" de Brooks

La arquitectura "embebida" de Brooks (*subsumption architecture*) no es exactamente un MSA. Ésta proporciona un ambiente computacional el cual puede ser utilizado como un medio para implementar sistemas de control robótico que engloben tareas de percepción y acción, además de manejar el problema de la emergencia de comportamientos. La arquitectura es una red completamente distribuida de máquinas de estado finito aumentadas, la cual proporciona un método incremental para la construcción de sistemas de control complejo en robots móviles. La construcción de un robot móvil a partir de esta arquitectura es un proceso incremental, en el cual pueden ser integradas una gran cantidad de entradas sensoriales y salidas (acciones). La arquitectura "embebida" proporciona mecanismos que permiten aumentar de una forma incremental la red de máquinas de estado finito con temporizadores internos. Este proceso incremental permite mejorar el desempeño del robot ante las tareas e incrementar el repertorio de las tareas (acciones) que el robot puede ejecutar.

Como se puede apreciar en la figura 1.5, una máquina de estado finito aumentada consiste de un conjunto de registros y un conjunto de temporizadores conectados a una máquina de estado finito convencional, la cual controla una red combinatoria alimentada por los registros. La función de los registros añadidos es permitir la llegada de mensajes (paso de mensajes) a la máquina de estado finito. Dos hechos pueden disparar un cambio de estado en la máquina de estado finito: la llegada de un mensaje desde otra máquina de estado finito o que expire el tiempo controlado por el temporizador.

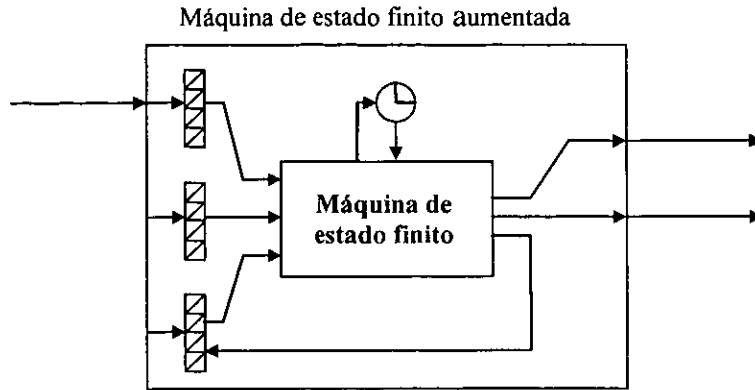


Figura 1.5. Una máquina de estado finito aumentada. Esquema basado en la representación original que aparece en (Brooks, 1989).

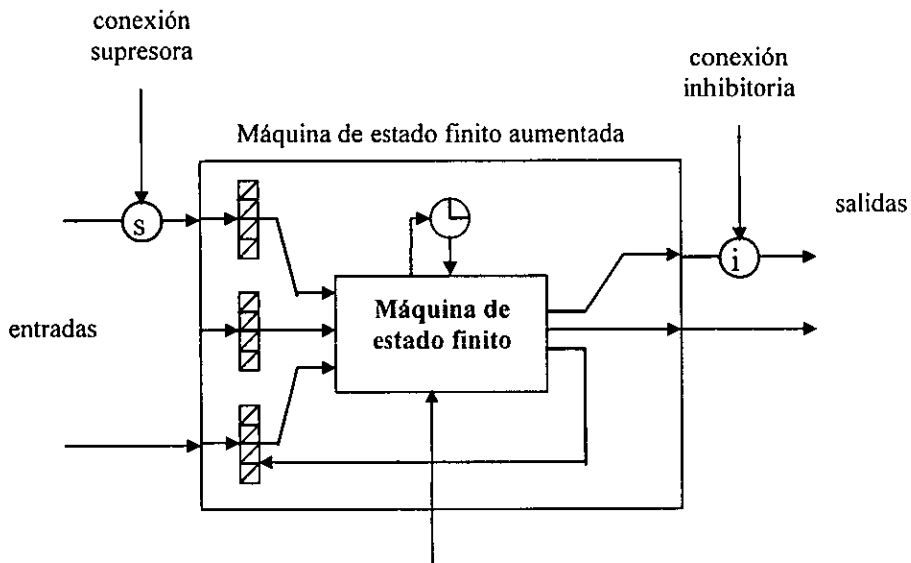


Figura 1.6. Formas en que una máquina de estado finito aumentada puede afectar a otra al ser adicionada a la red. Esquema basado en la representación original que aparece en (Brooks, 1989).

Sobre una configuración inicial de una red de máquinas de estado finito aumentadas, nuevas máquinas pueden ser adicionadas al conectar éstas a la estructura inicial. Como se muestra en la figura 1.6, cuando una nueva máquina es adicionada a la red, ésta puede afectar a la máquina a la cual ha sido conectada en varias formas: i. proporcionando una nueva entrada a la máquina a la cual ha sido conectada, ii. inicializando el estado de la máquina, iii. suprimiendo una de sus entradas, o iv. inhibiendo una de sus salidas.

Una red compuesta por 57 máquinas de estado finito aumentadas fue construida incrementalmente para controlar la locomoción de un robot móvil (Brooks 1989). El proceso incremental consistió en la adición de máquinas de estado finito aumentadas a una red inicial existente. La red resultante consistió de una colección de sistemas de control robótico para controlar las acciones que integran el sistema de locomoción del robot (pararse, andar simple, fuerza en el balanceo de las patas, levantarse, estabilizarse, deambular, deambular con dirección, entre otros).

La arquitectura "embebida" de Brooks ha demostrado ser una técnica muy útil para la implementación de sistemas de control de robots móviles. Los aspectos más relevantes de esta arquitectura resultan de mayor interés para el problema de la construcción de robots móviles que para el problema de la selección de acciones.

1.2.5. El modelo de Rosenblatt & Payton

El mecanismo propuesto por Rosenblatt & Payton es una red de nodos *feed-forward* jerárquica, diseñada para solucionar partes del PSA en un robot móvil. Este mecanismo resulta ser muy similar en muchos aspectos a los modelos jerárquicos propuestos por Tinbergen y Baerends. Al igual que en estos mecanismos, el nivel de mayor jerarquía de la red es el nivel sistema. En este nivel se encuentran los nodos que representan comportamientos más generales. Los restantes niveles sucesivos corresponden a los niveles subsistemas y de conceptos de menor abstracción. El nivel de menor jerarquía es el nivel de la vía final común conductual, los nodos ubicados en este nivel representan acciones.

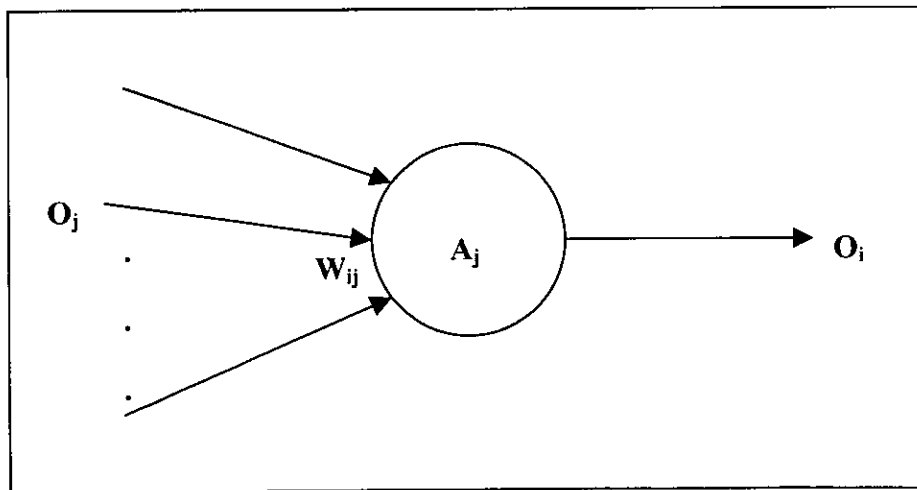


Figura 1.7. Un nodo en la red de Rosenblatt & Payton. Esquema basado en la representación original que aparece en (Rosenblatt & Payton, 1989).

Como se puede apreciar en la figura 1.7, los nodos en esta red son muy similares a las neuronas estándares utilizadas en las redes neuronales artificiales. En esta figura, O_i es la salida del nodo i , W_{ij} es el peso de la conexión entre el nodo i y el nodo j , $A_j = f_A(O_1 W_{1j}, \dots, O_n W_{nj})$ es el nivel de activación del nodo j , y $O_j = f_O(A_j)$ es la salida de la unidad j . Los parámetros que caracterizan a

un nodo son el vector de entrada, el vector de pesos, el nivel de activación y el valor de salida. La diferencia fundamental entre este tipo de nodos y las neuronas artificiales es que la activación en los primeros no se computa necesariamente como una suma pesada, ésta puede ser cualquier función de las entradas pesadas.

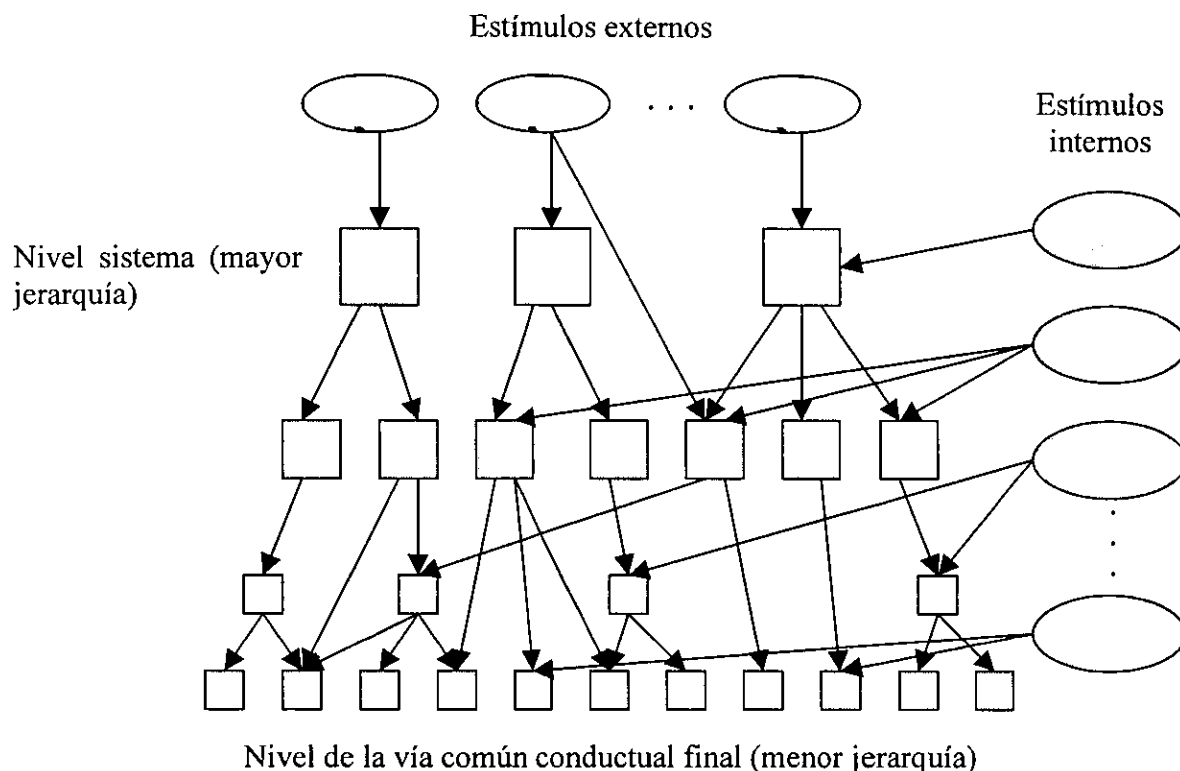


Figura 1.8. Ejemplo de una estructura de un MSA similar al propuesto por Rosenblatt & Payton.

La figura 1.8 ejemplifica el tipo de estructura que exhibe la red de Rosenblatt & Payton. Los nodos están organizados en niveles de jerarquía, desde el nivel sistema (nivel de mayor jerarquía) hasta el nivel acción (nivel de menor jerarquía). Los nodos ubicados en un nivel superior controlan la activación de sus nodos subordinados en el nivel o niveles inferiores. Un nodo en la red puede recibir activación desde los estímulos externos, los estímulos internos (motivaciones) y desde los nodos superordinados. Esta activación es propagada entre los nodos hasta llegar a los nodos que representan acciones, ubicados en el nivel vía común final conductual. En este nivel se lleva a cabo un proceso competitivo del tipo *winner-take-all*, para decidir cuál acción debe ser ejecutada.

No existen relaciones inhibitorias dentro de un mismo nivel de jerarquía, de forma tal que en un momento dado más de un nodo pudiera estar activo en un nivel. Por lo tanto, más de un nodo en un nivel jerárquico puede propagar activación a sus nodos subordinados en niveles inferiores. De esta forma, cada nodo activo en un nivel jerárquico superior puede "expresar un grado de preferencia" por cada uno de sus nodos subordinados. La preferencia de un nodo superordinado hacia un nodo subordinado puede ser positiva o negativa. El nivel de activación final de un nodo

subordinado se calcula en base a todas las preferencias recibidas por este nodo (este cómputo no tiene que ser precisamente una suma algebraica). Esta es una de las principales diferencias de este mecanismo con los mecanismos propuestos por Tinbergen y Baerends.

A pesar de que el mecanismo propuesto por Rosenblatt & Payton no aborda de forma explícita todos los detalles concernientes con el PSA, éste queda lejos de ser un modelo ambiguo dada la claridad de los aspectos definidos.

1.2.6. El mecanismo *bottom-up* de Maes

En el MSA de Maes una criatura artificial está compuesta por un conjunto de nodos, estructurados en una red distribuida y no jerárquica. Cada nodo consiste de una acción que la criatura puede ejecutar, y éstas representan alternativas consumatorias o apetitivas. Tres parámetros principales caracterizan a los nodos en la red: i. el nivel de activación; ii. un conjunto de condiciones perceptuales, las cuales tienen que ser satisfechas para que el comportamiento pueda ser "ejecutable"; y iii. un valor umbral.

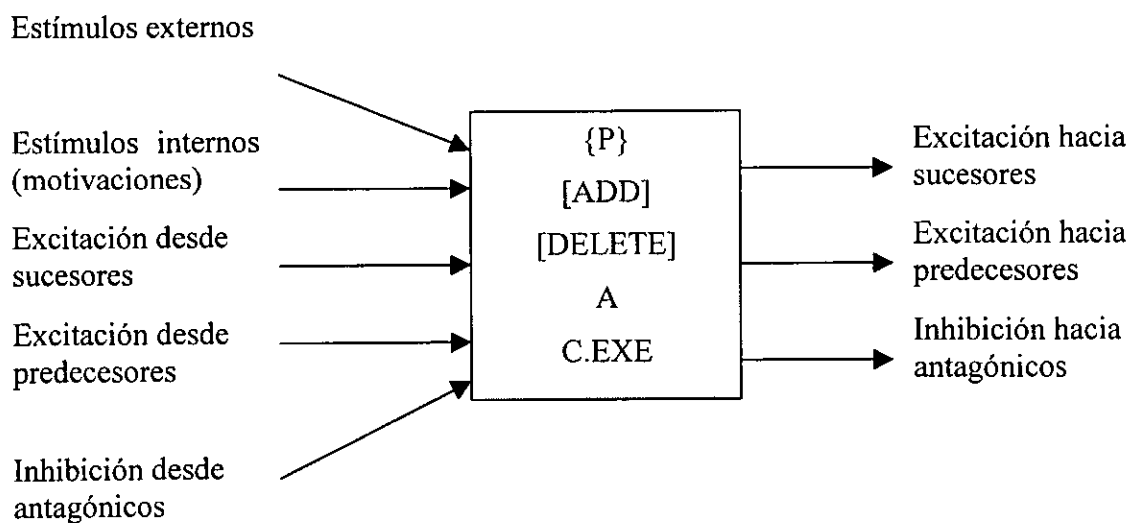


Figura 1.9. Un nodo en la red distribuida de Maes.

En la figura 1.9 se pueden apreciar las características de un nodo en la red de Maes. En esta figura, $\{P\}$ es el conjunto de precondiciones requeridas del medio externo para que el nodo se convierta en ejecutable, $[ADD]$ es la lista de condiciones del medio externo que el nodo probablemente podrá hacer verdaderas, $[DELETE]$ es la lista de condiciones del medio externo que probablemente se harán falsas una vez que el nodo sea ejecutado, A es el nivel de activación del nodo, y $C.EXE$ es el código que correrá (la especificación de la acción a ejecutar) si el nodo es ejecutado. La figura 1.10 muestra una red de nodos distribuida y no jerárquica similar a la propuesta por Maes. Un nodo en la red puede recibir dos tipos de conexiones externas: desde los sensores del medio externo (estímulos externos) y desde las motivaciones o metas (estímulos internos). También existen conexiones internas entre los nodos que pueden ser de tres tipos: ligas

predecesoras, sucesoras o antagónicas. Estas conexiones definen la entrada de nueva activación (energía) a los nodos de la red en la siguiente forma:

- Las condiciones perceptuales observadas incrementan el nivel de activación de los nodos que poseen estas condiciones.
- Las motivaciones o metas de la criatura incrementan el nivel de activación de los nodos consumatorios que están asociados con éstas.
- Los nodos "ejecutables" incrementan el nivel de activación de sus sucesores, mediante una fracción de su propio nivel de activación. La propagación de activación hacia los sucesores, contribuye a que éstos lleguen a ser "ejecutables" y puedan convertirse en activos.
- Los nodos "no ejecutables" incrementan el nivel de activación de sus predecesores, mediante una fracción de su propio nivel de activación. La propagación de activación hacia los predecesores, contribuye a que estos últimos puedan convertirse en "ejecutables" y posteriormente contribuyan a que los primeros lo sean también.

Cada nodo, independientemente de que sea "ejecutable" o no, decrementa el nivel de activación de sus antagónicos, mediante una fracción de su propio nivel de activación. Este proceso de inhibición contribuye a que no se conviertan en "ejecutables" nodos que pueden impedir que el nodo actual se vuelva activo. Un comportamiento puede activarse sólo cuando éste es "ejecutable" y su nivel de activación sobrepasa el valor umbral.

El algoritmo global utilizado para seleccionar el comportamiento a ejecutar es el siguiente:

1. Computar el nuevo nivel de activación teniendo en cuenta la excitación proveniente del medio exterior (mundo externo) y motivaciones.
2. Esparcir la activación hacia los nodos predecesores, sucesores y antagónicos.
3. Normalizar los niveles de activación de los nodos (por ejemplo, a partir de una función integradora con pérdida), de forma tal que el nivel de activación promedio se mantenga igual a cierta constante.
4. Activar el nodo que cumpla las siguientes tres condiciones:

El nodo es "ejecutable".

El nivel de activación del nodo sobrepasa su umbral.

El nodo posee el mayor nivel de activación entre todos los otros nodos que hayan cumplido las dos primeras condiciones.

Si no existe algún nodo que cumpla las condiciones 4.1 y 4.2 entonces reducir el umbral para todos los nodos en algún por ciento y volver al paso 1.

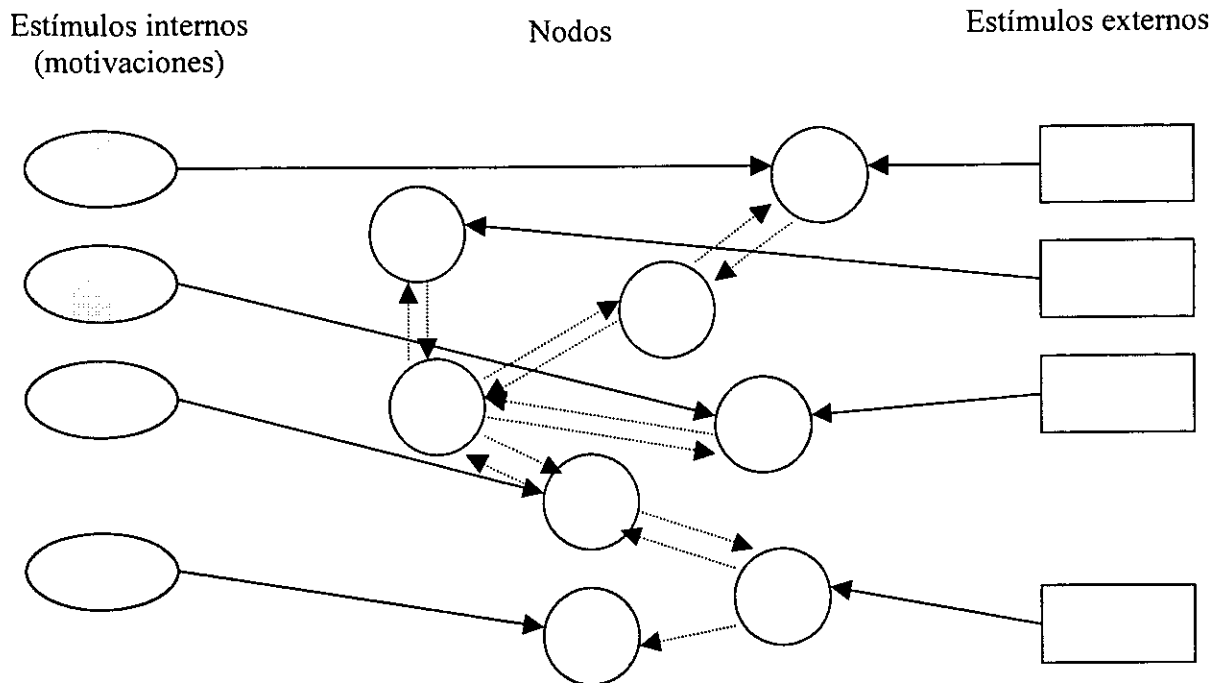


Figura 1.10. Una red de nodos distribuida y no jerárquica similar a la propuesta por Maes. Las flechas discontinuas son las ligas internas (ligas predecesoras, sucesoras y antagónicas) y las flechas continuas son las ligas externas (desde los medios interno y externo).

Algunos principios básicos del comportamiento animal y fenómenos de interés etológico observados en el mecanismo de Maes son los siguientes: i. el comportamiento no es estrictamente jerárquico; ii. parte del comportamiento es orientado por los estímulos externos (comportamiento reactivo); iii. parte del comportamiento es orientado por las motivaciones o metas internas; iv. el comportamiento es oportunístico, emerge al poder mostrar la criatura ambos comportamientos: reactivo y motivado; v. inhibición de comportamientos, los comportamientos más relevantes eliminan a los otros; vi. fatiga; vii. saciedad; y viii. variación de la atención.

1.2.7. El modelo neuronal de Beer

El modelo propuesto por Beer se centra fundamentalmente en el control motor de un insecto artificial, haciendo énfasis en los modelos neuronales que controlan los comportamientos de locomoción y alimentación. No obstante, en éste se pueden apreciar algunas cuestiones concernientes con la selección de acciones.

El insecto artificial propuesto por Beer es capaz de deambular, seguir-bordes y alimentarse (este último comportamiento tiene dos componentes: una apetitiva y una consumatoria). Varios de los comportamientos exhibidos por el insecto comparten el mismo aparato motor, por lo que resultan potencialmente incompatibles. Como se puede apreciar en la figura 1.11, el comportamiento alimentación toma precedencia sobre seguir-bordes, el cual a su vez toma precedencia sobre

deambular. La locomoción no es representada explícitamente en la figura 1.11 como un comportamiento independiente, ya que ésta es utilizada implícitamente por varios de los comportamientos.

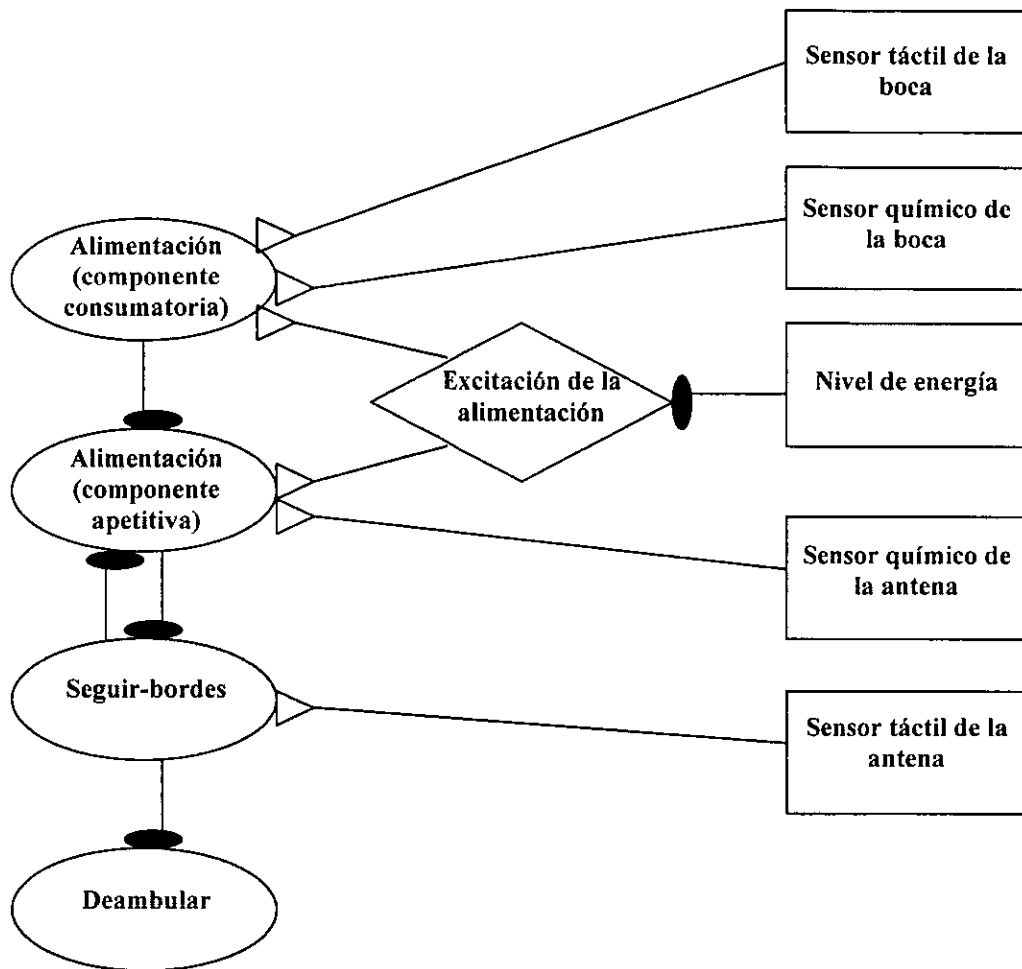


Figura 1.11. Organización de los comportamientos en el modelo de Beer. Esquema basado en la representación original que aparece en (Beer, Chiel y Sterling, 1990).

En la figura 1.11, las elipses representan los comportamientos que el insecto artificial es capaz de ejecutar; los rectángulos, los estímulos sensoriales responsables del comportamiento reactivo del insecto; y el diamante, el estado motivacional que ejerce influencia en las dos componentes del comportamiento alimentación. Los comportamientos interactúan entre sí a través de conexiones excitatorias e inhibitorias (las líneas que terminan en pequeñas elipses sólidas representan las conexiones inhibitorias).

A diferencia del modelo propuesto por Maes, los comportamientos en el modelo de Beer no son tratados como elementos individuales (nodos). Los comportamientos han sido implementados utilizando redes neuronales artificiales, algunas basadas sobre circuitos neuronales ya

especificados en varios animales (como es el caso de la red utilizada para implementar el comportamiento locomoción), mientras que otras son redes neuronales simples diseñadas *ad hoc*, capaces de generar los comportamientos deseados (por ejemplo, la red neuronal que implementa el comportamiento deambular).

El proceso de selección de acciones en el modelo de Beer puede ser explicado en los siguientes términos: en términos generales, siempre que un comportamiento de orden superior es disparado, los comportamientos de orden inferior son suprimidos. Sin embargo, esto no significa que el modelo propuesto sea estrictamente jerárquico, ya que la relación de precedencia entre algunos de los comportamientos depende fuertemente del contexto ambiental, por lo que dicha relación pudiera invertirse (este es el caso de las relaciones inhibitorias entre la componente apetitiva del comportamiento alimentación y el comportamiento sigue-bordes).

Algunas características básicas del comportamiento animal observadas en el mecanismo de Beer son las siguientes: (i) la ejecución de una acción dada no depende solo de la presencia de estímulos ambientales apropiados, sino también del estado motivacional interno del insecto artificial (comportamiento motivado); (ii) el comportamiento motivado que exhibe el insecto artificial se caracteriza por ser espontáneo, por la existencia de un nivel modulador de las respuestas y por ser persistente; (iii) el repertorio de comportamientos exhibe cierta organización, algunos comportamientos toman precedencia sobre otros, algunos comportamientos son mutuamente excluyentes y el cambio de un comportamiento a otro depende tanto de las condiciones ambientales como del estado motivacional. Por otra parte, a pesar de la tendencia hacia el comportamiento adaptativo que caracteriza al insecto artificial, éste no exhibe ninguna forma de plasticidad.

1.2.8. La red neuroconectora de Halperin

El MSA propuesto por Halperin es una red neuronal no supervisada *feed-forward*, llamada por la autora "red neuroconectora". Como se puede apreciar en la figura 1.12, en la red neuroconectora las neuronas se encuentran organizadas en tres estratos: un estrato sensorial (S), uno liberador (L) y uno de comportamiento (B), con conexiones de pesos ajustables entre los estratos S y R (estas *synapsis* definen el aprendizaje en el modelo), conexiones *feed-forward* y *feed-backward* de peso fijo entre los estratos R y B, y conexiones inhibitorias de peso fijo entre las neuronas del estrato B.

Las entradas provenientes del medio exterior (estímulos sensoriales) y las correspondientes a las metas internas o motivaciones del robot, son preprocesadas por una red autoorganizativa y pasadas a las neuronas del estrato sensorial (neuronas S_i). Por lo tanto, el patrón de disparo de las neuronas de este estrato en un momento dado reflejará el estado perceptual y motivacional actual del robot. Este patrón de disparo es transferido por igual a cada una de las neuronas del estrato liberador (neuronas L_k). La cantidad de entrada que recibirá cada neurona L_k dependerá del vector de pesos asociado a la neurona. Las neuronas que computen un mayor nivel de activación, a partir de las entradas recibidas, serán aquellas que tendrán mayor probabilidad de convertirse en activas. Cada neurona del estrato liberador envía su salida a la neurona correspondiente del estrato de comportamiento.

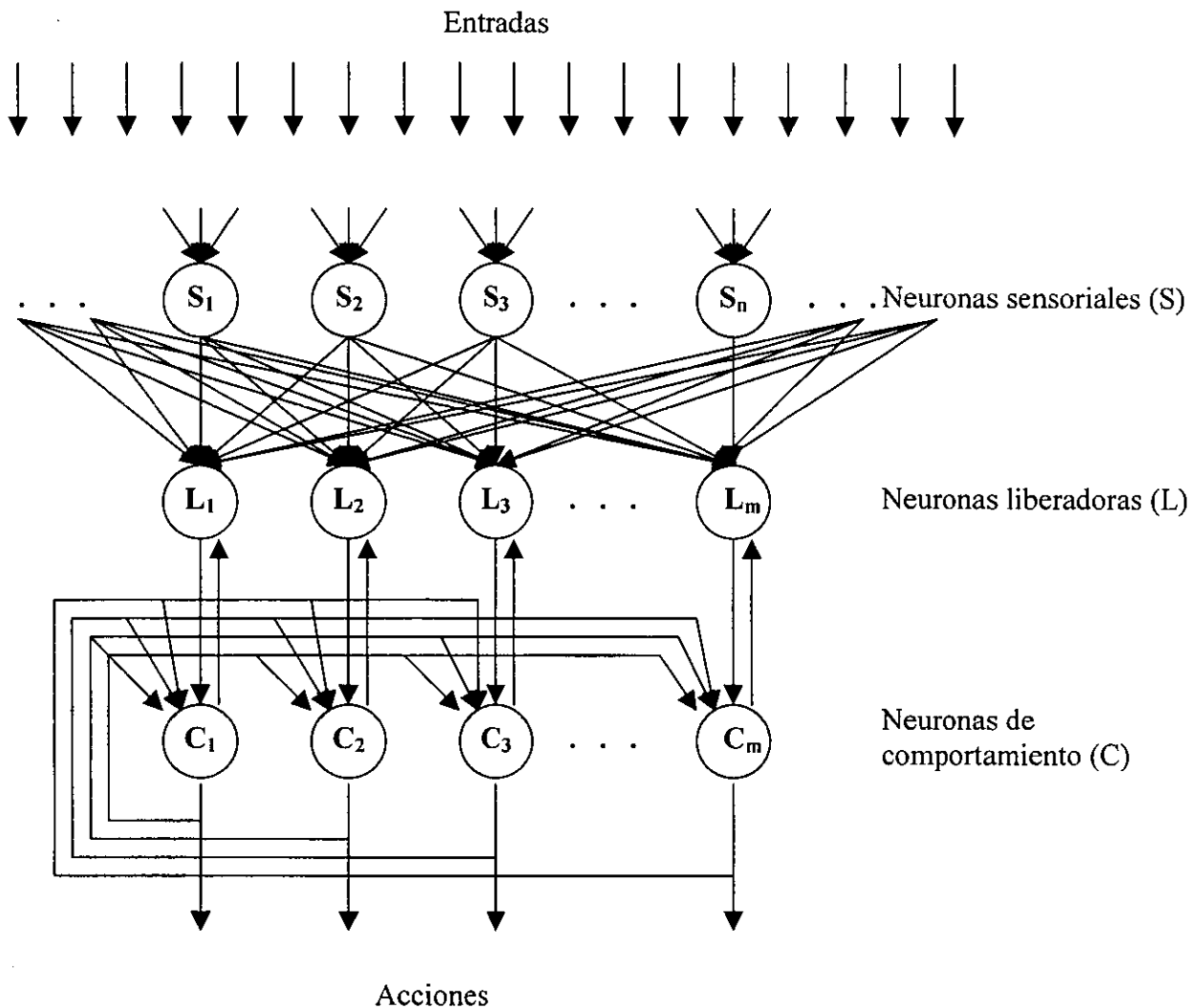


Figura 1.12. La Red Neuroconectora de Halperin. Esquema basado en la representación original que aparece en (Hallam, Halperin and Hallam, 1994).

Entre las neuronas C_i se lleva a cabo un proceso iterativo de tipo competitivo, esta competencia concluye con la selección de una sola neurona ganadora; la que persistió en mantener su disparo por encima de las restantes. La neurona ganadora C_j definirá entonces el comportamiento que será activado, mientras que la persistencia de este comportamiento queda definido por las conexiones *feedback* de peso fijo entre las neuronas del estrato liberador y las del estrato de comportamiento (conexiones uno a uno). El disparo de la neurona ganadora C_j finalizará, y por lo tanto la ejecución del comportamiento asociado, cuando cualquier otra neurona de este estrato se convierta en activa, y la cantidad de activación recibida sea suficiente como para inhibir a la actual ganadora.

En el modelo de Halperin se pueden observar los siguientes principios comunes al comportamiento animal y fenómenos de interés etológico: i. parte del comportamiento del robot es reactivo; ii. parte del comportamiento del robot es motivado, siendo característico en este comportamiento la espontaneidad, los cambios en la respuesta y la persistencia; iii. diferentes formas de plasticidad, entre las que se encuentran condicionamiento clásico (Pavloviano), condicionamiento instrumental, y condicionamiento diferido (tendencia a mostrar un comportamiento particular en respuesta a un estímulo previamente inefectivo pero que es presentado repetidamente antes de que la posterior descarga de este comportamiento finalice); y iv. influencias hormonales (no explicadas en el modelo).

1.2.9. El modelo neuro-humoral de Negrete

El modelo propuesto por Negrete constituye una versión neuronal del mecanismo *bottom-up* de Maes (Maes, 1990 y Maes, 1991), incorporando algunas nociones contemporáneas de los mecanismos neuro-humorales de la fisiología y neurofisiología del sistema nervioso central. Este mecanismo fue desarrollado para reproducir el comportamiento del forrajeo del mono aullador *Alouatta palliata*.

El modelo consiste en una red distribuida no jerárquica de neuronas neuro-humoral. Dos tipos de neuronas integran la red: neuronas perceptual-humoral y neuronas motor-humoral. A diferencia del mecanismo de Maes, donde cada nodo representa un tipo de comportamiento, en el modelo de Negrete un comportamiento es representado por un par de neuronas: una perceptual-humoral y una motor-humoral. En la figura 1.13 se muestran ambos tipos de neuronas y los flujos de señales de entrada y salida a cada una de éstas.

En un agente la transición entre comportamientos es determinada por la interacción dinámica de los siguientes elementos: el estado del mundo (condiciones del mundo externo percibidas por el agente, CE), las condiciones internas o motivaciones (MO) y un nivel de activación (NA) asociado a cada comportamiento. Al igual que en el mecanismo de Maes, un comportamiento en la red neuro-humoral (un nodo en la red de Maes) recibe activación desde las condiciones del mundo externo percibidas, desde los comportamientos sucesores y desde los comportamientos predecesores, y recibe inhibición desde los comportamientos antagónicos; a la vez que envía activación hacia sus comportamientos sucesores y predecesores, e inhibición hacia los comportamientos antagónicos. A diferencia del mecanismo de Maes, en la red neuro-humoral las condiciones internas o motivaciones constituyen un parámetro del propio comportamiento.

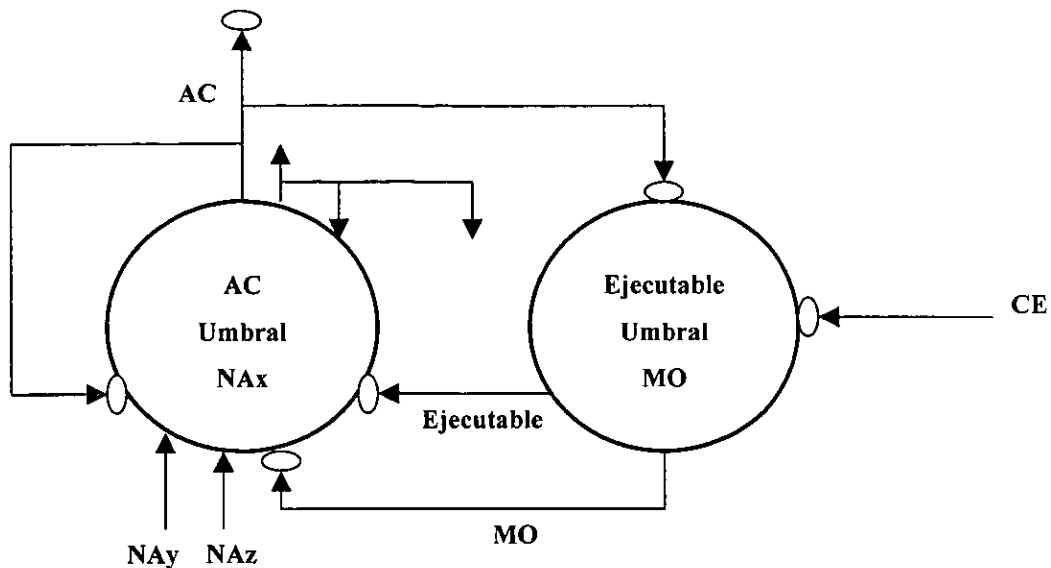


Figura 1.13. Un comportamiento en la red neuro-humoral. Esquema basado en la representación original que aparece en (Negrete y Martínez, 1996).

Tres parámetros caracterizan a una neurona perceptual-humoral: el parámetro Ejecutable, el cual toma valor de uno si las condiciones externas asociadas al comportamiento han sido percibidas o cero en otro caso; un valor umbral y la motivación. Una neurona motor-humoral se caracteriza por un valor umbral, un nivel de activación (NAx) y el parámetro AC (comportamiento atómico), el cual toma valor de uno si el nivel de activación es mayor que el umbral o cero en otro caso. Las neuronas neuro-humoral reciben entradas sinápticas y humorales y producen salidas sinápticas y al medio humoral. A nivel de cada comportamiento, las entradas externas, internas y humorales se combinan de forma aditiva.

1.2.10. La red recurrente de atractores de Goetz

La propuesta de Goetz consiste en considerar cada comportamiento de una red de comportamientos recurrente como un atractor de la red, de forma tal que los cambios repetitivos entre los diferentes comportamientos, en respuesta a pequeñas variaciones de las entradas externas e internas, puedan ser eliminados. Ejemplos de MSAs que exhiben algunas características de redes de comportamientos recurrentes son el mecanismo *bottom-up* de Maes, el insecto simulado de Beer y la red neuroconectora de Halperin.

Los nodos en una red recurrente pueden representar comportamientos, metas o sensores de estímulos externos. Cada comportamiento se caracteriza por un nivel de activación. Un nodo propaga parte de su activación sólo a una vecindad de nodos en la red recurrente. Este proceso de propagación de activación es repetido hasta que la red converge a un “mínimo de energía local”.

El hecho de considerar un comportamiento como un atractor significa que cuando el MSA ha seleccionado una acción que el agente debe ejecutar, entonces esta acción debe ser ejecutada persistentemente, sin que otras acciones, irrelevantes por el momento, perturben la ejecución de la acción seleccionada. Esto es, un atractor es por naturaleza estable a las perturbaciones.

Al imponer la no-linealidad como característica de la red de comportamientos se garantiza que más de un atractor pueda existir. Es decir, cualquiera de los comportamientos de la red puede convertirse en un atractor, en dependencia de las condiciones externas observadas y de las motivaciones del agente. La no-linealidad en la red de comportamientos también garantiza flexibilidad, de forma tal que la razón de cambio del nivel de activación de un nodo sea una función no lineal de los niveles de activación de los nodos vecinos. Esta no-linealidad asegura que cuando se requiera un cambio repentino en un comportamiento (por ejemplo, dada la aparición de un depredador), entonces no se necesiten de muchos ciclos de decisión antes que el nuevo comportamiento requerido sea activado.

La red recurrente de atractores fue utilizada para implementar una versión modificada del mecanismo *bottom-up* de Maes, y así determinar la relevancia de los atractores en la persistencia y flexibilidad de la selección de acciones.

1.3. Otros trabajos desarrollados

1.3.1. La aproximación “Sociedad de la Mente” a la selección de acciones

En el trabajo desarrollado por Cañamero (Cañamero, 1997) se propone una aproximación “Sociedad de la Mente” (Minsky, 1985) al problema de la selección de acciones. La característica relevante de esta aproximación es el uso de agentes motivación para coordinar y organizar el comportamiento de una sociedad. En este modelo la sociedad de la mente es llamada “Abbott”.

Un Abbott puede ser visto como una colección de agentes. Cada Abbott es construido sobre un estilo de arquitectura embebida (Brooks, 1989), en la cual los comportamientos más complejos son implementados añadiendo nuevos agentes a la sociedad, sin modificar los agentes ya existentes en ésta. Un Abbott está compuesto por una sociedad de nueve tipos diferentes de agentes, entre los que se encuentran los agentes comportamientos y los agentes emociones. Un agente comportamiento puede ser visto como un módulo de competencia en la arquitectura de red de agentes de Maes (Maes, 1989), mientras que un agente emoción es un agente que modifica el estado motivacional de la entidad. La selección de acciones en un Abbott se caracteriza por ser orientada a las metas y oportunística.

1.3.2. Los modelos de objetivos múltiples y de fusión de comportamientos usando votación de Pirjanian

El modelo de objetivos múltiples de Prijanian (Pirjanian, 1998 y Pirjanian, Christensen y Fayman, 1998) formaliza las ideas de control y síntesis de sistemas basados en el comportamiento utilizando la teoría de decisión de objetivos múltiples, donde cada comportamiento se modela como un estimador de funciones objetivos. La selección de acciones es determinada por la solución más satisfactoria, ya que podría no existir una solución óptima debido a conflictos entre los objetivos de los comportamientos que compiten. Prijanian analiza el modelo de Rosenblatt y Payton desde esta perspectiva.

El modelo de fusión de comportamientos (Pirjanian, 1998 y Pirjanian, Christensen y Fayman, 1998) utiliza técnicas de votación. En este modelo, un conjunto de comportamientos redundantes votan por las acciones posibles y la acción más favorecida es seleccionada.

1.3.3. Aprendizaje de nuevos comportamientos en un red de comportamientos

El mecanismo de selección de acciones propuesto por Negatu y Franklin (Franklin et. al., 1996 y Bogner et. al., 1999) es una red de comportamientos basada sobre una extensión de la red de comportamientos de Maes. El MSA propuesto es una colección de comportamientos conectados por ligas predecesoras y sucesoras, incluyendo un conjunto fijo de metas globales y una interfaz medioambiental. En el MSA propuesto por estos autores, un flujo de comportamientos es una colección de comportamientos los cuales sirven para satisfacer una única meta del agente. Estos flujos de comportamientos son aprendidos por el MSA, el cual los autores definen como adaptable.

El MSA incorpora algunos elementos de los sistemas basados en el conocimiento, como es el caso de una base de conocimiento y un planeador basado en casos. La base de conocimiento contiene conocimiento del dominio incluyendo los comportamientos. El planeador basado en casos es un sistema de razonamiento basado en casos para la generación de planes.

1.4. La combinación de estímulos externos e internos en los mecanismos revisados

En este capítulo hemos presentado una muestra representativa de los principales mecanismos de selección de acciones reportados en la literatura especializada. De forma breve, hemos descrito las características esenciales que exhiben estos mecanismos, orientando esta discusión hacia: ¿qué parte de la selección de acciones resuelve el mecanismo propuesto?, ¿cuál es la naturaleza o disciplina en la cual ha sido inspirado?, tipo de arquitectura en la que ha sido estructurado, nivel de detalle y/o explicación de los procesos que tienen lugar, formas de aprendizaje que logra exhibir el mecanismo, la forma en que se combinan las entradas externas e internas para la selección de acciones, y las nuevas propiedades que pueden emerger a partir de las características

de esta combinación. Tomando en cuenta estos aspectos, la tabla 1.1 presenta una comparación entre los diferentes mecanismos de selección de acciones que han sido presentados.

MSA	Disciplina en la que ha sido inspirado	Arquitectura	Combinación de entradas externas e internas	Formas de aprendizaje
Tinbergen	etología	red jerárquica de nodos, donde cada nodo representa un tipo de comportamiento	sumatoria	no exhibe
Lorenz	etología, hidrodinámica	modelo psico-hidráulico	sumatoria	no exhibe
Baerends	etología	red jerárquica de nodos, donde cada nodo representa un tipo de comportamiento	no especificada en el modelo	no exhibe
Brooks	robótica	red distribuida de máquinas de estado finito		
Rosenblatt y Payton	robótica y redes neuronales artificiales	red conexionista feed-forward jerárquica, los comportamientos son definidos por las conexiones entre los elementos de procesamiento	puede ser cualquier función de las entradas pesadas, aunque el modelo no especifica cual	no exhibe
Maes	etología, sistemas basados en el comportamiento	red de nodos distribuida no jerárquica, donde cada nodo representa un tipo de comportamiento	sumatoria	no exhibe
Beer	etología, neuroetología y redes neuronales artificiales	red de nodos semijerárquica, donde cada nodo es una red neuronal que implementa un tipo particular de comportamiento	sumatoria	no exhibe
Halperin	etología y redes neuronales artificiales	red conexionista feed-forward jerárquica, no supervisada	sumatoria	condicionamiento clásico, instrumental y diferido
Negrete	neurofisiología y etología	red distribuida no jerárquica de neuronas neuro-humorales	sumatoria	no exhibe
Goetz	redes neuronales artificiales recurrentes y teoría de atractores	red de nodos distribuida recurrente, donde cada nodo puede ser visto como un atractor	depende del MSA que se implemente en la red, para la implementación del mecanismo de MAES es la sumatoria	no exhibe

Tabla 1.1.

Como se puede apreciar en la tabla 1.1, la gran mayoría de estos mecanismos de selección de acciones proponen un modelo aditivo para la combinación de las entradas externas e internas. La adición de señales externas e internas pudiera interpretarse de muchas formas, pero la esencia es que este tipo de combinación de entradas modela mucho mejor un comportamiento con tendencia reactiva que un comportamiento motivado. Lo anterior puede ser explicado en los siguientes

términos: para valores grandes de las entradas externas, la combinación aditiva pudiera superar algún valor umbral previamente establecido y, por lo tanto, disparar una determinada acción externa, aún para valores no significativos de las entradas internas.

Por otra parte, en un modelo de combinación multiplicativo puro ambos tipos de entradas son necesarias para que una acción externa consumatoria de la entidad pueda ser ejecutada. Es decir, si consideramos que $\Sigma_j S_j$ es el conjunto de todas las entradas externas asociadas al estado interno E_i , entonces $\Sigma_j S_j * E_i \neq 0$ sólo si $\Sigma_j S_j \neq 0$ y $E_i \neq 0$. Los modelos multiplicativos puros tienen una razón límite a valores muy pequeños de las entradas externas y/o de las entradas internas, ya que no importa cuán grande es la entrada externa si la entrada interna es cercana a cero y viceversa.

Una alternativa para eliminar los efectos no deseados de las combinaciones aditivas y multiplicativas puras es considerar un único modelo que incorpore los elementos más valiosos de ambos tipos de combinación. En el presente trabajo nosotros proponemos un mecanismo de selección de acciones, el cual utiliza un esquema de combinación de entradas externas e internas donde ambos tipos de combinaciones tienen lugar. Es decir, en esencia en el esquema de combinación que se propone las entradas externas e internas interactúan multiplicativamente, lo cual puede ser interpretado al considerar que ambos factores son necesarios para disparar una acción externa consumatoria. Sin embargo, el esquema de combinación también contiene elementos aditivos, los cuales permiten que le sea atribuida una mayor importancia o peso a las entradas internas, y son finalmente los que determinan que propiedades tan interesantes como la selección de acciones dirigidas hacia el medio interno de la entidad y la ejecución de una conducta externa orientada a la búsqueda de una señal específica puedan emerger producto de este tipo de combinación.

Para la implementación del mecanismo de selección de acciones propuesto nos hemos basado en una arquitectura de pizarrón distribuida, la cual facilita la estructuración del mecanismo y la incorporación en un estilo de diseño *bottom-up* de nueva funcionalidad. En el siguiente capítulo se discuten las principales características estructurales y funcionales de la arquitectura de pizarrón y de las dos variantes desarrolladas por nosotros sobre esta arquitectura, las cuales proporcionan el sustrato básico para la estructuración del MSA propuesto.

CAPÍTULO 2

LA ARQUITECTURA DE PIZARRÓN Y LA SELECCIÓN DE ACCIONES

En el capítulo 1 fueron presentados algunos de los principales mecanismos de selección de acciones para un agente autónomo propuestos en la literatura. La descripción de estos mecanismos se basó en cuatro aspectos fundamentales: (1) disciplina o área en la cual fue inspirado el modelo, (2) arquitectura del mecanismo de selección de acciones, (3) forma de combinación de los estímulos externos e internos, y (4) principales características presentes en la selección de acciones. La discusión final del capítulo I se centró en la forma de combinación de los estímulos externos y externos utilizada por cada uno de estos MSAs, a partir de la clasificación de estos esquemas de combinación en dos principales aproximaciones: modelos aditivos y modelos multiplicativos; sin abordar en los tipos de arquitecturas más comúnmente usadas por estos mecanismos.

En el presente capítulo abordaremos el tema de las arquitecturas de los MSAs. Inicialmente identificaremos las principales clases de arquitecturas propuestas y las características más relevantes de éstas serán discutidas. Posteriormente describiremos la arquitectura que hemos seleccionado para la implementación del MSA propuesto en el capítulo III: la arquitectura de pizarrón y las dos principales variantes de esta arquitectura desarrolladas por nosotros (González y Negrete, 1997 y Negrete y González, 1998).

2.1. Arquitecturas de mecanismos de selección de acciones

Como se puede apreciar en la tabla 1.1, muchas de las arquitecturas de los MSAs propuestas tienen características en común. Por ejemplo, el mecanismo propuesto por Tinbergen, el modelo de Baerends y la red de Rosenblatt & Payton son mecanismos muy similares en muchos aspectos, pudiendo ser todos éstos clasificados como redes de nodos jerárquicas; la arquitectura embebida de Brooks, el mecanismo *bottom-up* de Maes y el modelo neuro-humoral de Negrete son todos ellos redes de nodos distribuidas no jerárquicas; mientras que el modelo de Beer y la red neuroconectora de Halperin comparten características de los modelos conexionistas.

De esta forma, las arquitecturas de MSAs propuestas pueden ser subdivididas en tres clases principales: (1) redes de nodos jerárquicas, (2) redes de nodos distribuidas y (3) redes conexionistas, las cuales a su vez pueden exhibir algún tipo de organización jerárquica, distribuida o ambas.

2.1.1. Redes jerárquicas

Este tipo de arquitectura propone una organización jerárquica de los diferentes comportamientos. Muchas de las arquitecturas jerárquicas propuestas están cercanamente inspiradas por modelos del comportamiento animal, como es el caso de los modelos propuestos por Tinbergen y Rosenblatt y Payton.

En una red de nodos jerárquica cada nodo representa un tipo particular de comportamiento. Los nodos se distribuyen en diferentes capas o estratos, donde cada capa representa un diferente nivel de jerarquía. Típicamente, las relaciones que se establecen entre estos niveles de jerarquía son relaciones de subordinación, donde un nodo en un nivel superior controla directamente sus nodos en el nivel (o niveles) inmediato inferior. En el nivel jerárquico superior se encuentran los nodos que representan comportamientos más generales (este nivel es comúnmente conocido como nivel sistema); los restantes niveles sucesivos corresponden a los niveles subsistemas y de conceptos de menor abstracción; mientras que en el nivel de menor jerarquía se encuentran las acciones que pueden ser ejecutables (acciones consumatorias).

Un nodo en un nivel de jerarquía intermedio puede recibir activación desde los estímulos externos, los estímulos internos (motivaciones), y desde los nodos superordinados; pudiendo también recibir entradas inhibitorias provenientes de otros nodos pertenecientes al mismo nivel de jerarquía. Comúnmente, los MSAs estructurados en redes jerárquicas usan algún tipo de selección de acciones en los niveles de mayor abstracción (niveles de mayor jerarquía) para preparar o sesgar la selección de acciones en los niveles más primitivos.

Una característica relevante de las redes jerárquicas es que este tipo de arquitectura tiene una tendencia a soportar motivaciones o metas más complejas que las redes distribuidas. Sin embargo, en muchas ocasiones la construcción de la red de nodos puede resultar un trabajo difícil, debido a que estos modelos basados en la etología tienden a tener una gran cantidad de parámetros, los cuales necesitan ser ajustados para obtener el comportamiento deseado.

2.1.2. Redes distribuidas

Una red distribuida puede ser vista como un conjunto de nodos competitivos (comúnmente conocidos como comportamientos), los cuales se comunican entre sí a través del paso de mensajes. Comúnmente estos mensajes consisten de energía de activación, o señales de supresión o inhibición. Entre los nodos de la red no se establecen relaciones jerárquicas por lo que, a diferencia de las redes jerárquicas, no es posible hablar de nivel sistema y nivel acciones. En este tipo de arquitectura, cada nodo representa un tipo de comportamiento (o acción) que la entidad puede ejecutar, y éstas representan alternativas consumatorias o apetitivas (Maes, 1990 y Maes, 1991). Ejemplos de MSAs estructurados en estas arquitecturas son el mecanismo *bottom-up* de Maes y la arquitectura embebida de Brooks.

Este tipo de arquitectura requiere que se especifique en un formalismo particular qué son las metas del agente, cómo las metas pueden variar en el tiempo, cómo las metas pueden ser reducidas a otras metas o a acciones, y qué son estas diferentes acciones (Maes, 1994). Es decir, las metas son explícitas en la especificación formal del agente.

Típicamente un nodo en la red se caracteriza por un nivel de activación y un valor umbral. Un nodo puede recibir señales desde los estímulos externos, desde los estímulos internos (metas o motivaciones) y desde otros nodos de la red. Las señales recibidas desde otros nodos de la red pueden incrementar o decrementar el nivel de activación del nodo, dependiendo si éstas son excitatorias o inhibitorias, respectivamente. Este proceso de propagación de activación entre los nodos de la red hace que los nodos más relevantes para las metas actuales incrementen su nivel de activación, y de esta forma puedan ejecutar la acción asociada a éstos.

A diferencia de las redes jerárquicas, las redes de nodos distribuidas ofrecen un modelo muy particular de relevancia de acciones. De aquí que la clase de agentes que puede ser construida con este tipo de arquitectura es restringida.

2.1.3. Redes conexionistas

A diferencia de las redes jerárquicas y distribuidas, los comportamientos en una red conexionista no son tratados como elementos individuales (nodos). En este tipo de arquitectura, los comportamientos son implementados utilizando redes neuronales artificiales, algunas basadas sobre conocidos circuitos neuronales de animales inferiores y otras son redes neuronales diseñadas *ad hoc*, capaces de generar los comportamientos deseados.

Dentro de esta arquitectura se han explorado dos variantes principales: (1) Una única red neuronal implementa todo el repertorio de comportamientos del agente, por lo que se pudiera decir que los comportamientos se encuentran distribuidos en la red. Un ejemplo de esta variante está dado por la red neuroconectora de Halperin y (2) Cada comportamiento es implementado utilizando una red neuronal particular, y entre las diferentes redes neuronales se establecen conexiones excitatorias e inhibitorias. Un ejemplo de esta variante lo constituye el insecto artificial desarrollado por Beer.

Comúnmente cuando un MSA es implementado utilizando redes conexionistas, entonces se requiere de algún subsistema o mecanismo que ejecute el preprocesamiento de las entradas generadas por los estímulos externos y por los estímulos internos, antes de que éstos sean presentados en la capa de entrada de la red. Aunque las redes conexionistas son por naturaleza sistemas distribuidos, este tipo de arquitectura permite que determinado grado de jerarquía sea especificado en el modelo. Finalmente, todas las propiedades inherentes al procesamiento de la información que se lleva a cabo en los modelos conexionistas pueden ser heredadas por este tipo de arquitectura.

2.2. La arquitectura de pizarrón

La arquitectura de pizarrón ha sido ampliamente usada en dominios de tareas donde los problemas son típicamente complejos y requieren de diversas fuentes de experticia para alcanzar la solución de los mismos. Como se podrá comprender más adelante, a la arquitectura de pizarrón

le es inherente la selección de acciones por naturaleza, aunque en el contexto en que este tipo de arquitectura ha sido más ampliamente utilizada estas acciones han respondido más al problema de qué conocimiento aplicar en un momento dado, que al problema de qué comportamiento ejecutar en un momento dado.

El concepto arquitectura de pizarrón fue concebido por investigadores en el campo de la Inteligencia Artificial en la década de los 70. La meta de esta investigación fue manejar el problema de la información compartida entre múltiples agentes expertos en la solución de problemas. El término arquitectura de pizarrón fue escogido para evocar una metáfora en la cual un grupo de expertos se reúnen alrededor de un pizarrón para solucionar cooperativamente un problema complejo. La arquitectura de pizarrón fue desarrollada por primera vez en el sistema para la comprensión del lenguaje HEARSAY II (Engelmore et. al., 1988; Erman et. al, 1988; Lesser y Erman, 1988 y Nii, 1989), posteriormente ha sido usada en una gran variedad de otros dominios de problemas, y abstraída en muchos medios ambientes para la construcción de sistemas.

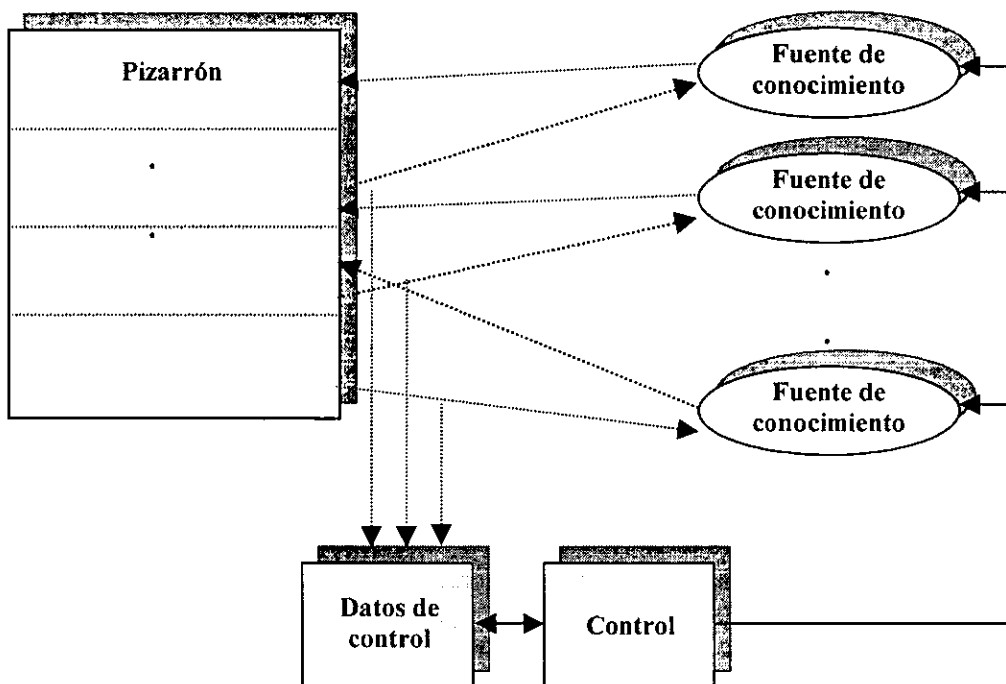


Figura 2.1. Componentes de la arquitectura de pizarrón

Como se puede apreciar en la figura 2.1, la arquitectura de pizarrón se define en términos de tres elementos básicos: un conjunto de módulos independientes, llamados fuentes de conocimiento, que contienen conocimiento específico acerca del dominio del problema; una estructura de datos compartida, llamada pizarrón, a través de la cual las fuentes de conocimiento se comunican entre

sí; y un mecanismo de control, o facilitador, el cual determina el orden en que las fuentes de conocimiento operarán sobre el pizarrón.

2.2.1. Las fuentes de conocimiento

Las fuentes de conocimiento son subsistemas de software modulares que representan diferentes puntos de vista, diferentes estrategias y diferentes tipos de conocimiento, acerca de cómo resolver un problema. Estos paradigmas para la solución de problemas incluyen: razonamiento basado en reglas, razonamiento basado en casos, razonamiento basado en modelos, redes neuronales, sistemas de lógica difusa, y algoritmos genéticos, entre otros. Las fuentes de conocimiento 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 solución sobre el pizarrón. La acción se vincula a la creación o modificación de elementos solución sobre el pizarrón.

En términos generales, la actividad de las fuentes de conocimiento es manejada 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 produce un único registro de activación de la fuente de conocimiento (RAFC). Sólo cuando un 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. Estas 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 con elementos solución al problema compartido.

2.2.2. El pizarrón

El pizarrón es utilizado como un almacén central para toda la información compartida. La información sobre el pizarrón representa hechos, asunciones y deducciones hechas por las fuentes de conocimiento durante el curso de la solución del problema. 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.

La información sobre el pizarrón es conocida como 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 de abstracción. Los diferentes niveles de abstracción representan la solución del problema en diferente cantidad de detalles, o en diferente contexto.

2.2.3. 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, comúnmente, comprende un detector de eventos y un administrador de agenda. Este maneja la interacción entre el pizarrón, las fuentes de conocimiento, y las fuentes externas tales como usuarios y subsistemas de control o adquisición de datos. El detector de eventos actualiza el pizarrón desde fuentes de entradas externas. El administrador de agenda decide de entre todas las fuentes de conocimiento participantes en la competencia, cuál ejecutará finalmente su acción sobre el pizarrón

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. Este pudiera 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, entonces 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, entonces 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, entonces 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 cuando 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 conocimiento 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.

Una sesión de solución de problemas inicia con el mecanismo de control (o algún otro mecanismo de interfaz) escribiendo la especificación del problema, con todos los hechos y asunciones, sobre el pizarrón, la cual es visible para todas las fuentes de conocimiento. Cada vez que una fuente de conocimiento reconoce oportunidades para aplicar su propio conocimiento especializado al estado actual de la solución del problema, ésta inscribe un registro de activación, en el cual describe las características del elemento solución que pretende crear. El mecanismo de control selecciona, entre todos los registros inscritos, aquel que represente la contribución más prometedora para el estado actual de solución del problema, por lo que la fuente de conocimiento asociada a este registro escribirá éste sobre el pizarrón. Este proceso continúa, iterativamente, hasta que el problema haya sido resuelto.

2.3. La arquitectura de REDSIEX

El sistema REDSIEX (González y Negrete, 1997) es un ambiente prototipo para el desarrollo de aplicaciones que requieren la solución cooperativa de problemas. REDSIEX (RED de Sistemas Expertos) es una red dinámica de nodos resolvedores de problemas, en la cual cada nodo es un sistema experto estructurado en una arquitectura de pizarrón. En REDSIEX, cada nodo puede trabajar independientemente en la solución de problemas, pero cuando el sistema se enfrenta a un problema global, entonces la cooperación entre los nodos es necesaria. Esto pudiera ocurrir cuando un único nodo no posee los recursos, el conocimiento o la suficiente experticia para resolver un problema de forma independiente. Por otra parte, diferentes nodos pudieran tener diferentes experticias para resolver diferentes partes de un problema, diferentes recursos disponibles para abordar la solución de un problema, y distintos conjuntos de conocimiento o puntos de vista acerca de un problema.

2.3.1. La solución cooperativa de problemas en REDSIEX

En REDSIEX la cooperación puede ser de dos tipos:

- cooperación intra-nodo
- cooperación inter-nodo

La cooperación intra-nodo se manifiesta a través del trabajo combinado de todas las fuentes de conocimiento locales del nodo. Este tipo de cooperación se caracteriza por ser un proceso oportunístico e incremental de creación y modificación de los elementos solución más prometedores sobre el pizarrón del dominio.

La cooperación inter-nodo establece que los nodos pueden intercambiar hipótesis y resultados parciales entre ellos. Este tipo de cooperación puede tomar lugar en tiempo real o en tiempo diferido. Lo anterior depende de la disponibilidad de tiempo y recursos de los nodos a los cuales

se les solicita la cooperación. La cooperación en tiempo real toma lugar cuando un nodo recibe una respuesta a su requerimiento de compromiso dentro del lapso de tiempo que este ha dedicado a los procesos de solución del problema, para el cual el compromiso fue requerido. La cooperación diferida toma lugar cuando el problema para el cual el compromiso fue requerido ha sido abandonado temporalmente, sin que haya sido recibida una respuesta para el mismo; este problema pudiera ser tomado nuevamente una vez que el nodo reciba una respuesta a su requerimiento solicitado.

2.3.2. Distribución de la información, procesamiento y control en REDSIEX

La distribución de la información, procesamiento y control en REDSIEX se ha llevado a cabo tomando en cuenta los siguientes criterios:

1. Distribución de la información.
 - Distribución del pizarrón: cada nodo tiene su propio pizarrón local.
 - Transmisión de hipótesis: un nodo transmite y recibe hipótesis hacia y desde un subconjunto local de nodos.
2. Distribución del procesamiento.
 - Distribución de las fuentes de conocimiento: cada nodo tiene su propio conjunto de fuentes de conocimiento.
 - Acceso al pizarrón por las fuentes de conocimiento: Las fuentes de conocimiento solo pueden acceder el pizarrón del nodo a las que éstas pertenecen. Este acceso puede ser del tipo lectura, modificación o creación de elementos solución.
3. Distribución del control.
 - Distribución de la activación de las fuentes de conocimiento: un evento sobre el pizarrón activa solo fuentes de conocimiento dentro del nodo local.
 - Distribución del mecanismo de control: cada nodo posee su propio mecanismo de control y lleva a cabo el control de su procesamiento sobre la base de su información local.

2.3.3. La arquitectura de los nodos en REDSIEX

Cada nodo en REDSIEX es un sistema pizarrón con elementos heredados de las arquitecturas de HEARSAY II (Engelmore, Morgan y Nii, 1988; Lesser y Erman, 1988 y Nii, 1989), la arquitectura del sistema DVMT (Lesser y Corkill, 1988) y la Arquitectura de Pizarrón para el Control (Hayes-Roth, 1985 y Hayes-Roth y Hewett, 1988); además de exhibir nuevos elementos definidos e incorporados por la propia arquitectura de REDSIEX. La figura 2.2 muestra los detalles de la arquitectura que caracteriza a los nodos en REDSIEX.

La estructura de cada nodo en la red comprende dos tipos de componentes fundamentales: componentes tipo memoria y componentes tipo función. Los componentes tipo memoria son estructuras de datos sobre las cuales los componentes tipo función ejecutan acciones de búsqueda, creación y modificación de elementos solución pertenecientes a dichas estructuras. Cada nodo en la red posee los siguientes componentes tipo memoria:

- Pizarrón del dominio
- Base de datos de control
- Lista de los registros de activación de las fuentes de conocimiento, la cual incluye:
 - Registros de activación de la fuentes de conocimiento de razonamiento (RAFC-R)
 - Registros de activación de la fuente de conocimiento receptora (RAFC-E)
 - Registros de activación de la fuente de conocimiento transmisora (RAFC-T)
 - Registros de activación de la fuente de conocimiento de procesamiento *backward* (RAFC-B)
- Árbol de búsqueda
- Buzón de mensajes de entrada

Los componentes tipo función ejecutan los procesos de búsqueda, creación y modificación de los elementos solución sobre las estructuras de datos y llevan a cabo los procesos relacionados con el control local del nodo.

- Fuentes de conocimiento de razonamiento
- Fuente de conocimiento de procesamiento *backward*
- Fuente de conocimiento receptora
- Fuente de conocimiento transmisora
- El planeador local
- Mecanismo de 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 unas con otras. 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 diferentes grados de detalle. El nivel de mayor abstracción es el nivel meta, mientras que el nivel de menor abstracción (o mayor cantidad de detalle) corresponde a estados iniciales del problema a solucionar. Cada hipótesis sobre el pizarrón del dominio está asociada con un identificador o número de hipótesis y con un valor de certidumbre (o grado de creencia de la hipótesis).

La base de datos de control

La base de datos de control (o base de meta-conocimiento) es una estructura de datos definida por la propia arquitectura de REDSIEX. Los elementos solución sobre la base de datos de control son registros cuyos atributos encierran información o decisiones relevantes al problema del control; elementos solución para la revisión de creencias y elementos solución para la investigación y comunicación de hipótesis. Para cada hipótesis sobre el pizarrón del dominio, un elemento solución sobre la base de datos de control es asociado con información correspondiente al valor de certidumbre (o creencia) de la hipótesis. Esta información es utilizada en el proceso de revisión de las creencias de las hipótesis. Los elementos solución para la comunicación e investigación de hipótesis constituyen parte del foco de control del nodo local, y son decisiones de control acerca de cuáles de los elementos solución sobre el pizarrón del dominio deben ser

investigados en un momento dado, cuáles hipótesis deben ser transmitidas como resultados parciales a otros nodos, o cuáles preguntas deben ser formuladas para pasarlas a otros nodos de la red.

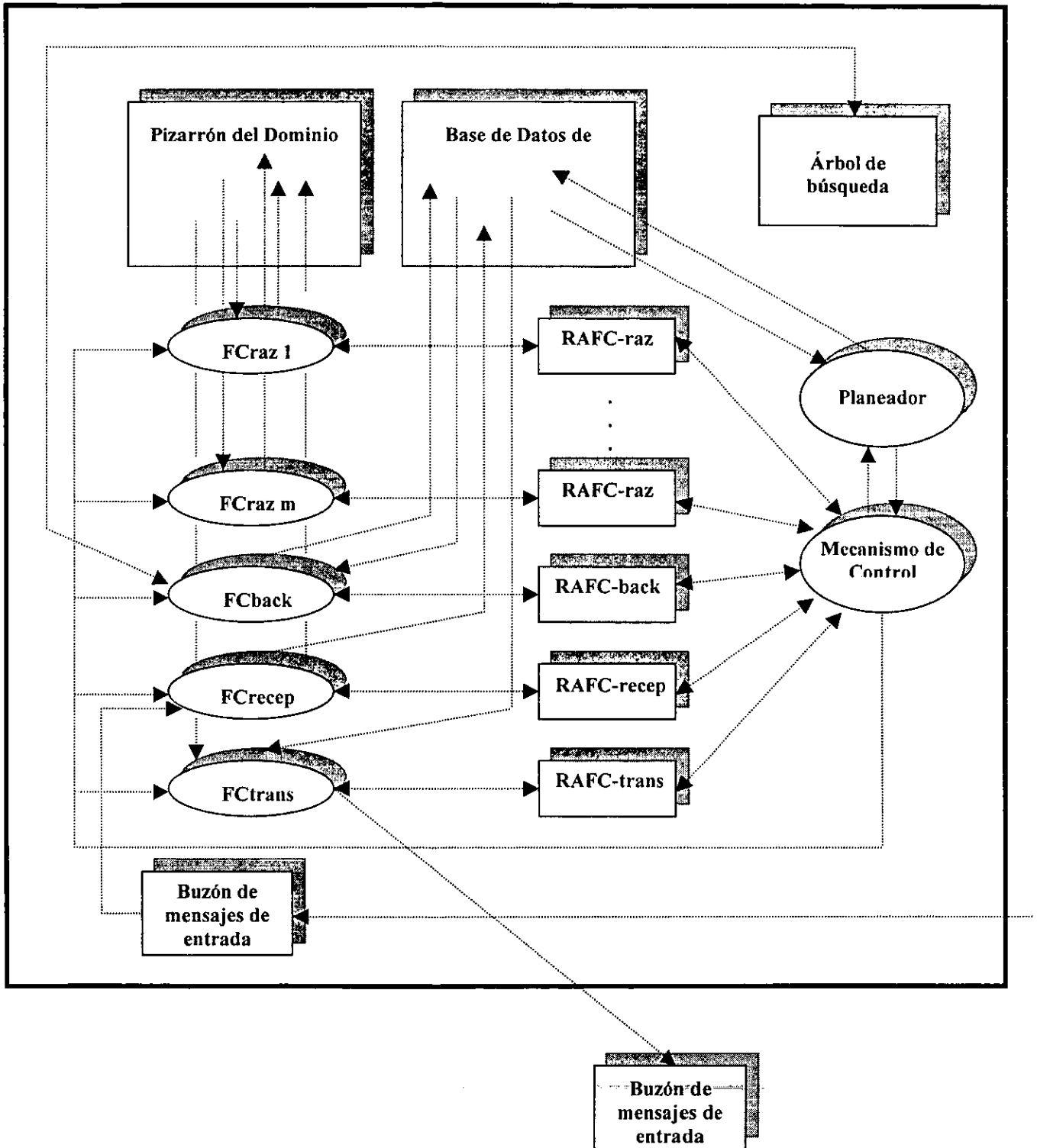


Figura 2.2. Arquitectura de un nodo en REDSIEX.

Los registros de activación de las fuentes de conocimiento

Cuando las condiciones o premisas de una fuente de conocimiento son satisfechas, ésta es activada, 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 de la fuente de conocimiento. Cuando un RAFC es seleccionado por el mecanismo de control, la acción de la fuente de conocimiento asociada con este RAFC es ejecutada, produciendo un nuevo evento sobre el pizarrón del dominio o sobre la base de datos de control, o actos de comunicación con otros nodos de la red, dependiendo de la naturaleza de la fuente de conocimiento que originó el RAFC. Para cada tipo de fuente de conocimiento, la arquitectura de REDSIEX define una estructura apropiada de RAFC.

El árbol de búsqueda

El árbol de búsqueda es una estructura de datos que permite la construcción de todas las posibles líneas de inferencia dirigidas a alguna hipótesis localizada en el nivel meta del pizarrón del dominio. Cada elemento del 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 las hipótesis que serán creadas o modificadas sobre el pizarrón una vez que la parte condición ha sido satisfecha. Una línea 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*, usa el árbol de búsqueda para la construcción de planes locales que indiquen al nodo cuáles acciones deben ser ejecutadas para arribar a determinadas metas.

El buzón de mensajes de entrada

Cada mensaje enviado al nodo por alguno de los otros nodos de la red es recibido en un buzón de mensajes de entrada, donde es creado un registro que describe los atributos del mensaje. Estos atributos son: el evento asociado al mensaje, el valor de creencia asociado al mensaje, el colaborador del mensaje y el tipo de mensaje.

Las fuentes de conocimiento de razonamiento

Una fuente de conocimiento de razonamiento (FCraz) contiene conocimiento específico acerca del dominio del sistema. Este conocimiento es proporcionado por las reglas de inferencia que componen cada FCraz. Una FCraz está compuesta por una parte identificación y una parte inferencial. La parte identificación describe los 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. La condición de cada regla de inferencia describe la configuración de elementos solución sobre el pizarrón que es necesaria para que la FCraz pueda contribuir a los procesos de solución del problema. La forma en la cual una FCraz contribuye a la solución de un problema es especificada en la acción de cada regla de inferencia. La acción de una regla de inferencia está ligada a la creación y/o modificación de elementos solución sobre el pizarrón. La actividad de una FCraz es direccionada por los eventos. Esto significa que cada cambio que es ejecutado sobre el pizarrón constituye un evento que, en

presencia de otra información específica sobre éste, puede activar una o más reglas de inferencia dentro de una FCraz. Cada activación de la FCraz produce un único RAFC-R.

La fuente de conocimiento de procesamiento bakward

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 *forward* dirigido a la creación de elementos solución sobre el pizarrón del dominio, para el problema que necesita ser resuelto. Al igual que las FCraz, 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 la FCback es determinada por la 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 *backward* orientado a la construcción de un árbol de secuencias para generar elementos solución. Las secuencias son iniciadas con el estado meta en la raíz del árbol. El próximo nivel del árbol es generado encontrando todas las reglas cuyas consecuencias o acciones aparezcan con el nodo raíz. Si fuera posible aplicar sólo estas reglas, entonces el estado meta deseado podría ser generado. Los antecedentes o condiciones de estas reglas son usados 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 cuyas acciones correspondan con éste, entonces las condiciones de las correspondientes reglas son usadas para generar los nuevos nodos en el siguiente nivel del árbol. El proceso continúa hasta que una configuración de elementos solución sólo correspondiente a estados iniciales haya sido generada.

Para construir el árbol de secuencias, la FCback usa la información registrada sobre el árbol de búsqueda del nodo, seleccionando de entre todas las vías de inferencia que se dirigen al nivel meta del pizarrón, sólo aquellas que generan los elementos solución que son necesarios para alcanzar la hipótesis meta deseada. La FCback tiene un formato condición-acción. La condición es satisfecha cuando un registro del tipo "investigar hipótesis" es creado sobre la base de datos de control.

La fuente de conocimiento receptora

La fuente de conocimiento receptora (FCrecep) decide cuáles mensajes, de los que hasta el momento han sido recibidos en el buzón de mensajes de entrada (BME), son de interés para la solución del problema a mano, y pone éstos a consideración del mecanismo de control. La FCrecep posee un formato condición. 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 FCrecep es satisfecha, ésta ejecuta dos acciones. La primera acción consiste en la inscripción de su correspondiente RAFC-E. La segunda acción es llevada a cabo cuando el mecanismo de control selecciona este RAFC-E para su ejecución. La ejecución de un RAFC-E depende del tipo de mensaje que éste contiene. Si el mensaje es del tipo "oferta" o "respuesta a un compromiso", entonces la ejecución del RAFC-E está asociada con la creación o modificación de elementos solución sobre el pizarrón del dominio. Si el mensaje es del tipo "requerimiento de compromiso",

entonces la ejecución del RAFC-E está asociada con la creación de un registro del tipo “investigar hipótesis” sobre la base de datos de control.

La fuente de conocimiento transmisora

La fuente de conocimiento transmisora (FCtrans) transmite mensajes del tipo “oferta”, “respuesta a compromiso” o “requerimiento de compromiso” a otros nodos de la red. Los mensajes del tipo “oferta” o “respuesta a compromiso” son hipótesis seleccionadas del pizarrón del dominio, para cada una de las cuales la FCtrans encuentra sobre la base de datos de control un registro del tipo “transmitir hipótesis”. Los mensajes del tipo “requerimiento de compromiso” están asociados con registros del tipo “preguntar hipótesis”, los cuales han sido previamente creados sobre la base de datos de control.

Al igual que la FCrecep, la FCtrans también posee un formato condición-acción. La condición de la FCtrans es satisfecha cuando un registro del tipo “transmitir hipótesis” o “preguntar hipótesis” es encontrado sobre la base de datos de control. La primera acción ejecutada por la FCtrans consiste en la creación de su correspondiente RAFC-T. Una vez que este RAFC-T es seleccionado por el mecanismo de control, la FCtrans ejecuta su acción final, la cual consiste en transmitir el mensaje asociado con el RAFC-T a algún nodo o subconjunto de nodos en la red.

El planeador local

La función del planeador local del nodo es la generación de planes locales dirigidos a alcanzar metas particulares. Un plan es una secuencia de actividades que deben ser llevadas a cabo por ciertas fuentes de conocimiento del nodo, con el objetivo de crear los elementos solución sobre el pizarrón que son necesarios para permitir al nodo alcanzar la meta deseada.

Para iniciar la creación de un plan local, el planeador usa a FCback para crear el árbol de secuencias para generar los elementos solución. Los nodos localizados en los diferentes niveles del árbol de búsqueda corresponden a elementos solución particulares, algunos de los cuales pudieran ya existir sobre el pizarrón del dominio, mientras que otros han de ser creados. La creación de cada elemento solución es una actividad que requiere el trabajo de una o varias fuentes de conocimiento específicas. Una vez que el árbol de secuencias ha sido creado, el planeador evalúa criterios heurísticos para decidir en qué orden ejecutar estas actividades. De esta forma, la generación de un plan local es concluida.

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 orientados a la búsqueda de una solución al problema actual se han detenido debido a la insuficiente información registrada sobre el pizarrón del dominio, o (2) sobre la base de datos de control un registro del tipo “investigar hipótesis” ha sido creado. Cuando la situación (1) ha tenido lugar, el planeador, haciendo uso de sus criterios heurísticos, determina cuál meta debe ser investigada y crea 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 en la creación de un plan apropiado, el cual concibe una relación de actividades y ejecutores para estas actividades.

El mecanismo de control

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 registrados hasta el actual ciclo de ejecución, sobre la base del cálculo de una medida de la importancia para cada RAFC, la cual llamaremos oferta. La fuente de conocimiento asociada con el RAFC de mayor oferta será seleccionada para ejecutar su acción final en este ciclo de ejecución.

El sistema de control está compuesto por tres fuentes de conocimiento de control: FCcal, FCsel y FCinv, las cuales iteran en un ciclo de tres pasos para ejecutar, respectivamente, las siguientes actividades:

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

Los RAFCs de un mismo tipo (asociados a un mismo tipo de fuente de conocimiento) se agrupan formando bloques de REACs dentro de una lista de registros, denotada L-REAC. Pudiera ocurrir que existan REACs compitiendo para ejecutar sus acciones finales en más de uno de estos bloques. Dada esta situación, una jerarquía ha sido establecida para seleccionar el bloque de RAFCs dentro del cual el registro con mayor oferta debe ser seleccionado. La jerarquía entre estos bloques de registros ha sido establecida de la siguiente forma:

- RAFC-R: prioridad 1 (mayor jerarquía)
- RAFC-E: prioridad 2
- RAFC-T: prioridad 3
- RAFC-B: prioridad 4 (menor jerarquía)

Consecuentemente, las fuentes de conocimiento que han registrado RAFCs de menor prioridad sólo podrán ejecutar sus acciones finales en ausencia de REACs registrados de mayor prioridad. La jerarquía establecida contribuye a maximizar la información interna contenida en cada nodo de la red (configuración de elementos solución sobre el pizarrón del dominio), antes que éste pueda establecer ligas de colaboración. Este criterio permite llevar a cabo una cooperación más coherente y robusta, evitando el envío de mensajes con información insegura o incompleta, así como la formulación de preguntas innecesarias.

2.3.4. Características de la cooperación en REDSIEX

La cooperación en REDSIEX tiene las siguientes características:

- Cada nodo en la red es un sistema pizarrón con sus propias fuentes de conocimiento, pizarrón y elementos de control.
- Cada nodo es un especialista en un diferente tipo de problema o subproblema.
- Cada nodo posee un planeador que desarrolla secuencias de actividades para la solución del problema basadas en la situación actual.

- Los nodos pueden transferir problemas o subproblemas no resueltos a otros nodos, los cuales poseen mayores habilidades en la solución.
- La cooperación es intra-nodo e inter-nodos.
- La cooperación inter-nodos es emergente y facultativa.
- La cooperación inter-nodos toma lugar en tiempo real o de forma diferida.
- Un proceso de revisión de creencias garantiza la consistencia de las hipótesis creadas sobre el pizarrón.

Finalmente, la arquitectura desarrollada en REDSIEX ha permitido probar una nueva aproximación para el trabajo cooperativo en una red de nodos resolventes de problemas. Esta aproximación a la solución cooperativa de problemas distribuidos está basada sobre las características y criterios de desempeño de una clase particular de aplicaciones distribuidas; la clase de los sistemas pizarrón distribuidos.

2.4. La arquitectura de ECN-MAES

Los sistemas expertos multi-agentes (MAES) implican mecanismos de interacción cooperativa, que permitan a sus múltiples agentes trabajar conjuntamente sobre la solución de un problema en común. Cuando el carácter de los agentes es heterogéneo, los agentes podrían tener diferentes tareas, las cuales pudieran ser conceptualizadas como “comportamientos”, en el sentido de que éstas tienen un mundo interno y/o externo que “percibir” y actúan autónomamente sobre estos mundos.

La red con control emergente de nodos MAES (ECN- MAES) (Negrete y González, 1998) es un sistema prototipo para el desarrollo de aplicaciones, el cual posee muchas características comunes con el sistema REDSIEX, presentado en el epígrafe anterior. ECN-MAES es una red dinámica de nodos resolventes de problemas, en la cual cada nodo es un sistema experto con arquitectura de pizarrón y control emergente *bottom-up*. Al igual que en la arquitectura de REDSIEX, en ECN-MAES cada nodo puede trabajar de forma independiente en la solución de un problema, pero cuando éstos se enfrentan a un problema global, entonces la cooperación es necesaria.

En ECN-MAES la cooperación puede ser de dos tipos: inter-agentes e inter-nodos. La cooperación inter-agentes se manifiesta a través del trabajo combinado de todos los comportamientos locales del nodo. En el tipo de cooperación inter-nodos, los nodos pueden intercambiar información entre sí.

2.4.1. La arquitectura de los nodos en ECN-MAES

Cada nodo en ECN-MAES es arquitectónicamente un sistema heterárquico. La figura 2.3 muestra los detalles de la arquitectura que caracteriza los nodos en ECN-MAES. La arquitectura de cada nodo en la red define dos tipos de componentes fundamentales: componentes tipo memoria y componentes tipo comportamiento.

Componentes tipo memoria

Los componentes tipo memoria son estructuras de datos sobre las cuales los comportamientos ejecutan acciones de búsqueda, creación y modificación de elementos pertenecientes a estas estructuras. Cada nodo en la red posee los siguientes componentes tipo memoria:

- Memoria de trabajo del dominio;
- Meta-memoria de trabajo;
- Lista de Registros de los Estados de Actividad de los Comportamientos (LEAC), la cual incluye los registros de los estados de actividad del comportamiento de razonamiento *forward*, los registros de los estados de actividad del comportamiento de razonamiento *backward*, y los registros de los estados de actividad de los comportamientos de comunicación.

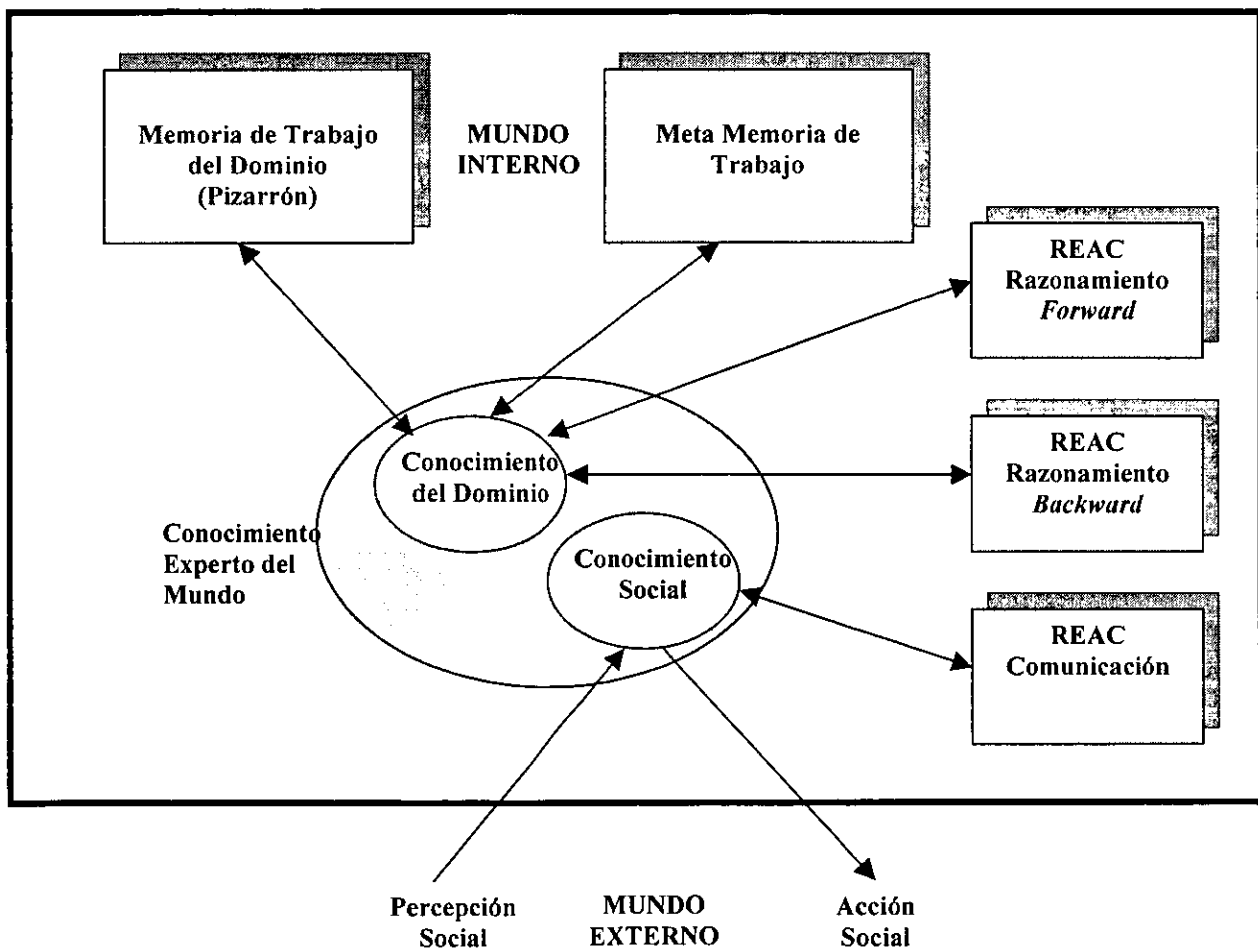


Figura 2.3. Arquitectura de un nodo ECN-MAES.

La memoria de trabajo del dominio (pizarrón del dominio) es una estructura de datos compartida a través de la cual los comportamientos de razonamiento *forward* se comunican entre sí. La memoria de trabajo del dominio organiza los elementos solución en diferentes niveles de abstracción. Estos niveles de abstracción representan la solución al problema en diferentes grados de detalles.

La meta-memoria de trabajo es una estructura de datos definida por la arquitectura de ECN-MAES. Los elementos solución en la meta- memoria de trabajo son registros cuyos atributos encierran información o decisiones relevantes para el problema de la selección de acciones. Sobre la meta-memoria de trabajo operan el comportamiento de razonamiento *backward* y los comportamientos de comunicación. Estos comportamientos se comunican entre sí a través de la creación de elementos solución sobre la meta-memoria de trabajo.

Cada comportamiento tiene un conjunto de condiciones, las cuales tienen que ser observadas para que el comportamiento sea “ejecutable” (Maes, 1991). Cuando las condiciones o premisas de un comportamiento son satisfechas, el comportamiento es activado, creando un registro del estado de actividad del comportamiento (REAC). Un REAC es una estructura de datos que describe las características de la acción activada por el comportamiento. Cuando un REAC es seleccionado por el mecanismo de selección de acciones (discutido más adelante), el comportamiento asociado con el REAC es ejecutado y comúnmente produce nuevos eventos sobre la memoria de trabajo del dominio o sobre la meta-memoria de trabajo.

Componentes tipo comportamiento

Los componentes tipo comportamiento del nodo ejecutan los procesos de razonamiento y comunicación. Cada nodo en la red posee los siguientes componentes tipo comportamiento:

- Comportamiento de razonamiento *forward*;
- Comportamiento de razonamiento *backward*.
- Comportamientos de comunicación, los cuales incluyen el comportamiento receptor y el comportamiento transmisor.

La acción ejecutada por el comportamiento de razonamiento *forward* está ligada a la creación y/o modificación de elementos solución sobre la memoria de trabajo del dominio. Al igual que las fuentes de conocimiento en REDSIEX, un comportamiento de razonamiento *forward* tiene una parte identificación y una parte inferencial, y éste se estructura a partir de un conjunto de reglas de inferencia. La condición de cada regla de inferencia describe la configuración de elementos solución sobre la memoria de trabajo del dominio – precondiciones en el mecanismo de Maes (Maes, 1991) – que es necesaria para que el comportamiento contribuya a los procesos de solución del problema.

Al igual que el comportamiento de razonamiento *forward*, el comportamiento de razonamiento *backward* también contribuye a la creación de elementos solución sobre la memoria de trabajo del dominio, aunque a diferencia del primero, la actividad de éste es determinada por las metas y no por los eventos que ocurren sobre la memoria de trabajo del dominio. El comportamiento de razonamiento *backward* tiene un formato condición-acción; la condición es satisfecha cuando un

registro del tipo “investigar hipótesis” es creado sobre la meta-memoria de trabajo. La actividad del comportamiento de procesamiento *backward* constituye un proceso de encadenamiento de reglas encaminado a la construcción de un árbol de secuencias para generar los elementos solución. El proceso para construir este árbol de secuencias es similar al ejecutado por la fuente de conocimiento de razonamiento *backward* en REDSIEX.

Cada nodo en la red recibe mensajes desde otros nodos y selecciona información relevante registrada en la memoria de trabajo del dominio para enviar a otros nodos. Para llevar a cabo estas actividades de comunicación, la arquitectura de ECN-MAES define dos comportamientos de comunicación para cada nodo de la red: el comportamiento receptor y el comportamiento transmisor.

El comportamiento receptor decide cuáles mensajes (de los que hasta el momento han sido recibidos por el nodo) son de interés en la solución del problema actual y pone éstos a consideración del mecanismo de selección de acciones. El comportamiento receptor posee un formato condición-acción. La condición es satisfecha cuando un mensaje de interés para la solución del problema actual es recibido. Cada vez que su condición es satisfecha, el comportamiento receptor ejecuta dos acciones. La primera acción consiste en la inscripción de su correspondiente REAC; mientras que la segunda acción es llevada a cabo cuando dicho REAC es seleccionado por el mecanismo de selección de comportamientos.

El comportamiento transmisor transmite mensajes a otros nodos de la red. Los mensajes transmitidos pueden ser del tipo “oferta de compromiso”, “respuesta a compromiso” o “requerimiento de compromiso”. Los mensajes del tipo “oferta” o “respuesta” son elementos solución seleccionados de la memoria de trabajo del dominio, para cada uno de los cuales el comportamiento transmisor encuentra sobre la meta-memoria de trabajo un registro del tipo “transmitir hipótesis”. Los mensajes del tipo “requerimiento” son asociados con registros del tipo “preguntar hipótesis”, los cuales han sido previamente creados sobre la meta-memoria de trabajo. Al igual que el comportamiento receptor, el comportamiento transmisor también posee un formato condición-acción. La condición es satisfecha cuando algún registro del tipo “transmitir hipótesis” o “preguntar hipótesis” es encontrado sobre la meta-memoria de trabajo. Cuando las condiciones del comportamiento transmisor son satisfechas, éste es activado creando inicialmente su correspondiente REAC. Una vez que dicho REAC es seleccionado, el comportamiento transmisor ejecuta su acción final, la cual consiste en la transmisión del mensaje asociado con el registro a un nodo particular o subconjunto de nodos en la red.

Comportamiento	Precondición	Acción final	Meta
Razonamiento <i>forward</i>	Memoria de Trabajo del Dominio (MTD): evento	MTD: creación y/o modificación	Permanente; búsqueda de datos
Razonamiento <i>backward</i>	No nuevos eventos sobre MTD	Meta Memoria de Trabajo (MMT): creación	Transiente; prueba de hipótesis
Transmisor	Evento de interés global o nuevos eventos sobre MTD	Nodo	Transiente; solución particular o global
Receptor	Requerimiento del nodo	MTD: creación	ninguna

Tabla 2.1.

La tabla 2.1 muestra las precondiciones, acción final y meta para cada uno de los comportamientos antes descritos.

2.4.2. El mecanismo de selección de comportamientos

Cuando un comportamiento se convierte en “ejecutable” éste crea un REAC, el cual describe las características asociadas con la acción que el comportamiento intenta ejecutar. En cada ciclo de ejecución, sólo un comportamiento es ejecutado. Para determinar el orden en el cual los comportamientos ejecutarán sus acciones finales, un proceso competitivo es iniciado entre los REACs inscritos, basado sobre el nivel de activación de cada uno. El comportamiento asociado con el REAC de mayor valor de activación será seleccionado para ejecutar su acción final en este ciclo de ejecución. en esta competencia no es utilizado un valor umbral.

El algoritmo para la selección de comportamientos ejecuta un ciclo, en el cual en cada momento las siguientes actividades toman lugar sobre todos los REACs:

1. Actualización del nivel de activación para cada REAC.
2. Selección del REAC con el mayor nivel de activación.
3. Ejecución del comportamiento asociado con el REAC seleccionado.

De esta forma, las tres actividades iteran en un ciclo de tres pasos, determinando en cada ciclo de ejecución el comportamiento que debe ser ejecutado y llevando a cabo su ejecución.

2.4.3. Propiedades de la selección de comportamientos en un nodo ECN-MAES

La selección de comportamientos que ocurre a nivel de un nodo ECN-MAES se caracteriza por las siguientes propiedades, algunas de las cuales son descritas utilizando la terminología introducida por Maes en (Maes, 1991):

- Los comportamientos de un nodo pueden ser vistos como agentes autónomos con características tales como: tipo de comportamiento, precondiciones y nivel de activación.
- El código ejecutable no está encapsulado dentro del comportamiento, y no están presentes las listas ADD y DELETE. La meta-memoria de trabajo juega este rol.
- Algo no deseable, presente en ACN-MAES, es que el mundo para algunos de los comportamientos es interno y declarativo: MTD y MMT.
- El tipo de razonamiento *forward* oportunístico es heredado de la arquitectura de pizarrón.
- Para la actual etapa del prototipo, la cuestión del razonamiento *backward* oportunístico es reducido a una condición por defecto (ver tabla 2.1).

- Desde el punto de vista de sus metas implícitas, los comportamientos de razonamiento *forward* y de comunicación son principalmente “apetitivos”, mientras que el razonamiento *backward* es “consumatorio” (Maes, 1991).
- El control interno de cada nodo en la red es impredecible para el usuario, y en este sentido emergente y *bottom-up*.

2.4.4. Características de la cooperación inter-nodos en ECN-MAES

Algunas características interesantes de la cooperación inter-nodos en ECN-MAES son las siguientes:

- La cooperación, en este nivel, es también emergente. Es decir, ésta es una propiedad no programada que surge dinámicamente (por ejemplo, una consulta no esperada pero aceptada de un nodo inactivo) en el proceso de interconsulta entre los nodos.
- La cooperación es facultativa. En ECN-MAES, la cooperación es una facultad que posee cada nodo en la red, y ésta no constituye una necesidad permanente. Esto significa que un nodo puede resolver un problema completo, si éste posee la capacidad y recursos necesarios para arribar a la solución.
- La cooperación puede ser ejecutada en tiempo real y en tiempo diferido. En un momento dado, cada nodo pudiera encontrarse trabajando en la solución de un problema diferente a los que están siendo resueltos por los otros nodos en la red. No obstante, aún bajo estas condiciones es posible que los nodos puedan establecer ligas cooperativas entre ellos mismos.

2.5. Desde REDSIEX y ECN-MAES a RCI

El problema del control de un agente autónomo envuelve cierta complejidad, la cual no es comúnmente encontrada en otras áreas de la investigación robótica. Algunos de los factores que determinan esta complejidad son: la naturaleza dinámica e impredecible del mundo externo del agente, la cual requiere de un sistema de control en tiempo real; la toma de decisiones en tiempo real, la cual debe ser basada sobre información sensada continuamente tanto del medio externo como del medio interno; y la continua exploración del agente, la cual hace que éste se enfrente a una amplia variedad de configuraciones externas posibles, por lo que se requiere de una estructura de control general y adaptable.

Nosotros postulamos que la arquitectura de pizarrón debe constituir un escenario ideal para la modelación, integración y control de los factores antes considerados, ya que este tipo de arquitectura posee la capacidad de poder coordinar e integrar varias actividades en tiempo real, proporciona una gran flexibilidad para la incorporación de nueva funcionalidad, además de manejar la selección de acciones como principal mecanismo para la aplicación del conocimiento

más adecuado a la solución del problema que se intenta resolver. De esta forma, nuestra intención es modelar y estructurar el control de la selección de acciones de un agente autónomo sobre una arquitectura de pizarrón.

En el siguiente capítulo se presenta el mecanismo de selección de acciones desarrollado por nosotros, el cual hemos nombrado Red de Conductas Internas (RCI). Como se verá en este capítulo, la arquitectura de la RCI hereda, redefine y extiende las principales características estructurales y funcionales de las arquitecturas de REDSIEX y ECN-MAES. De esta forma, de arquitectura base para la modelación de sistemas basados en el conocimiento, la arquitectura de pizarrón ahora se refina y extiende para proporcionar un escenario idóneo para la modelación de la selección de acciones en sistemas basados en el comportamiento.

CAPÍTULO 3

LA RED DE CONDUCTAS INTERNAS

El objetivo de este capítulo es presentar y discutir el modelo de selección de acciones que nosotros hemos propuesto. En la idea subyacente a nuestra propuesta han convergido dos direcciones básicas: (1) identificar todas aquellas propiedades no satisfechas o explicadas por los MSAs revisados en el capítulo I, las cuales consideramos de gran importancia para la selección de acciones, y lograr que nuestro modelo sea capaz de exhibirlas y explicarlas; y (2) tomar en cuenta varios de los principios exhibidos por el comportamiento animal que son requeridos por un agente autónomo y mapearlos directamente al modelo de selección de acciones propuesto.

Tomando en cuenta las anteriores ideas, hemos desarrollado un mecanismo de selección de acciones basándonos en una arquitectura de pizarrón distribuida. Al mecanismo de selección de acciones aquí propuesto lo hemos nombrado red de conductas internas implementada con nodos-pizarrón (RCI). Un nodo-pizarrón es un sistema pizarrón con elementos definidos en las arquitecturas de REDSIEX (González y Negrete, 1997) y ECN-MAES (Negrete y González, 1998) (ver anexos A y B). La arquitectura de la red de conductas internas puede ser apreciada en la figura 3.1.

Como se puede apreciar en la figura 3.1, la estructura actual que exhibe la red de conductas internas define dos nodos-pizarrón: el nodo cognitivo y el nodo motivacional. El nodo cognitivo es el que interactúa directamente con el medio externo, a través de sus sensores y actuadores. Es a nivel de este nodo donde podría ocurrir la adquisición de abstracciones que permitan “modelar” el medio externo de la entidad y usar los códigos así adquiridos en la selección de acciones mediatizada por las motivaciones (aprendizaje a nivel cognitivo). El nodo motivacional debe su nombre a los diferentes procesos que toman lugar a nivel de éste, en los cuales juegan un rol fundamental las motivaciones de la entidad. Entre estos procesos se encuentran: la representación de señales internas (motivaciones) en el nivel Percepciones Internas del pizarrón, la competencia a nivel motivacional de conductas externas incompatibles y la adquisición de significado motivacional por parte de señales que carecían inicialmente de significado para la entidad y su utilización en la selección de acciones.

Las tareas requeridas para el control de la selección de acciones sólo pueden ser satisfechas sobre la base del trabajo cooperativo de ambos nodos (cooperación ésta heredada de REDSIEX y ECN-MAES).

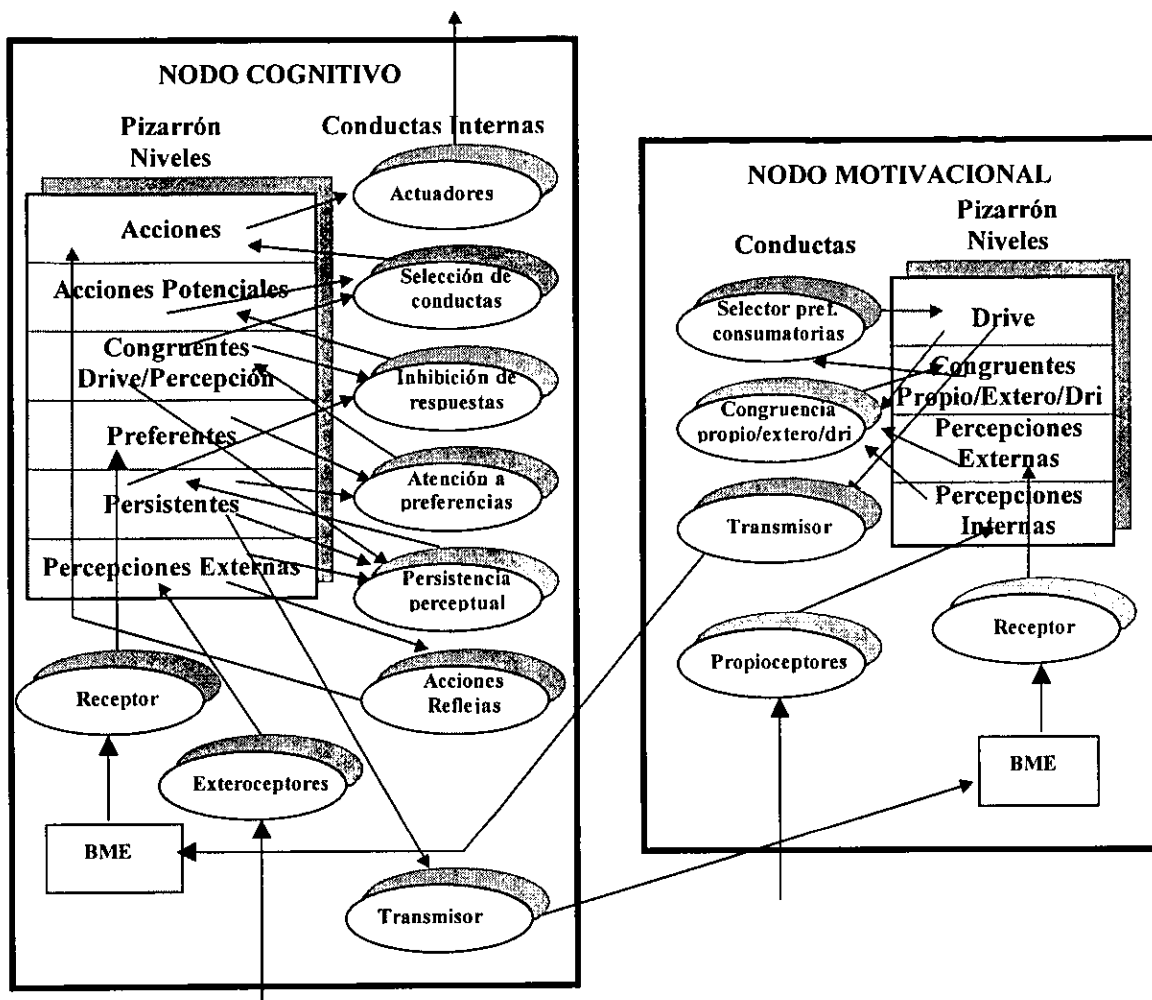


Figura 3.1. Estructura de la red de nodos pizarrón. En esta representación se han omitido los registros de estado de actividad de las conductas y las fuentes originadoras y receptoras de flujos de señales.

3.1. Los componentes de un nodo pizarrón

La estructura de un nodo-pizarrón se define a partir de cinco elementos básicos: el pizarrón, las conductas internas, los registros de los estados de actividad de las conductas internas, los mecanismos de interfaz/comunicación, y el mecanismo de competencia.

3.1.1. El pizarrón

El pizarrón es una estructura de datos compartida sobre la cual ejecutan sus acciones finales y se comunican entre si las conductas internas. Sobre el pizarrón también operan los mecanismos de

interfaz/comunicación. Las conductas internas producen cambios sobre el pizarrón, los cuales conducen incrementalmente a la formación de una solución para el problema que se intenta resolver: el problema de la selección de acciones. La información contenida en el pizarrón se conoce con el nombre de elementos solución. Estos elementos solución pueden ser datos de entrada, soluciones parciales, soluciones alternativas, o soluciones finales para el problema que se intenta resolver. El pizarrón organiza los elementos solución en diferentes niveles de abstracción, los cuales representan la solución del problema en diferente cantidad de detalles, o en diferente contexto. A cada elemento solución creado sobre el pizarrón se le asocia un identificador, comúnmente un índice que permite acceder dicho elemento, y un valor de certidumbre, el cual puede ser interpretado como la “intensidad” o “fuerza” del elemento solución. Como veremos más adelante, estos elementos solución no son más que señales provenientes de los medios externo e interno y nuevas señales creadas por las conductas internas que operan sobre el pizarrón.

3.1.2. Las conductas internas

El término “conducta interna” es utilizado para describir las acciones intrínsecas del procesamiento de información que ocurren a nivel del nodo-pizarrón, o sea, dentro de la entidad. (En REDSIEX, estas conductas internas vienen dadas por las fuentes de conocimiento de razonamiento (FCraz), mientras que en ECN-MAES éstas corresponden a los comportamientos de razonamiento). A diferencia de las conductas externas, cuyas acciones van dirigidas hacia el medio externo de la entidad, las conductas internas dirigen sus acciones hacia aquella parte del medio interno de la entidad representada por el pizarrón. Desde el punto de vista estructural, una conducta interna se define como un paquete de mecanismos más simples, comúnmente, todos del mismo tipo, y estructurados como reglas de producción (Si <condición> Entonces <acción>). A estos mecanismos simples les llamaremos conductas elementales (o simplemente reglas de producción). Conductas elementales de diferentes tipos se organizan formando “columnas conductuales” que atraviesan verticalmente varios niveles del pizarrón. El índice i (identificador de la conducta elemental) a través de varios niveles del pizarrón se encarga de definir esta columna. La estructura de una conducta elemental contiene tres elementos básicos: una lista de parámetros, una parte condición, y una parte acción.

La lista de parámetros especifica cuáles son los elementos solución condición, los elementos solución acción, y las fuerzas de apareo (elemento éste discutido más adelante) relacionados con la conducta elemental.

La condición de una conducta elemental describe la configuración de elementos solución sobre el pizarrón que es necesaria para que ésta contribuya a los procesos de solución del problema (es decir, cuál es la relación o composición que debe establecerse entre estos elementos). La forma en la cual una conducta elemental contribuye a la solución del problema es especificada en su acción, la cual está ligada a la creación o modificación de elementos solución sobre el pizarrón.

La fuerza de apareo es un vector $F_a = (F_{a_{i1}}, F_{a_{i2}}, \dots, F_{a_{in}})$ de n componentes reales, donde cada una de estas componentes representa la eficacia con la cual la conducta elemental i puede satisfacer la condición j ($j = 1, 2, \dots, n$). Dependiendo de la naturaleza de la conducta elemental, las componentes del vector F_a pueden ser de valor fijo o valor modificable en el tiempo, sin la restricción de que todas deban ser del mismo tipo. La existencia de fuerzas de apareo

modificables permiten la incorporación de procesos de aprendizaje para mejorar la selección de acciones, haciéndola más adaptativa. El vector Fa de fuerzas de apareo es para una conducta elemental lo mismo que es para una neurona artificial el vector de pesos.

3.1.3. Los registros de los estados de actividad

Cuando las condiciones de una conducta elemental son satisfechas, ésta es activada creando un registro de su estado de actividad (REAC). Un REAC es una estructura de datos que describe las características de la acción activada por la conducta. Cuando un REAC es seleccionado por el mecanismo de competencia (discutido más adelante), la acción de la conducta elemental asociada con el REAC es ejecutada, por lo que un nuevo elemento solución será creado sobre el pizarrón, o el valor de certidumbre de un elemento solución ya existente será modificado. No todos los REACs poseen la misma estructura de datos, para cada tipo de conducta interna es definida una estructura de datos apropiada para sus REACs.

3.1.4. Los mecanismos de interfaz/comunicación

Sobre el pizarrón también operan los mecanismos de interfaz/comunicación, los cuales también están estructurados por conductas elementales. Tres tipos de mecanismos de interfaz han sido definidos: (1) los *exteroceptores*, los cuales establecen la interfaz entre el sistema perceptual y el nodo cognitivo; (2) los *propioceptores*, para establecer la interfaz entre el medio interno y el nodo motivacional; y (3) los *actuadores*, los cuales definen la interfaz entre el nodo cognitivo y el sistema motor. Los mecanismos de comunicación son los encargados de ejecutar la recepción y transmisión de señales desde y hacia otros nodos de la red (en REDSIEX estas son las fuentes de conocimiento receptora y transmisora, mientras que en ECN-MAES estos son los comportamientos que poseen conocimiento social).

3.1.5. Los mecanismos de competencia

Una lista de REACs (L-REAC) es el conjunto definido por todos los REACs asociados a conductas elementales de un mismo tipo. A nivel de cada L-REAC se establece un proceso competitivo entre los REACs integrantes para decidir qué conducta(s) elemental(es) resultará(n) ganadora(s). Sólo las conductas ganadoras podrán ejecutar su acción final sobre el pizarrón. El proceso competitivo puede ser de dos tipos: como resultado de la competencia más de una conducta elemental puede resultar ganadora, o una sola conducta elemental será proclamada ganadora (*the winner takes all*).

Un principio novedoso y exclusivo del funcionamiento de los nodos en la red, es que las condiciones de las conductas elementales pueden resultar modificables en el tiempo. Un elemento que no ha sido necesario considerar en la estructura de los nodos es el metapizarrón (base de datos de control en REDSIEX o memoria de meta trabajo en ECN-MAES), estructura de datos asociada a los procesos de revisión de creencias e investigación de hipótesis.

En los próximos epígrafes se discuten las principales características estructurales y funcionales de las conductas internas y de los mecanismos de interfaz/comunicación de los nodos cognitivo y motivacional, y el rol que juegan éstos en el control de la selección de acciones; así como la naturaleza de los procesos competitivos a nivel de listas L-REACs. Para conocer otros detalles

sobre los principios que rigen el funcionamiento de este tipo de red, el lector puede referirse a (González y Negrete, 1997) y (Negrete y González, 1998) en los apéndices A y B.

3.2. El nodo cognitivo

La estructura completa del nodo cognitivo, considerando los registros de estado de actividad de las conductas internas y todos los flujos de señales entrantes y salientes, es mostrada en la figura 3.2. Para el nodo cognitivo han sido definidas las siguientes conductas internas: *persistencia perceptual*, *atención a preferencias*, *inhibición de respuestas reflejas*, y *selección de conductas externas*; los mecanismos de interfaz *exteroceptores* y *actuadores*; y los mecanismos de comunicación *receptor* y *transmisor*.

El nodo cognitivo recibe señales desde el sistema perceptual a través de los *exteroceptores* y del nodo motivacional por medio del mecanismo *receptor*, y envía señales al nodo motivacional a través del mecanismo *transmisor* y al sistema motor mediante los *actuadores*. El rol del nodo cognitivo abarca los procesos de representación de las señales perceptuales, integración de señales perceptuales externas e internas, inhibición de respuestas reflejas (inhibición de señales que viajan por vías directas desde el nivel Persistentes Perceptuales al nivel Acciones del pizarrón), y selección de la conducta externa que más se ajuste al estado interno actual.

3.2.1. El pizarrón del dominio

Como se puede apreciar en la figura 3.2, el pizarrón del dominio del nodo cognitivo organiza los elementos solución en seis niveles de abstracción:

- Primer nivel: Percepciones Externas
- Segundo nivel: Persistentes Perceptuales
- Tercer nivel: Preferentes Consumatorios
- Cuarto nivel: Congruentes Drive/Percepción
- Quinto nivel: Acciones Potenciales
- Sexto nivel: Acciones

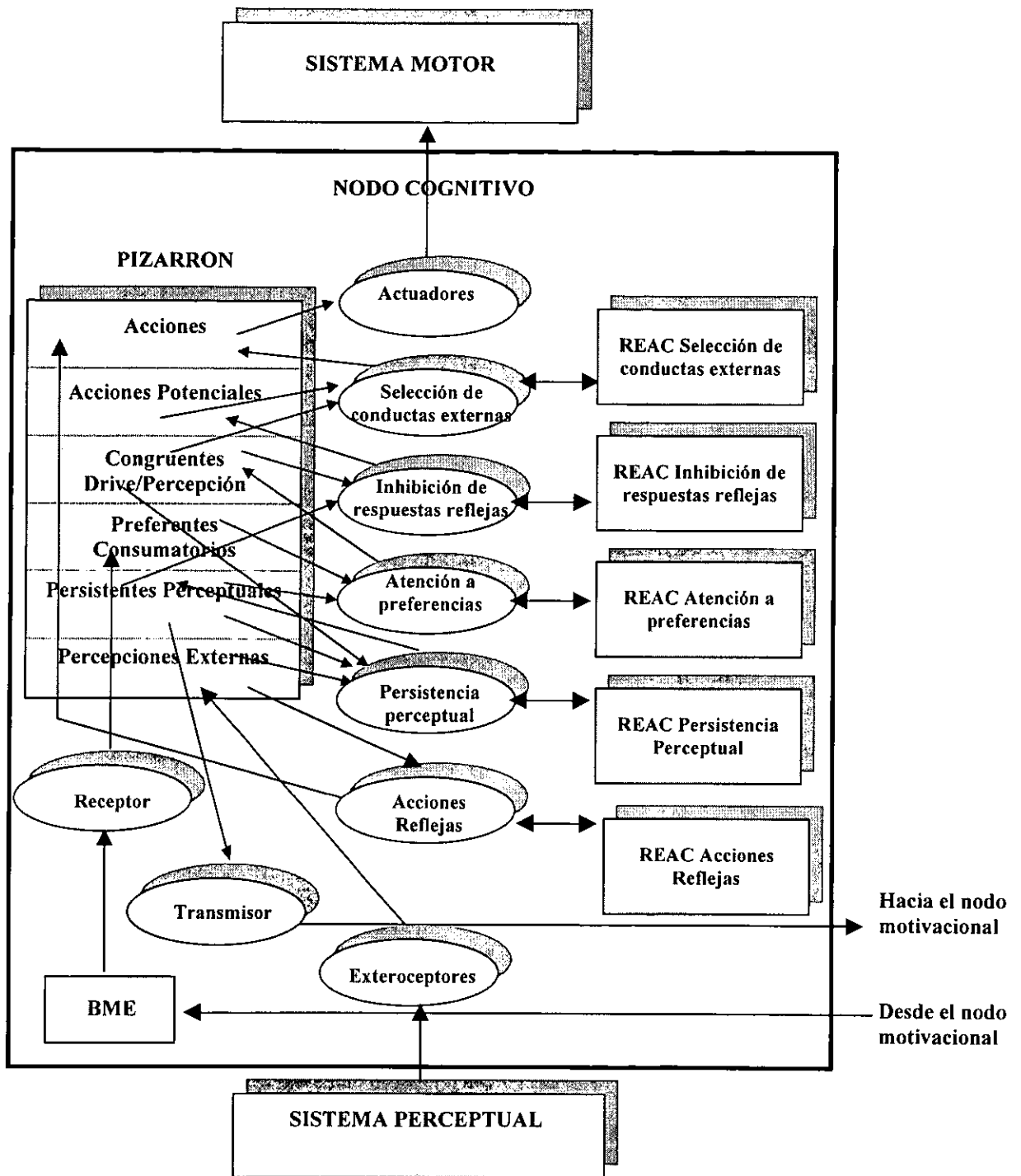


Figura 3.2. Estructura del nodo cognitivo. Se han considerado los registros de estados de actividad de las conductas internas, así como las fuentes originadoras y receptoras de flujos de señales.

Sobre el nivel Percepciones Externas se proyectan directamente las señales del medio externo sensatas y procesadas por el sistema perceptual. En el nivel Persistentes Perceptuales (o nivel de representación en memoria a corto plazo) persisten durante más tiempo las señales externas más fuertes y/o asociadas a estados internos predominantes. Sobre el nivel Preferentes Consumatorios se registran señales provenientes del nivel Drive del nodo motivacional. Las señales ubicadas en el nivel Congruentes Drive/Percepción son producto de la combinación de señales de los niveles Persistentes Perceptuales y Preferentes Consumatorios. Cuando en el nivel Acciones Potenciales

ha sido registrada una señal, entonces ocurrirá una de dos cosas: esta señal reforzará el disparo de la conducta externa iniciado por una señal del nivel Congruentes Drive/Percepción, o bien esta señal por sí sola podrá invocar una conducta externa al ser destruido todo el flujo de señales entrantes que provienen del nodo motivacional. Cuando un elemento solución es creado sobre el nivel Acciones, entonces la conducta externa asociada a este elemento será invocada, ejecutándose el código que implementa dicha acción.

En la tabla 3.1 se puede apreciar la notación utilizada para hacer referencia a elementos solución (señales) y valores de certidumbre (intensidad de la señal) en los diferentes niveles del pizarrón del nodo cognitivo.

Niveles del pizarrón	Elemento solución	Valor de certidumbre
Percepciones Externas	S_i	O_i^S
Persistentes Perceptuales	T_i	O_i^T
Preferentes Consumatorios	C_i	O_i^C
Congruentes Drive/Percepción	I_i	O_i^I
Acciones Potenciales	H_i	O_i^H
Acciones	M_i	O_i^M

Tabla 3.1.

3.2.2. Los exteroceptores

Los *exteroceptores* son los mecanismos que establecen la interfaz entre el sistema perceptual y el nodo cognitivo. La función de los *exteroceptores* es la creación de elementos solución en el nivel Percepciones Externas del pizarrón. Los elementos solución creados en este nivel corresponden precisamente a los a estímulos ambientales sentidos y procesados por el sistema perceptual. Es decir, los *exteroceptores* proyectan sobre el nivel Percepciones Externas las salidas producidas por el sistema perceptual en cada instante.

Al igual que las conductas internas, el mecanismo *exteroceptor* está compuesto por conductas elementales, cada una de las cuales exhibe una estructura de la forma “Si <condición> Entonces <acción>”. El nivel sobre el cual se aparean las condiciones de estas conductas elementales es el estrato de salida del sistema perceptual. Para cada unidad de salida U_i^S sobre este estrato, hay asociada una conducta elemental exteroceptora i , cuya condición es satisfecha cuando esta unidad U_i^S posee un valor diferente de cero ($Y_i^S \neq 0$).

La figura 3.3 muestra la estructura de una conducta elemental *exteroceptora*. En esta figura, F_{ii}^S es un valor constante que representa la fuerza de apareo de la conducta elemental; el procedimiento valor-salida(*,*) opera sobre el estrato de salida del sistema perceptual; y el procedimiento crear-elemento-sol(*,*) opera sobre el nivel Percepciones Externas del pizarrón, creando el elemento solución especificado en el primer parámetro con el valor de certidumbre especificado en el segundo parámetro.

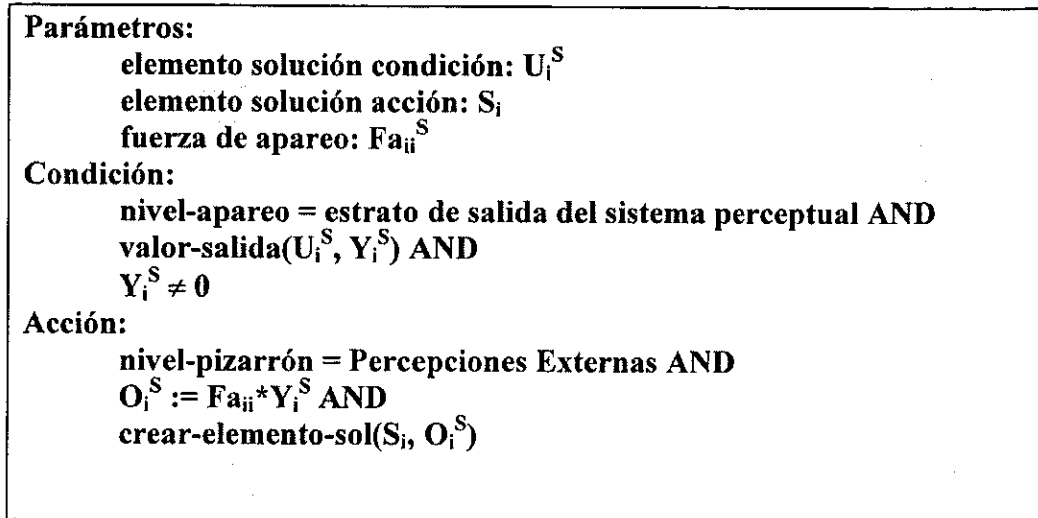


Figura 3.3. Estructura de un exteroceptor.

El mecanismo *exteroceptor* no requiere de la creación de registros de estados de actividad. Por lo tanto, todas las conductas elementales *exteroceptoras* que hayan satisfecho su condición en un instante dado, ejecutarán directamente sus acciones finales: la creación de un elemento solución (señal externa) sobre el nivel Percepciones Externas. De esta forma, todos los estímulos ambientales sensados y procesados por el sistema perceptual en un instante dado, participarán en el proceso de selección de acciones ejecutado por el trabajo conjunto de las conductas internas y mecanismos de interfaz/comunicación sobre el pizarrón. Es decir, la acción final ejecutada por el agente ha tomado en cuenta todos los estímulos ambientales percibidos por éste en ese instante.

3.2.3. La conducta interna *acciones reflejas*

El papel de la conducta interna *acciones reflejas* es permitir la activación inmediata de todas aquellas columnas conductuales que modelan acciones reflejas, las cuales no requieren de una entrada interna para la ejecución de la acción externa asociada a dicha columna.

La condición de una conducta elemental *acciones reflejas* es satisfecha cuando en el nivel Percepciones Externas del nodo cognitivo ha sido creado el elemento solución S_i con valor de certidumbre $O_i^S \neq 0$. La acción de esta conducta consiste en la creación del elemento solución M_i con valor de certidumbre O_i^M en el nivel Acciones de este mismo nodo.

Cuando la condición de una conducta elemental *acciones reflejas* es satisfecha, la conducta es activada creando su correspondiente REAC. La ejecución de la acción final de esta conducta depende completamente de la competencia que tienen lugar entre todos los REACs del mismo tipo creados durante el ciclo actual.

Debido a que las columnas conductuales en las cuales opera esta conducta no requieren de una entrada interna, entonces el proceso competitivo depende únicamente del valor de la entrada externa.

La Figura 3.4 muestra la estructura de una conducta elemental *acciones reflejas*. La estructura de un REAC para este tipo de conducta puede ser apreciado en la Tabla 3.2.

Parámetros:	
elemento solución condición:	S_i
elemento solución acción:	M_i
fuerza de apareo:	Fa_{ii}^S
Condición:	
nivel-pizarrón =	Percepciones Externas AND
cert(S_i , O_i^S) AND	
$O_i^S \neq 0$	
Acción 1:	
Calcular $A_i^M =$	$Fa_{ii}^S * O_i^S$
inscribir REAC _i	
Acción 2:	
nivel-pizarrón =	Acciones AND
crear-elemento-sol(M_i , O_i^M)	

Figura 3.4. Estructura de una conducta elemental *acciones reflejas*.

Atributo	Descripción
O_i^S	valor de certidumbre del elemento solución S_i
Fa_{ii}^S	fuerza de apareo
A_i^M	nivel de activación de la regla i
O_i^M	valor de certidumbre del elemento solución M_i (determinado durante la competencia)

Tabla 3.2.

Proceso competitivo entre conductas elementales acciones reflejas

Sólo una conducta elemental *acciones reflejas* podrá ejecutar su acción final sobre el nivel Acciones del pizarrón. El procedimiento para determinar la regla ganadora es el siguiente:

1. Para cada REAC inscrito se calcula el valor de certidumbre del elemento solución que la regla pretende crear (O_i^M) a partir de la siguiente expresión:

$$O_i^M = \begin{cases} 1 & \text{si } A_i^M = \max(A_j^M) \forall j. \\ 0 & \text{en otro caso} \end{cases} \quad (3.1)$$

2. Se selecciona el único REAC i con $O_i^M = 1$.

3. La regla asociada al REAC seleccionado ejecutará su acción 2 sobre el nivel Acciones del pizarrón.

3.2.4. La conducta interna persistencia perceptual

El rol principal de la conducta interna *persistencia perceptual* es la representación en el nivel Persistentes Perceptuales de las señales externas que han sido inicialmente proyectadas en el nivel Percepciones Externas del pizarrón. La condición de la *i*-ésima conducta elemental *persistencia perceptual* es satisfecha cuando al menos uno de los siguientes hechos ha tenido lugar: en el nivel Percepciones Externas del pizarrón ha sido creado el elemento solución S_i con certidumbre O_i^S , y/o en el nivel Persistentes Perceptuales se ha creado el elemento solución T_i con certidumbre O_i^T , y/o en el nivel Congruentes Drive/Percepción ha sido creado el elemento solución I_i con certidumbre O_i^I . Una condición necesaria para la ocurrencia de los dos últimos hechos es que el primero haya ocurrido.

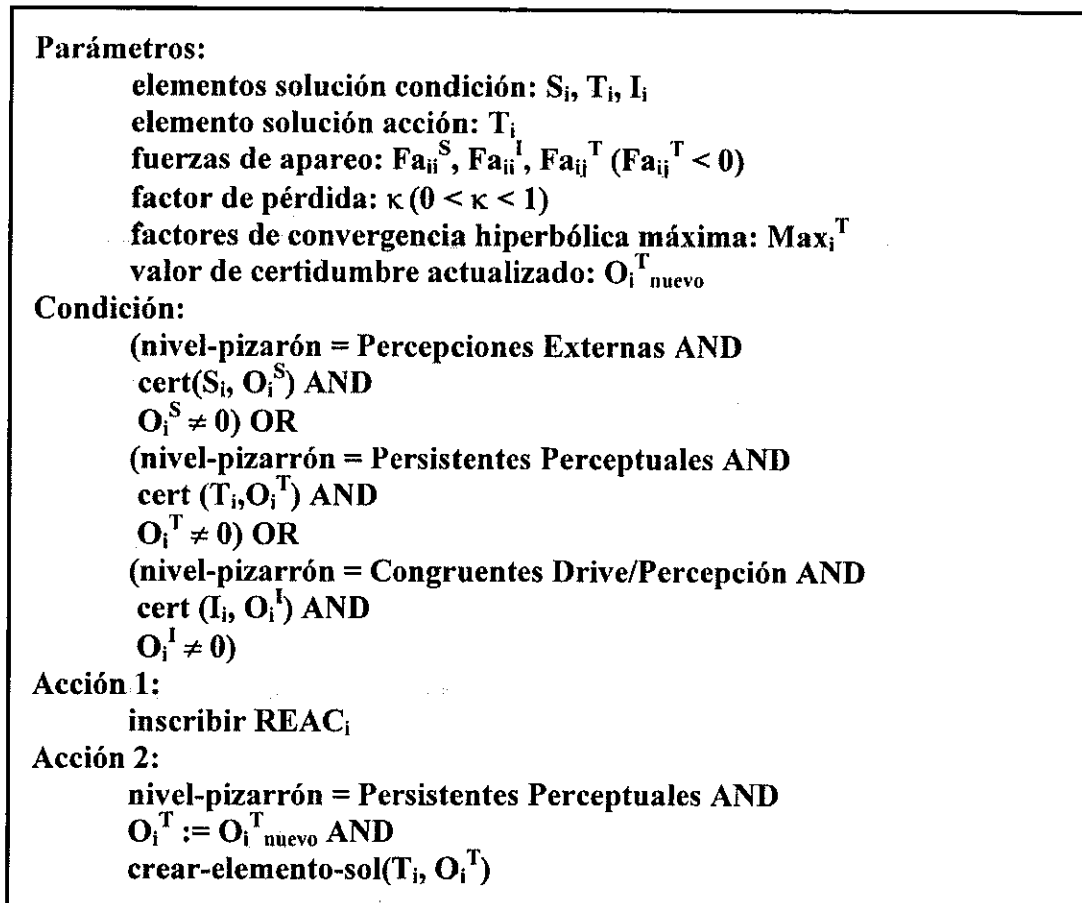


Figura 3.5. Estructura de una conducta elemental persistencia perceptual.

Cuando la condición de una conducta elemental *persistencia perceptual* es satisfecha, la conducta es activada creando su correspondiente REAC. La ejecución de la acción final de esta conducta es fuertemente dependiente del proceso competitivo que tiene lugar entre los REACs asociados a las restantes conductas elementales de este mismo tipo que han logrado satisfacer sus condiciones. Es decir, en cada etapa de la competencia, sólo conductas elementales *persistencia perceptual* asociadas a REACs ganadores podrán mantener representado en el nivel Persistentes Perceptuales el elemento solución especificado en sus partes acción. Ya a este nivel, el proceso de representación se traduce a la modificación del valor de certidumbre del elemento solución previamente creado por la conducta elemental (similar a un proceso de revisión de creencias (González y Negrete, 1997)).

La figura 3.5 muestra la estructura de una conducta elemental *persistencia perceptual*. La estructura de un REAC para este tipo de conducta puede ser apreciada en la tabla 3.3. El procedimiento cert (*,*) retorna en el segundo parámetro el valor de certidumbre del elemento solución especificado en el primer parámetro. $O_i^T_{nuevo}$ es el nuevo valor de certidumbre del elemento solución T_i , recalculado en el proceso competitivo.

Atributo	Descripción
O_i^S	valor de certidumbre del elemento solución S_i
O_i^T	valor de certidumbre del elemento solución T_i ($O_i^T = F_{a_{ij}}^S * O_i^S$ para $t=t_0$)
O_i^I	valor de certidumbre del elemento solución I_i ($O_i^I = 0$ para $t=t_0$)
$F_{a_{ij}}^I, F_{a_{ij}}^T$	fuerzas de apareo ($F_{a_{ij}}^T < 0, j=1, 2, \dots$)
κ	factor de pérdida ($0 < \kappa < 1$)
A_i^T	nivel de activación de la regla ($A_i^T = O_i^T$ para $t=t_0$)
$O_i^T_{nuevo}$	nuevo valor de certidumbre calculado para el elemento solución T_i

Tabla 3.3.

Proceso competitivo entre conductas elementales *persistencia perceptual*

Entre todos los REACs asociados a conductas elementales *persistencia perceptual* se lleva a cabo un proceso competitivo para determinar cuáles conductas elementales ejecutarán su acción final sobre el pizarrón. Al final de cada etapa del proceso competitivo, todas las conductas elementales asociadas a REACs ganadores ejecutarán su acción final sobre el pizarrón (la creación del elemento solución T_i especificado en la parte acción, o la modificación del nivel de certidumbre asociado a éste). El proceso para seleccionar la(s) conducta(s) ganadora(s) es el siguiente:

1. Para cada REAC inscrito se calcula su nivel de activación (A_i^T) a partir de las siguientes expresiones:

$$Atmp_i^T = (1-\kappa) \cdot A_i^T(t) + Fa_{ii}^S \cdot O_i^S + O_i^T(t) + Fa_{ij}^L \cdot O_j^L(t) + \sum_{j \neq i} Fa_{ij}^T \cdot O_j^T \quad (3.2)$$

$$A_i^T(t+1) = \begin{cases} (-1/(Atmp_i^T + (1/Max_i^T)) + Max_i^T & \text{si } Fa_i^S \cdot O_i^S > 0 \text{ y } Atmp_i^T > 0 \\ Atmp_i^T & \text{en otro caso} \end{cases} \quad (3.3)$$

donde: O_j^T es el nivel de certidumbre del elemento solución T_j registrado en el REAC $_j$, $Fa_{ij}^T < 0$, $Fa_{ii}^T = 0.0$; y para $t = 0$, $A_i^T(0) = 0$.

2. Para cada REAC inscrito se calcula el nuevo valor del nivel de certidumbre del elemento solución T_i ($O_i^T_{nuevo}$) a partir de la siguiente expresión:

$$O_i^T_{nuevo} = \begin{cases} A_i^T(t+1) & \text{si } A_i^T(t+1) > \theta^T \\ 0 & \text{en otro caso} \end{cases} \quad (3.4)$$

3. Se selecciona(n) la(s) regla(s) que cumpla(n) la condición $O_i^T_{nuevo} \neq 0$.
4. La(s) regla(s) seleccionada(s) ejecutará(n) su acción 2 sobre el nivel Persistentes Perceptuales del pizarrón.
5. Asignación de los nuevos valores calculados a los atributos:

$$A_i^T = A_i^T(t+1)$$

$$O_i^T = O_i^T_{nuevo}$$

Este proceso de competencia converge a un estado en el cual sólo una conducta elemental se mantendrá ejecutando su acción, mientras que el nivel de activación de las restantes ha caído por debajo de cero. El período durante el cual la conducta elemental ganadora i permanece ejecutando su acción, define el tiempo de representación en el nivel Persistentes Perceptuales del elemento solución T_i .

El tiempo durante el cual este elemento solución permanecerá activo en este nivel dependerá del valor asignado al factor de pérdida κ en la expresión 3.2, teniendo en cuenta que la relación entre el tiempo de persistencia del elemento solución T_i en el nivel Persistentes Perceptuales y el valor del factor de pérdida κ en esta expresión es inversamente proporcional.

La ecuación 3.3 determina que el nivel de activación de un estímulo T_i (A_i^T) converja hiperbólicamente al valor Max_i^T , evitando de esta forma que el nivel de activación crezca desproporcionadamente.

De esta forma, la conducta interna *persistencia perceptual* emula un “mecanismo de memoria a corto plazo”, imprescindible para los procesos de aprendizaje asociativo que ocurren a nivel de la red de conductas internas.

3.2.5. La conducta interna atención a preferencias

La función de la conducta interna *atención a preferencias* es la combinación de elementos solución registrados en los niveles Persistentes Perceptuales y Preferentes Consumatorios del pizarrón. Las conductas elementales que forman esta conducta interna funcionan como operadores AND u operadores OR, en dependencia de los valores que tomen los parámetros γ y ϕ . La acción final de la conducta elemental i consiste en la creación del elemento solución I_i en el nivel Congruentes Drive/Percepción.

Cuando las condiciones o premisas de una conducta elemental *atención a preferencias* han sido satisfechas ésta es activada, creando su correspondiente REAC. En la figura 3.6 se puede apreciar la estructura de una conducta elemental *atención a preferencias*. La estructura de un REAC para este tipo de conducta es mostrada en la tabla 3.4.

<p>Parámetros: elementos solución condición: T_i, C_j ($j = 1, 2, \dots$) elemento solución acción: I_i fuerzas de apareo: Fa_{ij}^T, Fa_{ij}^C ($j = 1, 2, \dots$) importancia de la señal proveniente del nodo motivacional: ϕ factor de reactividad: γ</p> <p>Condición: nivel-pizarrón = Persistentes Perceptuales AND cert(T_i, O_i^T) AND $O_i^T \neq 0$ AND ($\gamma > 0$ OR nivel-pizarrón = Preferentes Consumatorios AND cert(C_j, O_j^C) AND $O_j^C \neq 0$ para al menos una j)</p> <p>Acción 1: $O_i^I := Fa_i^T * O_i^T * (\gamma + \phi * \sum_j Fa_j * O_j^C / O_i^T)$ AND Inscribir REAC$_i$</p> <p>Acción 2: nivel-pizarrón = Congruentes Drive/Percepción AND crear-elemento-sol(I_i, O_i^I)</p>
--

Figura 3.6. Estructura de una conducta elemental atención a preferencias.

Los valores de los parámetros γ y ϕ en la expresión para el cálculo de O_i^I son utilizados para modular la reactividad en el comportamiento observado en la entidad. Para valores de ϕ cercanos a cero, se le atribuye una menor importancia a las señales provenientes del nodo motivacional y viceversa. Para valores de γ mayores que cero, la conducta *atención a preferencias* podría ser activada por la presencia de una entrada externa aunque la correspondiente entrada interna no

esté presente. Si γ es igual a cero, entonces para que sea activada una conducta *atención a preferencias* será necesario que ambos tipos de entradas estén presentes. Una discusión más detallada de estos parámetros es proporcionada en el epígrafe 4.3, donde se describe bajo que condiciones tiene lugar el comportamiento reactivo.

Atributo	Descripción
O_i^I	valor de certidumbre del elemento solución T_i
$O_j^C (j = 1, 2, \dots)$	valores de certidumbre de los elementos solución C_j
F_{ij}^C	fuerzas de apareo
O_i^I	valor de certidumbre del elemento solución I_i

Tabla 3.4.

Proceso competitivo entre conductas elementales *atención a preferencias*

Entre todos los REACs asociados a conductas elementales *atención a preferencias* e inscritos en un mismo ciclo se lleva a cabo una competencia para determinar cuáles conductas elementales ejecutarán su acción final sobre el pizarrón. En este proceso de selección más de una conducta elemental puede resultar ganadora. El criterio para la selección de conductas elementales ganadoras es el siguiente: el REAC i gana la competencia si $O_i^I > \theta^I$, donde θ^I es un valor umbral establecido. Todas las conductas elementales asociadas a REACs ganadores ejecutarán su acción 2 sobre el nivel Congruentes Drive/Percepción del pizarrón.

3.2.6. La conducta interna inhibición de respuestas reflejas

La conducta interna *inhibición de respuestas reflejas* implementa el principio de organización jerárquica discutido más adelante en el epígrafe 4.13. Es decir, la creación de un elemento solución en el nivel Persistentes Perceptuales no podrá implicar directamente la creación de un elemento solución en el nivel Acciones (y por lo tanto, la ejecución de una conducta externa) si durante este mismo ciclo ha sido creado un elemento solución en el nivel Congruentes Drive/Percepción. Cuando este último hecho ha tenido lugar, entonces la conducta *inhibición de respuestas reflejas* garantiza que el elemento solución que será creado en el nivel Acciones sea determinado por el elemento solución registrado en el nivel Congruentes Drive/Percepción.

La conducta interna *inhibición de respuestas reflejas* está compuesta por un conjunto de conductas elementales específicas (asociadas a necesidades específicas de la entidad) y una conducta elemental por defecto (la cual, en el caso de ciertos tipos de agentes, pudiera estar asociada a la actividad exploratoria). Las premisas de la i -ésima conducta elemental específica son satisfechas cuando en el nivel Persistentes Perceptuales ha sido creado el elemento solución T_i y/o en el nivel Congruentes Drive/Percepción se ha creado el elemento solución I_j . Ante esta situación, la conducta elemental i es activada ejecutando su primera acción, la cual consiste en la inscripción de su correspondiente REAC. Cuando $i = j$, entonces la acción final de la conducta elemental i consiste en la creación del elemento solución H_i en el nivel Acciones Potenciales, el

cual posteriormente contribuirá al reforzamiento del nivel de certidumbre del elemento solución M_i que será creado en el nivel Acciones. Cuando $i \neq j$, entonces la acción final de la conducta i es bloqueada por la presencia del elemento solución I_j .

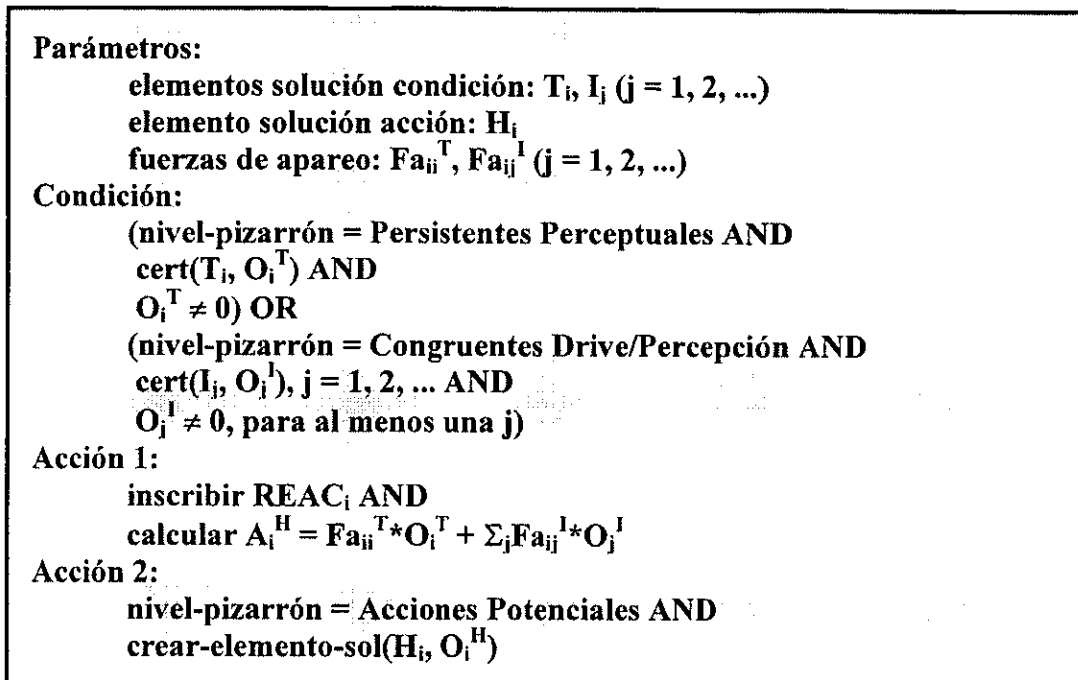


Figura 3.7. Estructura de una conducta elemental específica inhibición de respuestas reflejas.

Las premisas de la conducta elemental por defecto son satisfechas cuando en el nivel Preferentes Consumatorios ha sido creado el elemento solución C_k con certidumbre $O_k^C \neq 0$ y/o en el nivel Congruentes Drive/Percepción existe al menos un elemento solución I_j con certidumbre $O_j^I \neq 0$; mientras que su acción también consiste en la creación de un elemento solución en el nivel Acciones Potenciales. En la figura 3.7 se puede apreciar la estructura de una conducta elemental específica. Aquí $F_{a_{ij}^T}$ y $F_{a_{ij}^I}$ son las fuerzas de apareo del comportamiento i respecto a las condiciones O_i^T y O_j^I , respectivamente; $F_{a_{ij}^I}$ toma valores enteros negativos para $i \neq j$, siendo positivo para $i = j$; y el valor de O_i^H es determinado durante la competencia. En la figura 3.8 se puede apreciar la estructura de la conducta elemental por defecto. Aquí $F_{a_{kk}^C}$ y $F_{a_{kj}^I}$ son las fuerzas de apareo de la conducta k respecto a las condiciones O_k^C y O_j^I , respectivamente; $F_{a_{kj}^I}$ toma valores enteros negativos; y el valor de O_k^H es determinado durante la competencia. La estructura de un REAC para estos dos tipos de conductas internas es la misma y es mostrada en la tabla 3.5.

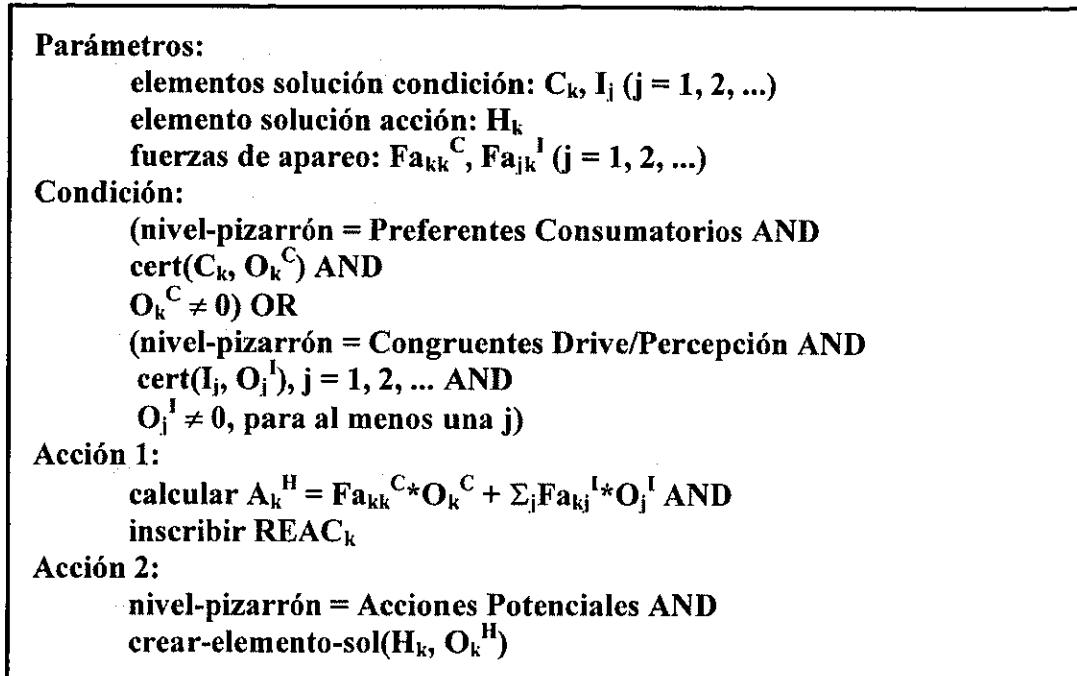


Figura 3.8. Estructura de la conducta elemental por defecto inhibición de respuestas reflejas.

Atributo	Descripción
O_i^I	valor de certidumbre del elemento solución T_i
O_j^I ($j = 1, 2, \dots$)	valores de certidumbre de los elementos solución I_j
Fa_{ii}^I, Fa_{ij}^I	fuerzas de apareo
A_i^H	nivel de activación de la conducta elemental i (determinado durante la competencia)
O_i^H	valor de certidumbre del elemento solución H_i (determinado al final de la competencia)

Tabla 3.5.

Proceso competitivo entre conductas elementales *inhibición de respuestas reflejas*

Para determinar qué conductas elementales *inhibición de respuestas reflejas* serán bloqueadas y cuál ejecutará su acción final, una competencia es establecida entre todos los REACs asociados a este tipo de conducta elemental e inscritos durante este ciclo. El procedimiento para seleccionar la conducta elemental ganadora es el siguiente:

1. Para cada REAC inscrito se calcula el valor de certidumbre del elemento solución que la conducta elemental asociada pretende crear (O_i^H) a partir de la siguiente expresión:

$$O_i^H = \begin{cases} A_i^H & \text{si } A_i^H > 0 \\ 0 & \text{en otro caso} \end{cases} \quad (3.5)$$

2. Se seleccionan los REACs que cumplan con la condición $O_i^H \neq 0$.
3. Las conductas elementales asociadas a los REACs seleccionados ejecutarán su acción 2 sobre el nivel Acciones Potenciales del pizarrón.

3.2.7. La conducta interna selección de conductas externas

El papel de la conducta interna *selección de conductas externas* es decidir qué conducta externa será ejecutada en el ciclo actual. Las conductas elementales *selección de conductas externas* funcionan como operadores OR inclusivo. La condición de la i -ésima conducta elemental es satisfecha cuando se cumple la condición O_i^I OR O_i^H ($O_i^I \neq 0$ o $O_i^H \neq 0$). Pero si tenemos en cuenta el principio de organización jerárquica que será discutido en el epígrafe 4.13, esta condición realmente debe comportarse como un operador AND, a menos que se hayan destruido todos los flujos de señales entrantes que provienen del nodo motivacional ($\phi = 0$). La acción final de la conducta elemental i consiste en la creación del elemento solución M_i con certidumbre O_i^M en el nivel Acciones.

En la figura 3.9 se puede apreciar la estructura de una conducta elemental *selección de conductas externas*. Aquí los Fa_{ij}^H , Fa_{ij}^I y Fa_{ik}^H son las fuerzas de apareo de la conducta elemental i respecto a las condiciones O_j^H , O_j^I y O_k^H , respectivamente; el segundo y tercer términos de la derecha en la expresión para el cálculo de A_i^M son mutuamente excluyentes; y el valor O_i^M es determinado durante la competencia. La estructura de un REAC para este tipo de conducta interna es mostrada en la tabla 3.6.

Existe una conducta elemental *selección de conductas externas* por defecto (la cual en el caso de ciertos tipos de agentes pudiera estar asociada a la actividad exploratoria). La premisa de esta conducta es satisfecha cuando en el nivel Acciones Potenciales ha sido creado el elemento solución H_k con valor de certidumbre $O_k^H \neq 0$. Al igual que en las restantes conductas elementales específicas la acción consiste en la creación del elemento solución M_k en el nivel Acciones. La estructura que posee esta conducta puede ser apreciada en la figura 3.10, mientras que la estructura de su REAC es la misma que se muestra en la tabla 3.6.

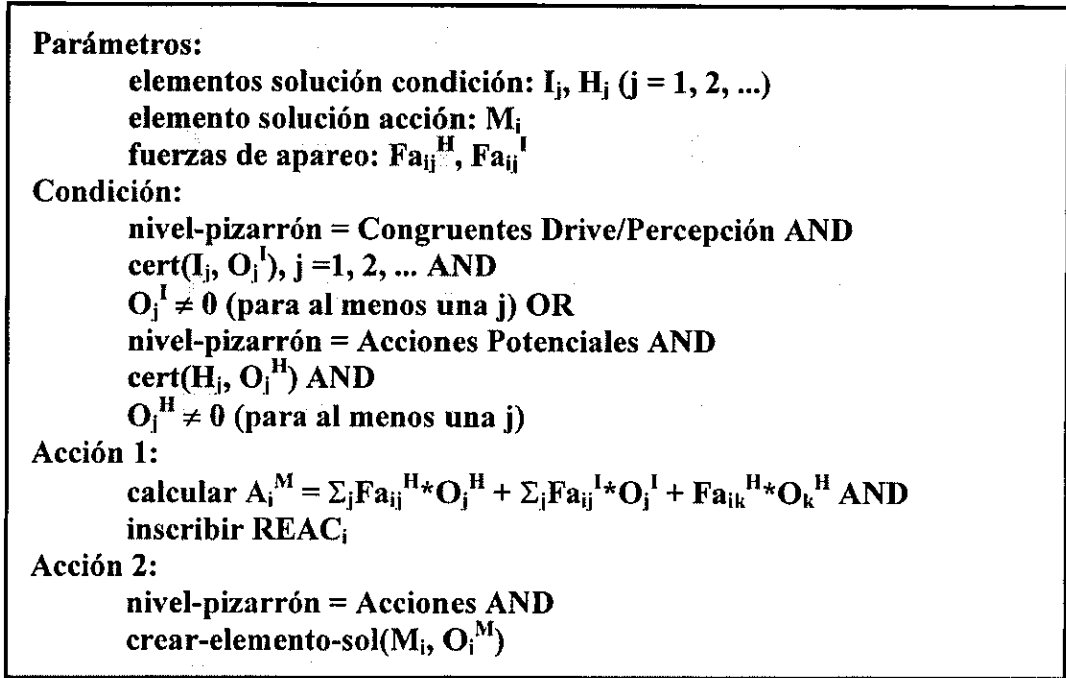


Figura 3.9. Estructura de una conducta elemental *selección de conductas externas*.

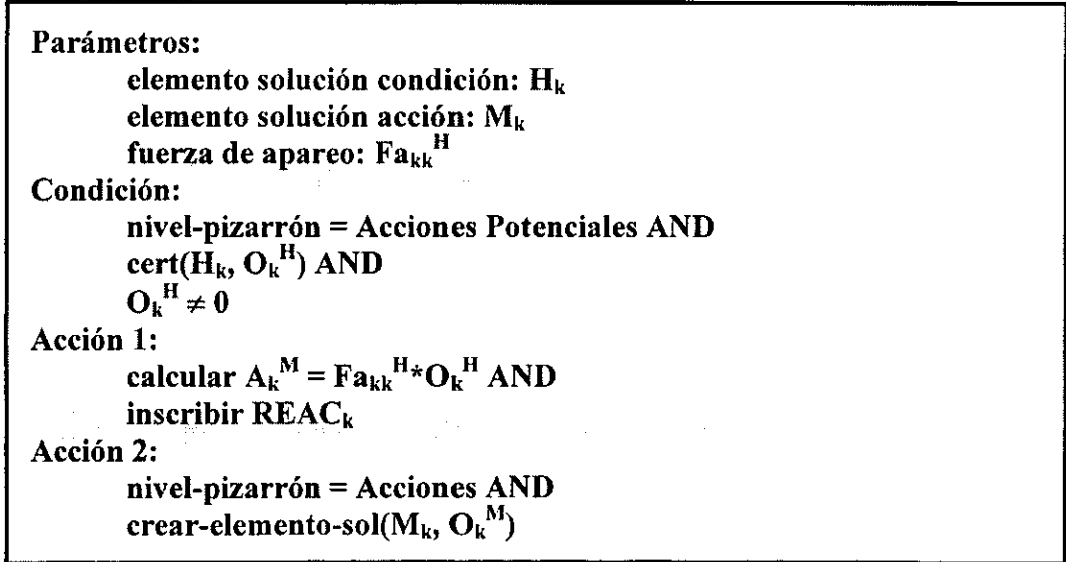


Figura 3.10. Estructura de la conducta elemental *selección de conductas externas por defecto*.

Atributo	Descripción
O_i^I	valor de certidumbre del elemento solución I_j
O_j^H	valor de certidumbre del elemento solución H_j
$F_{a_{ij}}^H, F_{a_{ij}}^I$	fuerzas de apareo
A_i^M	nivel de activación de la regla i
O_i^M	valor de certidumbre del elemento solución M_i (determinado durante la competencia)

Tabla 3.6.

Proceso competitivo entre conductas elementales *selección de conductas externas*

Sólo una conducta elemental *selección de conductas externas* podrá ejecutar su acción final sobre el nivel Acciones del pizarrón. El procedimiento para determinar la regla ganadora es el siguiente:

1. Para cada REAC inscrito se calcula el valor de certidumbre del elemento solución que la regla pretende crear (O_i^M) a partir de la siguiente expresión:

$$O_i^M = \begin{cases} 1 & \text{si } A_i^M = \max(A_j^M > \theta^M) \forall j \\ 0 & \text{en otro caso} \end{cases} \quad (3.6)$$

donde: θ^M es un valor umbral establecido.

2. Se selecciona el único REAC i con $O_i^M = 1$.
3. La regla asociada al REAC seleccionado ejecutará su acción 2 sobre el nivel Acciones del pizarrón.

3.2.8. Los actuadores

Los *actuadores* son los mecanismos que establecen la interfaz entre el nodo pizarrón y el sistema motor. Un *actuador* es una regla de producción, cuya condición es satisfecha sobre el nivel Acciones del pizarrón y su consecuencia se traduce en el desencadenamiento de una conducta externa determinada (acción motora). En cada ciclo de ejecución, sólo un elemento solución con certidumbre igual a uno ($O_i^M = 1$) podrá ser creado en el nivel Acciones, por lo que a lo sumo un *actuador* habrá satisfecho su condición, y será éste el que directamente invoque la acción motora especificada en su parte conclusión. En la figura 3.11 se muestra la estructura de un *actuador*. Aquí N_i es la i -ésima acción motora especificada en el sistema motor.

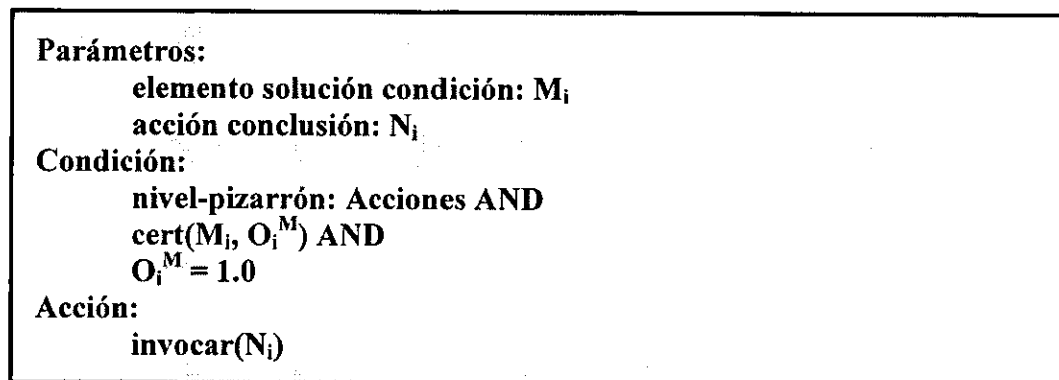


Figura 3.11 Estructura de un *actuador*.

3.2.9. El mecanismo receptor

La función del mecanismo *receptor* es registrar en el pizarrón del nodo cognitivo las señales que arriban a éste provenientes del nodo motivacional. Estas señales son elementos solución generados por conductas internas del nodo motivacional, y requeridos por conductas internas del nodo cognitivo para satisfacer sus premisas.

Las señales que llegan al nodo cognitivo son inicialmente registradas en el buzón de mensajes de entrada (BME) de este nodo por el mecanismo *transmisor* del nodo motivacional. El tiempo de permanencia de estas señales en el BME es instantáneo, ya que ante cada señal registrada en el buzón, el mecanismo *receptor* del nodo se activa, efectuando el registro final de la señal en el nivel adecuado del pizarrón. Es decir, el mecanismo *receptor* no requiere de la creación de REACs, cada vez que su condición es satisfecha éste ejecuta su acción final. La figura 3.12 muestra la estructura del mecanismo *receptor*. La estructura del BME es mostrada en la tabla 3.7.

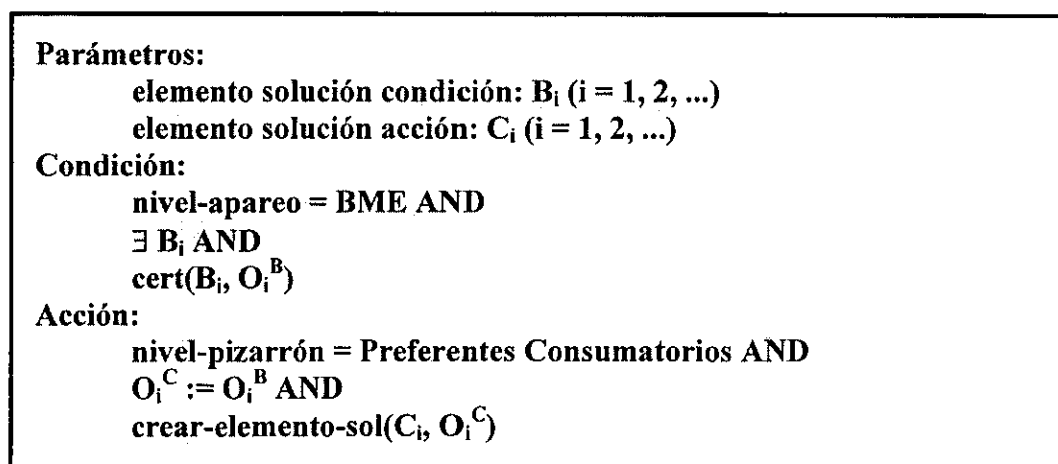


Figura 3.12. Estructura del mecanismo *receptor*.

Atributo	Descripción
B_i ($i = 1, 2, \dots$)	identificador de la señal de entrada
$\text{cert}(B_i)$	valor de certidumbre asociado a la señal de entrada
id-conducta	identificador de la conducta interna que creó el elemento solución
id-nodo	identificador del nodo emisor

Tabla 3.7.

Para la estructura actual que exhibe la red de nodos pizarrón, todos los mensajes que arriban al BME del nodo cognitivo proceden del nodo motivacional, y son finalmente registrados en el nivel Preferentes Consumatorios del pizarrón del nodo cognitivo. No obstante, el mecanismo *receptor* posee conocimiento para decidir en que nivel del pizarrón local debe registrar la señal receptionada en el BME, tomando como criterios los identificadores del nodo emisor y de la conducta interna que generó dicho elemento.

3.2.10. El mecanismo transmisor

El mecanismo *transmisor* envía al nodo motivacional aquellos elementos solución generados en el pizarrón del nodo cognitivo que son requeridos por conductas internas del nodo motivacional para satisfacer sus premisas. Para la estructura actual que exhibe la red de nodos pizarrón, la condición del comportamiento *transmisor* es satisfecha cuando en el nivel Persistentes Perceptuales del pizarrón ha sido creado un elemento solución T_i ($i = 1, 2, \dots$) o ha sido actualizado el valor de certidumbre de algún elemento solución T_j ($j = 1, 2, \dots$) previamente creado. El mecanismo *transmisor* no requiere de la creación de REACs, cada vez que su condición es satisfecha éste ejecuta su acción final, la cual consiste en registrar en el BME del nodo motivacional el elemento solución responsable de su disparo. En la figura 3.13 se puede apreciar la estructura del mecanismo *transmisor*.

Parámetros:
elemento solución condición: T_i ($i = 1, 2, \dots$)
elemento solución acción: B_i ($i = 1, 2, \dots$)
conducta interna que generó el elemento solución: id-conducta
nodo-emisor: id-nodo-e
nodo-receptor: id-nodo-r
Condición:
nivel-pizarrón = Persistentes Perceptuales AND
$\text{cert}(T_i, O_i^{\text{T act}})$ AND
$O_i^{\text{T act}} \neq 0$
Acción:
transmitir(id-nodo-r, T_i, $O_i^{\text{T act}}$, id-conducta, id-nodo-e)

Figura 3.13. Estructura del mecanismo *transmisor*.

La condición $O_i^{\text{T act}} \neq 0$ asegura que tanto los elementos solución recién creados, como los elementos solución que aún persisten en este nivel, disparen la acción del mecanismo *transmisor*.

Cuando la condición del mecanismo *transmisor* es satisfecha, el procedimiento *transmitir* es invocado, recibiendo como parámetros id-nodo-r, T_i , O_i^T _{act}, id-conducta e id-nodo-e. El parámetro id-nodo-r indica al procedimiento *transmitir* sobre qué BME deben ser registrados los restantes parámetros.

3.3. El nodo motivacional

En la figura 3.14 se puede apreciar la estructura del nodo motivacional. En esta ilustración se han incluido todos los componentes del nodo, así como las fuentes originadoras y receptoras de flujos de señales. Para este nodo han sido definidas las conductas internas *congruencia propio/extero/drive* y *selector de preferencias consumatorias*. El nodo motivacional recibe señales desde el medio interno a través de los *propioceptores* y del nodo cognitivo por medio del mecanismo *receptor*, y envía señales al nodo cognitivo a través del mecanismo *transmisor*.

El rol del nodo motivacional abarca la combinación de señales internas y externas, y la competencia a nivel motivacional de conductas motivacionalmente incompatibles. Es decir, la conducta externa final observada en la entidad es fuertemente dependiente de sus estados internos. Todos los estados internos, para los que existen señales externas aprovechables, compiten entre sí para determinar la conducta externa final que será ejecutada por la entidad. La competencia es del tipo *the winner takes all*.

3.3.1. El pizarrón del dominio

El pizarrón del dominio del nodo motivacional organiza los elementos solución en cuatro niveles de abstracción:

- Primer nivel: Percepciones Internas
- Segundo nivel: Percepciones Externas
- Tercer nivel: Congruentes Propio/Extero/Drive
- Cuarto nivel: Drive

Las señales registradas sobre el nivel Percepciones Internas son los valores actuales de las variables internas, sensados y preprocesados por el mecanismo *propioceptor*. En el nivel Percepciones Externas se registran los valores de las señales externas representadas en el nivel Persistentes Perceptuales del nodo cognitivo (transmitidas y recepcionadas por los mecanismos de comunicación). Las señales ubicadas sobre el nivel Congruentes Propio/Extero/Drive son producto de la combinación de señales de los niveles Percepciones Internas, Persistentes Perceptuales y Drive. Sobre el nivel Drive se registran los elementos solución creados por las conductas elementales que componen la conducta interna *selector de preferencias consumatorias*. Durante la competencia que se establece entre estas conductas elementales, el nivel Drive permanece cerrado en lectura para los mecanismos de comunicación. Al final de la competencia, a lo sumo sólo dos elementos solución podrán coexistir sobre este nivel. En la tabla 3.8 se

muestra la notación utilizada para hacer referencia a los elementos solución y valores de certidumbre en los diferentes niveles del pizarrón del nodo motivacional.

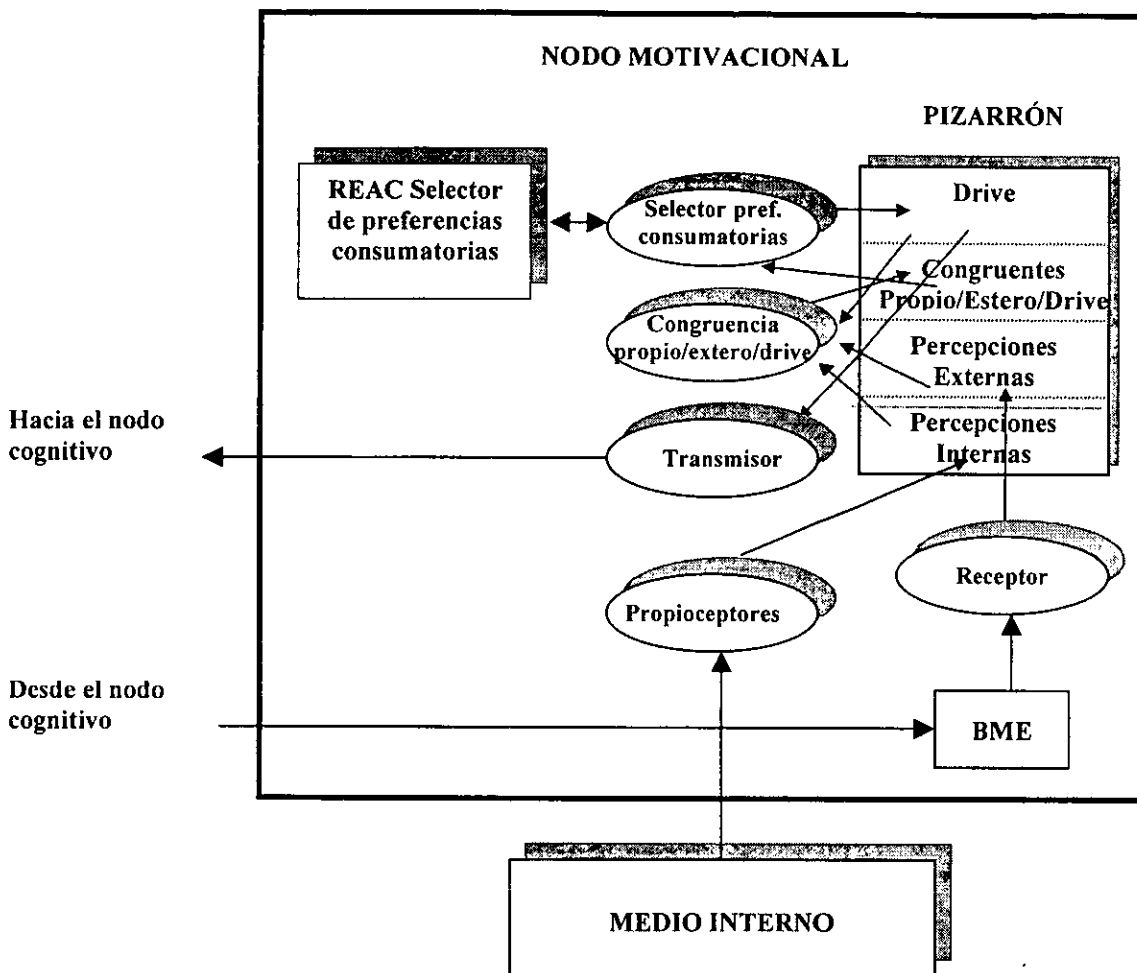


Figura 3.14. Estructura del nodo motivacional.

Nivel del pizarrón	Elemento solución	Valor de certidumbre
Percepciones Internas	E_i	O_i^E
Percepciones Externas	S_i	O_i^S
Congruencia Propio/Extero/Drive	C_i	A_i^C
Drive	D_i	O_i^D

Tabla 3.8.

3.3.2. Los propioceptores

Los *propioceptores* son los mecanismos que establecen la interfaz entre el nodo motivacional y el medio interno. La función de éstos se traduce en la creación de elementos solución en el nivel Percepciones Internas del pizarrón. Los elementos solución creados en este nivel son los valores actuales de las variables internas (necesidades internas de la entidad). Al igual que las conductas internas, el mecanismo *propioceptor* está compuesto por reglas de producción (SI <condición> ENTONCES <acción>). La condición de una regla *propioceptora* es satisfecha cuando la variable del estado interno asociada a ésta posee un valor diferente de cero.

El mecanismo *propioceptor* no requiere de la creación de REACs. Por lo tanto, todas las reglas componentes que hayan satisfecho sus condiciones en un instante dado, ejecutarán directamente sus acciones finales sobre el nivel Percepciones Internas del pizarrón. Pudiera decirse que el mecanismo *propioceptor* ejecuta un mapeo continuo entre los valores de las variables que definen el estado interno y los correspondientes elementos solución sobre el primer nivel del pizarrón. La figura 3.15 muestra la estructura de un *propioceptor*.

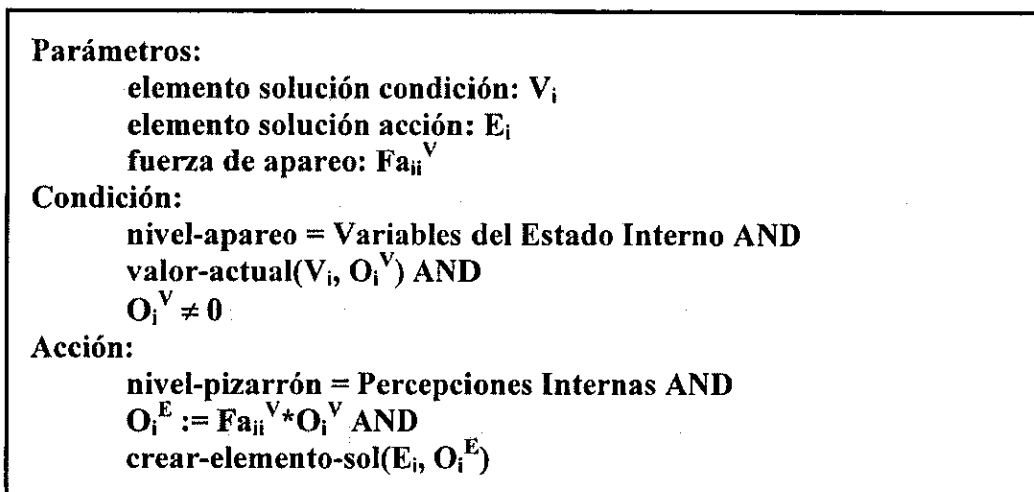


Figura 3.15. Estructura de un propioceptor.

3.3.3. La conducta interna congruencia propio/extero/drive

La conducta interna *congruencia propio/extero/drive* combina señales registradas en los niveles Percepciones Internas, Persistentes Perceptuales y Drive del pizarrón. El esquema de combinación de señales utilizado por esta conducta viene dado por la expresión (3.7).

$$A_i^C = O_i^E * (\alpha + \sum_j F_{a_{ij}}^S * O_j^S) + O_i^D \quad (3.7)$$

donde: A_i^C es el valor de certidumbre con el cual será creado el elemento solución C_i en el nivel Congruentes Propio/Extero/Drive del pizarrón, α es el peso o importancia atribuible al estado interno ($0 < \alpha \leq 1$), y $O_i^D = 0$, para toda i en el instante inicial $t = t_0$.

Cuando α es igual a cero, las conductas elementales que componen la conducta interna *congruencia propio/extero/drive* funcionan como operadores AND. Es decir, la condición de una regla es satisfecha sólo si la conjunción $O_i^E \text{ AND } \sum_j F_{ajj}^S * O_j^S$ es cierta ($O_i^E * \sum_j F_{ajj}^S * O_j^S \neq 0$). Un análisis más detallado de las propiedades de la conducta *congruencia propio/extero/drive* para diferentes valores de α será dado en los epígrafes 4.1 y 4.2.

<p>Parámetros: elementos solución condición: E_i, S_j, D_i ($j = 1, 2, \dots$) elemento solución acción: C_i fuerzas de apareo: F_{ajj}^S ($j = 1, 2, \dots$) importancia atribuible al estado interno: α</p> <p>Condición: $(\alpha = 0 \text{ AND}$ nivel-pizarrón = Percepciones Internas AND cert(E_i, O_i^E) AND $O_i^E \neq 0 \text{ AND}$ nivel-pizarrón = Percepciones Externas AND cert(S_i, O_i^S) AND $O_i^S \neq 0) \text{ OR}$ $(\alpha \neq 0 \text{ AND}$ nivel-pizarrón = Percepciones Internas AND cert(E_i, O_i^E))</p> <p>Acción: calcular $A_i^C = O_i^E * (\alpha + \sum_j F_{ajj}^S * O_j^S) + O_i^D$ AND nivel-pizarrón = Congruentes Propio/Extero/Drive AND crear-elemento-sol(C_i, A_i^C)</p>
--

Figura 3.16. Estructura de una conducta elemental *congruencia propio/extero/drive*

La conducta interna *congruencia propio/extero/drive* no requiere de la creación de REACs. Cada vez que la condición de una conducta elemental es satisfecha, ésta ejecuta su acción final, creando el elemento solución C_i con certidumbre A_i^C en el nivel Congruentes Propio/Extero/Drive del pizarrón. El nivel de certidumbre A_i^C representa la actividad de la conducta elemental i . La figura 3.16 muestra la estructura de una conducta elemental de este tipo.

Existe una conducta elemental *congruencia propio/extero/drive* asociada a una conducta elemental por defecto. En la figura 3.17 se muestra la estructura de esta conducta elemental.

Nótese que la expresión utilizada para calcular el nivel de activación de esta conducta ($A_k^C = \alpha * O_k^E$) es un caso particular de la expresión 3.7 cuando se toman $\alpha \neq 0$, $\sum_j F_{aj}^S * O_j^S = 0$ y $O_i^D = 0$.

<p>Parámetros: elemento solución condición: E_k elemento solución acción: C_k importancia atribuible al estado interno: α</p> <p>Condición: nivel-pizarrón = Percepciones Internas AND cert(E_k, O_k^E) AND $O_k^E \neq 0$</p> <p>Acción: calcular $A_k^C = \alpha * O_k^E$ AND nivel-pizarrón = Congruentes Propio/Extero/Drive AND crear-elemento-sol(C_k, A_k^C)</p>
--

Figura 3.17. Estructura de la conducta elemental *congruencia propio/ extero/drive* asociada a la conducta elemental por defecto.

3.3.4. La conducta interna selector de preferencias consumatorias

El papel de la conducta interna *selector de preferencias consumatorias* es “indicar” cuál es la conducta externa más adecuada que debe ejecutar el agente, dadas las actuales condiciones internas y externas. Esta conducta interna está compuesta por un conjunto de conductas elementales específicas, asociadas a necesidades específicas de la entidad, y una conducta elemental por defecto. La función de la conducta interna *selector de preferencias consumatorias* está fuertemente ligada al proceso competitivo que tiene lugar entre los REACs creados por las conductas elementales específicas que la integran.

La condición de una conducta elemental *selector de preferencias consumatorias* es satisfecha cuando en el nivel Congruentes Propio/Extero/Drive ha sido creado el elemento solución C_i o ha sido actualizado su valor de certidumbre A_i^C , y para cualquiera de estos casos el valor A_i^C supera el umbral θ^C establecido. Cuando este hecho ha tenido lugar, un REAC es creado describiendo las características de la conducta elemental activada. Entre los REACs inscritos en un mismo ciclo se establece un proceso de competencia del tipo *the winner takes all* para determinar el REAC ganador de la competencia. La acción final de una conducta elemental de este tipo consiste en la creación del elemento solución D_i en el nivel Drive del pizarrón. La figura 3.18 muestra la estructura de una conducta elemental *selector de preferencias consumatorias*. Para la conducta elemental por defecto k , se asume que la condición $A_k^C > \theta^C$ siempre es satisfecha. El valor de O_i^D es determinado durante la competencia. La estructura de un REAC para este tipo de conducta puede ser apreciada en la tabla 3.9.

Parámetros:	
elemento solución condición:	C_i
valor de certidumbre del elemento condición:	O_i^C
elemento solución acción:	D_i
valor de certidumbre actual del elemento acción:	O_i^D
fuerzas de apareo:	F_{ij}^C
Condición:	
nivel-pizarrón =	Congruentes Propio/Extero/Drive AND
cert(C_i, A_i^C) AND	
$A_i^C > \theta^C$	
Acción 1:	
$O_i^C := A_i^C$ AND	
inscribir REAC _i	
Acción 2:	
nivel-pizarrón =	Drive AND
crear-elemento-sol(D_i, O_i^D)	

Figura 3.18. Estructura de una conducta elemental selector de preferencias consumatorias.

Atributo	Descripción
O_i^C	valor de certidumbre del elemento solución C_i
O_i^D	valor de certidumbre del elemento solución D_i (determinado durante la competencia)
F_{ij}^C	fuerzas de apareo ($F_{ij}^C < 0$ para $i \neq j$, $F_{ij}^C = 1.0$ para $i = j$)
A_i^D	nivel de activación de la conducta elemental (determinado durante la competencia)

Tabla 3.9.

Proceso competitivo entre conductas elementales selector de preferencias consumatorias

En el proceso competitivo sólo participan los REACs asociados a conductas elementales específicas (se excluye la conducta elemental por defecto). La competencia es del tipo *the winner takes all*. Durante todo el tiempo que dura el proceso competitivo, el nivel Drive del pizarrón permanece cerrado en lectura para los mecanismos de comunicación. Para estos mecanismos la lectura en este nivel es abierta en el momento en que a lo sumo sólo una conducta elemental específica persiste en mantener registrado sobre este nivel el elemento solución asociado a su acción. Cuando este hecho ha tenido lugar, entonces el proceso competitivo habrá concluido. El mecanismo para seleccionar la conducta elemental ganadora es el siguiente:

1. Para cada REAC específico inscrito se calcula el nivel de activación (A_i^D) utilizando la siguiente expresión:

$$A_i^D = O_i^C + \sum_j F_{a_{ij}}^C * O_j^C \quad (3.8)$$

2. Para cada REAC específico inscrito se calcula el valor de certidumbre (O_i^D) del elemento solución que la conducta asociada pretende crear. La expresión utilizada para el cálculo es la siguiente:

$$O_i^D = \begin{cases} A_i^D & \text{si } A_i^D > 0 \\ 0 & \text{en otro caso} \end{cases} \quad (3.9)$$

3. Si ningún REAC cumple con la condición $O_i^D > 0$, entonces no hay conducta ganadora, por lo que la competencia finaliza sin que ninguna conducta específica haya ejecutado su acción final sobre el nivel Drive del pizarrón en esta etapa. Ir al paso 4.

Si sólo un REAC cumple con la condición $O_i^D \neq 0$, entonces la conducta elemental asociada a dicho REAC será la ganadora, por lo que ésta ejecutará su acción final sobre el nivel Drive del pizarrón, y la competencia finaliza. Ir al paso 4.

Si más de un REAC cumple con la condición $O_i^D \neq 0$, entonces todos ellos serán ganadores en esta etapa de la competencia, por lo que las conductas elementales asociadas a éstos ejecutarán su acción final sobre el nivel Drive del pizarrón. La competencia continúa hasta que sólo un REAC sea proclamado ganador. Regresar al paso 1.

4. La conducta elemental por defecto ejecuta su acción final sobre el nivel Drive del pizarrón, la cual consiste en la creación del elemento solución D_k con certidumbre $O_k^D = O_k^C$. El nivel Drive es abierto en lectura para los mecanismos de comunicación.

Al inicio de cada etapa de la competencia los valores O_i^C son actualizados por la actividad del comportamiento *selector de preferencias consumatorias*.

3.3.5. El mecanismo receptor

El mecanismo *receptor* registra en el pizarrón del nodo motivacional las señales que arriban a éste provenientes del nodo cognitivo. Estas señales son elementos solución generados por conductas internas del nodo cognitivo, y requeridos por conductas internas del nodo motivacional para satisfacer sus premisas.

El nivel de apareo del mecanismo *receptor* es el buzón de mensajes de entrada del nodo (BME). El mecanismo *receptor* no requiere de la creación de REACs. Ante cada señal inscrita en el BME, éste se dispara efectuando el registro final de la señal en el nivel adecuado del pizarrón. Para la estructura actual que exhibe la red de nodos pizarrón, todos los mensajes que arriban al BME del nodo motivacional proceden del nodo cognitivo, y son finalmente registrados en el nivel Percepciones Externas del pizarrón del nodo motivacional. En la figura 3.19 se presenta la

estructura del mecanismo *receptor*. La estructura del BME del nodo motivacional es similar a la del nodo cognitivo, mostrada en la tabla 3.7.

<p>Parámetros: elemento solución condición: B_i elemento solución acción: S_i</p> <p>Condición: nivel-apareo = BME AND $\exists B_i$ AND $\text{cert}(B_i, O_i^B)$</p> <p>Acción: nivel-pizarrón = Percepciones Externas AND $O_i^S := O_i^B$ AND crear-elemento-sol(S_i, O_i^S)</p>

Figura 3.19. Estructura del mecanismo receptor.

3.3.6. El mecanismo transmisor

La función del mecanismo *transmisor* es enviar al nodo cognitivo aquellos elementos solución generados en el pizarrón del nodo motivacional que son requeridos por conductas internas del nodo cognitivo para satisfacer sus premisas. Para la estructura actual que exhibe la red de nodos pizarrón, la condición del mecanismo *transmisor* es satisfecha cuando el nivel Drive es abierto en lectura para comportamientos de comunicación y en éste ha sido creado el elemento solución D_i con certidumbre O_i^D , para algún valor de i . El mecanismo *transmisor* no requiere de la creación de REACs, cada vez que su condición es satisfecha éste ejecuta su acción final, la cual consiste en registrar en el BME del nodo cognitivo el elemento solución responsable de su disparo. En la figura 3.20 se puede apreciar la estructura del comportamiento *transmisor*.

<p>Parámetros: elemento solución condición: D_i elemento solución acción: B_i comportamiento que generó el elemento solución: id-conducta nodo emisor: id-nodo-e nodo receptor: id-nodo-r</p> <p>Condición: nivel-pizarrón = Drive AND status-nivel = Abierto AND $\text{cert}(D_i, O_i^D)$ AND $O_i^D \neq 0$</p> <p>Acción: transmitir(id-nodo-r, D_i, O_i^D, id-conducta, id-nodo-e)</p>

Figura 3.20. Estructura del mecanismo transmisor.

3.4. Consideraciones finales

En este capítulo hemos presentado y discutido el modelo de selección de acciones que nosotros hemos propuesto: la red de conductas internas implementada con nodos pizarrón. Esta discusión se ha centrado fundamentalmente en las características estructurales y funcionales de cada uno de los nodos de la red, para lo cual se tomaron en cuenta los siguientes aspectos: (1) la estructura del pizarrón en términos de los diferentes niveles de análisis o abstracción que lo componen, así como las características de los elementos solución creados o modificados en cada uno de estos niveles; (2) la estructura y funcionamiento de las conductas internas que operan sobre el pizarrón del nodo; (3) la estructura y funcionamiento de los mecanismos de interfaz con los medios externo e interno; (4) el funcionamiento de los mecanismos de competencia entre conductas elementales de un mismo tipo; (5) el trabajo conjunto de todas las conductas internas y mecanismos de comunicación que operan a nivel del nodo; y (6) los procesos de comunicación que toman lugar entre los nodos.

Como se vio en el capítulo 2, tanto en las redes jerárquicas como en las redes distribuidas un nodo representa un comportamiento que la entidad puede ejecutar; mientras que en las redes conexionistas un comportamiento puede ser representado a través de toda una red neuronal. En la red de conductas internas, la representación de un comportamiento abarca dos columnas conductuales: una “columna conductual” completa del nodo cognitivo, incluyendo sus correspondientes actuador y acción motora, esta última no incluida en la columna conductual; y la correspondiente “columna conductual” del nodo motivacional.

En el capítulo 4 se discuten las propiedades que caracterizan la selección de acciones en la red de conductas internas, así como los procesos de aprendizaje asociativo que tienen lugar. Tanto la discusión de estas propiedades, como la explicación de los procesos de aprendizaje, permiten ilustrar de una forma más clara cómo es que se lleva a cabo el trabajo conjunto de las conductas internas en la selección de acciones.

CAPÍTULO 4

PROPIEDADES DE LA SELECCIÓN DE ACCIONES EN LA RED DE CONDUCTAS INTERNAS

El mecanismo de selección de acciones que implementa la red de conductas internas se caracteriza por las siguientes propiedades: (1) comportamiento motivado, (2) competencia a nivel motivacional de conductas externas incompatibles, (3) comportamiento reactivo, (4) direccionamiento por las metas, (5) no indecisión en la selección de acciones, (6) persistencia en la ejecución de una acción externa, (7) espontaneidad regulada en la ejecución de conductas externas, (8) saciedad, (9) cambios en el interés, (10) variación de la atención, (11) preactivación de conductas internas (selección de acciones hacia el medio interno de la entidad), (12) existencia de una conducta externa por defecto orientada a la búsqueda de una señal específica, (13) aprendizaje asociativo, y (14) inhibición de respuestas reflejas. A continuación pasaremos a discutir cada una de estas propiedades, indicando en cada caso cuáles conductas internas y procesos de la red de conductas internas generan dicha propiedad.

4.1. Comportamiento motivado

En la red de conductas internas la selección de acciones es orientada a la motivación. Es decir, si una acción externa puede contribuir a una fuerte motivación (necesidad interna), entonces esta acción externa tiene mucha más oportunidad de ser seleccionada que cualquier otra acción externa que contribuya a alguna motivación que por el momento es irrelevante.

Lo anterior puede ser fácilmente comprendido si analizamos nuevamente el esquema para la combinación de entradas externas e internas, utilizada por la conducta *congruencia propio/extero/drive* para calcular su nivel de activación. Este esquema de combinación ya fue discutido en el epígrafe 3.3.3 (expresión 3.7), no obstante, volveremos a replantear éste para facilitar el análisis que del mismo haremos a continuación.

$$A_i^C = O_i^{E*}(\alpha + \sum_j F a_{ij}^S * O_j^S) \quad (4.1)$$

Nótese que el término O_i^D ha sido eliminado en la expresión (4.1), ya que éste resulta irrelevante dentro de este análisis.

Como se puede apreciar en la expresión (4.1), para un valor de α igual a cero la entrada interna (O_i^E) y las entradas externas que pueden contribuir a la satisfacción de dicha motivación ($\sum_j F_{aj}^S * O_j^S$) interactúan de forma multiplicativa. Luego, si tenemos en cuenta que los modelos multiplicativos puros tienen una razón límite a valores muy pequeños de las entradas externas y/o de las entradas internas; entonces no importa cuán fuertes sean las entradas externas si el valor de la entrada interna es cercano a cero, ya que el valor de activación A_i^C también tenderá a cero. De esta forma, acciones externas que contribuyan a motivaciones muy débiles tendrán muy poca oportunidad de ser seleccionadas.

Al considerar un valor de α mayor que cero en la expresión 4.1, entonces al estado interno se le está atribuyendo mucho más peso o importancia que a las entradas externas; por lo que éste debe contribuir mucho más que sus correspondientes entradas externas al valor de activación (A_i^C) calculado. De esta forma, acciones que contribuyan a motivaciones muy fuertes tendrán mayor oportunidad de ser seleccionadas.

4.2. Competencia a nivel motivacional de conductas externas incompatibles

Si para el anterior esquema de combinación de entradas externas e internas (expresión 4.1) consideramos un valor de α igual a cero, entonces cualquier conducta elemental *congruencia/propio/extero/drive* podrá activarse, sólo si recibe conjuntamente entradas de los medios externo e interno. En este caso, por muy fuerte que sea el estado interno disponible para una conducta elemental de este tipo, ésta no podrá activarse si no existe una señal externa aprovechable.

Como fue establecido en el capítulo 3, diferentes tipos de conductas elementales asociadas a un mismo índice i establecen entre sí un mecanismo de encadenamiento de reglas hacia adelante (“columna conductual”). Por tal motivo, una conducta elemental *selector de preferencias consumatorias* puede satisfacer su condición sólo cuando la conducta *congruencia propio/extero/drive* encadenada a ésta ha ejecutado su acción, ya que ambas forman parte de la misma “columna conductual”. Entre todas las conductas elementales *selector de preferencias consumatorias* que han logrado satisfacer su condición se lleva a cabo un proceso de competencia para decidir cuál de éstas es la que liberará la señal *drive* enviada posteriormente al nivel Preferentes Consumatorios del pizarrón del nodo cognitivo. Cualquier conducta elemental *congruencia propio/extero/drive* con entradas desde el estado interno menos fuertes, pero con señales externas aprovechables, podrá activarse, y por lo tanto, la conducta elemental *selector de preferencias consumatorias* encadenada a ésta podrá participar en el proceso de competencia, e incluso ganar la competencia y activar una acción externa.

Cuando el valor de α es diferente de cero, entonces cualquier conducta elemental *congruencia/propio/extero/drive* podrá activarse si cumple al menos una de las dos siguientes

condiciones: $O_i^E \text{ OAND } O_i^S$ ($O_i^E * O_i^S \neq 0$), u O_i^E posee un valor muy elevado (esto es, la señal proveniente del estado interno asociado a esta conducta elemental es lo suficientemente fuerte). Al igual que en el caso anterior (cuando α es igual a cero), se establece un proceso competitivo entre todas las conductas elementales *selector de preferencias consumatorias* para decidir cuál de éstas liberará la señal *drive*. A diferencia del caso anterior, ahora si para una conducta elemental *congruencia propio/extero/drive* existe una señal muy fuerte desde el estado interno asociado (elemento solución O_i^E), entonces ésta podrá ser activada, aún en ausencia de entradas externas y por lo tanto, la conducta *selector de preferencias consumatorias* encadenada a la primera podrá participar en la competencia, e incluso ganar ésta.

El valor de α en la expresión 4.1 puede ser seleccionado de forma tal que, si una conducta elemental *congruencia propio/extero/drive* recibe una entrada suficientemente fuerte desde el estado interno y no recibe entradas externas, entonces la conducta *selector de preferencias consumatorias* encadenada a ésta podrá ganar la competencia si las restantes entradas externas están asociadas a conductas elementales con entradas internas muy débiles. Sin embargo, si existen conductas elementales *congruencia propio/extero/drive* con entradas internas menos fuertes (pero no débiles) y entradas externas aprovechables entonces, aunque la conducta elemental de este tipo que sólo recibe una fuerte entrada interna se haya activado, será muy probable que la conducta *selector de preferencias consumatorias* encadenada a esta última no resulte ganadora en el proceso competitivo.

4.3. Comportamiento reactivo

El comportamiento de una entidad es totalmente reactivo cuando las acciones externas ejecutadas por ésta dependen completamente de los estímulos ambientales observados. El comportamiento reactivo en la red de conductas internas puede ser modulado a partir del establecimiento de los siguientes tres parámetros: α , γ y ϕ .

Para un valor de α igual a cero en la expresión (4.1), entonces ambos tipos de entradas (externas e internas) son necesarias para que una conducta interna *congruencia propio/extero/drive* sea activada. De forma tal que, aunque exista una necesidad interna muy fuerte, si para ésta no existe su correspondiente señal externa, entonces la conducta interna asociada no podrá activarse, permitiendo que necesidades internas menos relevantes, pero para las cuales existen las señales externas disponibles, se activen. De esta forma, para valores de α iguales o cercanos a cero la conducta externa ejecutada por la entidad será menos motivada.

Un valor de ϕ igual a uno respeta el valor íntegro de la señal proveniente del nodo motivacional en la expresión (4.2), de forma tal que el comportamiento será motivado. Al reducirse el valor de ϕ , también se reduce la importancia que se le da a la motivación, haciendo el comportamiento observado en la entidad más reactivo. Si ϕ es igual a cero, no se tomará en cuenta el flujo de señales provenientes del nodo motivacional, por lo que la entidad no tendrá ningún conocimiento de sus necesidades internas, e independientemente de los estímulos (no reflejos) que perciba, siempre se ejecutará la conducta externa por defecto.

$$O_i^I := F_{a_i}^T * O_i^T * (\gamma + \phi * \sum_j F_{a_j} * O_j^C / O_i^T) \quad (4.2)$$

Como se puede apreciar en la expresión (4.2), para valores de γ mayores que cero, se le atribuye una mayor importancia a las entradas externas que a las provenientes del nodo motivacional, por lo que aún en ausencia total de señales provenientes de este nodo, una conducta interna *atención a preferencias* pudiera ser activada siempre que reciba su correspondiente señal externa.

4.4. Direccionamiento por las metas

La selección de acciones efectuada por la red de conductas internas responde a la satisfacción de las necesidades internas de la entidad. Se puede decir que las metas de la entidad van orientadas a la satisfacción de sus necesidades internas, por lo que la propiedad de direccionamiento por las metas emerge directamente del propio comportamiento motivado, explicado en el epígrafe 4.1. Por lo tanto, al exhibir la entidad un comportamiento motivado, también tendrá comportamiento direccionado por sus metas, las cuales van encaminadas a la satisfacción de sus necesidades internas.

4.5. No indecisión en la selección de acciones

El conjunto de todos los procesos competitivos que se efectúan en la red de conductas internas, así como la existencia de una conducta externa por defecto, evitan que haya indecisión en la selección de acciones. La competencia del tipo *the winner takes all*, que se efectúa entre las conductas elementales *selector de preferencias consumatorias* provoca que no haya más de una señal que se envíe al nivel Preferentes Consumatorios del pizarrón del nodo cognitivo. A su vez, la competencia efectuada entre las conductas elementales *selección de conductas externas* sólo produce un ganador, por lo que la red de conductas internas asegura que no habrá más de un elemento solución en el nivel Acciones del pizarrón del nodo cognitivo en cada ciclo de ejecución. Dada esta situación, en todo momento sólo existirá una única acción externa que podrá ejecutar la entidad, aunque esta pueda ser la conducta externa por defecto.

4.6. Persistencia en la ejecución de una acción externa

La competencia que se efectúa en el nodo motivacional entre las conductas elementales *selector de preferencias consumatorias* asegura que si hay una necesidad interna imperiosa, entonces no se ejecute una acción externa que no contribuya a la satisfacción de esa necesidad. En este hecho también incide la retroalimentación que es efectuada en el nodo cognitivo por la conducta interna *persistencia perceptual*, la cual toma para su competencia elementos solución previamente inscritos en el nivel Congruentes Drive/Percepción por la conducta interna *atención a preferencias*. Esto es, una conducta elemental *persistencia perceptual* tendrá más posibilidades

de ganar la competencia en un instante $t+1$ si también resultó ganadora en el instante t , y si había un elemento solución inscrito anteriormente en el nivel Congruentes Drive/Percepción correspondiente a su columna conductual. De esta forma, una vez que una acción externa ha sido seleccionada, ésta debe ser ejecutada persistentemente sin que otras acciones, irrelevantes por el momento, perturben la ejecución de la misma.

4.7. Espontaneidad regulada en la ejecución de conductas externas

La espontaneidad se da cuando se ejecuta un comportamiento en la ausencia completa del estímulo interno que debía generar éste. Si el parámetro γ es igual a cero, no habrá espontaneidad, ya que se necesitará que para una señal externa en el nivel Persistencia Perceptual exista su correspondiente señal motivacional en el nivel Preferentes Consumatorios. Al incrementarse el parámetro γ , se incrementará a su vez la espontaneidad en la ejecución de las conductas externas, siendo esta propiedad regulable en la red de conductas internas.

4.8. Saciedad

Cuando una conducta externa capaz de satisfacer un estado interno predominante ha sido ejecutada persistentemente, entonces dicho estado interno será reducido lo suficiente (no necesariamente hasta su valor mínimo, sino en dependencia del umbral θ^C y de la fuerza de la señal externa correspondiente a dicho estado interno), provocando la saciedad de la entidad, la cual es observable en el cambio de interés de la misma.

4.9. Cambios en el interés

Cuando la ejecución de una conducta externa satisface el estado interno de la entidad que motivó esa conducta, hay un decremento en la fuerza de dicho estado interno. Por lo tanto, una vez que el estado interno ha sido reducido lo suficiente, entonces la conducta elemental *selector de preferencias consumatorias* no podrá ganar la competencia en el nodo motivacional, por lo que la entidad manifestará un cambio en su interés, ejecutando la conducta externa más adecuada al medio interno actual. Si no hay necesidad interna que gane la competencia en el nodo motivacional, entonces se ejecutará la conducta externa por defecto hasta que haya alguna necesidad interna, lo cual provocará nuevamente un cambio en el interés de la entidad.

4.10. Variación de la atención

Los etólogos denotan la variación de la atención como la menor importancia que un animal otorga al peligro (por ejemplo, a un depredador) cuando el animal posee un estado de motivación extremo (por ejemplo, el animal tiene mucha hambre) (McFarland, 1981). Esta propiedad emerge directamente de la competencia que se efectúa a nivel motivacional. La importancia que la entidad le asigne al peligro será a su vez función de un estado interno existente, por ejemplo, seguridad. Este estado interno entrará a la competencia, la cual podría ser ganada por algún otro estado interno extremo (por ejemplo, mucha hambre o mucha sed), aún en la presencia de peligro para la entidad. Experimentos en los que se puede apreciar esta propiedad son presentados en los epígrafes 6.6, 6.7 y 6.8.2.

4.11. Preactivación de conductas internas

A nivel de la conducta interna *congruencia propio/externo/drive* del nodo motivacional se lleva a cabo la combinación de señales externas e internas para determinar la acción que debe ser ejecutada por la entidad. Estas señales son registradas en los niveles Percepciones Externas y Percepciones Internas, respectivamente, del nodo motivacional. Los elementos solución registrados en el nivel Percepciones Externas son originados por estímulos externos inicialmente proyectados en el nivel Percepciones Externas y posteriormente almacenados en el nivel Persistentes Perceptuales, del nodo cognitivo. Por otra parte, los elementos solución registrados en el nivel Percepciones Internas del nodo motivacional representan los valores de los estados internos capaces de satisfacer parte de las condiciones de determinadas conductas elementales *congruencia propio/externo/drive* que operan sobre este nivel.

Caso 1: $\alpha=0$

Cuando α es igual a cero, entonces la conducta elemental *congruencia propio/externo/drive* podrá activarse sólo cuando coinciden las señales internas y externas ($O_i^E \neq 0$ y $\sum_j F_{a_{ij}}^S * O_j^S \neq 0$) asociadas a ésta. Si no está presente el estado interno apropiado, entonces la señal externa por sí sola no es suficiente para evocar la acción motora, ya que la conducta interna *atención a preferencias* es activada sólo cuando coinciden sus correspondientes señal externa (elemento solución en el nivel Persistentes Perceptuales del nodo cognitivo) y señal *drive* (elemento solución en el nivel Preferentes Consumatorios del nodo cognitivo). Por otra parte, aunque exista una señal interna muy fuerte (correspondiente a una imperiosa necesidad de la entidad) si no está presente la señal externa aprovechable, la conducta elemental *congruencia propio/externo/drive* tampoco va a disparar. Luego, cuando $\alpha=0$, ambos factores (señal externa y señal interna) son necesarios para que se active la conducta interna *congruencia propio/externo/drive*. Fisiológicamente esto puede interpretarse al pensar que las entradas provenientes de los estados internos sensibilizan a esta conducta interna con las señales provenientes del medio externo.

Caso 2: $\alpha \neq 0$

Cuando α es diferente de cero, entonces al estado interno se le está otorgando un poco más de peso (o importancia) que a las entradas externas. De forma tal que, aún en ausencia total de entradas externas, la conducta elemental *congruencia propio/externo/drive* podría ser activada para un valor muy fuerte de las señales internas (cuando el valor de α es cercano a 1). Así, cuando la necesidad interna es extremadamente alta y no existen señales externas aprovechables, la conducta elemental *congruencia propio/externo/drive* podría activarse, y entonces provocar una especie de “preactivación” de las condiciones de las conductas elementales *atención a preferencias* (en particular, sólo una de estas conductas elementales será la que mayor preactivación reciba). Este es un ejemplo típico de una conducta interna cuyas acciones son dirigidas al medio interno. Este mecanismo constituye la base subyacente de la conducta exploratoria orientada hacia la búsqueda de un objetivo específico, explicada más adelante. Cuando existe una saciedad plena de la necesidad interna que provocó la activación de la conducta elemental *congruencia propio/externo/drive* ($O_i^E = 0$), entonces el nivel de activación de esta conducta elemental tenderá a cero ($A_i^C \approx 0$), independientemente de que hayan señales externas aprovechables. Para niveles de activación cero o muy cercanos a cero la conducta elemental no será activada. De esta forma la conducta externa observada siempre será adaptativa.

4.12. Existencia de una conducta externa “por defecto” orientada a la búsqueda de una señal específica

Si en el esquema de combinación de entradas externas e internas (expresión 4.1) es considerado un valor de α igual a cero, entonces la conducta elemental *selector de preferencias consumatorias* que gane la competencia liberará una señal *drive* que será aprovechable por todas las conductas elementales *atención a preferencias* que operan en el nivel Preferentes Consumatorios del nodo cognitivo. Sin embargo, al considerar en este esquema de combinación un valor de α diferente de cero, entonces pudiera ocurrir que la conducta elemental *selector de preferencias consumatorias* que gane la competencia sea aquella encadenada a una conducta elemental *congruencia propio/externo/drive* que esté recibiendo una entrada interna suficientemente fuerte, aunque para ésta no hayan señales externas aprovechables.

De ser ésta la conducta elemental ganadora, entonces no será una señal *drive* lo que recibirán las conductas elementales que operan en el nivel Preferentes Consumatorios del pizarrón del nodo cognitivo. En este caso, el elemento solución creado por la conducta elemental *selector de preferencias consumatorias* ganadora constituirá una señal de preactivación para aquellas conductas elementales *atención a preferencias* que en su condición requieran de dicho elemento solución. Esta señal de preactivación por sí sola no es capaz de activar a ninguna de las conductas elementales *atención a preferencias*, aunque en realidad “una conducta elemental *selector de preferencias consumatorias* haya disparado”. Esta preactivación, desde un punto de vista informático, consiste en que la condición de la conducta elemental está “a medio cumplir”, es decir, se cuenta con el elemento solución requerido en el nivel Preferentes Consumatorios, pero

falta el elemento solución requerido en el nivel Persistentes Perceptuales. En el momento en que este último elemento solución sea creado, la conducta elemental *atención a preferencias* será activada. Esta preactivación se diferencia de la activación en el hecho de que la conducta elemental *atención a preferencias* requiere de un AND entre sus correspondientes elementos solución condición generados en los niveles Preferentes Consumatorios y Persistentes Perceptuales. Luego, teniendo en cuenta lo anterior, no es suficiente que una conducta elemental *selector de preferencias consumatorias* haya disparado para asumir que una conducta externa será evocada como consecuencia de ese disparo. Una conducta externa será evocada sólo cuando una conducta elemental *atención a preferencias* dispare (la única que habrá recibido conjuntamente una señal externa y una señal *drive*).

Cuando ninguna de las conductas elementales *atención a preferencias* logra ser activada y existen fuertes necesidades internas, entonces debe ser liberada una conducta externa por defecto (en el caso de un agente autónomo esta conducta externa por defecto pudiera verse como una conducta exploratoria) orientada a la búsqueda de las señales externas necesarias. Para este fin, en la red de conductas internas que implementan los nodos cognitivo y motivacional se han considerado conductas elementales por defecto, encadenadas entre sí y asociadas a la conducta externa orientada a la búsqueda de una señal específica (o sea, una especie de “columna conductual por defecto”). Dicha conducta externa siempre será activada, cuando después de un proceso competitivo a nivel de las conductas elementales específicas *selector de preferencias consumatorias*, no ha sido liberada una señal *drive* hacia el nivel Preferentes Consumatorios del pizarrón del nodo cognitivo. Sin embargo, cuando una señal *drive* ha sido liberada, entonces una conducta elemental *selección de conductas externas* evocará una acción motora y por lo tanto, la acción de la conducta externa por defecto será inactivada. En la red de conductas internas esto último ha sido implementado introduciendo la conducta elemental por defecto *inhibición de respuestas reflejas*, la cual bloqueará la creación del elemento solución H_k en el nivel Acciones Potenciales, siempre que al menos una conducta elemental específica *atención a preferencias* haya sido activada.

La existencia de una conducta externa orientada a la búsqueda de una señal externa específica en la red de conductas internas se justifica por el hecho de que la conducta exploratoria en los animales no siempre es inespecífica, sino que se encuentra bastante orientada a la búsqueda de una señal específica que satisfaga su estado interno. Es decir, un animal realiza la búsqueda "agudizando" la atención en aquellas señales que puedan satisfacer su estado interno. Por ejemplo, un lobo con hambre explora atento a los olores de la comida, mientras que si tiene sed explora atento a ruidos (por ejemplo, borboteos o caída de agua) que indiquen la presencia del agua. Para lograr este efecto de “exploración orientada a la búsqueda de una señal específica”, en la red de conductas internas hemos considerado que un estado interno suficientemente alto puede activar la conducta elemental *selector de preferencias consumatorias* asociada a éste, y entonces provocar una especie de “preactivación” de las conductas elementales *atención a preferencias* asociadas con este estado interno. De esta forma, al arribar la señal externa al nivel Persistentes Perceptuales, el proceso competitivo ocurrirá solo a nivel cognitivo (nodo cognitivo), mientras que sin la preactivación el proceso competitivo será primero a nivel motivacional y después a nivel cognitivo. Es decir, sin la preactivación se requieren de dos competencias, lo cual incrementa el tiempo de respuesta al requerirse de más pasos de procesamiento desde el momento en el que el estímulo externo llega al nivel Persistentes Perceptuales del nodo cognitivo.

Luego, la existencia de este proceso de preactivación constituye la base subyacente para la existencia de la conducta “atención dirigida a un evento que no está ocurriendo en este momento en el medio externo”. Una simple comprobación experimental de esa preactivación podría estar en la obtención de tiempos de reacción más cortos en respuesta a una señal particular que a cualquier otra señal, si la necesidad imperiosa existente es satisfecha por la existencia de dicha señal particular. Es decir, de esta forma la exploración tiene un objetivo determinado por los estados internos.

4.13. Aprendizaje asociativo

En la red de conductas internas se pueden apreciar dos formas básicas de aprendizaje presentes en el comportamiento animal: el condicionamiento clásico primario y el condicionamiento clásico secundario.

Para lograr los procesos de condicionamiento clásico antes mencionados, en la red de conductas internas hemos considerado que los estímulos perceptuales que inicialmente se proyectan sobre el nivel Percepciones Externas del pizarrón del nodo cognitivo pueden ser de dos tipos: estímulos incondicionados (denotados por S_i^I , $i = 1, 2, \dots$) y estímulos condicionados (denotados por S_i^C , $i = 1, 2, \dots$). De forma tal que, para cada señal S_i^C proyectada en el nivel Percepciones Externas, podrá ser creada una señal T_i^C en el nivel Persistentes Perceptuales y una señal I_i^C en el nivel Congruentes Drive/Percepción, siempre que hayan sido satisfechas las mismas condiciones requeridas en el caso de los estímulos S_j^I (denotados inicialmente como S_i).

4.13.1. Condicionamiento clásico primario

El condicionamiento clásico primario puede ser explicado en los siguientes términos: si un estímulo que inicialmente es neutro (es decir, no posee significado alguno para la entidad), se presenta antes de cada presentación de un estímulo incondicionado (EI), este hecho origina que el estímulo neutro se asocie con el estímulo incondicionado, por lo que ahora éste será capaz de provocar la misma respuesta provocada por el estímulo incondicionado. A este estímulo que inicialmente era neutro, ahora se le llama estímulo condicionado (EC) (Kandel, 1976 y Kandel, 1978).

Los eventos que originan el condicionamiento clásico primario en la red de conductas internas son los siguientes:

- Proyección del estímulo neutro en el nivel Percepciones Externas del nodo cognitivo (elemento solución S_i^C , $i = 1, 2, \dots$), llegando esta señal al nivel Persistentes Perceptuales de este mismo nodo (elemento solución T_i , $i = 1, 2, \dots$) y desde aquí propagándose hacia el nivel Percepciones Externas del nodo motivacional (vías inespecíficas), logrando activar muy poco a las conductas elementales *congruencia propio/extero/drive* que sobre este nivel operan (aún existiendo una fuerte entrada interna para éstas); ya que las fuerzas de apareo de estas conductas elementales, aunque modificables, son inicialmente muy bajas, casi ceros. La señal T_i , $i = 1, 2, \dots$, almacenada en el nivel Persistentes Perceptuales del nodo cognitivo es requerida por la conductas elementales *atención a preferencias* que operan sobre este nivel para

satisfacer sus premisas. Sin embargo, esta señal inicialmente no logra activar a ninguna de estas conductas, pues estas últimas necesitan además de la existencia de la señal C_i , $i = 1, 2, \dots$, en el nivel Preferentes Consumatorios del nodo cognitivo, señal ésta que proviene del nivel Drive del nodo motivacional.

- Persistencia de la señal T_i en el nivel Persistentes Perceptuales del nodo cognitivo (en términos neuronales, este hecho se traduce en la reverberación del estímulo T_i en memoria a corto plazo) durante un lapso de tiempo determinado. Es necesario tener en cuenta que se está produciendo una competencia entre las diferentes conductas elementales *persistencia perceptual* que operan sobre el nivel Persistentes Perceptuales del nodo cognitivo. La conducta elemental que gane la competencia será la que haga persistir durante más tiempo sobre este nivel el elemento solución T_i especificado en su acción. Vamos a considerar, que los restantes eventos ocurren en paralelo al proceso de competencia que se lleva a cabo entre las conductas elementales *persistencia perceptual* o que ya alguna conducta elemental ganó la competencia.
- Proyección del estímulo incondicionado S_j^I en el nivel Percepciones Externas del nodo cognitivo, posteriormente a la presentación del estímulo neutro S_i^C , pero dentro del intervalo de tiempo en que este estímulo neutro persiste aún en el nivel Persistentes Perceptuales del nodo cognitivo (señal T_i).

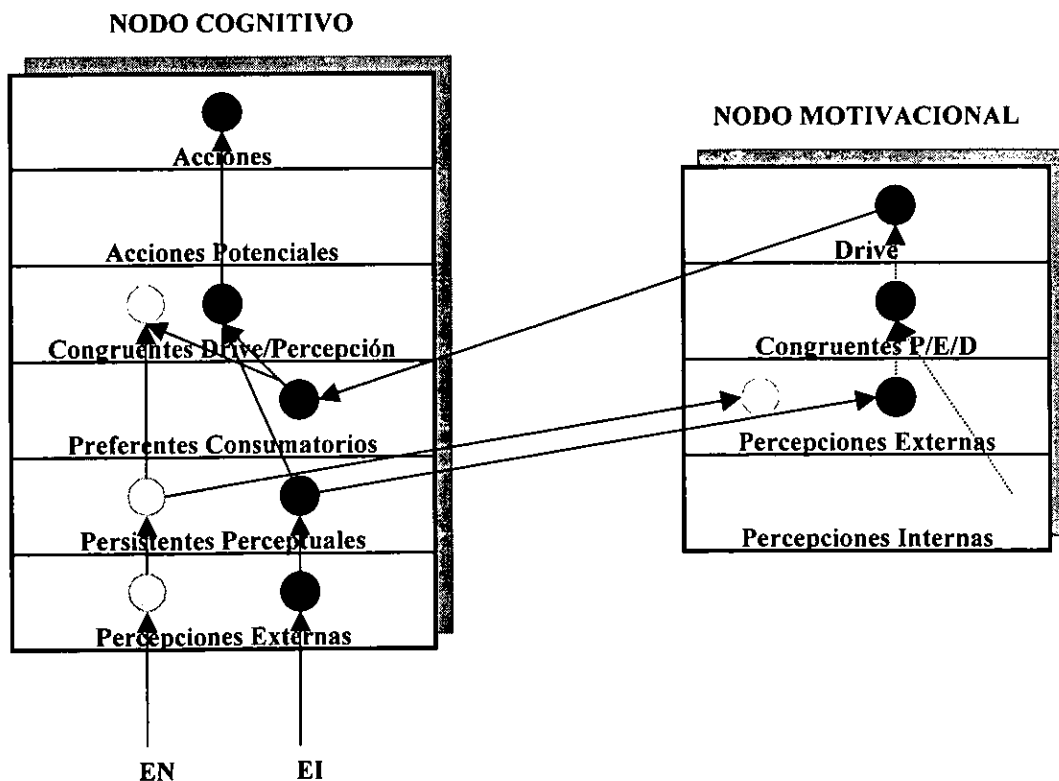


Figura 4.1. Trayectorias recorridas por el estímulo neutro (EN) y el estímulo incondicionado (EI) al inicio del proceso de condicionamiento. Los elementos solución generados a partir de la llegada del estímulo neutro son representados con círculos sólidos grises, mientras que los generados a partir de la llegada del estímulo incondicionado son representados con círculos sólidos negros. Las flechas representan las conductas elementales internas.

Una vez que un EI S_j^I ha sido proyectado sobre el nivel Percepciones Externas del nodo cognitivo, éste sigue exactamente la misma trayectoria que un estímulo en proceso de condicionamiento o ya condicionado, con la única diferencia de que las distintas conductas elementales que definen la trayectoria del estímulo S_j^I poseen desde un inicio valores muy elevados en sus fuerzas de apareo (parámetros $F_{a_{ij}}$). La trayectoria recorrida por el EI S_j^I es la siguiente: nivel Persistentes Perceptuales - nivel Percepciones Externas del nodo motivacional - nivel Congruentes Propio/Extero/Drive - nivel Drive - nivel Preferentes Consumatorios - nivel Congruentes Drive/Percepción - nivel Acciones. El segmento nivel Percepciones Externas - nivel Congruente Propio/Extero/Drive podrá ser ejecutado sólo si existen entradas internas asociadas a la señal externa S_j^I . El segmento nivel Congruentes Propio/Extero/Drive - nivel Drive podrá ser ejecutado sólo si la conducta elemental *selector de preferencias consumatorias* asociada a las señales S_j^I y E_j es la ganadora de la competencia que a este nivel se establece. Las trayectorias recorridas por el estímulo neutro y el estímulo incondicionado al inicio del proceso de condicionamiento pueden ser apreciadas en la figura 4.1.

Si se originó la señal desde el nivel Drive del nodo motivacional hacia el nivel Preferentes Consumatorios del nodo cognitivo, entonces ocurrirán los siguientes procesos de aprendizaje (los cuales se traducen en la modificación de las fuerzas de apareo de determinadas conductas elementales).

- **Aprendizaje del patrón de respuesta motora.** Este tipo de aprendizaje tiene lugar cuando alguna de las conductas elementales *selección de conductas externas* ha ejecutado su acción final sobre el nivel Acciones, dado que la conducta elemental *atención a preferencias*, con la cual la primera se encuentra encadenada ha ejecutado su acción final sobre el nivel Congruentes Drive/Percepción. La condición para que estas últimas conductas elementales ejecuten su acción es que reciban tanto señales específicas desde el nivel Persistentes Perceptuales del nodo cognitivo (elemento solución T_i^I), así como las señales inespecíficas que provienen del nivel Drive del nodo motivacional y son registradas en el nivel Preferentes Consumatorios del nodo cognitivo (elemento solución C_i). Cuando este hecho ha tenido lugar, entonces las fuerzas de apareo modificables de determinadas conductas elementales *selección de conductas externas* (específicamente, aquellas asociadas al estímulo en proceso de condicionamiento que aún está en el nivel Persistentes Perceptuales del nodo cognitivo) se ajustan (los $F_{a_{ij}}$ modificables inicialmente poseen valores muy bajos, cercanos a cero) aprendiéndose el patrón motor, el cual será evocado por estas conductas elementales por sí solas en una presentación posterior del estímulo S_i^C , el cual es ahora un estímulo en proceso de condicionamiento. En la figura 4.2 se puede apreciar la trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje del patrón de respuesta motora. En la arquitectura de la red de conductas internas, este tipo de aprendizaje puede ser visto como la “convergencia de columnas conductuales”.
- **Aprendizaje del significado biológico.** Este tipo de aprendizaje toma lugar a nivel del reforzamiento condicionado en las fuerzas de apareo modificables de determinadas conductas elementales congruencia propio/extero/drive. Dado que ha ganado la competencia la conducta elemental *selector de preferencias consumatorias*, asociada al EI S_j^I , y se están recibiendo además entradas desde el nivel Persistentes Perceptuales

del nodo cognitivo, correspondientes a algún estímulo en proceso de condicionamiento (elemento solución SiC), entonces se pueden ajustar las fuerzas de apareo de la conducta elemental congruencia propio/extero/drive capaz de establecer el segmento de trayectoria nivel Percepciones Externas - nivel Congruentes Propio/Extero/Drive para este estímulo en proceso de condicionamiento, de forma tal que el mismo comience a adquirir un significado específico para la entidad. En la figura 4.3 se muestra la trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje del significado biológico.

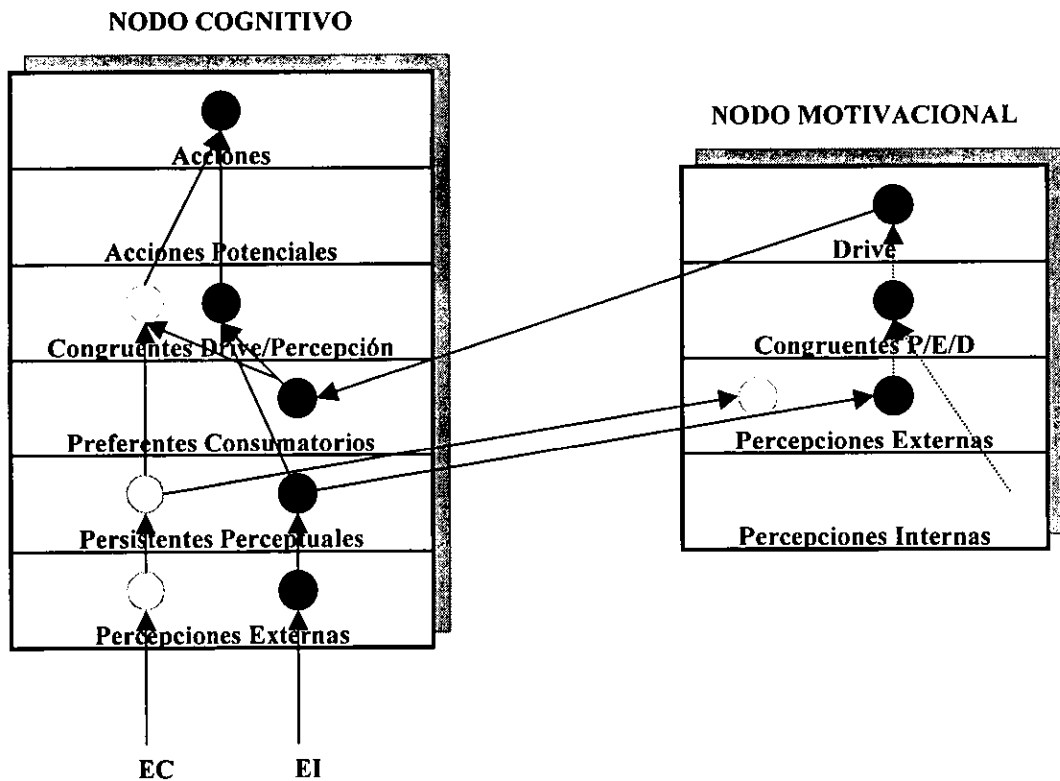


Figura 4.2. Trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje del patrón de respuesta motora. Se mantiene en esta representación la trayectoria recorrida por el estímulo incondicionado (EI).

- Aprendizaje a nivel motivacional** (aprendizaje de la señal drive). Este tipo de aprendizaje consiste en el ajuste de las fuerzas de apareo modificables de determinadas conductas elementales atención a preferencias. Al ejecutar su acción final la correspondiente conducta elemental atención a preferencias asociada al estímulo en proceso de condicionamiento (la única para la cual existen simultáneamente en ese momento una señal T_i asociada a un estímulo S_iC y una señal drive C_j aprovechable), entonces la fuerza de apareo F_{ijC} de la conducta elemental atención a preferencias asociada a la señal C_j es ajustada. En este punto es necesario aclarar lo siguiente: aunque las fuerzas de apareo F_{ijC} de las conductas elementales atención a preferencias asociadas a estímulos en proceso de condicionamiento son modificables, éstas poseen desde un inicio valores lo suficientemente grandes como para permitir la activación de todas aquellas conductas elementales para las cuales también existan señales T_i disponibles. El aprendizaje a nivel motivacional consiste precisamente en el reforzamiento de estos valores. En la figura 4.4 se ilustra la trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje a nivel motivacional.

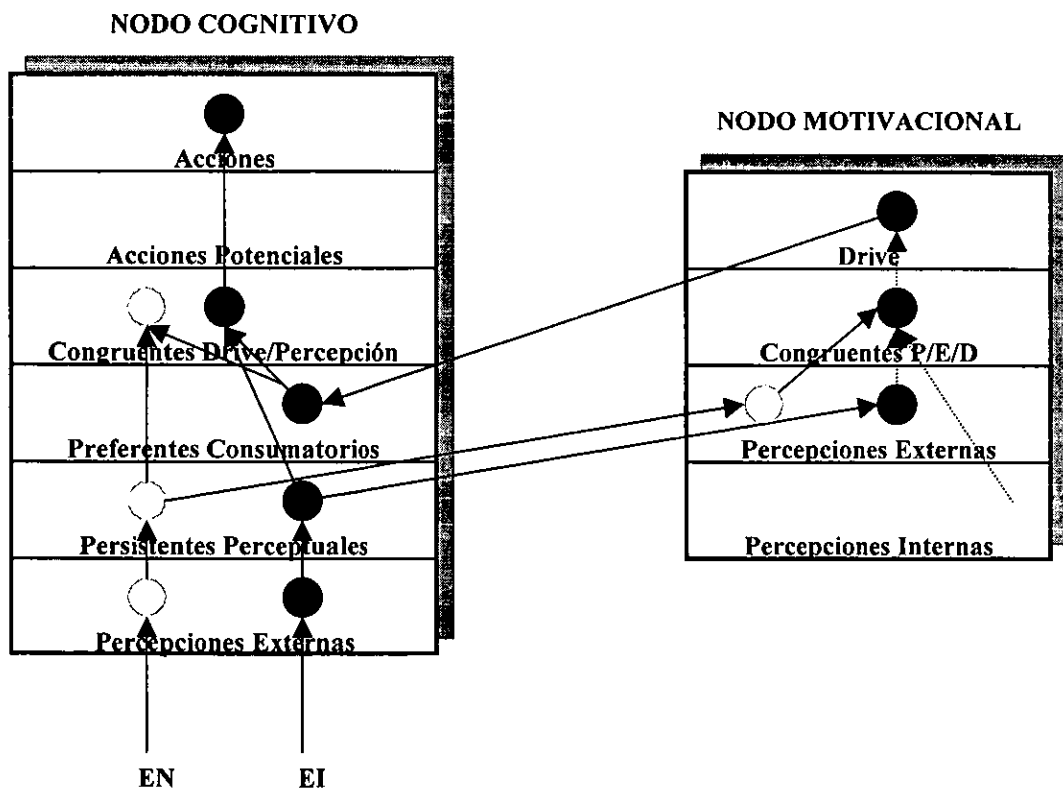


Figura 4.3. Trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje del significado biológico. Se mantiene en esta representación la trayectoria recorrida por el estímulo incondicionado (EI).

La regla para la modificación de las fuerzas de apareo utilizada en los tres tipos de aprendizaje antes descritos viene dada por la expresión (4.):

$$F_{ij}(t+1) = \begin{cases} (1 - \beta) * F_{ij}(t) + f_1(O_j^{in}(t)) * f_2(O_i^{out}(t)) * \lambda & \text{si } f_1(O_j^{in}(t)) > 0 \\ (1 - \mu) * F_{ij}(t) & \text{en otro caso} \end{cases} \quad (4.3)$$

donde: F_{ij} es la fuerza de apareo modificable de la conducta elemental i respecto al elemento solución j , O_j^{in} es el valor de certidumbre asociado al elemento solución condición j , O_i^{out} es el valor de certidumbre asociado al elemento solución acción i , β es un parámetro que determina la proporción que se mantiene de la fuerza de apareo correspondiente al instante anterior ($0 \leq \beta \leq 1$), λ es un factor para regular la velocidad del condicionamiento, y μ determina la velocidad de la pérdida del condicionamiento ($0 \leq \mu \leq 1$). La primera parte de la fórmula 4.3 regula el condicionamiento, mientras que la segunda parte regula la pérdida del condicionamiento. Para valores de β igual a μ , las dos partes de esta ecuación se podrían reducir a la primera.

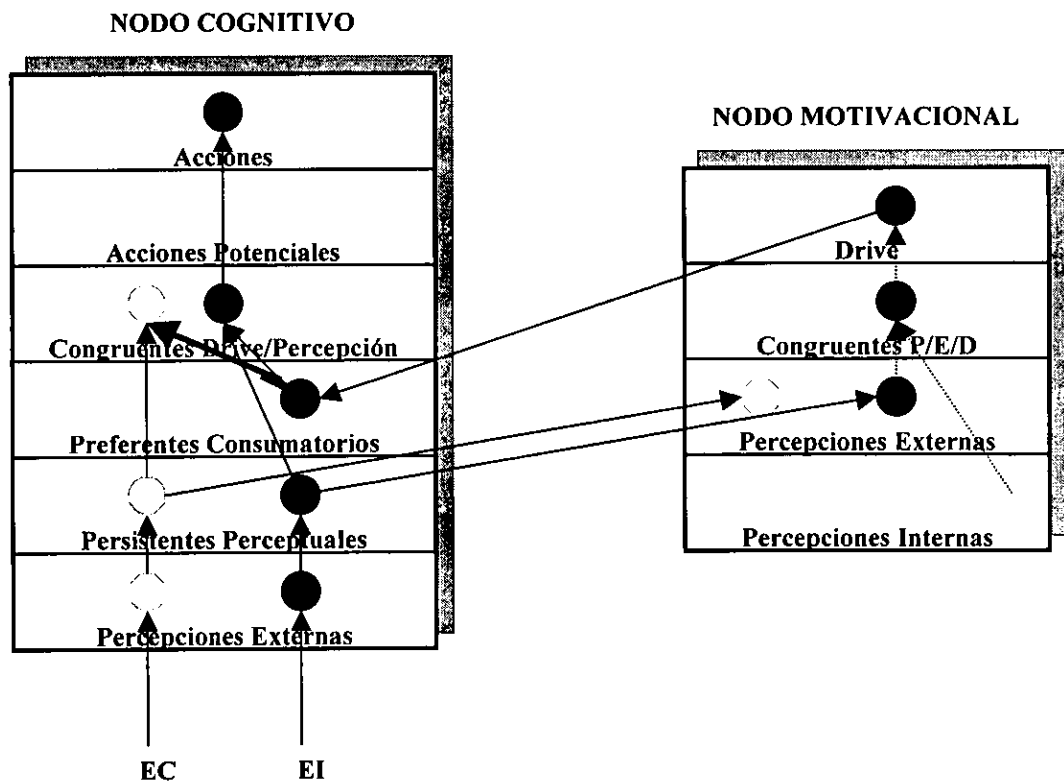


Figura 4.4. Trayectoria recorrida por el estímulo en proceso de condicionamiento (EC) una vez efectuado el aprendizaje a nivel motivacional. Se mantiene en esta representación la trayectoria recorrida por el estímulo incondicionado (EI). Nótese en la figura que este tipo de aprendizaje consiste en el reforzamiento de una de las fuerza de apareo de la conducta elemental *atención a preferencias*, la correspondiente al elemento solución condición generado en el nivel Persistentes Perceptuales.

Una vez que un estímulo neutro se ha condicionado, éste podrá evocar una conducta externa por sí mismo si existe una suficiente entrada de estado interno a la conducta elemental *congruencia propio/externo/drive* correspondiente a dicho estímulo. Los eventos que originan esta respuesta son los siguientes:

Proyección del estímulo condicionado en el nivel Percepciones Externas del nodo cognitivo (elemento solución S_i^C , $i = 1, 2, \dots$), llegando esta señal al nivel Persistentes Perceptuales de este mismo nodo (elemento solución T_i , $i = 1, 2, \dots$) y desde aquí propagándose hacia el nivel Percepciones Externas del nodo motivacional (vías inespecíficas), contribuyendo ahora fuertemente a la activación de la conducta elemental *congruencia propio/externo/drive* con la cual se asoció durante el proceso de condicionamiento clásico (o sea, ya ocurrió un aprendizaje del significado biológico a nivel de reforzamiento condicionado), de igual forma que lo haría un EI. De modo que, si existe suficiente entrada interna para la conducta elemental *congruencia propio/externo/drive*, entonces la conducta elemental *selector de preferencias consumatorias* asociada a ésta podría ganar la competencia, produciéndose la señal drive hacia el nivel Preferentes Consumatorios del nodo cognitivo. La señal T_i ($i = 1, 2, \dots$) almacenada en el nivel Persistentes Perceptuales del nodo cognitivo es requerida por la conductas elementales *atención a preferencias* que operan sobre este nivel para satisfacer sus premisas. Sin embargo, esta señal inicialmente no logra activar a ninguna de estas conductas, ya que estas últimas también requieren de la existencia de la señal C_i ($i = 1, 2, \dots$) en el nivel Preferentes Consumatorios del nodo cognitivo.

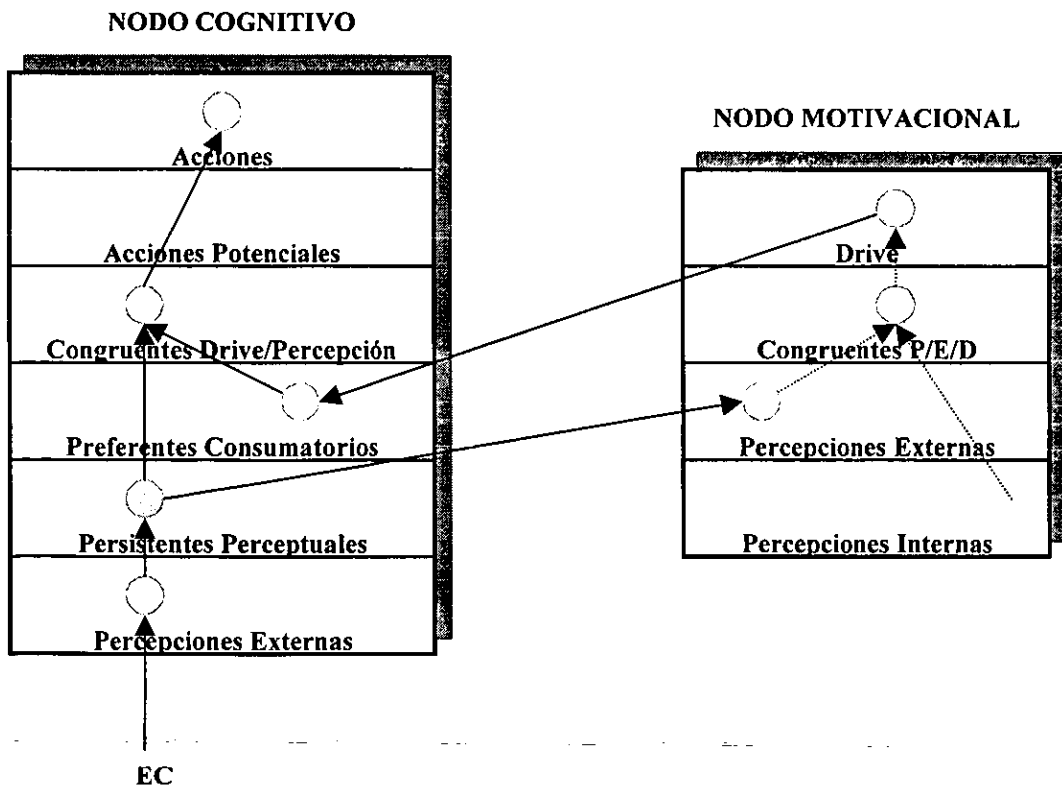


Figura 4.5. Trayectoria recorrida por el estímulo condicionado (EC) una vez efectuados los tres tipos de aprendizajes.

Si alguna conducta elemental *selector de preferencias consumatorias* resultó ganadora de la competencia, entonces se producirá una señal drive hacia el nivel Preferentes Consumatorios del nodo cognitivo, la cual llegará inespecíficamente a varias de las conductas elementales *atención a preferencias* que operan sobre este nivel. Sin embargo, esta señal llegará más fuertemente a la conducta elemental *atención a preferencias* asociada al estímulo condicionado (debido al aprendizaje a nivel motivacional que ya ocurrió) y más débil a las restantes conductas elementales. La existencia de vías de retroalimentación (señal I_i asociada a cada conducta elemental *persistencia perceptual*) contribuyen a que la conducta elemental *persistencia perceptual* asociada al estímulo condicionado gane la competencia sostenida con las restantes conductas elementales de este tipo. Esto hecho pudiera interpretarse como “fijar la atención”, pues de esta forma, será principalmente esta conducta elemental la que hará persistir en el nivel Persistentes Perceptuales el elemento solución especificado en su acción (señal T_i). Este “*attention feedback*” está mediatizado por la motivación. La señal drive registrada en el nivel Preferentes Consumatorios (señal C_j), conjuntamente con la señal registrada en el nivel Persistentes Perceptuales (señal T_i) provocan que la conducta elemental *atención a preferencias* asociada al EC se active, y como la conducta elemental *selección de conductas externas* encadenada a esta última ya se había aprendido el patrón motor (aprendizaje motor), entonces se evocará la respuesta motora. En la figura 5 se puede apreciar la trayectoria completa recorrida por el estímulo condicionado, una vez efectuados los tres tipos de condicionamiento.

Para que un EI pueda condicionar a un estímulo neutro tiene que existir suficiente entrada de estado interno a la conducta elemental *congruencia propio/extero/drive* asociada al EI. De igual forma, para que un EC (un estímulo previamente condicionado) pueda evocar la conducta externa a la cual fue condicionado, debe existir una suficiente entrada de estado interno a la conducta elemental *congruencia propio/extero/drive* correspondiente.

4.13.2. Condicionamiento clásico secundario

El condicionamiento clásico secundario consiste en lo siguiente: si un estímulo que inicialmente es neutro (es decir, carece de significado alguno para la entidad) se presenta antes de cada presentación de un estímulo que ya se condicionó (EC), este hecho originará que el estímulo neutro se convierta en un estímulo condicionado, por lo cual éste podrá evocar la respuesta que antes evocaba el EC. Es decir, en el condicionamiento secundario el papel de estímulo incondicionado (EI) lo juega el estímulo previamente condicionado (EC) (Kandel, 1976 y Kandel, 1978).

En la red de conductas internas los eventos que originan el proceso de condicionamiento clásico secundario son los mismos que ya fueron explicados en el condicionamiento clásico primario. La principal diferencia entre ambos procesos de condicionamiento puede ser explicada en los siguientes términos. En el condicionamiento clásico primario, el estímulo que juega el papel de condicionador es por naturaleza un EI, capaz de evocar una respuesta adaptativa (mediada por un estado interno) sin que haya tenido lugar un previo proceso de aprendizaje (condicionamiento). Por lo tanto, para este tipo de estímulo se han establecido desde un inicio los segmentos de trayectoria por donde el mismo ha de propagarse, los cuales están dados por los elevados valores que poseen las fuerzas de apareo de las conductas elementales asociadas a este tipo de estímulo. En el condicionamiento secundario, el estímulo que juega el papel de condicionador es un EC, o sea, un estímulo que inicialmente era neutro pero que fue condicionado por un EI en un previo

proceso de condicionamiento clásico primario. Aunque para este EC también se han establecido ya los segmentos de trayectoria por donde ha de propagarse, éstos no estaban preestablecidos desde un inicio, sino que fueron condicionados durante el proceso de condicionamiento clásico primario (aprendizaje del patrón de respuesta motora, aprendizaje del significado biológico y aprendizaje a nivel motivacional).

4.13.3. Principales propiedades de los procesos de condicionamiento clásico primario y secundario en la red de conductas internas

El intervalo inter-estímulo es el intervalo de tiempo existente entre la proyección de un estímulo neutro y de un estímulo incondicionado en el nivel Percepciones Externas del pizarrón del nodo cognitivo. La magnitud del intervalo inter-estímulo ejerce un considerable efecto en el proceso de competencia que se lleva a cabo entre las conductas elementales *persistencia perceptual*, para decidir cuál de ellas hará persistir durante más tiempo el elemento solución especificado en su parte acción sobre el nivel Persistentes Perceptuales.

Como lo han demostrado los datos experimentales (Sutton y Barto, 1981) la relación entre el tamaño del intervalo inter-estímulo y la fuerza del condicionamiento es representada gráficamente mediante una U invertida definida en el intervalo $[0, T]$. Según esta U invertida, para valores de intervalos inter-estímulos cero la fuerza del condicionamiento es cero, ocurriendo el fenómeno conocido como “bloqueo”; para valores del intervalo inter-estímulo cercanos a cero el condicionamiento es pobre; para valores del intervalo inter-estímulo muy altos (muy cercanos a T) el condicionamiento también es muy pobre, ocurriendo el fenómeno de “caída de la actividad del estímulo en el tiempo”; mientras que para valores del intervalo inter-estímulo intermedios ocurre el condicionamiento más fuerte Otra importante propiedad es la de dominancia (*overshadowing*) la cual ocurre cuando a un EI lo preceden varios EC.

Los procesos de condicionamiento clásico primario y secundario implementados en la red de conductas internas exhiben las propiedades antes descritas. A continuación vamos a describir bajo qué condiciones se manifiesta cada una de estas propiedades en la red de conductas internas.

Bloqueo

La propiedad de bloqueo en la red de conductas internas consiste en lo siguiente: si un estímulo neutro es presentado sincrónicamente con un EI o con un EC (un estímulo previamente condicionado), entonces no ocurrirá condicionamiento del estímulo neutro. Es decir, la representación del estímulo neutro en el nivel Persistentes Perceptuales es bloqueada por la representación del EI (o del EC) en este mismo nivel. Para comprender esta propiedad de bloqueo es necesario tener en cuenta que al inicio de la competencia ya el estímulo neutro, por muy intenso que éste sea, comienza con desventaja con respecto al EI (o al EC) debido a la fuerte retroalimentación positiva que llega a la conducta elemental *persistencia perceptual* asociada a este último, vía nivel Persistentes Perceptuales - nivel Congruentes Propio/Extero/Drive - nivel Drive - nivel Preferentes Consumatorios - nivel Congruentes Drive/Percepción - nivel Persistentes Perceptuales. De aquí que el estímulo neutro no llegue a persistir mucho tiempo en el nivel Persistentes Perceptuales. Luego, una vez que haya finalizado la competencia entre las

conductas elementales *persistencia perceptual*, la ganadora será la conducta elemental asociada al EI (o al EC) por lo que no ocurrirá condicionamiento del estímulo neutro.

Caída de la actividad del estímulo en el tiempo

Para intervalos inter-estímulos muy altos (cerca de T) ocurre un condicionamiento muy pobre del estímulo neutro. Esto es razonable si tenemos en cuenta que en la ecuación de la activación de las conductas elementales *persistencia perceptual* (expresión 3.1) existe un término de caída pasiva. Es decir, a pesar de la existencia de lazos positivos que mantienen al elemento solución T_i generado por la conducta elemental ganadora un tiempo en el nivel Persistentes Perceptuales, el factor de caída pasiva hace que poco a poco vaya disminuyendo la actividad de esta conducta elemental con el tiempo, aún sin la presencia de nuevas entradas a este nivel que originen una nueva competencia. Por lo tanto, cuando el intervalo inter-estímulo es muy alto (o sea, cuando el EI se demora mucho en aparecer) ya casi no hay representación en el nivel Persistentes Perceptuales del estímulo neutro, por lo que ocurre muy poco condicionamiento o ninguno.

Para valores intermedios del intervalo inter-estímulo ocurre el mayor condicionamiento

Cuando el intervalo inter-estímulo posee un valor intermedio ($I_{IE} \cong T/2 \pm \epsilon$), entonces al arribar un EI al nivel Persistentes Perceptuales (señal T_i), ya a la conducta elemental *persistencia perceptual* asociada al estímulo neutro le dio tiempo de ganar la competencia anterior, y como consecuencia de la propiedad de normalización de actividad que existe entre estas conductas, la conducta ganadora tendrá un valor de actividad muy alto. Si este hecho no le proporciona a esta conducta elemental una ventaja considerable, al menos le da una mayor resistencia a caer a cero como consecuencia de la aparición del EI (debido a la fuerte entrada de retroalimentación positiva que recibe la conducta elemental *persistencia perceptual* asociada a este último).

Queda claro que la conducta elemental *persistencia perceptual* asociada al estímulo neutro tendrá un valor mucho más alto de activación una vez que ha ganado una competencia que cuando se presenta inicialmente el estímulo neutro. Esto origina que aunque la conducta elemental *persistencia perceptual* asociada al EI compite con la conducta elemental *persistencia perceptual* asociada al estímulo neutro y la lleva a un estado en el cual esta última ya no podrá permanecer activa, la competencia durará más tiempo que si se hubieran presentado simultáneamente el estímulo neutro y el EI, y es precisamente en este tiempo que ocurre el condicionamiento del estímulo neutro. El proceso de condicionamiento del estímulo neutro puede ocurrir ya que están dadas todas las condiciones requeridas: está activa la conducta elemental *persistencia perceptual* asociada al estímulo neutro, y a la vez está activa la conducta elemental *selector de preferencias consumatorias* asociada al EI, y por consiguiente está activa la conducta elemental *atención a preferencias* asociada al estímulo neutro, y además está activo el patrón motor generado por la conducta elemental *selección de conductas externas* asociada al EI.

Como el tiempo que dura la competencia es en alguna medida función (en forma de U invertida) del intervalo inter-estímulo, y la cantidad de condicionamiento que ocurre es función (casi lineal) del tiempo que dure la competencia, entonces la cantidad de condicionamiento depende del intervalo inter-estímulo. El hecho de que haya ocurrido algún condicionamiento del estímulo neutro implicará que en asociaciones posteriores del estímulo neutro (quizás aún neutro, aunque

ya no tan neutro) y del EI, la conducta elemental *persistencia perceptual* asociada a este último no gane tan fácilmente la competencia, ya que en esta nueva competencia la conducta elemental *persistencia perceptual* asociada al estímulo neutro también está recibiendo alguna cantidad (aunque poca) de retroalimentación positiva.

Dominancia

La propiedad de dominancia significa que si al EI lo preceden varios estímulos neutros, entonces el estímulo neutro de mayor intensidad será el que mayor condicionamiento reciba. Para comprender mejor esta idea supongamos que al nivel Persistentes Perceptuales del nodo cognitivo arriban varios estímulos neutros (varias señales T_i , $i = 1, 2, \dots$), y uno de estos es más intenso que los restantes. Supongamos además que transcurre un intervalo inter-estímulo de magnitud intermedia y entonces se presenta el UCS. Luego, la conducta elemental asociada al estímulo neutro más intenso ya tuvo tiempo de disminuir la actividad de las otras conductas elementales asociadas a los restantes estímulos neutros y aumentar la suya (ganar la competencia). Por esta razón, cuando se libera la señal drive y ocurre el proceso de aprendizaje (modificación de las fuerzas de apareo), será precisamente el estímulo neutro más intenso el que mayor condicionamiento reciba.

Pérdida del condicionamiento (olvido)

Si en un escenario en el cual se encuentra un estímulo en proceso de condicionamiento o condicionado no se encuentra un EI que refuerce la columna a la cual se está condicionando, entonces se evalúa la segunda parte de la ecuación 4.3, provocando una pérdida del condicionamiento del EC, en términos del parámetro μ .

4.14. Inhibición de respuestas reflejas

Un EI es aquel tipo de estímulo capaz de evocar una respuesta adaptativa sin que haya mediado un aprendizaje (o condicionamiento) previo. Los EI son innatos y siempre producen la respuesta incondicionada al ser presentados (Kandel, 1976 y Kandel, 1978). Esta aproximación no establece claramente la relación de los EI con los estados internos, al pensar en que éstos pudieran actuar modificando el grado de la respuesta. Sin embargo, pudiéramos pensar que el concepto adaptativo lleva en sí mismo, por ejemplo, que si el animal no tiene hambre no coma, aunque esté presente la comida.

Esta primera aproximación pudiera llevarnos a establecer que un EI es aquel estímulo que no está mediatizado por el estado interno y que siempre produce la respuesta en el animal al ser presentado éste, y considerar dentro de estos estímulos aquellos bastantes primitivos (por ejemplo, un golpe o un pinchazo fuerte, el contacto con un objeto de elevada temperatura, etc.) donde sabemos que existen claros circuitos que conectan las entradas con las salidas.

Una segunda aproximación sería considerar a los estímulos que siempre se han tratado como incondicionados, pero que están fuertemente influenciados por los estados internos (por ejemplo, el olor a la comida), como estímulos condicionados (EC), los cuales se condicionaron en etapas

tempranas (durante los períodos críticos de maduración del animal). Este enfoque resulta plausible y ya ha sido planteado por otros autores (Knorsky, 1967). Sin embargo, hay algo que es importante tener en cuenta. Si el hecho de considerar esta segunda aproximación nos induce a pensar que no existen vías directas entre el estímulo y la salida, entonces estamos cometiendo un grave error, ya que los datos experimentales (Sutton y Barto, 1981) nos dicen que en animales descerebrados (animales en los que se destruyeron todos los centros donde se puede integrar el estado interno), al ponerles un alimento en la boca, evocan la salivación; al tocarles u oprimirles la mandíbula hacia abajo evocan la masticación, etc. Claro, estos son reflejos puramente consumatorios y no preparatorios, pero de todos modos nos dicen que sí existen conexiones directas entrada-salida. Este tipo de conexiones son bastante conocidas. Por ejemplo, los receptores de la boca envían, mediante el nervio lingual, información al núcleo del tractus solitario (en el tallo encefálico) y de aquí se envía información a las glándulas salivales. Esta vía conforma el arco reflejo que subyace en el reflejo de la salivación ante estímulos gustativos, el cual es un reflejo incondicionado.

En la red de conductas internas hemos considerado que en el caso de los EI ambas vías existen: vías directas entrada-salida y vías que incluyen a los estados internos (segunda aproximación). De esta forma, la ejecución de una acción motora estará influenciada por dos vías: una que llega directamente desde el nivel Percepciones Externas, no mediatizada por los estados internos, la cual constituye un primer lazo de nivel jerárquico inferior; y otra vía que llega desde el nivel Congruentes Drive/Percepción mediatizada por los estados internos, la cual constituye un segundo lazo de nivel jerárquico superior.

El carácter jerárquico de estos lazos consiste en que el lazo inferior será inhibido en presencia de señales provenientes desde el lazo superior. Es decir, si a la misma zona motora llegan señales provenientes desde el lazo inferior y desde el lazo superior, los comandos motores que se ejecutarán son los del lazo superior. En la red de conductas internas este principio jerárquico ha sido implementado a través de la conducta interna *inhibición de respuestas reflejas*, discutida ya en el epígrafe 3.2.6. El papel de esta conducta interna es bloquear la llegada a la zona motora (nivel Acciones del pizarrón) de señales provenientes del lazo inferior, siempre que existan señales provenientes del lazo superior. Este hecho es bastante consecuente con el principio de organización jerárquica del sistema nervioso. De este modo, cuando es bloqueado todo el flujo de señales que arriban al nodo cognitivo provenientes del nodo motivacional y se presenta un EI, el lazo de menor jerarquía posibilita la ejecución de la correspondiente acción motora, sin la intervención de los estados internos. Por otra parte, al no bloquearse este flujo de señales, los estados internos juegan el papel de moduladores o inhibidores de la intensidad de la ejecución de la acción motora.

Para comprender mejor este principio jerárquico consideremos el siguiente ejemplo. Supongamos que la red de conductas internas implementa el mecanismo de selección de acciones de un *animat* simulado en la pantalla de una computadora, y supongamos además que dentro del repertorio de comportamientos que éste es capaz de ejecutar se encuentran las acciones explorar, comer y tomar agua, entre otras. Consideremos ahora que hay presencia de “alimentos” y la conducta elemental *selector de preferencias consumatorias* asociada al estado interno “hambre” es la que gana la competencia (al existir un valor muy fuerte de este estado interno). Como resultado de esta situación, se va a fortalecer la selección de la conducta externa “comer”, ya que en la red de conductas internas se ha modelado una especie de aditividad entre las influencias provenientes de

ambos lazos. Consideremos ahora que continúa la presentación del estímulo "alimentos" pero que en esta ocasión la conducta elemental *selector de preferencias consumatorias*, asociada al estado interno "hambre" no es la que gana la competencia, pero sí existe otra conducta elemental de este mismo tipo ganadora. En este caso, la conducta elemental ganadora permitirá la activación de su correspondiente conducta elemental *selección de conductas externas* (al ser activada la conducta elemental *atención a preferencias* encadenada a ésta), la cual, aunque no sea específicamente antagonica con la conducta elemental *selección de conductas externas* encadenada a la conducta externa "comer", será quien active la conducta externa apropiada (recordemos que el lazo superior tiene prioridad sobre el lazo inferior). De este modo, aunque haya presencia de comida en el medio externo y el lazo inferior trate de tomar el control de la zona motora, si a nivel de las conductas elementales *selector de preferencias consumatorias* la competencia fue ganada, digamos por la conducta elemental asociada al estado interno "sed" (debido a la presencia de agua y a la existencia de un valor elevado en el estado interno "sed", y no hay "hambre"), entonces ocurrirá que, en virtud del papel de la conducta *inhibición de respuesta reflejas*, desde el lazo superior será evocada la conducta externa "tomar agua", mientras que será bloqueada la conducta externa "comer", la cual sería evocada por el lazo inferior.

Ahora bien, ¿qué ocurre con este proceso modulador o inhibitorio cuando no ha ganado la competencia ninguna conducta elemental específica *selector de preferencias consumatorias*? Continuando con el ejemplo anterior, este es el caso en el que existen señales de alimentos en el medio externo y no hay señales que puedan satisfacer otras necesidades internas, por ejemplo, aunque aún exista demasiada sed, no hay agua presente. Es decir, la situación es la siguiente:

Estados Internos: NO hay hambre, SI hay sed.

Señales Externas: SI hay alimentos, NO hay agua.

En este caso, en virtud del lazo inferior, se ejecutaría la conducta de "comer", aunque no haya hambre, lo cual no es adaptativo. Lo adaptativo sería que, en función del estado interno, la conducta fuera dirigida a la búsqueda del agua. En la red de conductas internas esta situación ha sido resuelta considerando una "columna conductual" por defecto. Es decir, una cadena formada por diferentes conductas elementales por defecto, todas ellas asociadas a una conducta externa orientada a la búsqueda de una señal externa específica, la cual en el ejemplo del *animat* es la actividad exploratoria. Cuando ninguna de las conductas elementales *atención a preferencias* ha logrado ejecutar su acción final, entonces será completada la activación de toda la "columna conductual" por defecto, y como consecuencia será invocada la conducta externa "explorar", la cual ejerce el efecto de supresión de los actos del lazo inferior. De este modo, la conducta externa a ejecutarse siempre será una conducta determinada por los estados internos.

4.15. Consideraciones finales

En este capítulo han sido presentadas las principales propiedades que caracterizan la selección de acciones en la red de conductas internas. Algunas de estas propiedades han sido ya explicadas por varios de los mecanismos de selección de acciones revisados en el capítulo 1, aunque ninguno de

éstos explica el conjunto completo de todas estas propiedades heredadas de los principios que caracterizan el comportamiento animal, las cuales enriquecen considerablemente la selección de acciones. Por otra parte, otras de las propiedades explicadas en este capítulo resultan ser muy particulares de la red de conductas internas; este es el caso de propiedades tales como la preactivación de conductas internas, la existencia de una conducta externa por defecto orientada a la búsqueda de una señal externa específica, el papel de los lazos jerárquicos superior e inferior; así como la relación explícita de la selección de acciones con los procesos de aprendizaje, y la emergencia de una gran cantidad de propiedades no “alambradas” en la red de conductas internas, siendo estas últimas discutidas en la parte final del trabajo. La gran mayoría de las propiedades que caracterizan a la red de conductas internas serán ilustradas en los experimentos desarrollados en el capítulo 6.

CAPÍTULO 5

LA SIMULACIÓN

Para comprobar cuándo la red de conductas internas era capaz de producir por sí misma los efectos aclamados por ésta, fue escrito un programa implementando este modelo y utilizado por un robot móvil autónomo (*animat*) simulado en la pantalla de una computadora. El robot simulado fue inspirado en una idea original desarrollada por Negrete (Negrete y Martínez 1996).

A continuación se ilustran los componentes de la simulación: el robot simulado, compuesto por el sistema perceptual, el medio interno, el mecanismo de selección de acciones (la red de conductas internas), y el sistema motor; el medio ambiente simulado; y el repertorio de comportamientos. En la parte final de éste capítulo se exponen los parámetros y las fuerzas de apareo que se han ajustado específicamente para la simulación del *animat*.

5.1. El robot simulado

Como se puede apreciar en la figura 5.1, la representación gráfica del robot simulado (*animat*) es un cono con una esfera semicircunscrita. La punta del cono indica la dirección del *animat*.

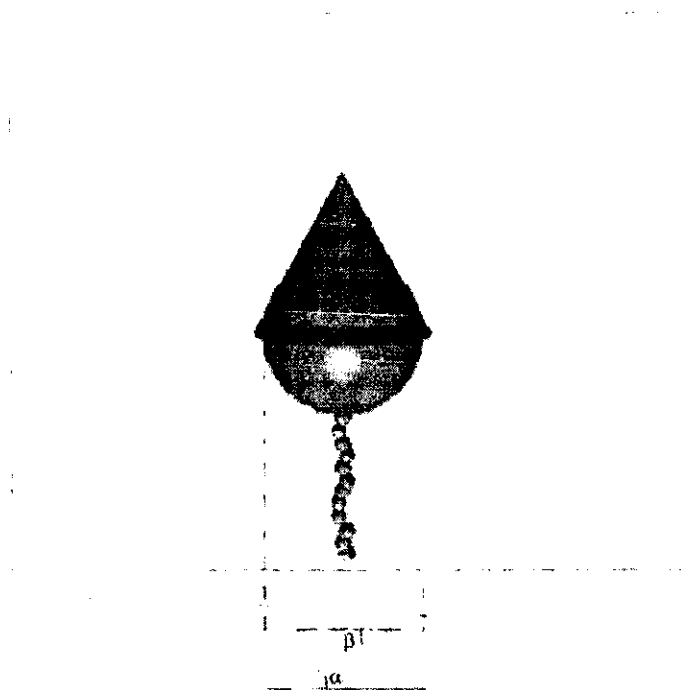


Figura 5.1. Representación gráfica del robot simulado (*animat*).

La estructura interna del *animat* puede ser descrita a partir de los cuatro componentes básicos que la integran: el sistema perceptual, el medio interno, la red de conductas internas, y el sistema motor. En la figura 5.2 se pueden apreciar estos componentes y las interrelaciones que se establecen entre los mismos.

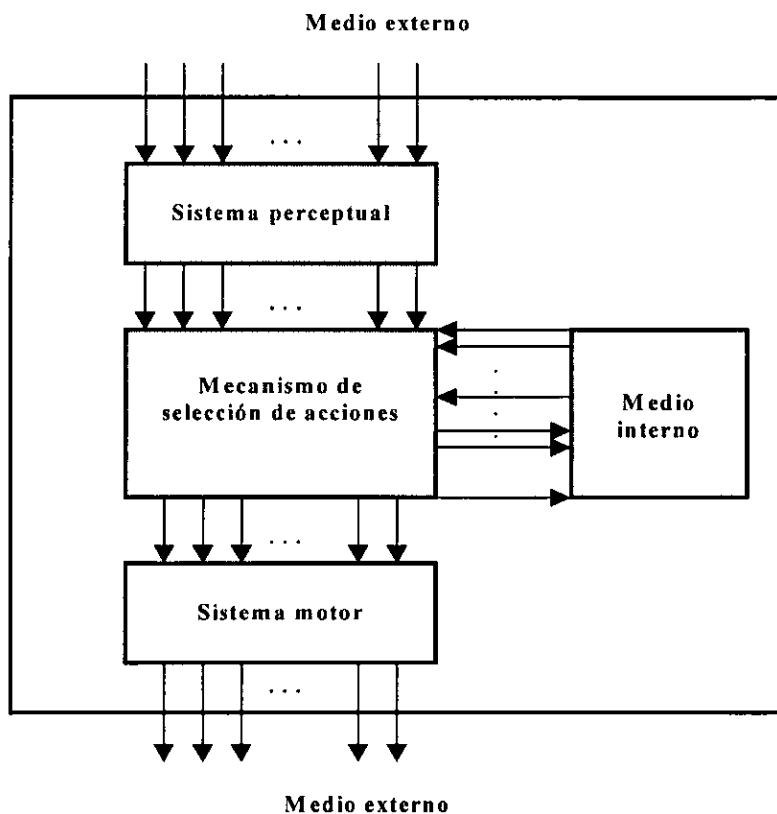


Figura 5.2. Componentes que integran la estructura interna del animat y las relaciones que se establecen entre éstos.

5.1.1. El sistema perceptual

El *animat* está situado en un plano (z,x) de un espacio (x,y,z) . En este plano se pueden encontrar tres tipos de representaciones de estímulos externos: agua, representada por círculos azules; comida, representada por esferas verdes; obstáculos fijos, representados por paralelepípedos cafés; pasto, representado por círculos verdes texturizados; estímulos aversivos, representados por elipsoides negros; manchas rojas; y manchas amarillas. El *animat* sólo percibe aquellos estímulos externos que caen dentro de la región perceptual (R_p) definida por el semicírculo determinado por la fórmula 5.1.

$$R_p = \begin{cases} ((z-z_a)^2+(x-x_a)^2 < r_p^2) \cap (x > x_a + \tan((th+\pi/2)*(z-z_a))) & \text{si } 0 < th \leq \pi \\ ((z-z_a)^2+(x-x_a)^2 < r_p^2) \cap (x < x_a + \tan((th+\pi/2)*(z-z_a))) & \text{si } \pi < th \leq 2\pi \end{cases} \quad (5.1)$$

La expresión $(z-z_a)^2+(x-x_a)^2 < r_p^2$ determina el conjunto de todos los puntos contenidos en el círculo con centro en (z_a, x_a) y radio r_p (r_p es el radio de percepción). La expresión $x > x_a + \tan((th+\pi/2)*(z-z_a))$ determina el conjunto de todos los puntos por encima de la recta perpendicular a la orientación del animat (esta última denotada por th) y que pasa por el punto (z_a, x_a) , mientras que la expresión $x < x_a + \tan((th+\pi/2)*(z-z_a))$ determina el conjunto de todos los puntos por debajo de dicha recta. Cuando $0 < th \leq \pi$, la región perceptual R_p del animat es el semicírculo definido por la intersección $((z-z_a)^2+(x-x_a)^2 < r_p^2) \cap (x > x_a + \tan((th+\pi/2)*(z-z_a)))$; mientras que cuando $\pi < th \leq 2\pi$, la región perceptual R_p del animat es el semicírculo definido por la intersección $((z-z_a)^2+(x-x_a)^2 < r_p^2) \cap (x < x_a + \tan((th+\pi/2)*(z-z_a)))$. La región perceptual del animat se puede apreciar en la figura 5.3. Una vez determinados los estímulos que caen dentro de la región perceptual, son eliminados aquellos que se encuentran detrás de algún obstáculo.

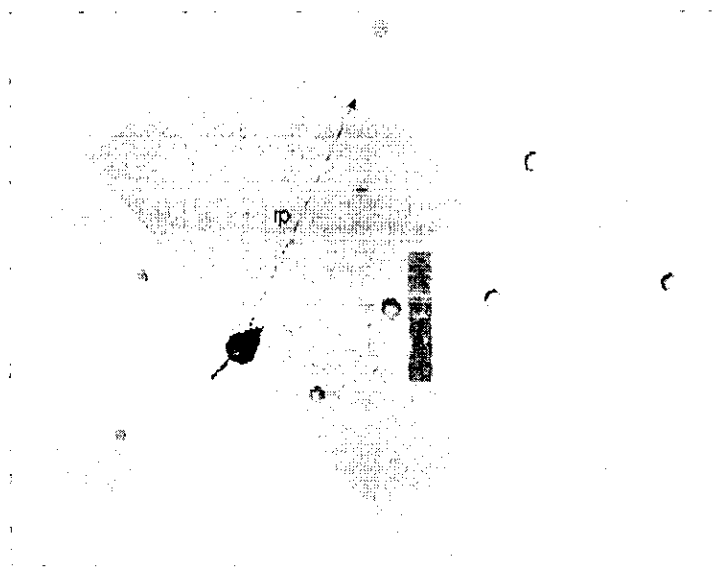


Figura 5.3. Región perceptual del animat.

El sistema perceptual capta la magnitud de los estímulos, y la distancia a la que éstos se encuentran del animat. Posteriormente, éste envía al MSA un valor ponderado correspondiente a cada tipo de estímulo percibido, el cual es función de la relación entre magnitud y distancia para todos los estímulos de un mismo tipo. Este valor ponderado representa la fuerza con la cual el tipo de estímulo será registrado en el nivel Percepciones Externas del nodo cognitivo.

El sistema perceptual cuenta con un sistema simple de memoria a corto-mediano plazo, mostrado en la expresión 5.2. El funcionamiento de éste consiste en disminuir el valor de un estímulo que

ha dejado de percibirse en términos de un parámetro \mathcal{N} . Cuando el valor del estímulo es menor a \mathcal{N} , el estímulo sale de su memoria, olvidándolo.

$$F_e(t + 1) = F_e(t) - \mathcal{N} \quad (5.2)$$

donde F_e representa la intensidad de un estímulo externo, y \mathcal{N} es un factor de olvido, $0 \leq \mathcal{N}$.

El *scanning* que efectúa el *animat* durante su percepción puede verse como un “proceso de atención”, por lo que el radio perceptual puede verse como un parámetro de atención. La región perceptual del *animat* es considerada como el “escenario percibido” por éste en cada momento en el tiempo. El *animat* al desplazarse cambia de escenario percibido, por lo que los estímulos ambientales que el *animat* acaba de dejar fuera de su escenario y se encuentran en su memoria a corto-mediano plazo forman parte de su “escenario recordado”.

5.1.2. El sistema motor

Como se puede apreciar en la figura 5.1, la marcha del *animat* es comandada por los pasos angulares α y β , con centro en los extremos del diámetro de la proyección de la esfera en el plano (z,x) perpendicular a la orientación del *animat*. La conducta externa *deambular* se produce cuando los ángulos α y β toman valores aleatorios entre 0 y 1 (radianes), lo cual describe una trayectoria aleatoria. Por otra parte, la conducta externa *exploración orientada a la búsqueda de una señal específica* se produce cuando los ángulos α y β toman valores iguales, lo cual describe la trayectoria del *animat* como una línea recta. El tamaño de los pasos es proporcional a la fuerza del *animat*. Las trayectorias descritas por ambos tipos de conductas externas pueden ser apreciadas en las figuras 5.4.y 5.5, respectivamente.

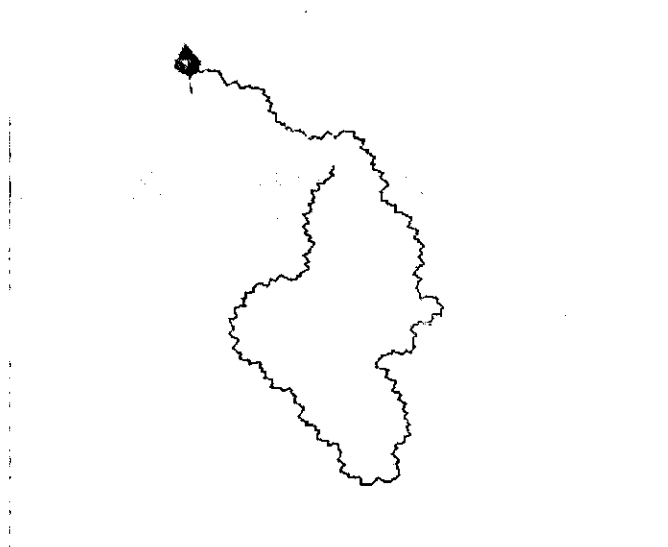


Figura 5.4. La conducta externa *deambular*.

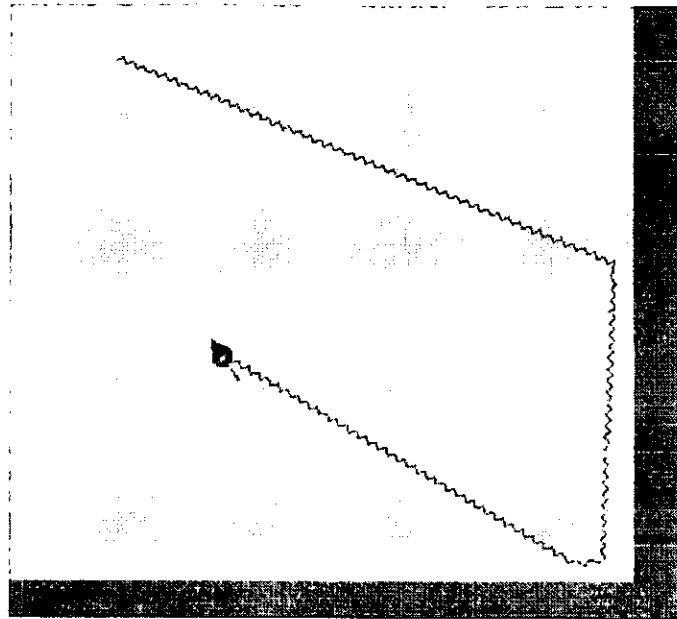


Figura 5.5. La conducta externa *exploración orientada a la búsqueda de una señal específica.*

5.1.3. El medio interno

El medio interno del *animat* es expresado a través de un conjunto de variables, cada una de las cuales representa una necesidad o estado interno. Como se muestra en la figura 5.6, los estados internos pueden ser inicializados o establecidos en los valores requeridos, en dependencia del tipo de experimento o prueba que se desee ejecutar.

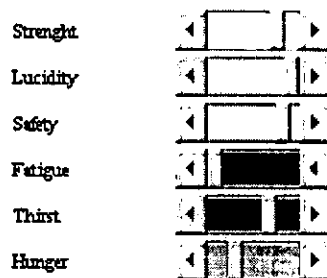


Figura 5.6. Establecimiento de los estados internos del *animat*.

Para la simulación actual se han considerado los estados internos fuerza, lucidez, seguridad, fatiga, sed y hambre. Como se puede apreciar en la figura 5.2, el mecanismo de selección de acciones es el único componente de la estructura interna del *animat* que establece una interfaz directa con el medio interno. El MSA percibe en cada instante el valor actual de cada una de las variables del medio interno a través de los propioceptores, y actualiza dicho valor en una proporción determinada, siempre que la conducta externa asociada al estado interno en cuestión

haya sido ejecutada. Por ejemplo, la ejecución de los comportamientos consumatorios “comer” y “tomar agua” disminuirán los valores de los estados internos “hambre” y “sed”, respectivamente.

La simulación permite la opción de que el *animat* deje una huella de la trayectoria recorrida, la cual facilita el análisis del patrón de comportamientos ejecutado por éste. El color de la huella toma valores RGB (rojo-verde-azul) con las intensidades de los estados internos fatiga, hambre y sed, respectivamente. Esto permite visualizar también la dinámica de los estados internos del *animat*.

El estado interno fuerza indica qué tan amplios serán los pasos angulares del animat. Al tener menos fuerza, el animat dará pasos más pequeños, y viceversa. El estado interno lucidez es un análogo a la fuerza en el campo perceptivo. Esto es, al ser la lucidez menor, la región perceptual del *animat* disminuirá (en términos del radio de percepción). El estado interno seguridad es constante durante la simulación. Esto es, no hay acciones externas que alteren este estado interno, el cual puede ser determinado por el usuario. Los estados internos fatiga, sed y hambre se incrementan a medida que transcurre la simulación, y son decrementados cuando se ejecuten sus comportamientos consumatorios correspondientes: descansar, tomar agua y comer. A su vez, cuando estos tres estados internos son elevados, la fuerza y la lucidez disminuyen. Cuando son mínimos, la fuerza y la lucidez se recuperan lentamente.

5.2. El medio ambiente simulado

El ambiente externo del *animat* está definido por un plano (z,x) correspondiente a un espacio (x,y,z). El plano (z,x) está delimitado por un marco. En el área definida por este marco se pueden crear diferentes objetos, tales como esferas verdes, círculos azules, paralelepípedos cafés círculos verdes, elipsoides negros, círculos rojos y círculos amarillos; los cuales representan los estímulos externos comida, agua, obstáculos fijos, pastos, estímulos aversivos, manchas rojas y manchas amarillas, respectivamente. El marco que delimita el plano (z,x) también es considerado un obstáculo fijo. La figura 5.7 muestra una vista aérea del medio ambiente simulado, mientras que en la figura 5.8. se puede apreciar una perspectiva isométrica de este medio ambiente.

Los estímulos externos pueden ser generados tanto de forma determinista como de forma aleatoria. Cuando un estímulo externo es generado de forma determinista, entonces es necesario especificar la localización de dicho estímulo sobre el plano (z, x), así como las dimensiones del mismo. Al inicio de cada corrida de la simulación, la ubicación del animat es establecida en las coordenadas (0.0, 0.0) del plano (z, x), aunque esta ubicación también puede ser reestablecida en las coordenadas deseadas. En la figura 5.9 se puede apreciar la interfaz a través de la cual es posible establecer las coordenadas y dimensiones para los estímulos externos a ser generados en el medio ambiente del *animat*, así como las coordenadas iniciales para la ubicación de este último.



Figura 5.7. El medio ambiente simulado del *animat* en vista aérea.

Para algunos estímulos externos la magnitud o valor de existencia es invariable, mientras que para otros estímulos este valor puede cambiar debido a la ejecución de alguna acción motora del *animat*. Por ejemplo, el estímulo “obstáculo fijo” debe tener asociado una magnitud invariable, ya que su existencia no está asociada a la ejecución de acciones motoras del *animat*. Sin embargo, el estímulo “agua” debe tener asociado una magnitud variable, ya que ante la ejecución de la acción externa “tomar agua” por parte del *animat*, la existencia del estímulo “agua” debe disminuir.

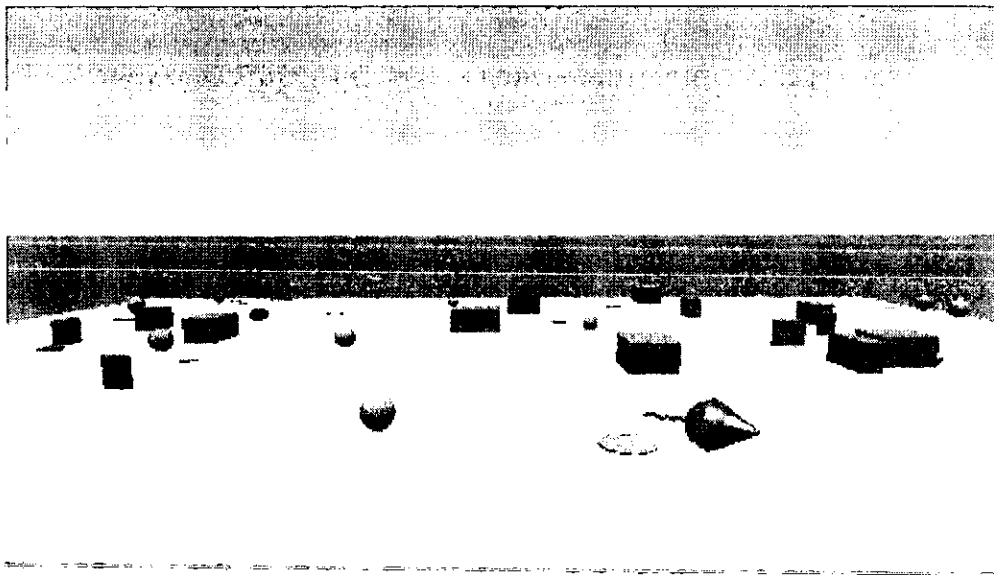


Figura 5.8. El medio ambiente simulado del *animat* en perspectiva isométrica.

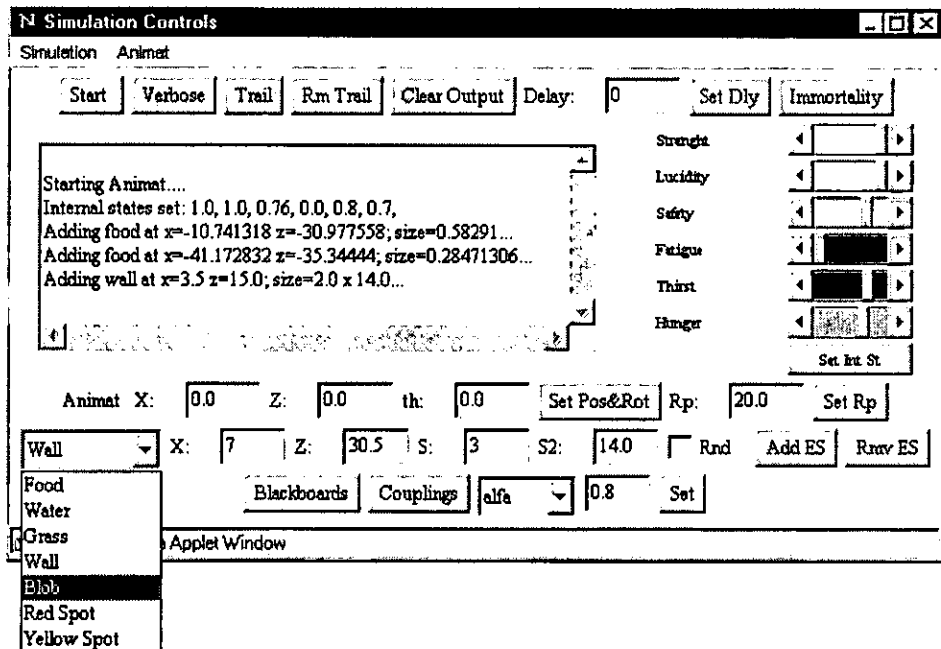


Figura 5.9. Interfaz de la simulación. Esta interfaz permite la generación de estímulos externos, el establecimiento de los estados internos, y el ajuste de parámetros.

Atendiendo al significado interno que puedan tener para el *animat*, los estímulos ambientales pueden ser de dos tipos: estímulos que ya poseen un significado para el *animat* y estímulos neutros. Los primeros son aquellos que poseen un significado interno determinado para el *animat*, siendo capaces de satisfacer en un momento dado sus necesidades internas (estos son los estímulos incondicionados vistos ya en la red de conductas internas). Ejemplos de estos estímulos son fuente de agua, fuente de comida, pasto y estímulo aversivo. Los estímulos neutros son aquellos que inicialmente no poseen algún significado interno para el *animat*, pero que pudieran llegar a asociarse a estímulos con significado positivo. Ejemplos de estos estímulos son manchas rojas y manchas amarillas.

5.3. El repertorio de comportamientos

El repertorio de comportamientos (acciones externas) que el *animat* puede ejecutar es mostrado en la tabla 5.1. Como se puede apreciar en esta tabla, muchos de estos comportamientos pueden ser ejecutados sólo cuando ambas condiciones externa e interna han sido satisfechas. Es decir, existe una necesidad interna elevada y la entrada externa capaz de satisfacer dicha necesidad ha sido observada. Este es el caso de los comportamientos “aproximarse a la comida”, “comer”, “aproximarse al agua”, “tomar agua”, y “aproximarse a la comida y al agua”.

Comportamiento	Entrada externa	Entrada interna
Evitar obstáculo	Obstáculo al alcance	Ninguna
Deambular	Ninguna	Ninguna
Explorar	Ninguna	Sed y/o hambre y/o fatiga
Aproximarse a la comida	Comida percibida	Hambre
Comer	Comida al alcance	Hambre
Aproximarse al agua	Agua percibida	Sed
Tomar agua	Agua al alcance	Sed
Aproximarse a la comida y al agua	Comida y agua percibidas	Hambre y sed
Descansar	Pasto percibido	Fatiga
Aproximarse al pasto	Pasto al alcance	Fatiga
Alejarse de estímulo aversivo	Estímulo aversivo percibido	Seguridad

Tabla 5.1.

Según McFarland (McFarland 1981, Maes 1991), los comportamientos asociados con una o más motivaciones (o necesidades internas) son conocidos como comportamientos consumatorios, mientras que los que no están asociados directamente con alguna motivación se conocen como comportamientos apetitivos. De esta forma, un comportamiento consumatorio es aquel que un animal (o entidad artificial) realmente desea satisfacer cuando la motivación asociada a éste es elevada; mientras que un comportamiento apetitivo solo contribuye a que un comportamiento consumatorio pueda ser ejecutado.

En la simulación del *animat*, nosotros hemos considerado que ambos tipos de comportamientos existen: consumatorios y apetitivos. Sin embargo, a diferencia de McFarland (McFarland 1981), nosotros hemos asumido que tanto los comportamientos consumatorios como los apetitivos se encuentran directamente asociados con alguna motivación. La diferencia entre ambos tipos de comportamientos radica en que un comportamiento consumatorio es aquel que ejecuta la acción final requerida por el *animat* para satisfacer una necesidad interna elevada, y comúnmente reduce la fuerza de dicha necesidad; mientras que un comportamiento apetitivo contribuye a que el consumatorio pueda ser ejecutado, pero la ejecución del primero no reduce directamente el nivel de la motivación o necesidad interna a la que se encuentra asociado. Por ejemplo, si consideramos que el comportamiento “aproximarse al agua” es apetitivo y que el comportamiento “tomar agua” es consumatorio, entonces ambos comportamientos estarán asociados a la necesidad interna “sed”; y el *animat* solo se aproximará al agua si tiene sed, contribuyendo de esta forma a que la acción final deseada por el *animat*, “tomar agua”, sea ejecutada.

Los comportamientos apetitivos de la simulación son: “aproximarse a la comida”, “aproximarse al agua”, “aproximarse a la comida y al agua” y “aproximarse al pasto”. Los comportamientos consumatorios son: “comer”, “tomar agua” y “descansar”.

El comportamiento “evitar obstáculo” es una conducta refleja, por lo que ésta siempre tomará precedencia ante cualquier otra conducta externa ejecutable, siempre que un obstáculo se interponga en la trayectoria del *animat*. Lo anterior ocurre en virtud de un principio implementado en la red de conductas internas, el cual establece conexiones directas entre los

niveles Percepciones Externas y Acciones del nodo cognitivo, para ciertos tipos de estímulos externos.

El comportamiento “deambular” es la conducta externa por defecto que será observada en el *animat*, siempre que no existan necesidades externas que satisfacer. Como se puede observar en la figura 5.4, la trayectoria seguida por el *animat*, cuando esta conducta externa es ejecutada, es completamente aleatoria, ya que dicha conducta no está orientada a la búsqueda de alguna señal en particular.

El comportamiento “explorar” responde a uno de los principios novedosos implementados en la red de conductas internas: la existencia de una conducta externa orientada a la búsqueda de una señal externa específica. El comportamiento “explorar” será activado cuando exista una imperiosa necesidad interna para la cual no se han percibido señales externas capaces de satisfacer la misma. Es decir, siempre que a nivel motivacional gane la competencia una necesidad interna para la cual no existan señales externas aprovechables, entonces el comportamiento “explorar” será ejecutado, actuando como un comportamiento apetitivo orientado a la satisfacción de dicha necesidad interna. Lo anterior justifica el hecho de que en la tabla 5.1 se asocian al comportamiento “explorar” las entradas internas hambre, sed y fatiga.

La conducta por defecto de la red de conductas internas puede ser en un momento dado “deambular” o bien “explorar”. Si no hay necesidades internas que satisfacer (no se genera una señal Drive en el nodo motivacional asociada a una necesidad interna), entonces la conducta externa que se ejecutará será “deambular”. La conducta externa “explorar” se ejecutará si hay una necesidad interna que satisfacer en el nivel Drive del nodo motivacional y no hay un estímulo externo capaz de satisfacer esta necesidad.

5.4. Ajuste de parámetros

Para permitir una mayor flexibilidad al momento de realizar experimentos, la simulación permite que varios de los parámetros definidos en la red de conductas internas puedan ser ajustados a través de la interfaz que ésta proporciona. La interfaz puede apreciarse en la figura 5.10.

Los parámetros denotados por letras griegas son los que pertenecen al mecanismo de selección de acciones, los cuales fueron discutidos en los capítulos 3 y 4; mientras que los denotados por letras hebreas pertenecen al sistema perceptual, los cuales fueron discutidos en el epígrafe 5.1.1.

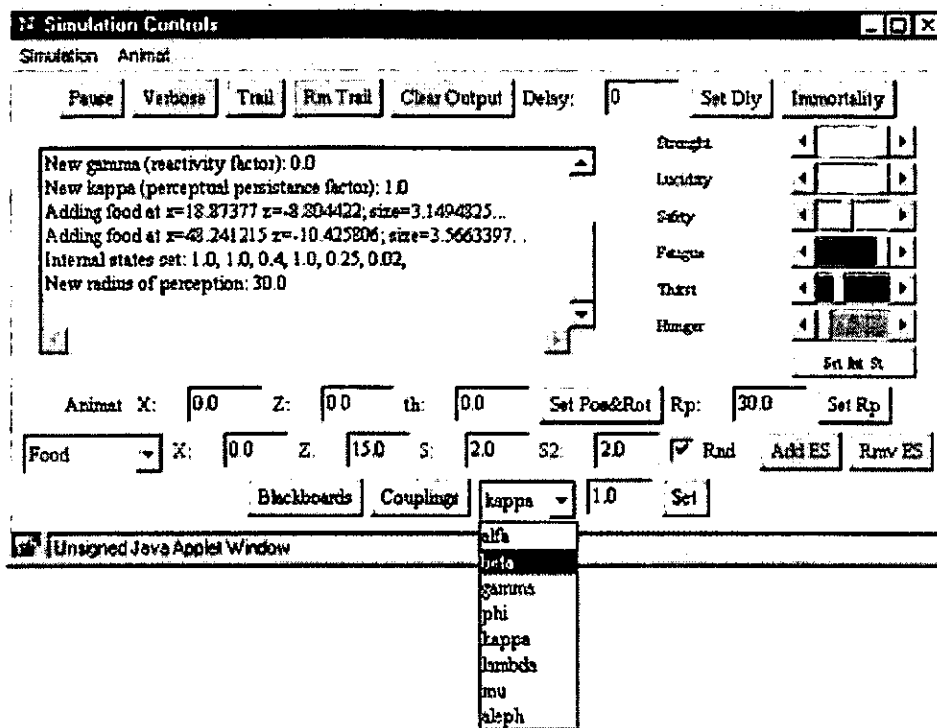


Figura 5.10. Ajuste de parámetros a través de la interfaz de la simulación.

En la tabla 5.2 se pueden apreciar estos parámetros, con una breve descripción de qué papel juegan, y el valor por defecto que poseen éstos en la simulación.

Parámetro	Papel	Valor por defecto	Expresión en la que aparece
α	Factor de motivación y preactivación	0.8	(3.7)
β	Razón de pérdida en la fuerza de apareo	0.001	(4.3)
ϕ	Importancia atribuible a la señal del nodo motivacional a nivel del nodo cognitivo	1.0	(4.2)
γ	Factor de reactividad a nivel del nodo cognitivo	0.0	(4.2)
κ	Factor de pérdida en el nivel persistencia perceptual	0.15	(3.2)
λ	Razón de velocidad del aprendizaje	0.1	(4.3)
μ	Razón de pérdida del aprendizaje	0.02	(4.3)
r_p	Radio perceptual	20.0	(5.1)
\aleph	Razón de olvido	0.1	(5.2)

Tabla 5.2.

En la tabla 5.3 se pueden apreciar los valores de los umbrales utilizados en la simulación.

Umbral	Valor
θ^T	0.001
θ^I	0.001
θ^M	0.2
θ^C	$\alpha + 0.1$ para el estado interno seguridad; 0.2 para cualquier otro estado interno

Tabla 5.3

El vector Max_i^T en la expresión (3.3) toma valores de 3.0 para estímulos incondicionados, valores de 2.0 para estímulos neutros y 0.0 para estímulos que activan conductas reflejas.

5.4.1. Inicialización de las fuerzas de apareo

A continuación se presentan las fuerzas de apareo de las conductas internas que han sido establecidos para la simulación. Los valores fijos de las fuerzas de apareo que son representadas por vectores en el nodo cognitivo pueden apreciarse en la tabla 5.4. En esta tabla se muestran las fuerzas de apareo definidas para las conductas internas y mecanismos del nodo cognitivo dependiendo de cada conducta externa.

Conductas internas y mecanismos del nodo cognitivo	Conducta por defecto	Aproximarse a la comida	Comer	Aproximarse al agua	Tomar agua	Aproximarse al agua y comida	Aproximarse al pasto	Descansar	Alejarse de estímulo aversivo	Conducta asociada a mancha roja	Conducta asociada a mancha amarilla	Evitar obstáculos
Exteroceptores	0.0	0.93	0.96	0.94	0.97	0.95	0.87	0.90	0.98	0.92	0.91	1.0
Acciones Reflejas ^S	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Persistencia Perceptual ^S	0.0	0.93	0.96	0.94	0.97	0.95	0.87	0.90	0.98	0.92	0.91	0.0
Persistencia Perceptual ^I	0.0	0.93	0.96	0.94	0.97	0.95	0.87	0.90	0.98	0.92	0.91	0.0
Atención a Preferencias ^T	1.0	1.08	1.11	1.09	1.12	1.10	1.02	1.05	1.13	1.07	1.06	0.0
Inhibición de Respuestas Reflejas ^I	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
Actuadores ^M	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Tabla 5.4.

Las F_{ij}^T de la conducta interna *persistencia perceptual* toman valores de 0 cuando $i = j$, valores de -1.0 cuando i es una columna conductual que corresponde a un estímulo neutro, y valores iguales a -0.15 para los otros casos. Las F_{ij}^C de la conducta interna *atención a preferencias* toman valores de 1.0 para conductas i asociadas a estados internos j del nodo motivacional (aproximarse a la comida-hambre, comer-hambre, alejarse de estímulo aversivo-seguridad, etc.), valores iniciales de 0.001 para columnas conductuales i que corresponden a un estímulo neutro a condicionarse a estados internos j (por medio de los procesos de aprendizaje), y valores de 0.0 para los otros casos. Las F_{ij}^I de la conducta interna *inhibición de respuestas reflejas* toman valores de 1.0 cuando $i = j$ y de -8.0 para los otros casos. Las F_{ik}^I de la conducta interna *inhibición de respuestas reflejas* por defecto toman valores de -10.0, mientras que la F_k^C de la conducta interna *inhibición de respuestas reflejas* por defecto toma un valor de 1.0. Las F_{ij}^I de la conducta interna *selección de conductas externas* toman los siguientes valores: si $i = j$, 9.0 para conductas consumatorias y 3.0 para las demás conductas incondicionadas; si i corresponde a un estímulo neutro o en proceso de condicionamiento con un estímulo incondicionado j , toma valores iniciales de 0.001, modificables por los procesos de aprendizaje; y 0.0 en otro caso. Las F_{ij}^H de la conducta interna *selección de conductas externas* toman valores de 1.0 para $i = j$, siendo i un estímulo incondicionado; y 0.0 en otro caso.

Las columnas del nodo motivacional representan los siguientes estados internos: estado interno por defecto, fatiga, sed, hambre, sed-hambre, y seguridad. Las fuerzas de apareo F_{ij} de los *propioceptores*, F_{ij}^E y F_{ij}^D de la conducta interna *congruencia propio/extero/drive* toman valores de 1.0. Las F_{ij}^C de la conducta interna *selector de preferentes consumatorios* toma valores de 1.0 para $i = j$ y valores de -0.15 para otros casos. Las F_{ij}^S de la conducta interna *congruencia propio/extero/drive* toman valores de 1.0 para estados internos i correspondientes a conductas externas j del nodo cognitivo (hambre-aproximarse a la comida, hambre-comer, seguridad-alejarse de estímulo aversivo, etc.); valores iniciales de 0.001 para estados internos i a asociarse a estímulos neutros o en proceso de condicionamiento j por medio de los procesos de aprendizaje; y 0.0 en otro caso.

5.5. Consideraciones finales

El objetivo de la simulación desarrollada consiste en comprobar las propiedades que caracterizan la selección de acciones en la red de conductas internas, las cuales fueron discutidas en el capítulo 4, y no el desempeño de los sistemas perceptual y motor. Aunque estos sistemas responden a los requerimientos actuales que permiten evaluar el desempeño de la selección de acciones en la red de conductas internas, los mismos pueden mejorarse aún más. En el siguiente capítulo se ilustran y discuten los experimentos que fueron realizados a partir de esta simulación.

CAPÍTULO 6

PRUEBAS Y EXPERIMENTOS

Consideremos un medio ambiente como el descrito en la sección 5.2, en el cual se han creado objetos O_i^+ que poseen significado específico para el *animat*, tales como fuente de agua, fuente de comida, y obstáculos fijos; y objetos O_i^n , los cuales pueden adquirir un nuevo significado para el *animat*, tales como manchas rojas y manchas amarillas. Sean las posiciones en el medio, el cual describiremos plano, definidas por el par (z, x) . Sea la entidad *animat* definida por: (1) un conjunto de acciones primitivas que ésta puede ejecutar, como las descritas en la sección 5.1.2; (2) un mecanismo de selección de acciones, como el descrito en los capítulos 3 y 4; (3) un sistema perceptual, como el descrito en la sección 5.1.1; y (4) un conjunto de sensores de “necesidades”, que captan estados propios del medio interno. Consideremos además que las acciones ejecutadas por el *animat* se dividen en conductas externas dirigidas hacia el medio externo y conductas internas dirigidas hacia el medio interno de la entidad, las cuales en turno logran activar alguna conducta externa particular. Dado este sistema, las siguientes pruebas y experimentos tienen lugar.

6.1. La conducta externa observada es influenciada por los estados internos de la entidad

Como ya se vio en el epígrafe 3.3.3, es a nivel de la conducta interna *congruencia propio/extero/drive* del nodo motivacional que se lleva a cabo la combinación de señales externas e internas, para determinar la acción externa que el *animat* debe ejecutar. Según el esquema de combinación utilizado por las conductas elementales *congruencia propio/extero/drive* (expresión 3.7), para valores del estado interno O_i^E igual a cero o muy cercanos a cero, el valor de activación A_i^C creado por dichas conductas en el nivel Congruentes Propio/Extero/Drive del nodo motivacional también será cero o muy cercano a cero. Como consecuencia de este hecho, la conducta elemental *selector de preferencias consumatorias* que integra esta columna conductual no podrá inscribir su REAC, y por lo tanto, tampoco podrá participar en el proceso competitivo que se lleva a cabo entre conductas elementales de este tipo. Es decir, según este esquema de combinación, son los valores significativos de los estados internos ($O_i^E \gg 0$) los que determinan que una acción externa diferente a “deambular” pueda ser observada en el *animat*. Lo anterior puede ser apreciado en las figuras 6.1 y 6.2.

En la figura 6.1 se modela una situación en la cual para los estados internos fatiga, sed y hambre se han establecido valores iguales a cero, es decir, existe una satisfacción total de los estados internos del *animat*; mientras que éste está percibiendo abundantes fuentes de comida y de agua,

así como lugares para descansar. Como se puede apreciar en esta figura, la acción externa ejecutada por el *animat* fue la conducta externa por defecto “deambular”, ya que al no existir necesidades internas ninguna acción consumatoria pudo ser ejecutada.



Figura 6.1. No existen necesidades internas y el *animat* está percibiendo abundantes fuentes de comida y agua.

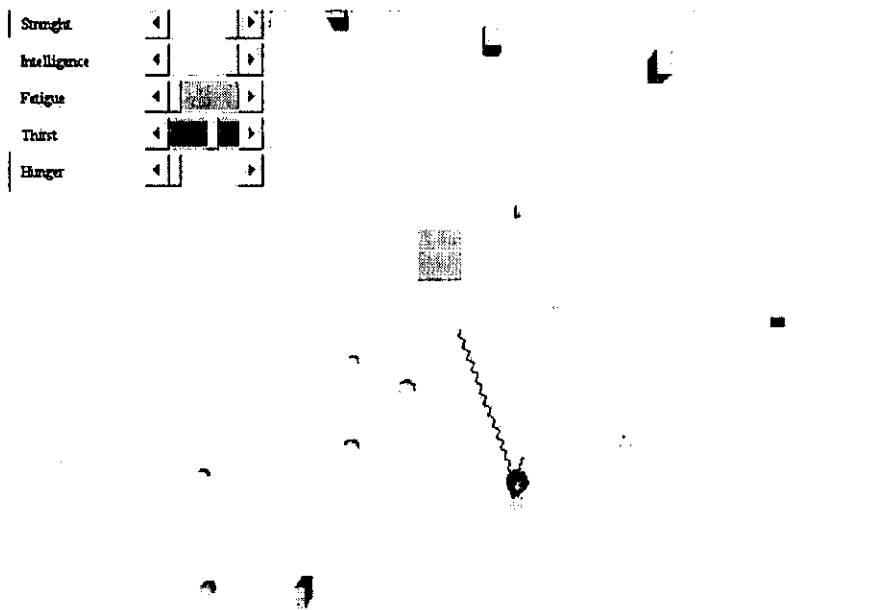


Figura 6.2. Existe una necesidad interna muy elevada, sed, y el *animat* está percibiendo abundantes fuentes de comida y agua.

Por otra parte, la figura 6.2 modela una situación en la cual el *animat* posee una necesidad interna elevada, sed; mientras que para los restantes estados internos se han establecido valores iguales a cero. Nuevamente, en esta situación, el *animat* está percibiendo abundantes fuentes de comida y de agua, así como lugares para descansar. Como se puede apreciar en esta figura, las acciones externas ejecutadas por el *animat* fueron inicialmente “aproximarse al agua”, y posteriormente “tomar agua”, ya que tanto “aproximarse al agua” como “tomar agua” son comportamientos asociados al estado interno “sed”; siendo el primero un comportamiento apetitivo que permite que el segundo, el cual es un comportamiento consumatorio, pueda ser ejecutado.

Luego, como se pudo apreciar en las figuras 6.1 y 6.2, la conducta externa observada en el *animat* siempre es influenciada por los valores de sus estados internos.

6.2. Conductas incompatibles motivacionalmente compiten entre sí, y la competencia se efectúa a nivel motivacional

Como fue establecido en el epígrafe 4.2., entre todas las conductas elementales *selector de preferencias consumatorias*, que han logrado satisfacer su condición, se lleva a cabo un proceso competitivo del tipo *the winner takes all* para decidir cuál de estas conductas será la que determine la acción externa que posteriormente ejecutará el *animat*. Las conductas que participan en esta competencia son conductas incompatibles, en el sentido de que sólo una de éstas podrá liberar una señal *drive* dirigida hacia el nodo cognitivo, hecho éste que determinará la acción externa que debe ser ejecutada.

Para ilustrar lo anterior se ha modelado una situación en la cual el *animat* posee varios estados internos con valores significativos ($O_i^E \gg 0$, para algunos valores de i) y diferentes entre sí; a la vez que éste está percibiendo señales externas capaces de satisfacer cada uno de estos estados internos. La situación modelada puede ser apreciada en las figuras 6.3 y 6.4

La figura 6.3 muestra un estado inicial dado por la posición del *animat* en el medio ambiente, la ubicación y magnitud de los estímulos ambientales, y los valores iniciales de los estados internos sed, hambre y cansancio. Es decir, este estado inicial define la situación existente antes de que se lleve a cabo el proceso competitivo que ocurre a nivel motivacional. Nótese la imperiosa necesidad de “tomar agua” dado el elevado valor del estado interno sed. Como se puede apreciar en esta figura, el *animat* está percibiendo abundantes fuentes de comida, de agua y lugares donde éste pudiera descansar.

Debido a que todas las necesidades internas actuales del *animat* pueden ser satisfechas, ya que se han percibido los estímulos externos necesarios, entonces un proceso competitivo tiene lugar entre las conductas elementales *selector de preferencias consumatorias*, para decidir cuál conducta externa debe ser ejecutada. En la figura 6.4 se muestra un estado en el cual la conducta externa “aproximarse al agua” fue ejecutada hasta que la fuente de agua se encontró al alcance del *animat* y éste pudo ejecutar la acción externa consumatoria “tomar agua”. Como consecuencia de la ejecución de esta última acción, la fuerza del estado interno sed fue reducida.

Nótese en esta figura la disminución del tamaño del estímulo externo “fuente de agua”, y la consiguiente reducción del nivel del estado interno sed.

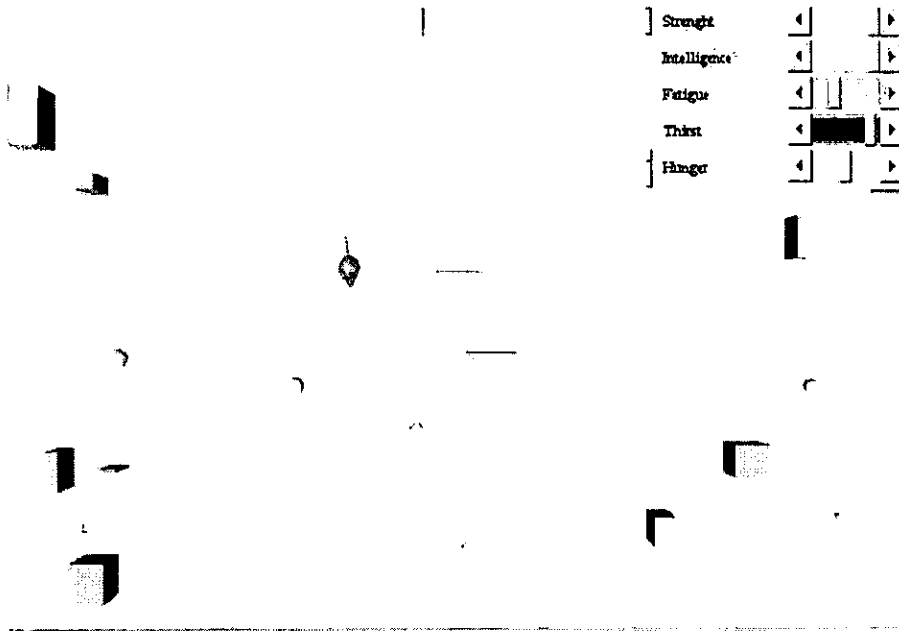


Figura 6.3. Estado inicial del *animat*. Hay sed, hambre y fatiga; a la vez que se perciben abundantes fuentes de agua, comida y lugares para descansar.

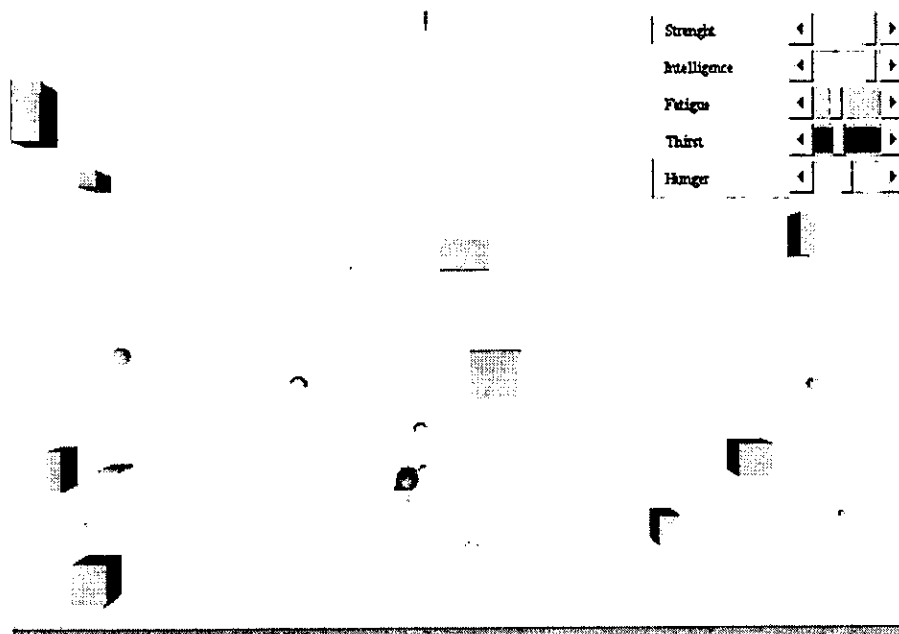


Figura 6.4. Estado alcanzado una vez que fueron ejecutadas las conductas externas “aproximarse al agua” y “tomar agua”, después que los correspondientes procesos competitivos a nivel de las conductas elementales *selector de preferencias consumatorias* activadas han tomado lugar.

Luego, como se ha podido apreciar en el experimento ilustrado en las figuras 6.3 y 6.4, siempre que dos o más conductas externas hayan logrado satisfacer sus condiciones en un momento dado, entonces una competencia a nivel motivacional tendrá lugar para decidir cuál es la conducta externa más apropiada que el *animat* debe ejecutar en ese momento.

6.3. La conducta exploratoria orientada a la búsqueda de una señal específica logra tiempos de reacción del animat mucho menores

El objetivo de este experimento es comprobar que la existencia de una conducta exploratoria orientada a la búsqueda de una señal específica mejora la selección de acciones, al lograr tiempos de reacción del *animat* mucho menores entre la presentación de un estímulo externo, para el cual existe una necesidad interna muy elevada, y la consumación de la acción asociada a la satisfacción de esta necesidad.

Para llevar a cabo la anterior comprobación, es suficiente demostrar la siguiente hipótesis: “el valor de α en la expresión 4.1 está estrechamente relacionado con el tiempo de reacción del *animat* entre la presentación de un estímulo externo, para el cual existe una necesidad interna muy elevada, y la consumación de la acción asociada a la satisfacción de esta necesidad, siendo esta relación inversamente proporcional. Es decir, si el valor de α aumenta entonces el tiempo de reacción disminuye, y si el valor de α disminuye entonces el tiempo de reacción aumenta”.

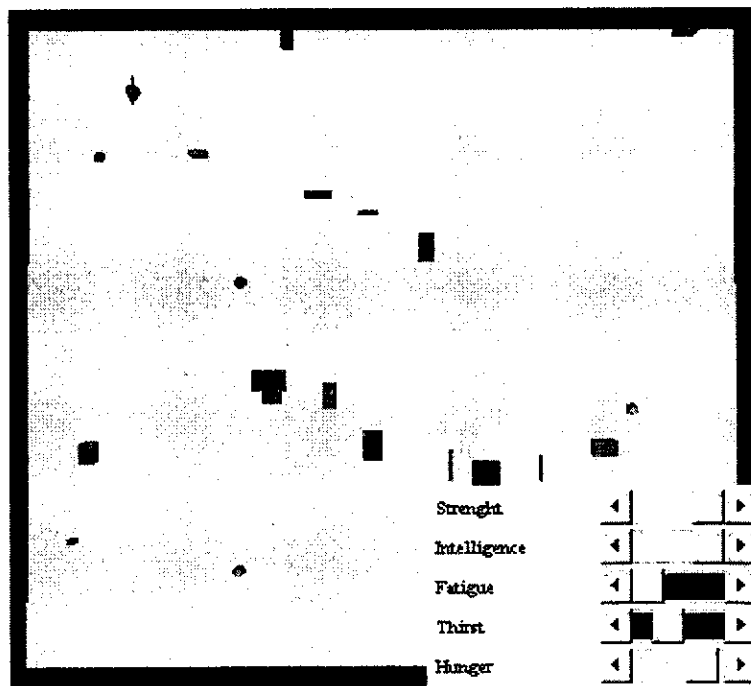


Figura 6.5. Estado inicial: fuente de agua percibida, fuente de comida no percibida, poca sed y mucho hambre.

Consideremos ahora un ambiente simulado como el que se muestra en la figura 6.5, en el cual el estímulo “fuente de comida” no es percibido; mientras que varios estímulos “fuente de agua” pueden ser percibidos. Consideremos además que existe una imperiosa necesidad de “comer”, mientras que la necesidad de “tomar agua” no es relevante, tal como se refleja en los valores de los estados internos en esta misma figura. Teniendo en cuenta estas consideraciones, analicemos ahora cuál será la acción ejecutada por el *animat* en cada uno de los casos $\alpha = 0.0$ y $\alpha \approx 1.0$.

Caso 1: $\alpha = 0$

Recordemos que cuando α es igual a 0, una conducta elemental *congruencia propio/externo/drive* podrá activarse sólo cuando coinciden las señales externas e internas asociadas a ésta. Luego, en este caso la conducta elemental *congruencia propio/externo/drive* asociada al estado interno “sed” podrá ser activada, ya que para la misma existe una entrada externa aprovechable y un estado interno, que aunque no es muy elevado, es diferente de cero; mientras que la conducta elemental *congruencia propio/externo/drive* asociada al estado interno “hambre” no podrá ser activada, ya que aunque el valor de este estado interno es muy elevado, no existe una señal externa capaz de satisfacer al mismo (recordemos que la comida no ha sido percibida, ésta se encuentra retirada).

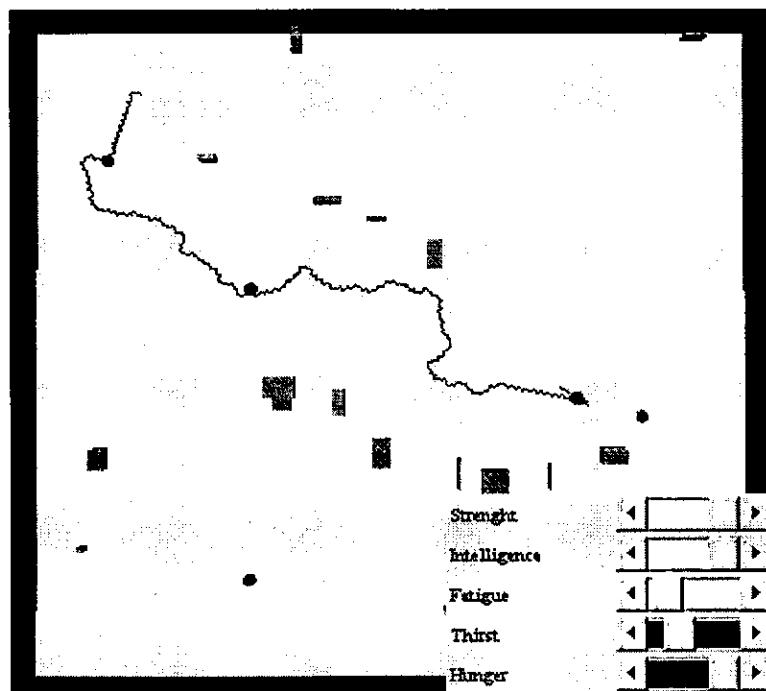


Figura 6.6. Trayectoria y acciones ejecutadas por el *animat*, considerando el estado inicial mostrado en la figura 5.14, y un valor de α igual a 0.

En virtud de lo anterior, y como se puede apreciar en la figura 6.6, lo que ha ocurrido es que la conducta elemental *congruencia propio/externo/drive* asociada a la entrada interna “sed” triunfó a nivel motivacional, por lo que la conducta externa “tomar agua” fue ejecutada, y el *animat* ha perdido tiempo en encontrar la comida, mientras que el estado interno “hambre” continúa incrementándose.

Caso 2: $\alpha \approx 1$

Aquí debemos tener en cuenta que cuando $\alpha \approx 1$, al estado interno asociado a una conducta elemental *congruencia propio/externo/drive* se le está otorgando un poco más de peso que a las entradas externas asociadas a ésta, de forma tal que, aún en ausencia total de entradas externas, esta conducta elemental podría ser activada para un valor muy fuerte del estado interno. Por lo tanto, en este caso tanto la conducta elemental *congruencia propio/externo/drive* asociada al estado interno “sed”, como la asociada al estado interno “hambre” podrán activarse, por lo que en el proceso competitivo que se lleva a cabo a nivel motivacional participarán ambas conductas elementales.

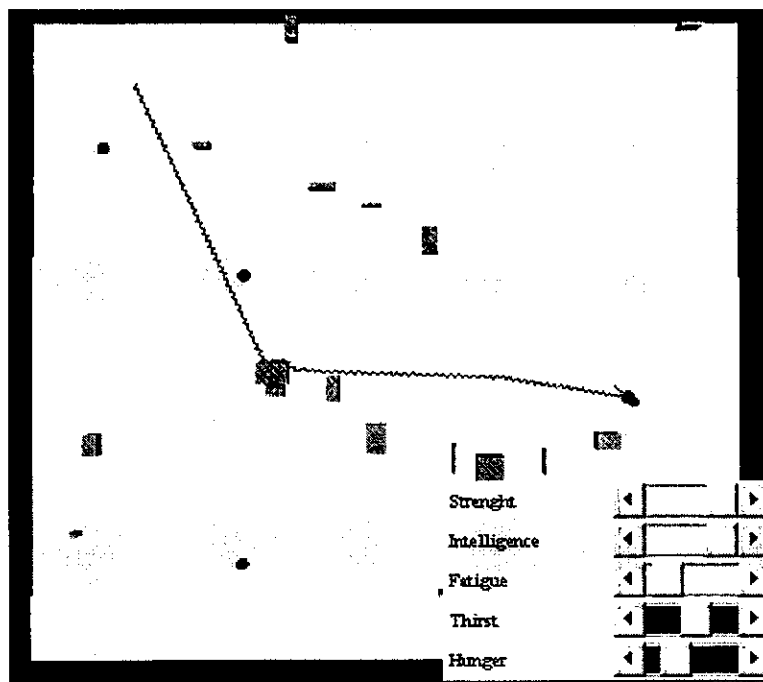


Figura 6.7. Trayectoria y acciones ejecutadas por el *animat*, considerando el estado inicial mostrado en la figura 5.14, y un valor de α muy cercano a 1.

En virtud de lo anterior, y como se puede apreciar en la figura 6.7, para un valor de α muy grande (cercano a 1), la conducta elemental *congruencia propio/externo/drive* asociada a la entrada interna “hambre” resultó ganadora en la competencia, dado el bajo valor de la entrada interna

“sed”. Este hecho provocó la “preactivación” de las condiciones de las conductas elementales *atención a preferencias* que operan sobre el pizarrón del nodo cognitivo, sin que ninguna de éstas lograra ser aún activada. Como consecuencia de lo anterior, fue activada la conducta externa exploración orientada a la búsqueda de la señal externa necesaria, siendo en este caso “fuente de comida”.

Luego, al generarse la conducta exploratoria orientada a la búsqueda de una señal específica, el *animat* tiene una mayor probabilidad de encontrar la fuente de comida mucho más rápido que en el caso anterior, cuando α era igual a 0, lo cual resulta ser más adaptativo ya que su necesidad imperiosa es comer. A esto hay que añadir que el tiempo de reacción entre el momento en que se percibe la fuente de comida directamente y el momento en que la conducta “comer” es ejecutada es mucho menor, aunque hayan otras entradas internas presentes.

6.4. Estabilidad en la selección y persistencia en la ejecución de las conductas externas

Consideremos un estado similar al mostrado en la figura 6.8, en el cual el *animat* tiene sed y hambre, tanta sed como hambre; y éste se encuentra posicionado a la misma distancia de una fuente de comida que de una fuente de agua, las cuales está percibiendo.

La estabilidad en la selección de acciones puede ser explicada en los siguientes términos. Una vez que se haya iniciado un proceso competitivo entre conductas elementales *selector de preferencias consumatorias* a nivel del nodo motivacional, entonces este proceso siempre convergerá a un estado en el cual una y sólo una conducta elemental será proclamada ganadora. Como consecuencia de lo anterior, sólo una conducta externa será ejecutada en un momento dado, ya sea ésta apetitiva o consumatoria.

En la simulación esta estabilidad es lograda atribuyendo un poco más de importancia a ciertos tipos de estímulos ambientales respecto a otros, importancia ésta que se refleja en las fuerzas de apareo de las conductas elementales *congruencia propio/extero/drive*. Es así como, para el estado inicial mostrado en la figura 6.8, la conducta externa “aproximarse al agua” será preferida en lugar de la conducta externa “aproximarse a la comida”, dado que se le ha atribuido algo más de importancia al estímulo fuente de agua que al estímulo fuente de comida. Este resultado es mostrado en la figura 6.9. De esta forma, la estabilidad en la selección de acciones lleva implícita la “no indecisión” respecto a qué conducta externa ejecutar, cuando varias conductas externas igualmente requeridas por el *animat*, han logrado satisfacer sus condiciones ambientales.

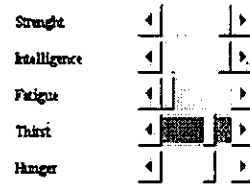


Figura 6.8. Estado inicial que muestra que el *animat* tiene tanta sed como hambre, y que éste se encuentra ubicado a igual distancia de la fuente de comida que de la fuente de agua.

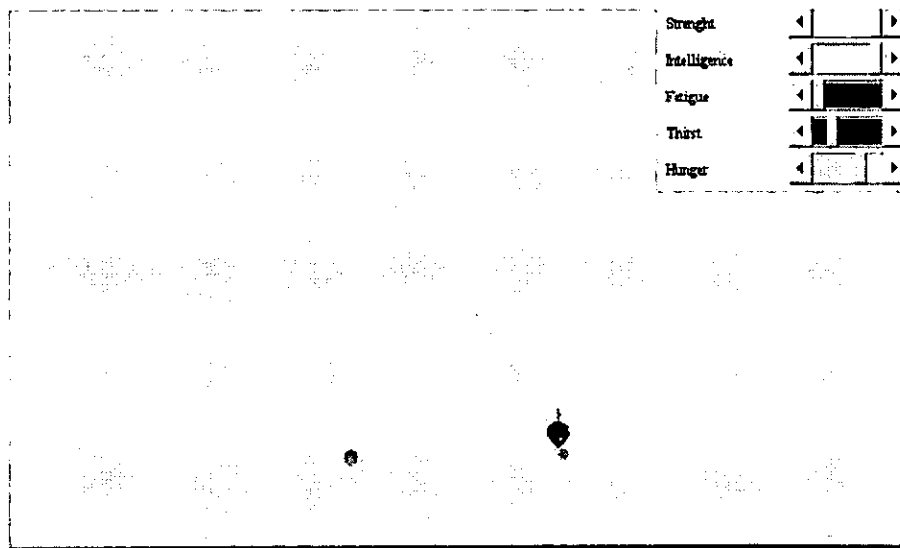


Figura 6.9. Estado en el cual la conducta externa "aproximarse al agua" ha sido ejecutada, y la conducta "tomar agua" se está ejecutando de forma persistente.

La persistencia en la ejecución de las conductas externas significa que una vez que la ejecución de una conducta externa ha sido iniciada, el *animat* siempre intentará finalizar ésta, evitando que otras posibilidades distraigan su atención. Para comprender las consecuencias originadas por la falta de persistencia en la ejecución de acciones, consideremos el siguiente ejemplo.

Supongamos un estado inicial como el mostrado en la figura 6.8 y supongamos, además, que la conducta externa “aproximarse al agua” fue ejecutada, y como resultado de dicha acción la fuente de agua se encuentra al alcance del *animat*, por lo que la conducta externa “tomar agua” es ahora ejecutada. Pero una vez que el *animat* consume una pequeña cantidad de la fuente de agua, éste se encuentra ahora con más hambre que sed, por lo que se aproxima a la fuente de comida y toma parte de ésta. Ahora el *animat* está con más sed que hambre, por lo que se aproxima a la fuente de agua y consume otra pequeña cantidad de ésta, y continúa iterando entre la fuente de agua y la fuente de comida, hasta que posiblemente el agotamiento lo detenga.

Para evitar esta continua iteración (*switching*) entre conductas externas, provocada por pequeños cambios en la motivaciones y en los estímulos ambientales, la persistencia en la ejecución de las conductas externas ha sido modelada en la red de conductas internas. Como se puede apreciar en las figuras 6.9 y 6.10, el *animat* es persistente en la ejecución de las conductas externas. En la figura 6.9 se observa como el *animat* finaliza la ejecución de la conducta “tomar agua”, sin que la presencia del estímulo “fuente de comida” distraiga la atención del *animat*, aunque este último estímulo está siendo percibido. La finalización de esta conducta se puede apreciar claramente en la considerable reducción de la fuerza de la necesidad interna sed. La figura 6.10 muestra como una vez que la ejecución de la conducta externa “tomar agua” ha finalizado, entonces el *animat* se aproxima a la comida e inicia la ejecución de la conducta externa “comer” hasta que la fuerza del estado interno hambre es reducida lo suficiente.

A pesar de que otros tipos de estímulos ambientales no han sido considerados en esta simulación, por ejemplo, la presencia de un depredador, en la red de conductas internas también es posible modelar la no persistencia en la ejecución de una conducta externa, cuando otra conducta mucho más relevante debe ser ejecutada.

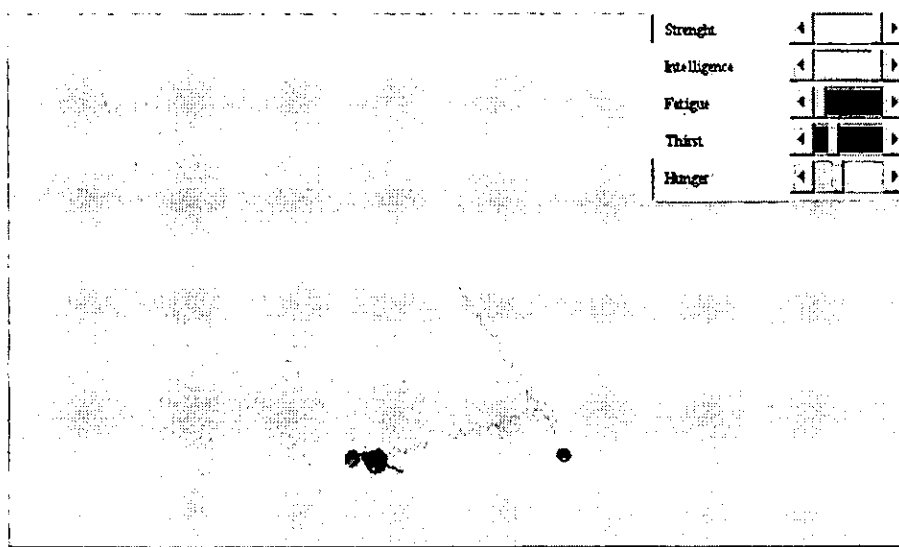


Figura 6.10. Estado en el cual las conductas externas “tomar agua” y “aproximarse a la comida” han sido ejecutadas, y la conducta “comer” se está ejecutando de forma persistente.

6.5. Discriminación entre diferentes estímulos tomando en cuenta la calidad de los mismos

Supongamos un estado inicial como el que se muestra en la figura 6.11, en el cual el *animat* tiene sed y hambre, tanta sed como hambre; y éste percibe fuentes de agua y de comida, en particular, una fuente de agua aislada, una fuente de comida aislada y una fuente de agua junto a una fuente de comida. Dada la cercanía de la fuente de agua a la fuente de comida, consideraremos a ambas fuentes formando un nuevo tipo de estímulo, fuente de agua y comida.

La propiedad “discriminación entre diferentes estímulos tomando en cuenta la calidad de los mismos” establece, para la situación antes descrita, que el mejor de estos tres estímulos es la fuente de agua y comida, dados los valores actuales de los estados internos sed y hambre del *animat*. Por lo que debía esperarse que la primera acción externa que ejecute el *animat*, una vez llevado a cabo el proceso competitivo a nivel motivacional, sea “aproximarse a la fuente de agua y comida”.

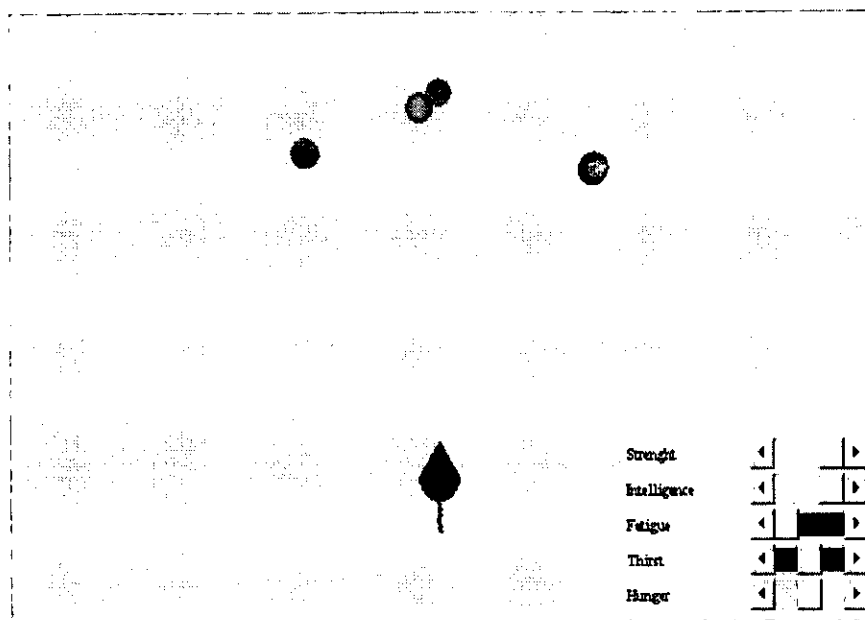


Figura 6.11. Estado inicial del *animat*. Hay sed y hambre, tanta sed como hambre; a la vez que se perciben una fuente de agua, una fuente de comida, y una fuente de agua y comida.

En la figura 6.12 se muestra un estado en el cual la conducta externa “aproximarse a la comida y al agua” fue ejecutada hasta que la fuente de agua y comida se encontró al alcance del *animat*. Una vez que el estímulo fuente de agua y comida fue alcanzado por el *animat*, fue ejecutada entonces la conducta externa “tomar agua”, ya que ganó la competencia a nivel motivacional la conducta elemental *selector de preferencias consumatorias* asociada al estado interno sed. Como resultado de la persistencia de la acción consumatoria “tomar agua”, la fuerza del estado interno sed fue decrementada, lo cual en turno permitió que la conducta elemental *selector de preferencias consumatorias* asociada al estado interno hambre ganara en esta ocasión la

competencia a nivel motivacional, y como resultado de esto la conducta externa “comer” fuera ejecutada, hasta que la fuerza del estado interno hambre disminuyera lo suficiente.

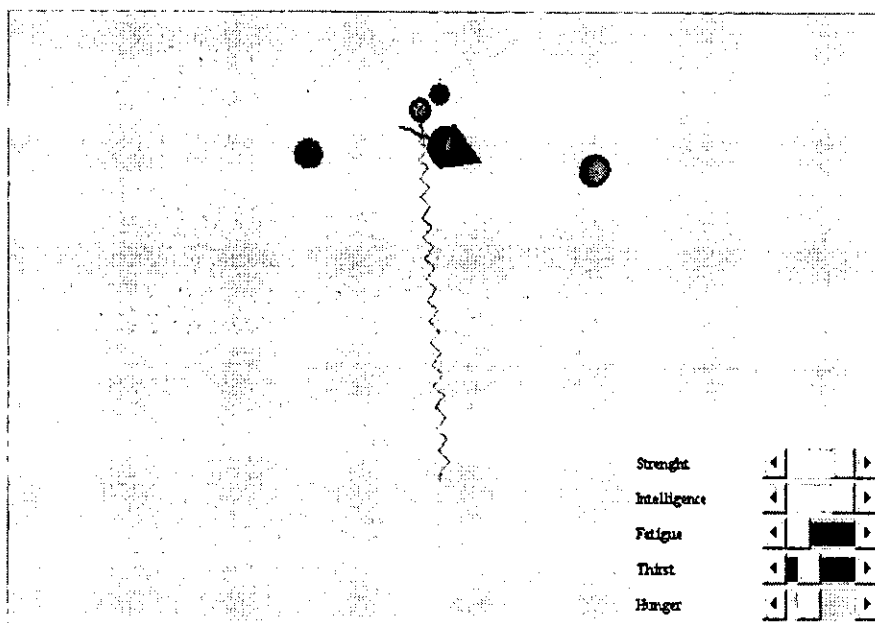


Figura 6.12. Estado alcanzado una vez ejecutadas las conductas externas “aproximarse a la comida y al agua”, “tomar agua” y “comer”, después que los correspondientes procesos competitivos a nivel de las conductas elementales selector de preferencias consumatorias activadas han tomado lugar.

La consecuencia directa de que el *animat* haya considerado el estímulo fuente de agua y comida como el de mayor calidad entre todos los estímulos percibidos en ese momento, fue una reducción en la cantidad de acciones externas que éste debía ejecutar para satisfacer sus dos necesidades internas predominantes, sed y hambre. Es decir, una vez que el estado interno sed fue satisfecho, el *animat* no tuvo que ejecutar la conducta externa “aproximarse a la comida”, para posteriormente ejecutar la conducta externa “comer”, y satisfacer de esta forma el estado interno hambre; ya que la comida se encontraba al alcance y, por lo tanto, la conducta externa consumatoria “comer” pudo ser ejecutada sin necesidad de que la precediera una conducta externa apetitiva.

6.6. Evitar estímulos aversivos

La propiedad “evitar estímulos aversivos” puede ser explicada en los siguientes términos: siempre que un estímulo aversivo es percibido por el *animat*, entonces la conducta elemental *selector de preferencias consumatorias* asociada al estado interno “seguridad” entrará a participar en el proceso competitivo que ocurre a nivel del nodo motivacional, resultando ganadora de dicha competencia siempre que la relación dada por la magnitud del estímulo aversivo y la distancia entre éste y el *animat* constituya un riesgo potencial para este último.

Para ejemplificar esta propiedad consideremos un estado inicial como el mostrado en la figura 6.13, en el cual el *animat* tiene mucho hambre; y éste percibe una fuente de comida, sin lograr aún percibir un estímulo aversivo que se encuentra ubicado en una zona posterior a la fuente de comida, respecto a la posición en la que se encuentra el *animat*. La distancia entre el estímulo aversivo y la fuente de comida es aproximadamente igual a la distancia entre esta última y el *animat*.

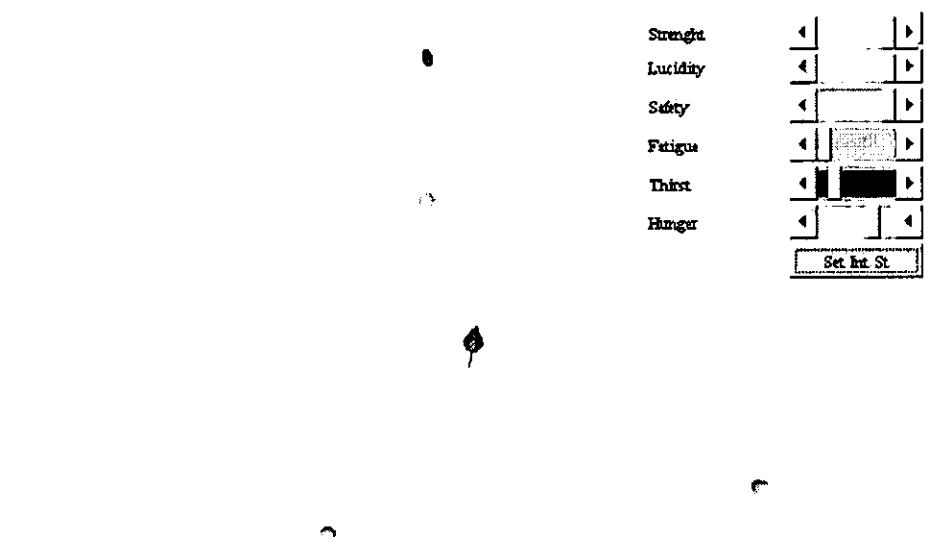


Figura 6.13. Estado inicial en el cual el *animat* tiene mucho hambre y percibe una fuente de comida.

En la figura 6.14 se muestra un estado en el cual la conducta externa “aproximarse a la comida” ha sido ejecutada, aunque no concluida. Como resultado de la ejecución de esta acción, el *animat* se encuentra ahora más cerca de la fuente de comida y por lo tanto también del estímulo aversivo, sin lograr aún percibirlo, ya que éste cae fuera de su región perceptual. Luego, en virtud de la propiedad que establece el carácter persistente de la ejecución de las acciones externas, la ejecución de la conducta externa “aproximarse a la comida” no será interrumpida, a menos que otra conducta externa mucho más relevante para la situación actual deba ser ejecutada.

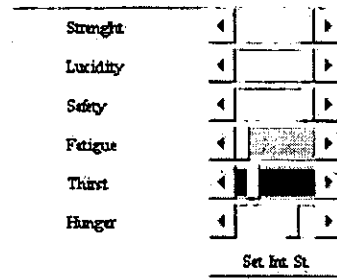


Figura 6.14. La conducta externa “aproximarse a la comida” está siendo ejecutada. El *animat* aún no ha percibido el estímulo aversivo.

Como se puede apreciar en la figura 6.15, el *animat* continúa la ejecución de la conducta externa “aproximarse a la comida” hasta que el estímulo aversivo es percibido por éste. Como consecuencia de la relación dada por la proximidad del estímulo aversivo a la fuente de agua y la magnitud de éste, la conducta externa “alejarse de estímulo aversivo” es disparada, interrumpiéndose de esta forma la ejecución de la acción “aproximarse a la comida”.

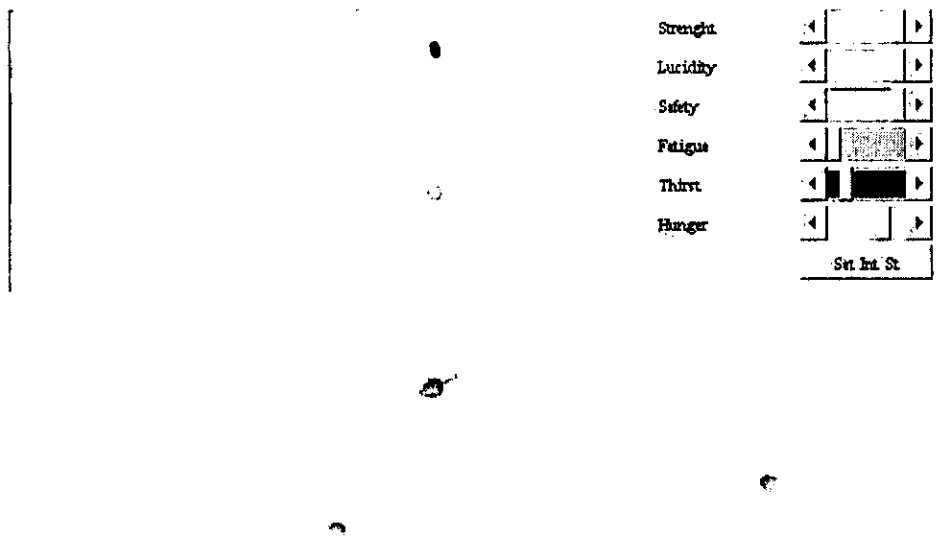


Figura 6.15. La acción “aproximarse a la fuente de comida” es interrumpida al ser percibido un estímulo aversivo cercano a ésta.

Una vez que el *animat* se ha alejado lo suficiente del estímulo aversivo, nuevamente la conducta externa “explorar” es activada para encontrar una fuente de comida que pueda satisfacer la imperiosa necesidad interna del mismo. La figura 6.16 muestra un patrón de comportamientos en el cual se pueden apreciar todas las acciones externas ejecutadas por el *animat*: “aproximarse a la comida”, “alejarse de estímulo aversivo”, “explorar en busca de comida”, aproximarse a la comida” y “comer”. Como resultado de la ejecución de esta última acción, el elevado nivel de la necesidad interna hambre ha sido disminuido considerablemente, en virtud del carácter persistente de la ejecución de las acciones.

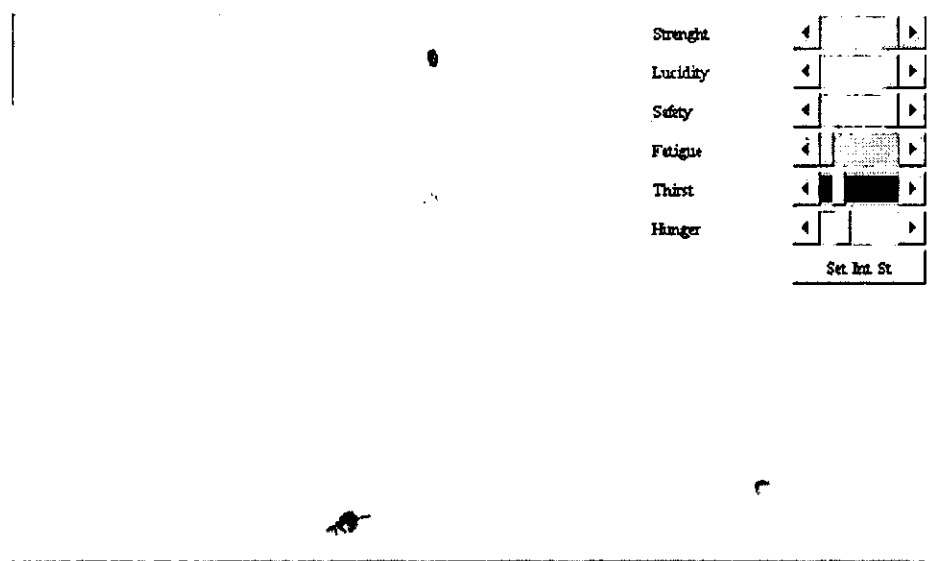


Figura 6.16. Patrón de comportamientos que muestra todas las acciones externas ejecutadas por el *animat*.

6.7. La no persistencia en la ejecución de una acción consumatoria ante la percepción de un estímulo aversivo

Como ya se mostró en el experimento 6.4, la persistencia en la ejecución de las conductas externas significa que una vez que la ejecución de una conducta externa ha sido iniciada, el *animat* siempre intentará finalizar ésta evitando que otros estímulos, irrelevantes por el momento, distraigan su atención. Sin embargo, la no persistencia en la ejecución de una acción consumatoria siempre tendrá lugar cuando un estímulo aversivo ha sido percibido y la relación dada por la magnitud del estímulo aversivo y la distancia entre éste y el *animat* constituya un riesgo potencial para este último.

Para ejemplificar esta propiedad, consideremos un estado inicial como el que se muestra en la figura 6.17, en el cual el *animat* tiene mucha sed y está percibiendo una fuente de agua y no percibe aún un estímulo aversivo que se encuentra también en el medio ambiente, más cercano a la fuente de agua que al *animat*.

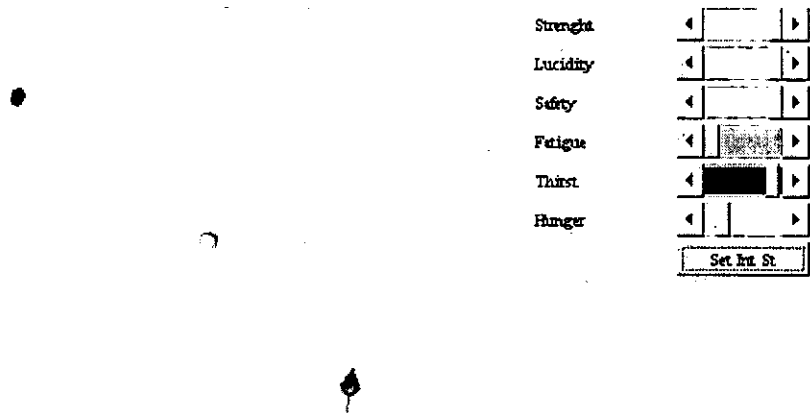


Figura 6.17. Estado inicial en el cual el *animat* tiene mucha sed y percibe una fuente de agua.

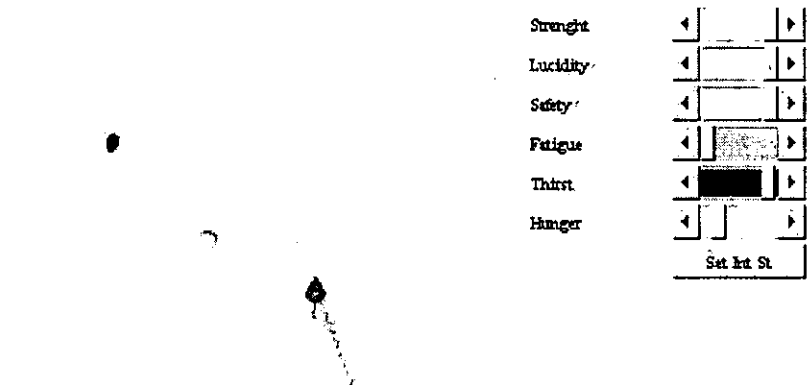


Figura 6.18. La conducta externa "aproximarse al agua" está siendo ejecutada. El *animat* aún no ha percibido al estímulo aversivo.

En la figura 6.18 se muestra un estado en el cual la conducta externa “aproximarse al agua” ha sido ejecutada, por lo que el *animat* se encuentra ahora más cerca de la fuente de agua, pero aún no ha percibido al estímulo aversivo, el cual se ha desplazado y también se encuentra ahora más próximo a la fuente de agua. Luego, en virtud de la propiedad que establece el carácter persistente de la ejecución de las acciones externas, la ejecución de la conducta externa “aproximarse al agua” continuará siendo ejecutada, a menos que otra conducta externa mucho más relevante para la situación actual, tal como “alejarse de estímulo aversivo”, deba ser ejecutada.

Como se puede apreciar en la figura 6.19, el *animat* continuó la ejecución de la conducta externa “aproximarse a la fuente de agua” y una vez que esta última estuvo a su alcance inició la ejecución de la conducta externa “tomar agua”, la cual tuvo que ser interrumpida antes de que concluyera, dada la proximidad del estímulo aversivo a la fuente de agua. Como consecuencia, la conducta externa “alejarse de estímulo aversivo” fue ejecutada. En la figura 6.19 se puede observar como el estado interno sed fue disminuido como resultado de la ejecución de la conducta “tomar agua”, aunque esta última no fue lo suficientemente persistente como para reducir aún más el nivel de esta necesidad interna.

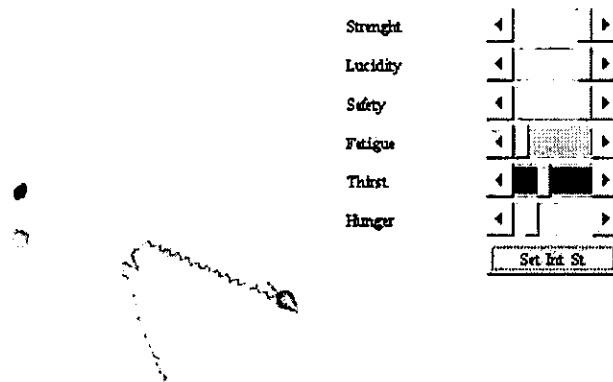


Figura 6.19. La acción “tomar agua” es interrumpida dada la cercanía del estímulo aversivo a la fuente de agua. La acción “alejarse de estímulo aversivo” es ejecutada.

Como muestra la figura 6.20, una vez alejado del estímulo aversivo, el *animat* comienza nuevamente a explorar en busca de una fuente de agua para completar la consumación de la conducta “tomar agua”. Una fuente de agua es percibida y el *animat* se aproxima a ésta y completa la acción interrumpida por la presencia del estímulo aversivo.

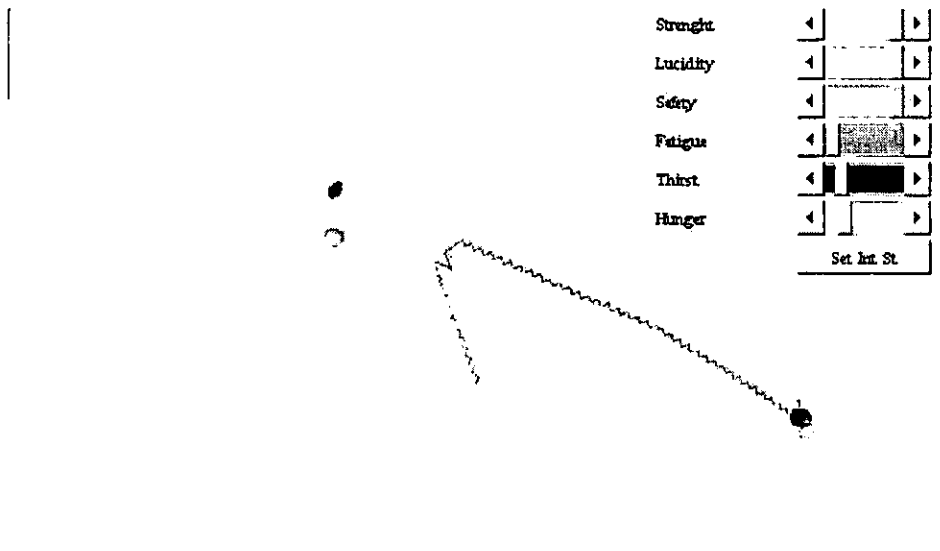


Figura 6.20. Patrón de comportamientos que muestra todas las acciones externas ejecutadas por el animat.

6.8. Condicionamiento clásico primario

En la red de conductas internas se pueden apreciar dos formas de aprendizaje asociativo: el condicionamiento clásico primario y el condicionamiento secundario. Como ya se discutió en los epígrafes 4.13.1 y 4.13.2, ambas formas de aprendizaje tienen lugar a través de la modificación de las fuerzas de apareo de determinadas conductas internas de los tipos *selección de conductas externas* (Fa SCE4), *congruencia propio/externo/drive* (Fa CPED2) y *atención a preferencias* (Fa AP3); de forma tal que “columnas conductuales” que antes permanecían inactivas ante la proyección en el nivel Percepciones Externas de sus correspondientes estímulos externos, ahora logran volverse activas, permitiendo que estos estímulos ambientales (inicialmente neutros) ahora cobren un significado específico para la entidad.

Las fuerzas de apareo SCE4, CPED2 y AP3 poseen desde un inicio valores muy bajos cercanos a cero (cada una de estas fuerzas de apareo es inicializada en 0.001), de forma tal que aunque determinados segmentos de una potencial columna conductual logren ser mapeados sobre los pizarrones de los nodos cognitivo y motivacional; la columna conductual completa no podrá ser activada hasta que estas fuerzas de apareo alcancen valores lo suficientemente fuertes.

6.8.1. Asociación de un estímulo neutro con una fuente de comida

Las figuras 6.21, 6.22 y 6.23 ilustran diferentes etapas dentro del proceso de condicionamiento clásico primario. En este experimento se ha considerado un determinado número de situaciones, en las cuales el estado interno predominante del animat es hambre y en la gran mayoría de estas situaciones, antes de que el estímulo fuente de comida sea percibido por el animat, éste siempre percibe una mancha roja muy próxima a la fuente de comida.

La figura 6.21 muestra los valores de las fuerzas de apareo SCE4, CPED2 y AP3 después de un reducido número de presentaciones de pares mancha roja - fuente de comida, para un predominante estado interno hambre. Como se puede apreciar en esta figura, ya ha iniciado el proceso de condicionamiento clásico primario, expresado en el reforzamiento de estas fuerzas de apareo. No obstante, aunque los valores de las Fa modificables han sido reforzados, éstas no poseen aún valores lo suficientemente fuertes como para que el estímulo en proceso de condicionamiento, mancha roja, evoque una conducta externa por sí mismo, en este caso “aproximarse a la comida”, dado el predominante estado interno hambre.

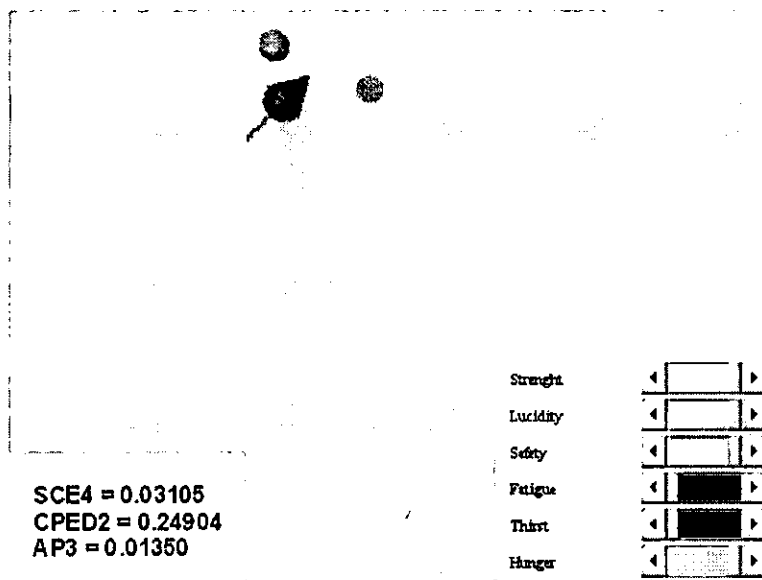


Figura 6.21. Valores de las fuerzas de apareo SCE4, CPED2 y AP3 después de un reducido número de presentaciones de pares mancha roja - fuente de comida.

En la figura 6.22, se puede apreciar una etapa muy avanzada del proceso de condicionamiento clásico primario, teniendo en cuenta que los valores máximos de las Fas SCE4, CPED2 y AP3 son 3.0, 1.0 y 1.0, respectivamente. Como ilustra esta imagen, el *animat* ha continuado percibiendo el par mancha roja – fuente de comida en este mismo orden, de forma tal que en las Fas se puede apreciar una tendencia a sus respectivos valores límites.

Para los valores actuales de las Fas mostrados en la figura 6.22, el estímulo mancha roja ha dejado de ser un estímulo neutro o en proceso de condicionamiento para convertirse en un estímulo ya condicionado, por lo que para un fuerte estado interno hambre, éste será capaz por sí sólo de evocar la conducta externa que evocaba el estímulo incondicionado (fuente de comida) con el cual se condicionó (es decir, aproximarse a la comida). Lo anterior puede ser apreciado en la figura 6.23. En este caso nuevamente se ha considerado que el estado interno predominante es hambre, y una situación inicial en la cual el *animat* no percibe la fuente de comida.

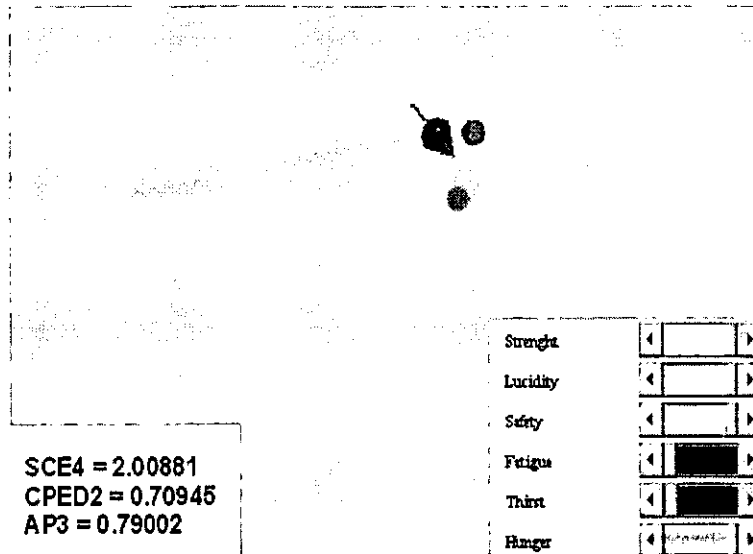


Figura 6.22. Valores de las fuerzas de apareo SCE4, CPED2 y AP3 en una etapa avanzada del proceso de condicionamiento.

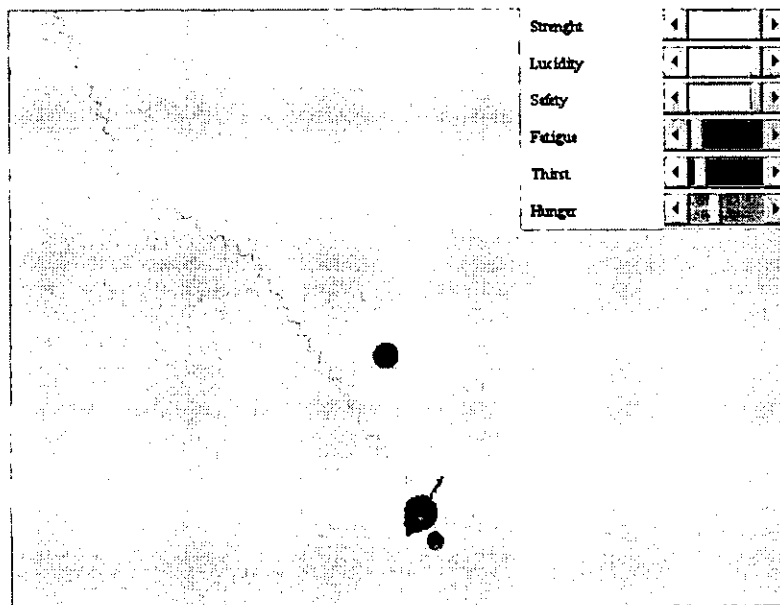


Figura 6.23. Patrón de comportamientos ejecutado por el *animat* una vez que el estímulo mancha roja se ha convertido en un estímulo condicionado.

El patrón de comportamientos ilustrado en la figura 6.23 explica claramente lo ocurrido. El estado interno hambre gana la competencia a nivel motivacional, por lo que la conducta externa explorar es activada. El *animat* inicia la exploración hasta que la mancha roja es percibida, por lo que la conducta externa aproximarse a la comida es activada, en virtud de que el estímulo mancha

roja se condicionó con fuente de comida y por lo tanto éste se ha convertido ahora en una señal externa aprovechable para el estado interno hambre. Luego, la conducta externa aproximarse a la comida (la cual se traduce en aproximarse a la mancha roja) es ejecutada hasta que el *animat* percibe una fuente de comida cercana a la mancha roja y entonces nuevamente la conducta aproximarse a la comida es ejecutada, pero esta vez teniendo como estímulo blanco la fuente de comida. Una vez que la fuente de comida se encuentra al alcance, la conducta externa comer inicia su ejecución y como consecuencia el nivel del estado interno hambre es decrementado.

Es de esta forma, como el condicionamiento clásico primario logra mejorar la selección de acciones haciéndola más adaptativa. Es decir, estímulos que inicialmente no poseen significado alguno para la entidad, pueden llegar a jugar un papel determinante en la satisfacción de imperiosas metas de la entidad.

6.8.2. Asociación de un estímulo neutro con un estímulo aversivo

En el experimento anterior se pudo comprobar la asociación entre un estímulo neutro y un estímulo incondicionado, el cual es capaz de satisfacer una necesidad interna del animat. En este experimento se ilustrará el proceso de condicionamiento entre un estímulo neutro y un estímulo incondicionado, siendo en este caso el estímulo incondicionado una señal aversiva que el animat intentará evitar.

En la figura 6.24 se ilustra una situación en la cual el animat tiene sed, y está explorando en busca de agua. Durante la trayectoria de la exploración, se encuentra con una mancha roja, la cual está próxima a un estímulo aversivo. Primero percibe la mancha roja, y después el estímulo aversivo, lo que inicia el condicionamiento entre estos dos. Al percibir el estímulo aversivo, el animat ejecuta la conducta externa alejarse de estímulo aversivo.

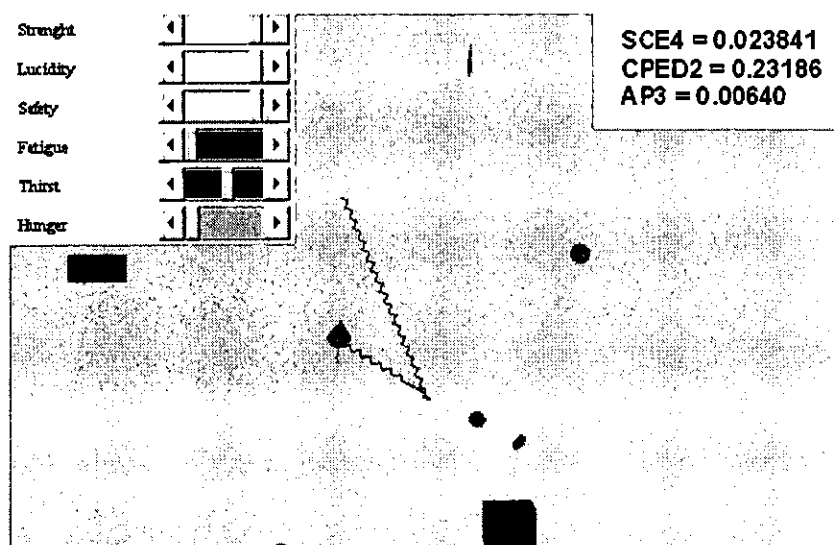


Figura 6.24. El *animat* percibe una mancha roja cerca de un estímulo aversivo. La conducta externa alejarse de estímulo aversivo es ejecutada.

Consideremos ahora que en varios experimentos subsecuentes, el animat ha encontrado manchas rojas cerca de estímulos aversivos durante la ejecución de determinadas conductas externas, tales como explorar, deambular, o aproximarse a un estímulo particular. Este hecho ha reforzado el condicionamiento entre ambos estímulos. Por otra parte, también se considera la existencia de un fuerte condicionamiento entre los estímulos mancha amarilla y fuente de agua.

En la figura 6.25 se ilustra la siguiente situación: el animat tiene sed, y está explorando en busca de una fuente de agua. Éste percibe una mancha amarilla, la cual está condicionada con el estímulo fuente de agua, por lo que se aproxima a ella. Una vez cerca de la mancha amarilla, el animat logra percibir una fuente de agua, por lo que se aproxima a su vez a ésta. La conducta aproximarse al agua es interrumpida debido a que el animat percibe una mancha roja, la cual se ha condicionado con el estímulo aversivo, y aunque este último no es percibido por el animat, éste ejecuta la acción alejarse de estímulo aversivo.

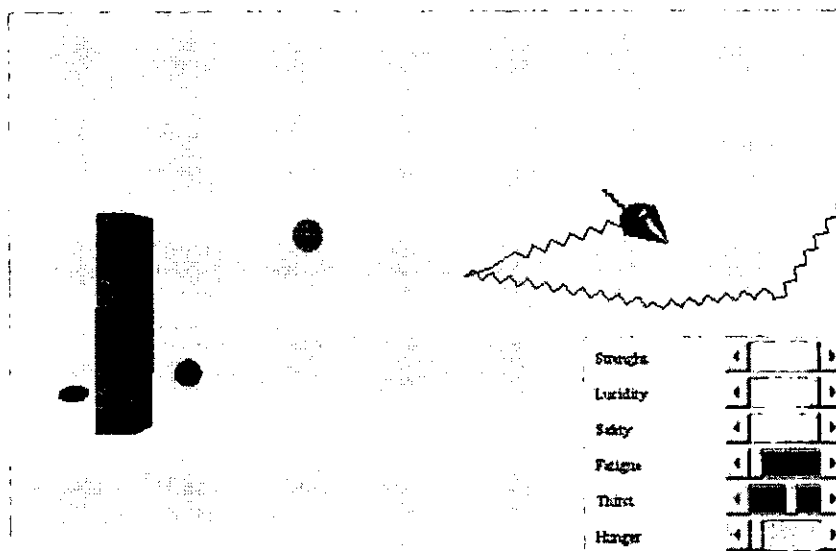


Figura 6.25. El animat tiene sed y está explorando en busca de una fuente de agua, al percibir una mancha roja, la conducta evitar estímulo aversivo es ejecutada.

De esta forma, el condicionamiento clásico primario puede tener lugar entre estímulos neutros y estímulos incondicionados capaces de satisfacer fuertes necesidades internas de la entidad; y entre estímulos neutros y estímulos que representan señales aversivas.

6.9. Condicionamiento secundario

Para comprobar la existencia del condicionamiento clásico secundario en la red de conductas internas, se tuvieron en cuenta tres tipos de estímulos externos: un estímulo incondicionado (fuente de comida), un estímulo previamente condicionado (mancha roja) y un estímulo neutro (mancha amarilla); considerando nuevamente al hambre como la necesidad interna más fuerte.

La figura 6.26 muestra una situación en la cual el condicionamiento para el estímulo mancha amarilla está teniendo lugar. De la misma forma en la que la mancha roja fue condicionada con la fuente de comida, ocurre el condicionamiento de la mancha amarilla con la mancha roja, jugando en este caso la mancha roja el papel de estímulo condicionador. Lo anterior puede ser apreciado teniendo en cuenta los valores actuales de las fuerzas de apareo para el estímulo mancha amarilla (los valores que aparecen después de la coma).

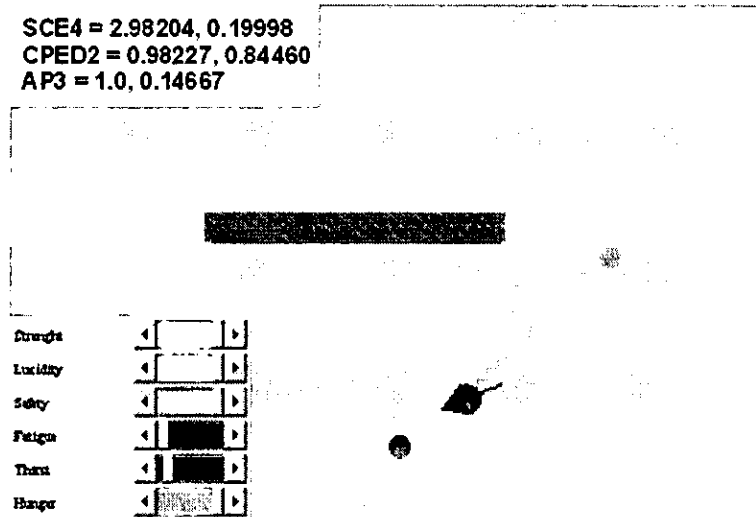


Figura 6.26. Situación en la cual el condicionamiento para el estímulo mancha amarilla está teniendo lugar.

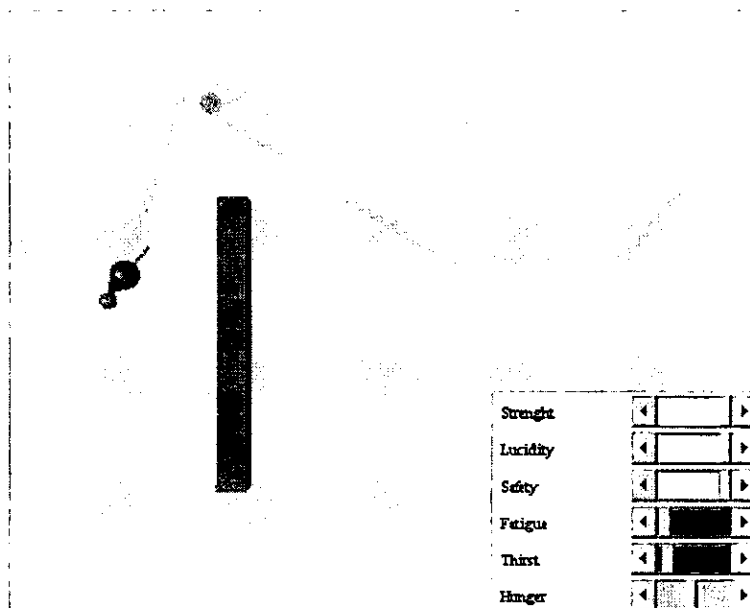


Figura 6.27. Patrón de comportamientos ejecutado por el animat, una vez que el condicionamiento entre los estímulos mancha amarilla y mancha roja ha ocurrido.

Una vez que el estímulo mancha amarilla se ha condicionado lo suficiente, éste puede evocar la misma conducta externa que evocaba por sí solo el estímulo mancha roja. La figura 6.27 muestra claramente esta situación. El animat al estar explorando en busca de comida, percibe una mancha amarilla (sin lograr percibir aún una fuente de comida oculta tras un obstáculo fijo), por lo que la conducta externa aproximarse a la comida (aproximándose a la mancha amarilla) es ejecutada. Antes de alcanzar la mancha amarilla, la mancha roja es percibida, por lo que la conducta aproximarse a la comida (en esta ocasión aproximándose a la mancha roja) es nuevamente ejecutada. Una vez que el animat alcanza la mancha roja, éste permanece buscando la comida en las vecindades de la mancha roja, y percibe la fuente de comida, ahora visible para el animat. Éste se aproxima a la comida para lograr que se haga ejecutable la conducta externa consumatoria comer; la cual, en turno, es ejecutada.

6.10. Propiedades del condicionamiento clásico en la red de conductas internas

Entre las principales propiedades que caracterizan el condicionamiento clásico en la red de conductas internas, se encuentran el bloqueo, la caída de la actividad del estímulo en el tiempo, la dominancia, la interrupción del condicionamiento y la pérdida del condicionamiento (olvido). Los siguientes experimentos van encaminados a comprobar estas propiedades. En cada uno de estos experimentos se ha considerado nuevamente que el estado interno predominante es hambre, el estímulo incondicionado es fuente de comida, y los estímulos neutros son ahora manchas rojas y manchas amarillas.

6.10.1. Bloqueo

Para comprobar la propiedad de bloqueo, se presentaron en varias pruebas es estímulo neutro mancha roja y el estímulo incondicionado fuente de comida sincrónicamente; para un fuerte estado interno hambre.

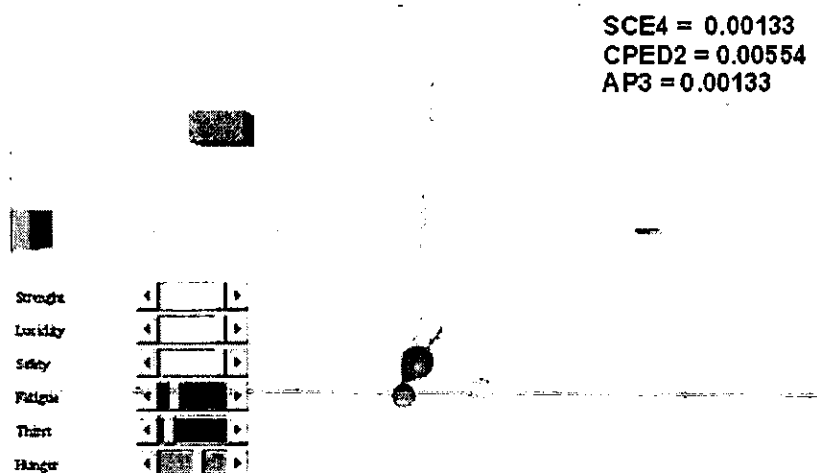


Figura 6.28. El animat percibe el estímulo neutro mancha roja y el estímulo fuente de comida sincrónicamente.

Como se aprecia en la Figura 6.28, el reforzamiento de las fuerzas de apareo es casi nulo, debido a que el estímulo incondicionado bloquea la representación del estímulo neutro en el nivel Persistentes Perceptuales al encontrarse en el mismo escenario, por lo que a éste último no se condiciona.

Después de repetir varias pruebas en diferentes contextos, pero siempre presentando ambos estímulos de forma sincrónica, tampoco fue posible obtener condicionamiento. Esto se puede observar en la Figura 6.29, donde el animat tiene hambre, y está en busca de comida. Hay manchas rojas, pero estas no tienen ningún significado para él, por lo que continúa explorando.

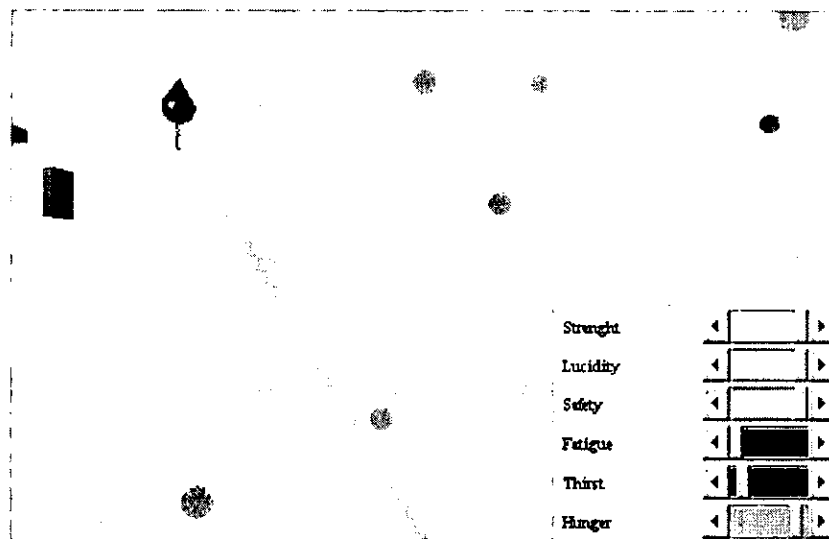


Figura 6.29. El animat percibe manchas rojas durante su exploración en busca de comida, pero éstas no se han condicionado a la fuente de comida.

6.10.2. Caída de la actividad del estímulo en el tiempo

Para comprobar la caída de la actividad del estímulo en el tiempo se han considerado varias situaciones en la cuales los estímulos neutros manchas rojas aparecen mucho antes de que aparezca el estímulo incondicionado fuente de comida, esto es, en escenarios distintos. Luego, en virtud de esta propiedad, cuando el estímulo fuente de comida es creado en el nivel Persistentes Perceptuales del pizarrón del nodo cognitivo, ya las representaciones en este mismo nivel de los estímulos neutros mancha roja que lo precedieron son demasiado débiles, por lo que el condicionamiento que estos reciben es muy pobre o ninguno.

La figura 6.30 muestra los valores de las fuerzas de apareo SCE4, CPED2 y AP3 después de varias presentaciones de pares mancha roja - fuente de comida, para valores muy altos del intervalo inter-estímulo. Como se puede apreciar en esta figura, el reforzamiento de las Fas ha

sido muy pobre (los valores de las Fas continúan siendo cercanos a 0.001), a pesar de todas las presentaciones ocurridas.

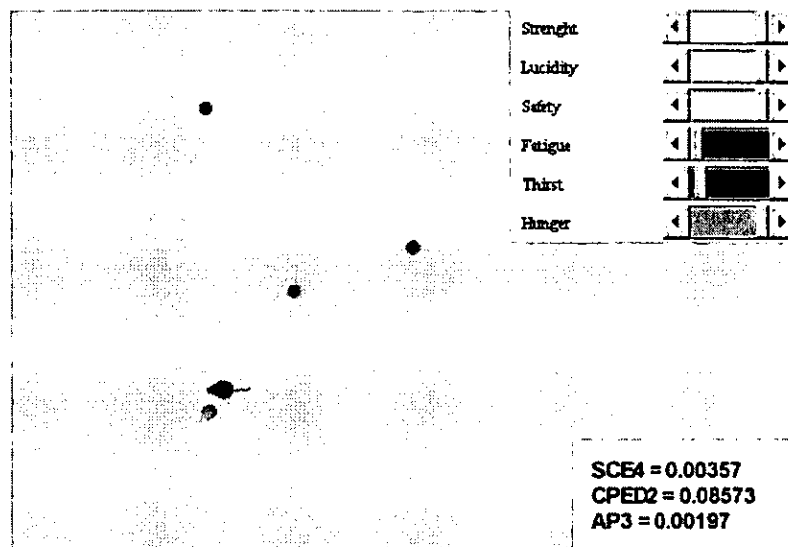


Figura 6.30. Valores de las fuerzas de apareo SCE4, CPED2 y AP3 después de varias presentaciones de pares mancha roja - fuente de comida, para valores muy altos del intervalo inter-estímulo.

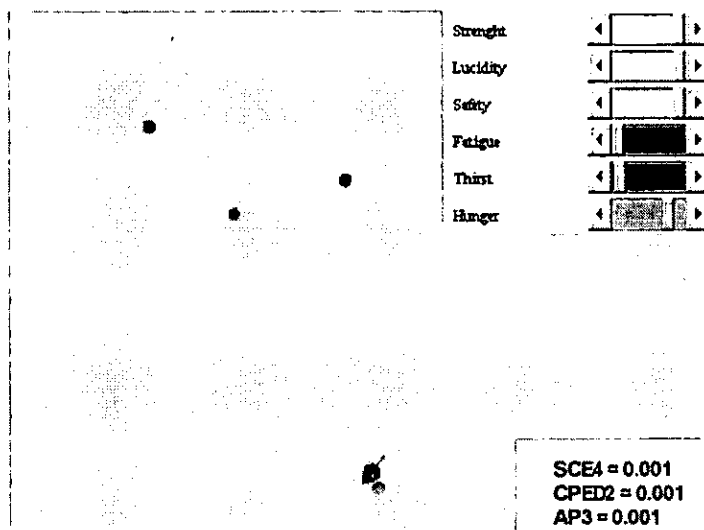


Figura 6.31. Patrón de comportamientos ejecutado por el *animat* después de muchas presentaciones del par mancha roja - fuente de comida, para valores grandes del intervalo inter-estímulo.

La figura 6.31 muestra lo ocurrido después de un considerable número de presentaciones de estímulos manchas rojas precediendo al estímulo fuente de comida, para valores del intervalo inter-estímulo muy elevados. Como se puede apreciar en esta figura, una vez que el estado interno hambre ganó la competencia a nivel motivacional, la conducta externa explorar es ejecutada y ésta continúa su ejecución en busca de una fuente de comida aunque las manchas rojas hayan sido percibidas por el *animat*. Es decir, las manchas rojas no han sido capaces de condicionarse con la fuente de comida, por lo que aún carecen de significado particular para la entidad, y como resultado éstas no son capaces de evocar por sí solas la conducta externa aproximarse a la comida. Luego, en virtud de esta propiedad, para valores muy elevados del intervalo inter-estímulo ocurre muy poco o ningún condicionamiento del estímulo neutro.

6.10.3. Dominancia

Para comprobar la propiedad de dominancia, se ejecutaron varias pruebas en cada una de las cuales al estímulo incondicionado fuente de comida lo precedían los estímulos neutros mancha roja y mancha amarilla. Para cada una de estas pruebas se consideró que la intensidad y cercanía a la fuente de comida del estímulo mancha roja fueran mayores que las correspondientes al estímulo mancha amarilla, aunque para ambos casos se consideró un intervalo inter-estímulo de magnitud media.

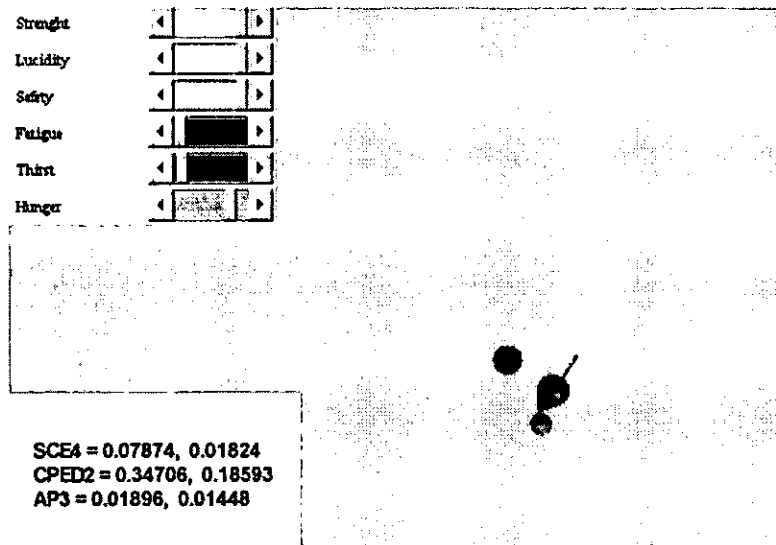


Figura 6.32. Valores de las Fas SCE4, CPED2 y AP3 para los estímulos mancha roja y mancha amarilla en una etapa intermedia del proceso de condicionamiento.

La figura 6.32 muestra los valores de las Fas SCE4, CPED2 y AP3 para los estímulos mancha roja y mancha amarilla en una etapa intermedia del proceso de condicionamiento. El valor antes de la coma corresponde al estímulo en proceso de condicionamiento mancha roja. Como se puede

apreciar en esta figura, el estímulo mancha roja es el que mayor condicionamiento ha recibido después de la ejecución de varias presentaciones. Esto significa que de continuar el proceso de condicionamiento, entonces será la mancha roja la que más fuertemente se condicione a la fuente de comida. En virtud de lo anterior, en una posterior presentación de la mancha roja y un predominante estado interno de hambre, ésta será capaz por sí misma de invocar la conducta externa aproximarse a la comida, mientras que la mancha amarilla no lo podrá lograr dado el pobre condicionamiento que ha recibido.

6.10.4. Interrupción del condicionamiento

La interrupción del condicionamiento puede ocurrir en cualquier etapa del proceso de condicionamiento. Para comprobar esta propiedad emergente del condicionamiento clásico, hemos considerado una etapa intermedia en el proceso de condicionamiento, en la cual el estímulo mancha roja ha comenzado a dejar de ser un estímulo neutro para convertirse en un estímulo condicionado por una fuente de comida, dado un fuerte valor de la necesidad interna hambre. La causa que genera esta interrupción del condicionamiento es la presencia de un estímulo aversivo.

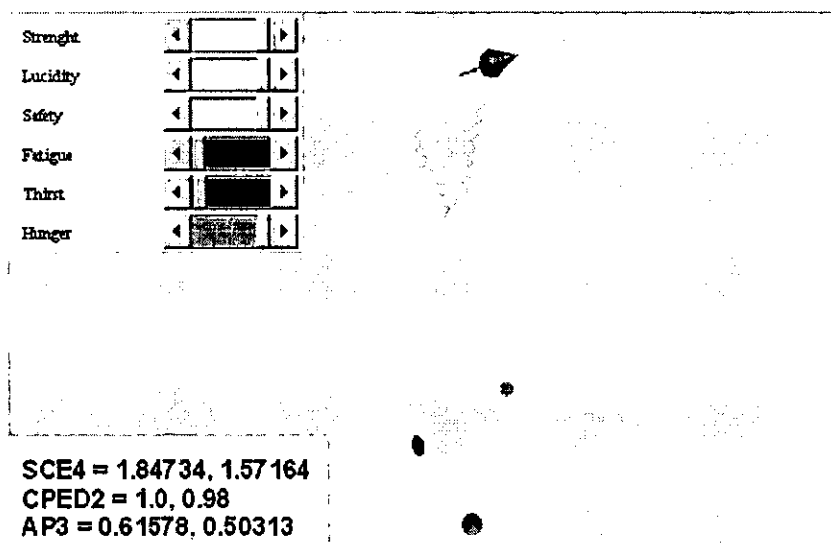


Figura 6.33. Interrupción del condicionamiento dada la presencia de un estímulo aversivo.

La figura 6.33 ilustra esta situación. El animat tiene hambre y al ser percibida la mancha roja, el animat comienza a aproximarse a la misma, dado el significado adquirido por ésta. Durante la ejecución de esta conducta, un estímulo aversivo es percibido cercano a la mancha roja, por lo que la conducta externa evitar estímulo aversivo es ejecutada. Como consecuencia de este *switching* entre conductas externas, el proceso de condicionamiento es interrumpido temporalmente, y como resultado las fuerzas de apareo sufren un ligero debilitamiento, tal como se puede apreciar en esta figura.

6.10.5. Pérdida del condicionamiento

Para comprobar la propiedad de pérdida del condicionamiento se consideró la existencia de un fuerte condicionamiento entre el estímulo mancha roja y el estímulo fuente de comida, a partir de pruebas ejecutadas previamente. Las figuras 6.34, 6.35 y 6.36 ilustran bajo qué condiciones tiene lugar el olvido o pérdida del condicionamiento.

La figura 6.34 muestra una situación en la cual el estado interno hambre gana la competencia a nivel motivacional, por lo que la conducta externa explorar es ejecutada, hasta que una mancha roja es percibida por el *animat*, por lo que la conducta aproximarse a la comida es ejecutada, dado el fuerte condicionamiento aún existente entre la mancha roja y la fuente de comida.

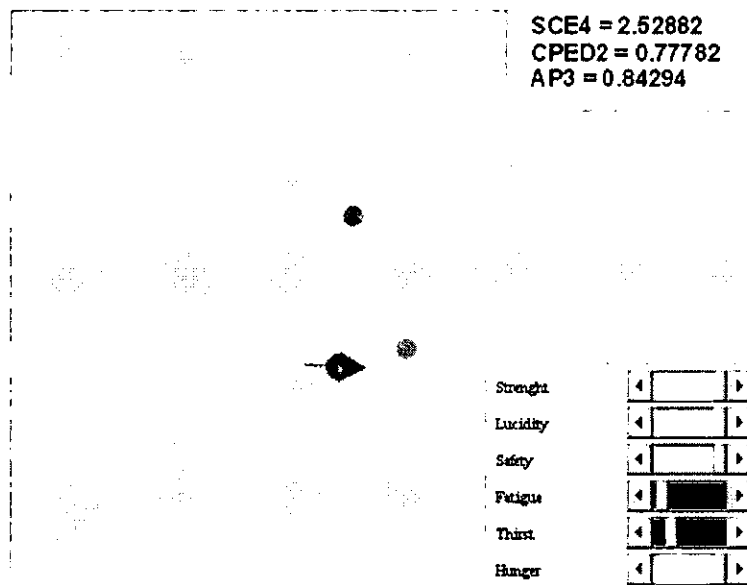


Figura 6.34. Ejecución de la conducta aproximarse a la comida (aproximándose a la mancha roja) para un fuerte estado interno hambre.

En la Figura 6.35, el animat ha llegado hasta la mancha roja sin que logre percibir una fuente de comida cercana a ésta, por lo que permanece buscado una fuente de comida en las vecindades de la mancha roja. Al no encontrar el estímulo al que la mancha roja se ha condicionado (fuente de comida) en sus escenarios, inicia el proceso de pérdida del condicionamiento entre ambos estímulos; el cual es definido por la expresión (4.3). En virtud de la aplicación de esta fórmula, las fuerzas de apareo SCE4, CPED2 y AP3 comienzan a perder fuerza, tal como se puede apreciar en la figura 6.35.

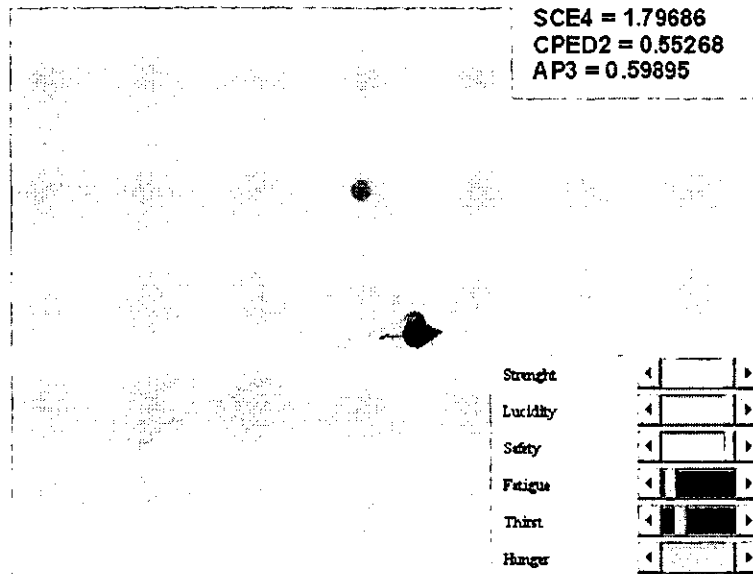


Figura 6.35. Mancha roja alcanzada por el animat sin que éste logre percibir una fuente de comida cercana a la misma. El proceso de pérdida del condicionamiento ha debilitado las fuerzas de apareo.

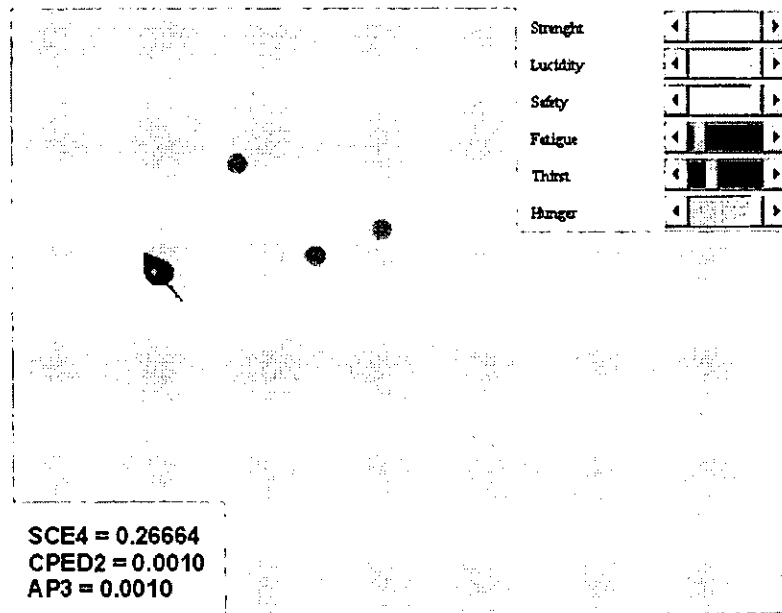


Figura 6.36. Patrón de comportamientos ejecutados por el animat una vez que se ha perdido el condicionamiento entre los estímulos mancha roja y fuente de comida.

La Figura 6.36 muestra lo ocurrido después de ejecutadas varias pruebas, para un estado interno predominante hambre y un medio en el cual la(s) mancha(s) roja(s) fue(ron) percibida(s), pero la fuente de comida no. Como se puede apreciar en esta figura, la necesidad interna predominante continúa siendo hambre, por lo que la conducta externa explorar ha sido ejecutada, pero esta vez aunque el animat percibe las manchas rojas no se aproxima a éstas ya que las fuerzas de apareo han disminuído lo suficiente y como consecuencia se ha perdido el condicionamiento entre ambos estímulos.

6.11. Acerca de los experimentos realizados

La amplia gama de experimentos realizados nos ha permitido comprobar la riqueza del conjunto de propiedades que caracterizan la selección de acciones en la red de conductas internas. Como ya se explicó en el capítulo 4, muchas de estas propiedades se derivan directamente del esquema de combinación de estímulos externos e internos utilizado por la red de conductas internas; mientras que otras propiedades, “no alambradas en la red”, emergen directamente de las distintas interacciones que tienen lugar entre las diferentes conductas elementales y mecanismos que componen la red de conductas internas (la emergencia de estas propiedades se discute en el epígrafe 7.3).

Especial énfasis fue hecho en el grupo de experimentos realizados para comprobar el rol de los procesos de aprendizaje asociativo en la mejora de la selección de acciones: aprendizaje clásico primario y secundario. A partir de estos experimentos fue posible comprobar cómo los nuevos significados adquiridos durante el proceso de aprendizaje participan como “elementos informativos activos” en el proceso de selección de acciones, garantizando una selección de acciones más adaptativa en la red de conductas internas.

Al comparar los resultados obtenidos en esta etapa de experimentación con los resultados presentados en otros trabajos desarrollados, entre los que se encuentran el mecanismo *bottom-up* de Maes, la red neuroconectora de Halperin, el insecto simulado de Beer y la red neuro-humoral de Negrete; podemos establecer que la gran mayoría de las propiedades que caracterizan a estos mecanismos de selección de acciones son un subconjunto de las propiedades explicadas y comprobadas que caracterizan la selección de acciones en la red de conductas internas, mientras que otras propiedades discutidas e ilustradas durante la presente etapa de experimentación sólo son atribuibles a la red de conductas internas.

Un valor agregado de toda esta etapa de experimentación ha sido la identificación de aquellos aspectos que deben ser mejorados o resueltos en el selección de acciones que tiene lugar en la red de conductas internas, algunos de los cuales son relacionados más adelante en el epígrafe 7.7.

CONCLUSIONES

Acerca de la arquitectura de la Red de Conductas Internas

En la arquitectura propuesta se combinan elementos de las tres arquitecturas más ampliamente utilizadas por los MSAs, las cuales fueron revisadas en el capítulo II: red de nodos jerárquica, red de nodos distribuida y red conexionista. El carácter jerárquico de la arquitectura viene dado por el control que establece el lazo jerárquico superior sobre el lazo jerárquico inferior (discutidos en el epígrafe 4.13), para garantizar el comportamiento motivado en la RCI. La naturaleza distribuida de esta arquitectura se puede apreciar en dos aspectos fundamentales: en el carácter distribuido e independiente de las conductas internas a nivel de cada nodo (las cuales, haciendo abstracción, pueden ser vistas como agentes); y en las propiedades de los diferentes nodos de la red, cada uno de los cuales puede ser visto como una entidad que posee su propia información, procesamiento y control (a su vez, éstos pueden ser vistos como agentes). Muchas características de la arquitectura de la RCI pueden ser mapeadas directamente a un modelo conexionista. Al igual que estos últimos, la arquitectura desarrollada es en sí, un sistema de procesamiento distribuido paralelo (PDP). Las conductas elementales pueden ser vistas como neuronas artificiales, también caracterizadas por una función de activación (cálculo del nivel de activación de la conducta elemental), una función de transferencia (el registro del elemento solución sobre el correspondiente nivel del pizarrón), un vector de pesos (dado por el vector de las fuerzas de apareo) y un valor umbral (parámetros θ). Las conductas elementales que operan sobre un mismo nivel (cada conducta interna) pueden ser vistas como capas de neuronas; mientras que los elementos solución creados o modificados en un mismo nivel del pizarrón pueden ser vistos como las salidas producidas por cada capa neuronal.

Computacionalmente, la red de conductas internas puede verse como un conjunto de algoritmos ejecutándose en paralelo. Cada conducta elemental puede verse como una secuencia de pasos encaminada a la ejecución de una acción final (la creación o modificación de un elemento solución sobre el pizarrón). A su vez, cada conducta interna puede ser vista como la implementación de un algoritmo el cual invoca a un subconjunto de algoritmos más simples (las conductas elementales).

Sobre el modelo de combinación de estímulos externos e internos propuesto

Uno de los principales resultados derivado del esquema de combinación de entradas externas e internas propuesto, ha sido comprender que la diferencia esencial entre los diversos MSAs revisados va más allá del tipo de arquitectura en la que éstos han sido estructurados, o de la disciplina en el cual éstos han sido inspirados. La diferencia esencial radica en la forma en que estos MSAs combinan los estímulos externos e internos para seleccionar una acción determinada, y por lo tanto, en el diverso repertorio de propiedades que caracteriza esta combinación.

El modelo de combinación propuesto combina características de modelos multiplicativos puros y aditivos puros. A diferencia de ambos tipos de modelos, en el modelo de combinación propuesto, el grado en que la conducta externa observada depende de los estados internos de la entidad no está “alambrado”, este puede ser regulado con el parámetro α . De esta forma, cuando α sea cercana a 0, el comportamiento observado tendrá una tendencia a la reactividad; mientras que cuando sea cercana a 1, el comportamiento observado será motivado. Dependiendo del dominio del problema en el que se encuentre el agente implementando la RCI, el parámetro α otorga una flexibilidad para determinar el tipo de relación a establecerse entre las entradas internas y externas, dándoles una importancia mayor o menor a cada una de éstas.

En la RCI, un complemento de este modelo de combinación viene dado por el ajuste de los parámetros ϕ y γ , los cuales regulan la reactividad del comportamiento atendiendo a otros criterios.

Propiedades emergentes de la Red de Conductas Internas

Algunas de las propiedades que caracterizan el comportamiento observado en la red de conductas internas emergen de las interacciones que tienen lugar entre las diferentes conductas elementales y mecanismos que componen la red de conductas internas. Ejemplos de estas propiedades son los siguientes:

- Direccionamiento por las metas, el cual emerge del comportamiento motivado
- No indecisión en la selección de acciones, la cual emerge de los procesos de competencia de las conductas internas *selector de preferentes consumatorios* y *selección de conductas internas*.
- Persistencia en la ejecución de una acción externa, la cual emerge como resultado del proceso competitivo que tienen lugar entre conductas elementales del tipo *selector de preferentes consumatorios* y de la competencia entre conductas elementales del tipo *persistencia perceptual*.
- Espontaneidad regulada en la ejecución de conductas externas, la cual emerge de la conducta interna *atención a preferencias* y del valor del parámetro γ .

- Saciada, la cual emerge de los procesos de competencia de las conductas internas *selector de preferentes consumatorios*. Cuando no hay ningún ganador en esta competencia significa que la entidad ha saciado todas sus necesidades internas.
- Cambios en el interés, emerge del proceso competitivo que ocurre a nivel del nodo motivacional.
- Variación de la atención, emerge del proceso competitivo que ocurre a nivel del nodo motivacional, y depende de los niveles de los estados internos.
- Propiedades emergentes del aprendizaje asociativo:
 - Bloqueo, emerge de la competencia entre conductas elementales *persistencia perceptual*.
 - Caída de la actividad del estímulo en el tiempo, emerge de la competencia entre conductas elementales *persistencia perceptual*.
 - Dominancia, emerge de la competencia entre conductas elementales *persistencia perceptual*.
 - Interrupción del condicionamiento, emerge del proceso competitivo que tiene lugar entre conductas elementales *selector de preferentes consumatorios*.

Mejorando la selección de acciones con el aprendizaje asociativo

La selección de acciones ha sido mejorada en la red de conductas internas con la incorporación de los procesos de aprendizaje asociativo. Esto significa que estímulos ambientales que inicialmente carecían de significado para la entidad, pueden asociarse con otros estímulos que ya poseen un significado para la misma. De esta forma, una vez que un estímulo ha adquirido un significado particular, éste jugará un rol determinante durante la búsqueda de estímulos externos requeridos para satisfacer determinada necesidad interna de la entidad.

Es así como los significados adquiridos durante el proceso de aprendizaje participan como elementos informativos activos en el proceso de selección de acciones, garantizando una selección más adaptativa del selector.

La simulación desarrollada: una aproximación a un “laboratorio virtual de conductas”

La simulación desarrollada implementando la red de conductas internas brinda una gran flexibilidad para la experimentación y prueba de una amplia gama de comportamientos y propiedades deseables en la selección de acciones. Esta flexibilidad en parte es expresada por los siguientes aspectos:

- La facilidad que brinda la simulación para el ajuste de los diferentes parámetros que regulan aspectos tales como: grado de motivación, grado de reactividad, velocidad de

condicionamiento y de pérdida del condicionamiento, persistencia de señales en memorias a corto y a corto-mediano plazo.

- La visualización de los diferentes estados internos de la red de conductas internas, tales como: niveles de pizarrón, fuerzas de apareo modificables.
- El establecimiento y la visualización de los estados internos y de la posición en el medio ambiente de la entidad.
- La generación y modificación de ambientes externos con características determinadas *a priori*.
- El almacenamiento y recuperación de medios ambientes previamente creados.
- El almacenamiento y recuperación de las propiedades del agente simulado.

De esta forma, la simulación desarrollada constituye en sí una aproximación a un “laboratorio virtual de conductas”, en el cual se pueden simular y comprobar propiedades intrínsecas del comportamiento animal.

Aplicaciones

Una aplicación actual de la red de conductas internas se está llevando a cabo en la modelación de sistemas biológicos, en particular en la construcción de modelos celulares, simulando algunos de los principales procesos que ocurren a nivel celular, tales como: transducción de señales, cascadas de fosforilación, controles de replicación y control de expresión. Lo anterior va encaminado a la comprensión y predicción en dichos sistemas. Para tal propósito se intenta construir un *celulat*, una entidad celular cuyo mecanismo de selección de acciones sea implementado por la red de conductas internas. El dominio inicial del *celulat* será la predicción de blancos terapéuticos carcinogénicos potenciales.

Trabajo futuro: extensiones y refinamientos

Aún quedan muchos aspectos que pueden ser mejorados en la red de conductas internas. Nuestro trabajo futuro en este sentido va dirigido hacia:

- Lesiones en diferentes estructuras y procesos de la red de conductas internas para comprender mejor su funcionamiento.
- Aprendizaje dinámico de los valores de los parámetros α , γ , ϕ y κ , para cada columna conductual.
- Aprendizaje a nivel cognitivo. En este punto se pretende utilizar un paradigma de aprendizaje no-supervisado, el cual también se defina en términos de la modificación de las fuerzas de apareo. La idea subyacente es que el *animat* aprenda a reconocer nuevos estímulos ambientales.
- La inclusión del papel de las emociones en la selección de comportamientos, considerando un nodo emocional en la red de conductas internas para modelar emociones (Gershenson, 1999).

- La construcción de una sociedad artificial de agentes autónomos, donde cada uno de éstos implemente la red de conductas internas, y la descripción de las nuevas propiedades que emerjan.

REFERENCIAS

- Baerends, G. (1976) The functional organization of behavior. *Animal Behavior*, 24, 726-735.
- Beer, R. (1990) Intelligence as Adaptive Behaviour: an Experiment in Computational Neuroethology. Academic Press.
- Beer, R., Chiel, H. y Sterling, L. (1990) A biological perspective on autonomous agent design. *Robotics and Autonomous Systems*, 6, 169-186.
- Bogner, M., U. Ramamurthy y S. Franklin (1999) "Consciousness" and conceptual learning in a socially situated agent. Dautenhahn, Kerstin (Ed.). Human cognition and social agent technology. Amsterdam: John Benjamins Publishing Company.
- Brooks, R. A. (1986) A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*. RA-2, April, pp. 14-23.
- Brooks, R. A. (1989) A robot that walks: Emergent behavior from a carefully evolved network. *Neural Computation*, 1, 253-262.
- Brooks, R. A. (1991) How to build complete creatures rather than isolated cognitive simulators. En K. VanLehn (ed.), *Architectures for Intelligence*, pp. 225-239, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Cañamero, D. (1997). Modeling Motivations and Emotions as a Basis for Intelligent Behavior. In Proceedings of the First International Symposium on Autonomous Agents, AA'97, Marina del Rey, CA, February 5-8, The ACM Press.
- Engelmores, R., A. J. Morgan, y H. P. Nii (1988) HEARSAY II. En R. Engelmores y T. Morgan (ed.), *Blackboard Systems*, Addison-Wesley Publishing Company.
- Erman, L. D., F. Hayes-Roth, V. R. Lesser, y D. R. Reddy (1988) The HEARSAY II: speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Survey* 12, pp. 213-253.
- Feigenbaum, E. A. (1985) Paper presented at the Fifth International Joint Conference on Artificial Intelligence, Los Angeles, CA.
- Franklin, S., A. Graesser, B. Olde, H. Song y A. Negatu (1996) Virtual Mattie: An intelligent Clerical Agent. AAI Symposium on Embodied Cognition and Action, Cambridge, MA.

- Gershenson, C. (1999) Modelling Emotions with Multidimensional Logic. Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS '99), pp. 42-46. New York City, NY.
- Goetz, P. y D. Walters (1997) The dynamics of recurrent behavior networks. *Adaptive Behavior*, 6(2), 245-282.
- González, P.P. y J. Negrete (1997) REDSIEX: A cooperative network of expert systems with blackboard architectures. *Expert Systems*, 14(4), 180-189.
- Hallam, B.E, J. Halperin y J. Hallam (1994) An Ethological Model for Implementation in Mobile Robots. En *Adaptive Behaviour*, volumen 3, número 1, pp 51-79.
- Kandel, E. (1976) *Cellular Basis of Behavior: An introduction to Behavioral Neurobiology*. Freeman, W.H., San Francisco.
- Kandel, E. (1978) A cell biological approach to Learning . Grass Lecture No. 1. Society of Neurosciences, Bethesda.
- Konorsky, J. (1967) *Integrative Activity of the Brain*. The University of Chicago Press, Chicago.
- Lesser, V. R. y D. D. Corkill (1988) The distributed vehicle monitoring testbed: a tool for investigating distributed problem solving networks. En R. Englemore y T. Morgan (ed.), *Blackboard Systems*, Addison-Wesley Publishing Company.
- Lorenz, K. (1950) The comparative method in studying innate behaviour patterns. *Symposia of the Society for Experimental Biology*, 4, 221-268.
- Lorenz, K. (1981) *Foundations of Ethology*. Springer-Verlag.
- Maes, P. (1990) Situated agents can have goals. *Journal of Robotics and Autonomous Systems*, 6(1&2).
- Maes, P. (1989) The Dynamics of Action Selection. Proceedings of the IJCAI-89 conference, Detroit.
- Maes, P. (1991) A bottom-up mechanism for behaviour selection in an artificial creature. En J.A. Meyer y S.W. Wilson (ed.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behaviour* MIT Press/Bradford Books.
- Maes, P. (1994) Modeling Adaptive Autonomous Agents. *Journal of Artificial Life*, volumen 1, números 1 y 2, MIT Press.
- McFarland, D. (1981) *The Oxford Companion to Animal Behavior*. Oxford University Press.
- Minsky, M. (1985). *The Society of Mind*. New York: Simon and Schuster.

- Negrete, J. y M. Martínez (1996) Robotic Simulation in Ethology. Proceedings of the IASTED International Conference: Robotics and Manufacturing, Honolulu, Hawaii, USA, pp. 271-274.
- Negrete, J. y P.P. González (1998) Net of multi-agent expert systems with emergent control. *Expert Systems with Applications*, 14(1) 109-116.
- Nii, H. P. (1989) Blackboard systems. En A. Barr, P. R. Cohen, y E. A. Feigenbaum (ed.), *The Handbook of Artificial Intelligence, volumen IV*, Addison-Wesley Publishing Company.
- Pirjanian, P. (1998) Multiple Objective Action Selection & Behavior Fusion using Voting, Ph.D.-thesis, Faculty of Technical Sciences, Aalborg University Denmark.
- Pirjanian P., Christensen H. I. and Fayman J. A. (1998) Application of Voting to Fusion of Purposive Modules: An Experimental Investigation, Invited paper in *Journal of Robotics & Autonomous Systems*, Vol. 23, No. 4, July 1998. pp 253-266.
- Rosenblatt, K. y D. Payton (1989) A fine-grained alternative to the subsumption architecture for mobile robot control. En *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, IEEE.
- Sutton, R.S. y A. Barto (1981) A toward a modern theory of adaptive networks: expectation and prediction. *Psychological Review*, 88, 135-170.
- Tinbergen, N. (1950) The hierarchical organization of mechanisms underlying instinctive behavior. *Experimental Biology*, 4, 305-312.
- Tinbergen, N. (1951) *The Study of Instinct*. Claredon Press.

APÉNDICES

REDSIEX: A cooperative network of expert systems with blackboard architectures

Pedro Pablo González Pérez

Instituto de Investigaciones Biomédicas/UNAM
Apartado Postal 70228
Ciudad Universitaria, 04510 México D.F.
Tel: +52 622 38 70, Fax: +52 550 00 48
e-mail: ppgp@servidor.unam.mx

José Negrete Martínez

Maestría en Inteligencia Artificial/UV
Xalapa Centro
Xalapa, VERACRUZ, México

Abstract: *The blackboard architecture, originally developed for the system that permits the comprehension of language, HEARSAY II, has later been used in a great variety of domains and in various environments for the construction of systems. From the classic architecture of HEARSAY II, many applications, generalizations, extensions and refinements have been developed. In this paper we present REDSIEX, (RED de Sistemas EXpertos) which is a network of expert systems within a blackboard architecture, for the cooperative solution of distributed problems. The REDSIEX system inherits various of the elements defined by the architecture of HEARSAY II and incorporates new components and organization. These produce a very characteristic and exclusive global work style in the solution of problems, within a conceptual framework of emergent control. The main structural and functional characteristics of REDSIEX are discussed.*

Keywords: Blackboard architecture, cooperative systems, distributed artificial intelligence, emergent cooperation

1. Introduction

A cooperative expert systems approach requires mechanisms of cooperative interaction that permit multiple expert systems to work together on the solution of a problem in common. When the character of the expert systems is heterogeneous, each expert may have different goals, different approaches to the solution of the problem, different evaluation criteria for the solution of the problem and different internal representations of solutions and problems. Achieving cooperation within this framework is much more complex than just providing these experts with a common language of communication. In order for these experts to be able to cooperate, each one of them needs information about the type of knowledge, goals, plans and preferences of the remaining experts. This information allows the emergence of a new control mechanism in the global solution of the problem.

In this paper, we present REDSIEX (RED de Sistemas EXpertos), a prototype system for the development of applications that require the cooperative solution of problems. REDSIEX is a dynamic network of problem-solving nodes, in which each node is an expert system within a blackboard architecture. In REDSIEX each node can work independently on the solution of a problem, but when faced with a global problem, cooperation is necessary. This may be when a single node does not possess the resources, the knowledge or sufficient expertise to solve a problem independently. Similarly, different nodes may have different expertise for solving different parts of a problem, different resources available for tackling the solution to a problem and distinct sets of knowledge or points of view about a problem.

As will be shown in the final comments of this paper, we are not dealing with a simple development of what already exists, but rather an investigation that touches upon issues that are fundamental to Artificial Intelligence such as the emergence of expertise control amongst autonomic agents.

2. REDSIEX and the cooperative solution of problems

If a set of expert systems is to work effectively in a cooperative manner, then it is necessary to overcome its reasoning limitations in terms of knowledge, processing and interaction. In REDSIEX these limitations are remedied by introducing specifications of:

- Communication
- Organization
- Planning
- Modelling

The specifications of communication, planning and modelling are discussed later in this paper in section 4.2 where the activity of the communication knowledge sources, and the local planner are described. In a cooperative problem-solving network, organization specifies in concrete terms which are the responsibilities and the patterns of interaction for each one of the nodes in the network. Organization can improve the performance of the network by reducing the responsibilities of each node in the solution of problems.

In REDSIEX, the organization is specified on the basis of the following criteria:

- (1) Assigning problem-solving roles to nodes, which renders each node a specialist in a different kind of problem or subproblem.
- (2) Indicating to the nodes the connections for the flow of information among them, in such a way that they may transfer unsolved problems or subproblems to other nodes which possess greater solution abilities.

With the use of specifications, nodes achieve real coordination of their activities and a cooperative manner of working.

In REDSIEX the cooperation is intranode and internode. Intranode cooperation manifests itself through the combined work of all the node's local knowledge sources. This process is characterized by being an opportunistic and incremental process of creation and modification of the most promising solution elements on the domain blackboard.

Internode cooperation is favoured because nodes can exchange hypotheses and partial results. This type of cooperation can take place in real or deferred time. This depends on the availability of the time and resources of those nodes which have been asked to commit themselves. Cooperation in real time takes place when a node receives a reply to its request for commitment within a time lapse that it has dedicated to the problem-solving processes, for which the commitment was requested. Deferred cooperation takes place when the problem for which the commitment was requested has been abandoned temporarily, without having received a reply; this problem can be taken up again once the node receives an answer to the request for commitment. These characteristics of the cooperation in REDSIEX are covered in greater detail in section 4.2

3. Distribution of information, processing and control in REDSIEX

There exist various dimensions for the breakdown of a centralized blackboard system in a distributed environment and various options within each one of these (Erman, Hayes-Roth, Lesser and Reddy, 1988). The structure of REDSIEX takes into account the following dimensions and options for each dimension:

- (1) Information Dimension
 - (a) Distribution of the blackboard: each node has its own local blackboard.
 - (b) Transmission of hypotheses: a node transmits and receives hypotheses to and from a local subset of nodes.
- (2) Knowledge Dimension
 - (a) Distribution of the knowledge sources: each node has a subset of knowledge sources.
 - (b) Access to the blackboard by knowledge sources: a knowledge source activation can access only the blackboard in its local node.
- (3) Control Dimension
 - (a) Distribution of the knowledge sources activation: an event on the blackboard activates only knowledge sources within the local node.
 - (b) Distribution of the dispatch mechanism (scheduling), control database and the planner: each node possesses its own dispatch mechanism, and carries out the dispatch based on local information. Each node possesses a control database (control blackboard) and a planner to execute local control tasks.

4. The architecture of the nodes in REDSIEX

Each node in REDSIEX is a blackboard system (Engelmore, Morgan and Nii, 1988; Nii, 1989) with elements inherited from the architecture of HEARSAY II (Engelmore, Morgan and Nii, 1988; Lesser and Erman, 1988; Nii, 1989), the architecture of the DVMT system (Lesser and Corkill, 1988) and the Control Blackboard Architecture (Hayes-Roth, 1985; Hayes-Roth and Hewett 1988) as well as new elements defined by the particular architecture of REDSIEX. Figure 1 shows the details of the architecture that characterizes the nodes in REDSIEX. The structure of each node in the network comprises two types of fundamental components: data-base type components and program type components.

4.1. Database type components

The database type components are structures of data on which the program type components execute actions of search, creation and modification of elements belonging to the said structures. Each node in the network possesses the following database type components:

- Domain Blackboard
- Control database
- List of Knowledge Source Activation Registers (LKSAR), which includes:
 - Reasoning Knowledge Source Activation Registers (RKSAR)
 - Receiver Knowledge Source Activation Registers (EKSAR)
 - Transmitter Knowledge Source Activation Registers (TKSAR)

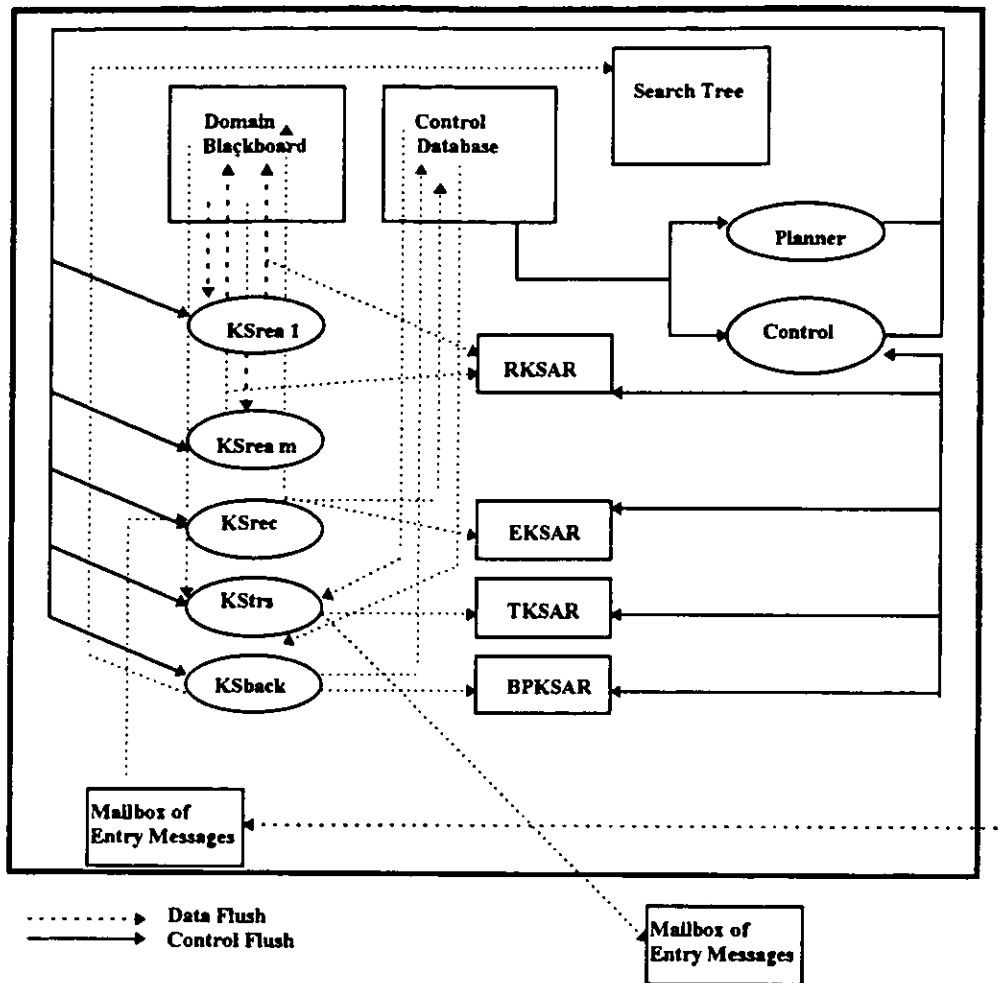


Figure 1: REDSIEX node architecture

Backward Processor Knowledge Source Activation Registers (BPKSAR)

- Mailbox for Entry Message (MEM)
- Search Tree

4.1.1. *The domain blackboard* The domain blackboard is a shared data structure through which the knowledge sources of the domain (reasoning knowledge sources) communicate with each other. The domain blackboard organizes the solution elements, also called hypotheses, at different levels of abstraction. These levels of abstraction represent the solution to the problem at different grades of detail. Each hypothesis on the domain blackboard is associated with the following information:

- Identifier or number of the hypothesis
- Certainty or grade of belief associated with the hypothesis

4.1.2. *The Control Database* The control database (or metaknowledge base) is a data structure defined by the architecture of REDSIEX. The solution elements on the

control data base are registers whose attributes enclose information or decisions relevant to the problem of control; solution elements for the revision of beliefs and solution elements for the investigation and communication of hypotheses. For each hypothesis on the domain blackboard, a solution element in the control database is associated with information corresponding to the certainty or the belief of the hypothesis. This information is used in the process of revising the hypothesis beliefs (described later in this paper). The solution elements for the investigation and communication of hypotheses constitute part of the control focus of the local node and are control decisions about which of the solution elements on the domain blackboard should be investigated at a given moment (in accordance with the node's particular resources), which hypotheses should be transmitted as partial results to other nodes, or which questions should be formulated to pass to other nodes in the network. The receiver knowledge source, transmitter knowledge source and backward processor knowledge source operate on the control database together with the local planner which is also characterized by a con-

dition-action structure. These knowledge sources communicate with each other through the creation of solution elements on the control database.

4.1.3. The List of Knowledge Sources Activation Registers

When the conditions or premises of a knowledge source are fulfilled, the source is activated, creating an knowledge source activation register (KSAR). A KSAR is a data structure that describes the characteristics of the action of the knowledge source. When a KSAR is selected by the control mechanism (discussed later) the action of the knowledge source associated with the KSAR is executed in the context of the activated information and typically produces new events on the domain blackboard or on the control data base, or acts of communication with other nodes on the network, depending on the nature of the knowledge source that originated the KSAR.

Not all the KSARs possess the same data structure. An appropriate data structure for KSARs is defined for each type of knowledge source. The types of KSARs defined for each node are as follows:

- Reasoning Knowledge Sources Activation Registers (RKSARs)
- Receiver Knowledge Source Activation Registers (EKSARs)
- Transmitter Knowledge Source Activation Registers (TKSARs)
- Backward Processor Knowledge Source Activation Registers (BPKSARs)

The data structure of a RKSAR is shown in Table 1.

4.1.4. Entry Message Mailbox Each message sent to the node by any of the other nodes in the network is received in an entry message mailbox creating a register for that

message (EMR). An EMR is a data structure with attributes as designed in Table 2.

4.1.5. Search Tree The search tree is a data structure that permits the construction of all the possible lines of inference that lead to any of the localized hypotheses at the goal level of the domain blackboard. Every element of the search tree has a condition part and an action part. The condition part describes a particular hypothesis configuration on the blackboard. The action part specifies the hypothesis that will be created or modified on the blackboard once the condition part has been fulfilled. A line of inference is constructed from the concatenation of elements of the search tree. The final element of the chain formed is the goal hypothesis that is desired to be reached. The node's local planner, through the backward processor knowledge source, uses the search tree for the construction of local plans that indicate to the node which actions must be executed to arrive at determined goals.

4.2. The node program type components and their functioning

Node program type components execute the processes of search, creation and modification of solution elements on the data structures and carry out the processes related to the local control of the node. Each node in the network possesses the following program type components:

- Reasoning Knowledge Sources
- Receiver Knowledge Source
- Transmitter Knowledge Source
- Backward Processor Knowledge Source
- Local Planner
- Control Mechanism, which includes:
 - Knowledge Source of Control I

Table 1: RKSAR data structure

Attribute	Description
Activation cycle	Number of the execution cycle in which the activation register was created
Source Identifier	Number or identifier of the reasoning source that created the register
Rule identifier	Number or identifier of the activated inference rule
Action event	Blackboard hypothesis that is to be created or modified
Matching level	Blackboard level at which the condition of the activated rule was matched
Synthesis level	Blackboard level at which the action of the activated rule will be executed
Rule matching strength	Certainty or belief value with which the condition of the rule was fulfilled
Rule importance	Certainty or belief value of the rule proposed by the expert
Rule efficiency	Certainty calculated for the rule based on its matching strength and importance
Final belief of the event	Certainty or final belief value calculated for the created event
Offer	Value calculated by the control mechanism based on the remaining parameters

Table 2: EMR data structure

Attribute	Description
Event associated with the message	Number or identifier of the hypothesis for which information is proposed or requested
Belief value associated to the message	Certainty or belief value of the hypothesis associated with the message
Message collaborator	Number or identifier of the node that sends the message
Message type	Specifies the type of information contained in the message (offer, answer or request for commitment)

- Knowledge Source of Control II
- Knowledge Source of Control III

4.2.1. Reasoning Knowledge Sources The reasoning knowledge sources (KSrea) contain specific knowledge about the system's domain. This knowledge is provided by the inference rules that make up each KSrea. A KSrea is made up of one identification part and one inferential part. The identification part describes the identification elements of the KSrea, such as: identifier, matching level and synthesis level. The inferential part specifies the characteristics of the inference rules defined by the KSrea. The condition of each inference rule describes the configuration of solution elements on the blackboard that is necessary for the KSrea to contribute to the problem-solving processes. The way in which a KSrea contributes to the solution of a problem is specified in the action of each inference rule. The action of an inference rule is linked to the creation or modification of solution elements on the blackboard. The activity of the KSrea is driven by the events. This means that every change that is executed on the blackboard constitutes an event that, in the presence of other specific information on this, can activate one or more inference rules within a KSrea. Each activation produces a unique KSrea activation register (RKSAR).

4.2.2. The communication knowledge sources Each node in the network receives messages from other nodes and selects hypotheses on the blackboard to send to other nodes. In order to carry out these activities of communication, the architecture of REDSIEX defines for each node of the network two communication knowledge sources: the receiver knowledge source (KSrec) and the transmitter knowledge source (KStrs).

The receiver knowledge source (KSrec). The KSrec decides which messages, from those which up to the moment have been received in the mailbox of entry messages (MEM), are of interest in the solving of the problem in hand and it places them at the consideration of the control mechanism. The KSrec possesses a condition-action format. The condition of the KSrec is fulfilled when a message of interest for the solving of the present problem is found in the MEM. Each time the condition of the KSrec has been

fulfilled this executes two actions. The first action consists of the inscription of a EKSAR on the list of KSARs. The second action is carried out when the control mechanism selects the recorded EKSAR for its execution. The execution of a EKSAR depends on the type of message it contains. If the message is of the type *offer* or *reply to commitment*, then the execution of the EKSAR is associated with the creation or modification of solution elements on the domain blackboard. If the message is of the type *request for commitment*, then the execution of the EKSAR is associated with the creation of a register of the type *investigate hypothesis* on the control database.

The transmitter knowledge source (KStrs). The KStrs transmits messages of the type *offer*, *reply to commitment* or *request for commitment* to other nodes of the network. Messages of the type *offer* or *reply to commitment* are hypotheses selected on the domain blackboard, for each one of which the KStrs finds on the control blackboard a register of the type *transmit hypothesis*. Messages of the type *request for commitment* are associated with registers of the type *ask hypothesis*, which have been previously created on the control database. Like the KSrec, the KStrs also possesses a condition-action format. The condition of the KStrs is fulfilled when any register of the type *transmit hypothesis* or *ask hypothesis* is found on the control database. When the KStrs manages to fulfill its condition, this is activated creating initially a TKSAR on the list of KSARs. Once the TKSAR is selected by the control mechanism, the KStrs executes the final action, which consists of transmitting the message associated with the TKSAR to any node or subset of nodes in the network.

When a TKSAR has been selected for execution by the control mechanism, the KStrs decides to which node or nodes of the network it should send the message associated with the TKSAR, by consulting the model of the outside world which the node possesses. The model of the outside world is a data structure defined by the local node, to which only the KStrs have access. It describes the main characteristics, abilities, knowledge and resources of the remaining nodes in the network.

4.2.3. The backward processing knowledge source The activity of the reasoning knowledge sources (KSrea) is

directed by the events that occur on the domain blackboard. This type of activity constitutes a process of forward rules chaining leading to the creation of solution elements on the domain blackboard, for the problem that needs to be solved. Like the KSreas, the backward processor knowledge source (KSback) also contributes to the creation of solution elements on the domain blackboard although, unlike the former, the activity of KSback is determined by goals and plans, and not by events that occur on the blackboard.

The activity of the KSback constitutes a process of backward rules chaining leading to the construction of a sequence tree to generate solution elements. The sequences are initiated with the configuration or goal state at the root of the tree. The next level of the tree is generated by finding all the rules whose consequent or actions match in the root node. Were it possible to apply only these rules then the desired goal state could be generated. The antecedents or conditions of these rules are used to generate the nodes at this second level of the tree. The next level of the tree is generated by taking each node at the previous level and finding all the rules whose consequent or actions correspond to this, then the antecedents of the corresponding rules are used to generate the new nodes at the next level of the tree. The process continues until a configuration of solution elements only corresponding to initial states has been generated.

To construct the sequence tree, the KSback uses the information registered on the node's search tree, selecting from among all the lines of inference that lead to the goal level of the blackboard only those that generate the solution elements that are necessary for reaching the desired goal hypothesis. The KSback has a condition-action format. The condition is fulfilled when a register of the type *investigate hypothesis* is created on the control database.

4.2.4. The local planner The function of the planner is to generate local plans directed towards achieving of particular goals. A plan is a sequence of activities that must be carried out by certain knowledge sources of the node, with the purpose of creating the solution elements on the blackboard that are necessary to permit the node to reach the desired goal. To initiate the creation of a local plan, the planner uses a KSback to create the sequence tree to generate solution elements. The nodes located at the different levels of the search tree correspond to particular solution elements, some of which may already exist on the domain blackboard while others have to be created. The creation of each solution element is an activity that requires one or various specific knowledge sources to work. Once the sequence tree has been created, the planner evaluates heuristic criteria to decide in which order to execute these activities. In this way, the generation of the local plan is concluded. Like the knowledge sources, the local planner possesses a condition-action structure. The condition is fulfilled when any of the following situations has occurred:

- (1) The processes leading to the search for the solution to the problem have stopped on account of insufficient information on the domain blackboard. This means that the work of the reasoning knowledge sources, directed by the events, cannot be carried out, since it is necessary to have goal directed reasoning that permits the creation of new solution elements on the domain blackboard.
- (2) On the control database a register of the type *investigate hypothesis* has been created.

When situation 1 has taken place, the planner, making use of its heuristic criteria, determines which goal it should investigate and so creates a register of the type *investigate hypothesis* on the control database. Once the condition of the local planner has been fulfilled, the action manifests itself creating an appropriate plan which conceives a relation of activities and executors for these activities. The first heuristic criterion that the local planner evaluates is the calculation of a measure of resolution for each node not resolved on the sequence tree. As its name indicates, the measure of resolution reflects the degree in which a node has been resolved, being a value between 0 and 1, and its calculation is given by the following expression:

$$R_{ij} = \frac{CNR_{j-1}}{CTN_{j-1}}$$

where:

R_{ij} is the measure of resolution of the node i belonging to the level j of the sequence tree.

CNR_{j-1} is the quantity of nodes resolved at the level $j-1$ corresponding to the antecedent of a rule whose action is the creation of the node i at the level j .

CTN_{j-1} is the total quantity of nodes at the level $j-1$ corresponding to the antecedent of a rule whose action is the creation of the node i at the level j .

There might be more than one line of inference (rule) that would permit the creation of a node i at the level j . This in its turn would imply the existence of more than one measure of resolution for the node i . When this is the case, the measure of resolution of the node i will be taken to be the maximum from among the measurements of resolution contributed by the group of rules. Once the measure of resolution is calculated for all the inferable nodes belonging to the level j of the tree, there follows the application of one of the remaining heuristic criteria contributed by the local planner. These heuristic criteria have a common objective – that of selecting those rules of inference that contribute a greater quantity of information for solving the problem and which, therefore, will contribute to the optimization of the problem-solving.

4.2.5. The control mechanism When a knowledge source is activated, it creates a register of activation (RKSAR,

EKSAR, TKSAR, or BPKSAR) which describes the characteristics associated with the action that the knowledge source intends to execute. In each execution cycle only one KSAR is executed. To determine the order in which the knowledge sources will execute their final actions, the control system establishes a competition among the registered KSARs, based on the calculation of an offer for each KSAR. The knowledge source associated with the highest-offered KSAR will be selected to execute its final action in that execution cycle.

The control system is made up of three basic control knowledge sources (Hayes-Roth and Hewett, 1988): knowledge source of control I, knowledge source of control II and knowledge source of control III, which iterate in a three-step cycle to execute respectively the following activities:

- (1) Calculation of the offer for each registered KSAR.
- (2) selection of the highest-offered KSAR.
- (3) Invoking of the knowledge source associated with the selected KSAR.

Since there are various types of registers of activation (RKSAR, EKSAR, TKSAR and BPKSAR) it is possible for there to exist registers of activation competing to execute their actions in more than one of these structural blocks. Given this situation, a hierarchical selection has been established to select the block of KSAR within which the highest-offered register must be selected. The hierarchy has been established as follows:

- RKSAR : priority 1 (greatest hierarchy)
- EKSAR : priority 2
- TKSAR : priority 3
- KSARB : priority 4 (least hierarchy)

Consequently, knowledge sources that have recorded registers of lesser priority will only execute their action in the absence of registers of greater priority. The order or hierarchy that has been established contributes to maximizing the internal information contained in each node of the network, before establishing collaboration links. This enables a more coherent and robust cooperation, avoiding the sending of messages with incomplete or unsure information and the formulation of unnecessary questions.

Calculating the offer for registers of activation. This process is carried out once the block of KSARs within which the register that will execute its action has been selected. The offer is carried out by the knowledge source of control I. The condition of this knowledge source is satisfied when there exist registers recorded within a block for which it is necessary to calculate or recalculate their offer. This involves the calculating or recalculating of the offer for all the registers of the selected block. The offer is a function of the parameters that make up the register. Therefore, the form of calculating the offer will depend on the type of register selected.

Selection of the executable KSAR. Once the offer has been calculated for all the registers corresponding to the selected block (according to the described hierarchical criteria) then the knowledge source of control II carries out the selection of the highest-offered register for its execution.

Execution of the selected KSAR. When the highest-offered register has been selected for its execution, the knowledge source of control III invokes the knowledge source responsible for the creation of that register so that it executes its final action: the creation or modification of the corresponding solution element on the domain blackboard.

In this way, the three knowledge sources of basic control iterate in a three-step cycle, determining in each execution cycle which register of activation is to be executed and carrying out its execution.

5. The revision of beliefs

One of the principal characteristics that make REDSIEX a consistent problem-solving system is its process of revision of beliefs. The revision of beliefs is the process of adjusting the belief values of determined hypotheses, created on the domain blackboard based on modifications belief values of one or more hypotheses, that have contributed evidence for its creation. In other words, if the belief value of a hypothesis on the blackboard is modified, then the belief values of the hypotheses found on all the lines of inference, with forward chaining starting with this hypothesis, will also be modified in accordance with this new value.

When the modification of a belief value of a hypothesis on the blackboard has taken place, the system checks whether this hypothesis has contributed to the creation of any other hypothesis at that same level or at higher levels of the blackboard. If this is the case, then a forward processing is executed, through which the rules whose conditions have already been fulfilled given the previous belief value of the hypothesis are activated again. The process of revision of beliefs requires the consultation of the belief revision type elements, registered on the control database.

6. The internode cooperation in REDSIEX

Some of the most notable characteristics of the observed internode cooperation in REDSIEX are the following:

- (1) The cooperation is emergent, that is to say it is a non-programmed property that arises dynamically in the process of interconsultation among nodes. It could be said also that REDSIEX exhibits an emergent control strategy.
- (2) The cooperation is facultative. In REDSIEX, cooperation is a faculty possessed by each blackboard

node in the network, and does not constitute a permanent necessity. This means that a blackboard node can solve a complete problem, if this node possesses the necessary capacity and resources to arrive at the solution. When this is not the case, then the node that is responsible for the problem can ask for cooperation from other nodes in the network. The REDSIEX architecture does not require an administrator node with faculties for the breaking down of a task into sub-tasks to distribute among different nodes in the network for their execution. Instead of this, the task is requested from a particular node in the network (which is presumed to possess the best abilities, knowledge and resources for carrying this out) making it responsible for the task and deciding if in any part of the problem-solving process, the help from any other node in the network is required. Therefore, the REDSIEX system can be seen as a collection of expert systems facultatively associated in a cooperative network.

- (3) Cooperation can be executed in real time or deferred time. At any given time each blackboard node may find itself working on the solution of a different problem from those that are being solved by the other nodes in the network. However, even under these conditions it is possible for the nodes to establish cooperative links among themselves. This is achieved through two cooperation mechanisms defined in REDSIEX: cooperation in real time and cooperation in deferred time. When the cooperation takes place in real time, the node that requests the commitment receives the reply within a time lapse that the node has dedicated to the processes of problem-solving. Deferred cooperation occurs when the reply to the requested commitment cannot be elaborated and sent within a given time lapse. This may occur because the node, which has received the request for commitment, is busy with the solution of another problem which for it has greater priority and decides to conclude some process already begun before attending to the request received. When the node that requested the commitment receives the reply, it may be working on another problem and then it will decide whether to continue the processes leading to the solution of its present problem, or to take up again the unconcluded problem for which it had requested the commitment.

7. A REDSIEX implementation in the medical interconsultation domain

Medical diagnosis, as a problem-solving process, may be approached as an incremental generation process of solution elements through various levels of abstraction. Starting at a level of minimum abstraction, which contains primary information corresponding to signs and symptoms,

and running through other intermediate levels at which specific syndromes are determined or particular etiological and physiopathological entities are inferred, a superior level is reached at which a conclusion is made as to the nosological entity that is present and as to which should be the adequate treatment. To this end, the blackboard architecture provides an ideal scenario for modeling in a coherent manner the relevant medical knowledge and expertise in the solution of diagnostic problems.

INTERMED system (González, 1995; González and Negrete, 1996) is a cooperative network of expert systems that carries out the work of a group of medical specialists in medical consultation and interconsultation. We have developed INTERMED within REDSIEX architecture. Each expert system in INTERMED possesses knowledge, reasoning methods and expertise to make diagnoses within a limited range of diseases belonging to a specific speciality.

In INTERMED each expert system or specialist is capable of carrying out alone a complete task of diagnosis, whenever it possesses all the knowledge, skill and resources that are necessary to find the solution to the problem in question. When this is not the case, the specialist carrying out the medical consultation can request help from the other specialists in the network, to complete the problem-solving processes and thus arrive at an accurate diagnosis. In this way, each specialist or medical consultant in INTERMED is capable of executing diagnostic tasks within its domain or area of interest, while at the same time possessing faculties that permit it to establish cooperative processes leading to the solution of its problem or to problems that concern other specialists in the network.

So far the INTERMED application covers four expert systems which have been defined following the subdomains of interest that make up the global domain of the system; diagnosis and treatment of diseases of the digestive system, respiratory system and cardiovascular system, and analysis and interpretation of clinical laboratory tests. For the expert systems in the diagnosis and treatment of diseases, the domain blackboard organizes the solution elements in five levels of abstraction; coarse grain signs and symptoms, fine grain signs and symptoms, physiopathological entities, nosological entities, and therapeutic (higher abstraction level). For the expert system in the analysis and interpretation of clinical laboratory tests, the solution elements are organized at four levels of abstraction; laboratory test, laboratory data, diagnosis, and causes. In INTERMED, the process of medical diagnosis begins with the selection of an expert node (medical specialist) which is presumed to possess the greatest knowledge related to the patient's problem. The system initiates the gathering of signs and symptoms present in the patient and then refines them, to obtain signs and symptoms at two levels of granularity; coarse grain signs and symptoms and fine grain signs and symptoms (semiology). The collection of signs and symptoms in coarse and fine grain allows the system to formulate

diagnoses at different depth levels, mainly at etiopathological level and nosological/causal level. However, as occurs in clinical practice, in order to arrive at this level, assistance is frequently required from the clinical laboratory.

When the diagnosis problem so requires, the different expert nodes can establish cooperative links, in order to achieve a definitive diagnosis. Once a diagnosis is reached, as occurs in practice, an adequate treatment is proposed. This does not depend only on the diagnosis made, but also on the characteristics of the patient (age, sex, biological stage, pregnancy, and senility among others) and on the patient's hereditary-family and personal pathological history. When all these characteristics have been evaluated, it is possible to propose a complete handling in accordance with the diagnosis made for a specific patient.

All structural and functional characteristics of REDSIEX architecture are present in INTERMED. The system was programmed in C++ for Windows. Each node was resident in a different PC. This implementation solved expediently several problems of medical cooperation, but the final evaluation is waiting for its use in a medical service.

8. Related work

The work described in this article is perhaps most closely related to the large body of research on distributed architectures of blackboard systems. Various of the REDSIEX components are inherited from the architectures of HEARSAY II (Engelmore, Morgan and Nii, 1988; Lesser and Erman, 1988; Nii, 1989), DVMT (Lesser and Corkill, 1988) and BB1 (Hayes-Roth, 1985; Hayes-Roth and Hewett, 1988). This means that some particular aspects, like the characteristics and the work style of REDSIEX, are similar to those of these other systems. However, the new components and work strategies incorporated in the architecture of REDSIEX make its global work style on the solving of problems characteristic and exclusive to it.

The REDSIEX system belongs to the class of distributed blackboard systems, as well as the systems BLOBS (Zanconato, 1988), an object-oriented blackboard system framework for reasoning in time; GUARDIAN (Hayes-Roth, Hewett, Washington, Hewett and Seiver, 1985), a system for intensive-care monitoring with distributed artificial intelligence architecture based on the blackboard control architecture; POLIGON (Nii, Aiello and Rice, 1988; Nii, Aiello and Rice, 1989), a framework for concurrent problem solving; DVMT (Lesser and Corkill, 1988), a framework where general approaches for distributed problem solving can be developed and evaluated; and TRICERO (Nii, 1989), a system for airy space monitoring developed within distributed blackboard architecture. Generally speaking, the systems classified in this category are characterized by being made up of a set of nodes or agents, where each of these is a blackboard system with its own knowledge sources, blackboard and control elements. In

this type of system the nodes communicate with each other through passing messages.

The main characteristics of REDSIEX that make it different to the remaining systems in its class, are the following:

- (1) The specification of organization in REDSIEX guarantees that each node can work independently on the solution of a whole problem, if the node has the resources, knowledge and skill to solve the problem independently. Frequently the nodes do not solve a single problem.
- (2) Depending on the availability of time and resources of the nodes, when cooperation is necessary it can be executed in real time or in deferred time.

Other recent related works are the following; The Architecture for Adaptive Intelligent Systems (Hayes-Roth, 1995), where the required task, configuration of available resources, contextual conditions, and the performance criteria of blackboard agents can vary dynamically; The OPUS Pipeline: A Partially Object-Oriented Pipeline System (Rose, Akella, Binigar, Choo, Heller-Boyer, Hester, Hyde, Perrine, Rose and Steurman, 1995), an object-oriented blackboard architecture for concurrent problem solving; and Using a Blackboard to Integrate Multiple Activities and Achieve Strategic Reasoning for Mobile-Robot Navigation (Liscano, Manz, Stuck, Fayek and Tigli, 1995), an activity-based blackboard system to control mobile robot navigation. In REDSIEX, the emergent cooperation at node level has the spirit of this last work.

9. Conclusions

In this article, we have presented the REDSIEX system, a cooperative network of expert systems nodes. The cooperative network presented has the following features:

- (1) Each node is a blackboard system with its own knowledge sources, blackboard, and control elements.
- (2) Each node is a specialist in a different kind of problem or subproblem.
- (3) Each node has a planner that develops sequences of problem-solving activities based on the current situation.
- (4) The nodes may transfer unsolved problems or subproblems to other nodes, which possess greater solution abilities.
- (5) The cooperation is intranode and internode.
- (6) The internode cooperation is emergent and facultative.
- (7) The internode cooperation can take place in real or deferred time.
- (8) A process of revision of belief values guarantees the consistency of the hypotheses created on the blackboard.

With this proposed architecture, we think we have tested a new approach for cooperative work in a network of problem-solving nodes. Our approach to cooperative solution of distributed problems is based on the characteristics and performance criteria of a particular class of distributed applications; the class of distributed blackboard systems. A practical development of REDSIEX has been implemented in the domain of medical interconsultation. This implementation solves expediently several problems of medical cooperation. Finally, we think that the architecture proposed in REDSIEX constitutes a valid new approach towards the cooperative solution of distributed problems.

References

- ENGELMORE, R., A.J. MORGAN and H.P. NII (1988) HEARSAY II, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- ERMAN, L.D., F. HAYES-ROTH, V.R. LESSER and D.R. REDDY (1988) The Hearsay II Speech-Understanding System: Integrating knowledge to Resolve Uncertainty, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- GONZÁLEZ, P.P. (1995) 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*, UNAM, México.
- GONZÁLEZ, P.P. and J. NEGRETE (1996) INTERMED: A cooperative network of expert systems for medical consultation and interconsultation, in *Memorias del V Congreso Iberoamericano de Inteligencia Artificial: IBERAMIA 96*, 469-483, México.
- HAYES-ROTH, B. (1985) A blackboard architecture for control, *Artificial Intelligence*, **26**, 251-321.
- HAYES-ROTH, B. (1995) An architecture for adaptive intelligent systems, *Artificial Intelligence*, **72**, 329-375.
- HAYES-ROTH, B. and M. HEWETT (1988) BBI: An implementation of the Blackboard Control Architecture, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- HAYES-ROTH, B., M. HEWETT, R. WASHINGTON, R. HEWETT and A. SEIVER (1985) Distributing intelligence within an individual, in L. Gasser and M. Huhns (Eds.), *Distributed Artificial Intelligence, volume II*, Pitman Publishing, London.
- LESSER, V.R. and D.D. CORKILL (1988) The Distributed Vehicle Monitoring Testbed: A tool for investigating distributed problem solving networks, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- LESSER, V.R. and L.D. ERMAN (1988) A retrospective view of the Hearsay II Architecture, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- LISCANO, R., A. MANZ, E.R. STUCK, R.E. FAYEK and J. TIGLI (1995) Using a blackboard to integrate multiple activities and

achieve strategic reasoning for mobile-robot navigation, *IEEE Expert*, April, 24-35.

- NII, H.P. (1989) Blackboard Systems, in A. Barr, P.R. Cohen and E.A. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence, volume IV*, Addison-Wesley Publishing Company, Stanford, Calif.
- NII, H. P., N. AIELLO and J. RICE (1988) Frameworks for concurrent problem solving: A report on CAGE and POLIGON, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.
- NII, H. P., N. AIELLO and J. RICE (1989) Experiment on CAGE and POLIGON: Measuring the performance of parallel blackboard systems, in L. Gasser and M.N. Huhns (Eds.), *Distributed Artificial Intelligence, volume II*, Pitman Publishers, London.
- ROSE, J., R. AKELLA, S. BINEGAR, T.H. CHOO, C. HELLER-BOYER, T. HESTER, P. HYDE, R. PERRINE, M.A. ROSE and K. STEURMAN (1995) The OPUS Pipeline: A Partially Object-Oriented Pipeline System, in R.A. Shaw, H.E. Payne and J.J.E. Hayes (Eds.), *ASP Conference Series*, vol. 77.
- ZANCONATO, R. (1988) BLOBS: An Object-Oriented Blackboard System Framework for reasoning in time, in R. Englemore and T. Morgan (Eds.), *Blackboard Systems*, Addison-Wesley Publishing Company, London.

The authors

Pedro Pablo González Pérez

Pedro Pablo González Pérez is currently a researcher at the Instituto de Investigaciones Biomédicas, UNAM, Mexico. He is a PhD candidate in Biomedical Science, within the Artificial Intelligence Area, and he received his MS in Biomedical Science, within the Artificial Intelligence Area from Instituto de Investigaciones Biomedicas, UNAM in 1995. He has been active in such diverse research fields as distributed expert systems, cooperative systems and emergent cooperation. His current research interests concern distributed artificial intelligence, artificial neural networks, adaptive behaviour, and robotics.

José Negrete-Martínez

José Negrete-Martínez received an MD in 1952 from the Facultad de Medicina UNAM, Mexico. He is currently a researcher at the Instituto de Investigaciones Biomedicas UNAM, Mexico and a professor in the Maestria en Inteligencia Artificial at Universidad Veracruzana, Xalapa, Mexico. He spent two years at the Laboratory of Neurocybernetics at MIT and two years at the Hertz Institute in Berlin. He is interested in Computational Neuro-Ethology and Artificial Intelligence.

Net of multi-agent expert systems with emergent control

José Negrete Martínez* and Pedro Pablo González Pérez

Instituto de Investigaciones Biomédicas/UNAM, Apartado Postal 70228, Ciudad Universitaria, Mexico City, 04510, México

Abstract

An aperture to expertise as a social outcome is to minimize supervised control: control of an expert system society and control of an agent's society in multi-agent expert systems. The present paper starts this aperture with autonomous behavioral agents within Expert Systems that exhibit bottom-up emergent control. The behavior-agents are either reasoning agents or communicating agents. The reasoning agents have a blackboard Domain Working Memory or a Meta Working Memory. The Expert Systems communicate with each other in a net for the solution of a problem. A practical development has been made in the medical domain of interconsultation that shows expedient problem-solving behavior. © 1998 Elsevier Science Ltd. All rights reserved

1. INTRODUCTION

Multi-agent Expert Systems (MAES) imply mechanisms of cooperative interaction that allow its multiple agents to work together on the solution of a problem in common. When the character of the agents is heterogeneous, the agents might have different tasks that could be conceptualized as behaviors in the sense that they have an internal or external world to 'perceive' and they act autonomously upon these worlds. What these worlds are, which behaviors there are and how they should act in the internal worlds of the MAES or in its outside world so to generate, without supervision, the solution of a problem is the subject of this communication. The outside world of these MAES are other MAES of the same kind and the user.

In this paper, we present a MAES net with Emergent Control prototype system for the development of applications: ECN-MAES. This is a dynamic network of problem-solving nodes, in which each node is an Expert System with Blackboard Architecture (Engelmore et al., 1988a,b; Nii, 1989; González & Negrete, 1996a) with bottom-up emergent control (Maes, 1990, 1991a,b) among its agents (or behaviors). In ECN-MAES each node can work independently in the solution of a problem, but when faced with a global problem, cooperation is necessary. This may be when a single

node does not possess the resources, the knowledge or sufficient expertise to solve a problem independently. Similarly, different nodes may have different expertise for solving different parts of a problem, different resources available for tackling the solution to a problem and distinct sets of knowledge or points of view about a problem.

This research touches upon issues that are fundamental to Artificial Intelligence such as the emergence of expertise control among autonomic nodes when each node (MAES) has an emergent control mechanism of its agents. The following four sections describe the kind of cooperative behavior in ECN-MAES, the dimensions and options adopted in its blackboard system, the classification of the memory components, the mechanism for behavior selection, and the revision of beliefs implemented.

2. ECN-MAES AND THE COOPERATIVE SOLUTION OF PROBLEMS

If a set of expert systems is to work effectively in a cooperative manner, then it is necessary to start with a minimum of reasoning competence, and a minimum of knowledge, processing and interaction. In ECN-MAES these minima are specifications of:

- communication;
- organization;
- reasoning;
- control;

* Author for correspondence. Present address: Maestría en Inteligencia Artificial/UV. Sebastián Camacho 5, Xalapa Centro, Xalapa, Veracruz, México.

- distribution.

The specifications of communication, reasoning and control are discussed later in this paper (Section 3.2). In a cooperative problem-solving network, organization specifies, in concrete terms, which are the responsibilities and the patterns of interaction for each one of the nodes in the network. Organization can set up a starting point for the performance of the network.

In ECN-MAES, the organization was specified on the basis of the following criteria:

- (1) assigning problem-solving behaviors to the nodes (MAES), which renders each node a specialist in a different kind of problem or subproblem;
- (2) indicating for the nodes the connections for the flow of information among them, in such a way that they may transfer unsolved problems or subproblems to other nodes which possess greater abilities for their solution.

In ECN-MAES the cooperation takes place through interagents and internode. The interagents' cooperation manifests itself through the combined work of all the node's local behaviors. This process is characterized by being an opportunistic and incremental process of creation and modification of the most promising solution elements on the Domain Working Memory (blackboard).

Internode cooperation is favored because the nodes can exchange hypotheses and partial results. This type of cooperation can take place in real or deferred time. This depends on the availability of time and resources of those nodes which have been asked to commit themselves. The cooperation in real time takes place when a node receives a reply to its request for commitment within a time lapse that it has dedicated to the problem-solving processes, for which the commitment was requested. Deferred cooperation takes place when the problem for which the commitment was requested has been abandoned temporarily, without having received a reply; this problem can be taken up again once the node receives an answer to the request for commitment.

There exist various dimensions for the breakdown of a centralized blackboard system in a distributed environment and various options within each of these (Erman et al., 1988). The structure of ECN-MAES takes into account the following dimensions and options for each dimension:

- (1) Information Dimension (a) Distribution of the blackboard (Domain Working Memory in ECN-MAES): the scope of a node's local blackboard defines its area of interest. (b) Transmission of hypotheses: a node transmits and receives hypotheses to and from a local subset of nodes.
- (2) Knowledge Dimension (a) Distribution of the knowl-

edge sources (behaviors in ECN-MAES): each node has a sub-set of knowledge sources. (b) Access to the blackboard by knowledge sources: the activation of a knowledge source may be reached only by the blackboard in each MAES.

- (3) Control Dimension (a) Distribution of the knowledge source activation: an event on the blackboard can activate only knowledge sources within the local node. (b) Distribution of the dispatch mechanism. Each node possesses its own dispatch mechanism, and carries out the dispatch based on local information.

3. THE ARCHITECTURE OF THE NODES IN ECN-MAES

Each node in ECN-MAES is architectonically an heterarchical system. Figure 1 shows the details of the architecture that characterizes the nodes in ECN-MAES. The structure of each node in the network defines two types of fundamental components: data base type components and behavior type components.

3.1. Memory Type Components

Some of the memory components are data base type components: structures of data on which the behaviors execute actions of search, creation and modification of elements belonging to the said structures. Each node in the network possesses the following data base type components.

- Domain Working Memory;
- Meta Working Memory;
- List of Behavior State of Activity (LBSA), which includes among others: Forward Reasoning Behavior

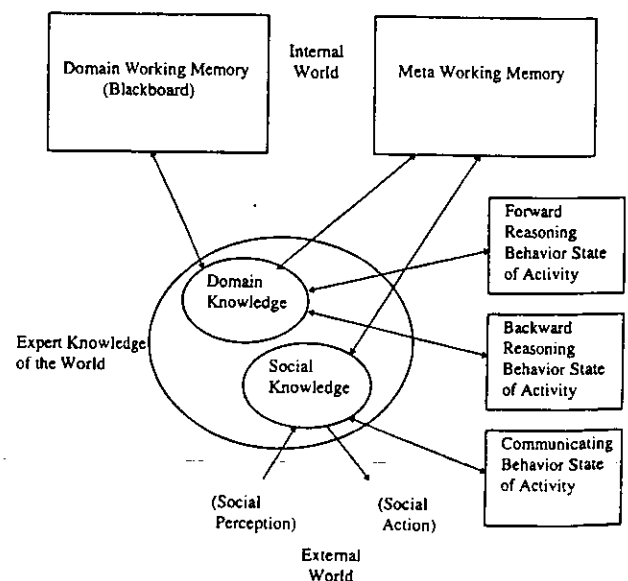


FIGURE 1. MAES memory architecture.

State of Activity, Backward Reasoning Behavior State of Activity, and Communicating Behavior State of Activity.

3.1.1. The Domain Working Memory. The Domain Working Memory (domain blackboard) is a shared structure of data through which the forward reasoning behaviors communicate with each other. The Domain Working Memory organizes the solution elements, also called hypotheses, at different levels of abstraction. These levels of abstraction represent the solution to the problem in different degrees of detail. Each hypothesis on the domain blackboard is associated with an identifier or number of the hypothesis and certainty or grade of a belief associated with the hypothesis.

3.1.2. The Meta Working Memory. The Meta Working Memory is a data structure defined by the architecture of ECN-MAES. The solution elements on the Meta Working Memory are registers whose attributes enclose information or decisions relevant to the problem of the action selection: solution elements for the revision of beliefs and solution elements for the investigation and communication of hypotheses. For each hypothesis on the Domain Working Memory, a solution element in the Meta Working Memory is associated with information corresponding to the certainty or the belief of the hypothesis. This information is used in the process of revising the hypothesis beliefs, which is described later in this paper. The solution elements for the investigation and communication of hypotheses constitute part of the control focus of the local node and are decisions about which of the solution elements on the Domain Working Memory should be investigated at a given moment (in accordance with the node's particular resources), which hypotheses should be transmitted as partial results to other nodes, or which questions should be formulated to pass to other nodes in the network. The communicating behaviors and backward reasoning behavior operate on the Meta Working Memory. These behaviors communicate with each other through the creation of solution elements on the Meta Working memory.

3.1.3. The List of Behavior State of Activity. Every behavior has a set of conditions, which have to be observed for the behavior to be 'executable' (Maes, 1991a). When the conditions or premises of a behavior are fulfilled, the behavior is activated, creating a behavior state of activity register (BSAR). A BSAR is a data structure that describes the characteristics of the action activated by the behavior. When a BSAR is selected by the action selection mechanism (discussed later) the behavior associated with the BSAR is executed in the context of the activated information and typically produces new events on the Domain Working Memory or on the Meta Working Memory, or else 'acts of communication' with other nodes on the network, depending on the nature of the behavior that originated the BSAR.

Not all the BSARs possess the same data structure. An appropriate data structure for BSARs is defined for each type of behavior. The types of BSARs defined for each node are as follows:

- Forward Reasoning Behavior State of Activity Register;
- Backward Reasoning Behavior State of Activity Register;
- Communicating Behavior State of Activity Register.

The data structure of a forward reasoning behavior state of activity register is shown in Table 1.

3.2. The Node Behavior Type Components and their Functioning

Node behavior type components execute the reasoning and communicating processes. Each node in the network possesses the following behavior type components:

- Forward Reasoning Behavior;
- Backward Reasoning Behavior;
- Communicating Behavior, which includes Receiver Behavior and Transmitter Behavior.

TABLE 1
Data Structure of a Forward Reasoning Behavior State of Activity Register

Attribute	Description
Activation cycle	Number of the execution cycle in which the behavioral state of activity was created
Behavior identifier	Number or identifier of the behavior that created the register
Rule identifier	Number or identifier of the activated inference rule
Action event	Domain Working Memory hypothesis that is to be created or modified
Matching level	Domain Working Memory level at which the condition of the activated rule was matched
Synthesis level	Domain Working Memory level at which the action of the activated rule will be executed
Rule matching strength	Certainty or belief value with which the condition of the rule was fulfilled
Rule importance	Certainty or belief value of the rule proposed by the expert
Rule efficiency	Certainty calculated for the rule based on its matching strength and importance
Final belief of the event	Certainty or final belief value calculated for the created event
Activation level	Value calculated for behavior based on the remaining parameters

The preconditions, final action and goal of every behavior are shown in Table 2.

3.2.1. Forward Reasoning Behaviors. The action executed by a forward reasoning behavior is linked to the creation or modification of solution elements on the Domain Working Memory. It is mediated by inference rules, which contain specific knowledge about the system's domain. A forward reasoning behavior has one identification part and one inferential part. The identification part describes the identification elements of the reasoning behavior, such as: identifier, matching level and synthesis level. The inferential part specifies the characteristics of the inference rules defined by the reasoning behavior (condition and final action). The condition of each inference rule describes the configuration of solution elements on the Domain Working Memory—preconditions in Maes' mechanism (Maes, 1991a,b)—that is necessary for the behavior to contribute to the problem-solving processes. The way in which a forward reasoning behavior contributes to the solution of a problem is specified in the action of each inference rule. The activity of the forward reasoning behaviors is driven by the events. This means that every change that is executed on the Domain Working Memory constitutes an event that, in the presence of other specific information on it, can activate one or more inference rules within a forward reasoning behavior. Each activation produces a unique forward reasoning behavior state of activity register.

3.2.2. Communicating Behaviors. Each node in the network receives messages from other nodes and selects hypotheses on the Domain Working Memory to send to other nodes. To carry out these activities of communication, the architecture of ECN-MAES defines two communicating behaviors for each node of the network: the receiver behavior and the transmitter behavior.

The receiver behavior decides which messages (from

those which up to the moment have been received by the node) are of interest in the solving of the problem in hand and it places them at the consideration of the selection mechanism. The receiver behavior possesses a condition-action format. The condition is fulfilled when a message of interest for the solving of the present problem is received. Each time its condition has been fulfilled the receiver behavior executes two actions. The first action consists of the inscription of a communicating behavior state of activity register on the list of behavior state of activity. The second action is carried out when its associated register has been selected. The execution of a communicating behavior depends on the recorded type of message. If the message is of the type 'offer' or 'reply to commitment', then the execution of the communicating behavior is associated with the creation or modification of solution elements on the Domain Working Memory. If the message is of the type 'request for commitment', then the execution of the communicating behavior is associated with the creation of a register of the type 'investigate hypothesis' on the Meta Working Memory.

The transmitter behavior transmits messages of the type 'offer commitment', 'reply to commitment' or 'request for commitment' to other nodes of the network. Messages of the type 'offer' or 'reply' are hypotheses selected on the Domain Working Memory, for each one of which the transmitter behavior finds on the Meta Working Memory a register of the type 'transmit hypothesis'. Messages of the type 'request' are associated with registers of the type 'ask hypothesis', which have been previously created on the control data base. Like the receiver behavior, the transmitter behavior also possesses a condition-action format. The condition is fulfilled when any register of the type 'transmit hypothesis' or 'ask hypothesis' is found on the Meta Working Memory. When the conditions of the transmitter behavior has been fulfilled, this is activated creating initially a state of activity register. Once the said register is selected, the transmitter behavior executes the final

TABLE 2
The Behaviors and Its Preconditions, Final Action and Goal

Behavior	Precondition	Final action	Goal
Forward reasoning	Domain Working Memory (DWM): event	DWM: writing	Permanent; data search
Backward reasoning	Not new events on DWM	Meta Working Memory (MWM): writing	Transient; support hypothesis
Explain	User request	Answer to the user	none
Transmit present state of the solution	Important partial or global solution	Information to the user	Permanent; global solution
Question to the user	Question type register on MWM	Question to the user	Transient; particular solution
Transmit to node	Global interest event or not new event on DWM	Node	Transient; particular or global solution
Receive from user	User request	DWM: writing	none
Receive from node	Node request	DWM: writing	none

action, which consists of transmitting the message associated with the register to any node or subset of nodes in the network.

3.2.3. Backward Reasoning Behavior. As we have already mentioned, the activity of the forward reasoning behaviors is directed by the events that occur on the Domain Working Memory. This type of activity constitutes a process of chaining forward the rules leading to the creation of solution elements on the Domain Working Memory, for the problem that needs to be solved. Like the forward reasoning behaviors, the backward reasoning behavior also contributes to the creation of solution elements on the Domain Working Memory although, unlike the former, the activity of this is determined by goals and not by events that occur on the Domain Working Memory.

The backward reasoning behavior has a condition-action format; the condition is fulfilled when a register of the type 'investigate hypothesis' is created on the Meta Working Memory. The activity of the backward reasoning behavior constitutes a process of rules-chaining leading to the construction of a sequence tree to generate solution elements. The sequences are initiated with the configuration of a goal state at the root of the tree. The next level of the tree is generated by finding all the rules whose consequent or actions match in the root node; were it possible to apply only these rules then the desired goal state could be generated. The antecedents or conditions of these rules are used to generate the nodes at this second level of the tree. The next level of the tree is generated by taking each node at the previous level and finding all the rules whose consequent or actions correspond to this, then the antecedents of the corresponding rules are used to generate the new nodes at the next level of the tree. The process continues until a configuration of solution elements only corresponding to initial states has been generated.

4. THE MECHANISM FOR BEHAVIOR SELECTION

When a behavior is 'executable', it creates a register of state of activity which describes the characteristics associated with the action that the behavior intends to execute. In each execution cycle only one behavior is executed. To determine the order in which the behaviors will execute their final actions, a competition is started among the registered states of activity, based on the activation level of each one. The behavior associated with the register of highest activation value will be selected to execute its final action in that execution cycle. No threshold value is used in this competition.

The behavior selection algorithm performs a loop, in which at every time step the following activities take place over all of the registered behavior states of activity:

- (1) Updating of the activation level for each registered state of activity.
- (2) Selection of the behavior state of activity with the highest activation level. Once the activation level has been updated for all the registered behavior states of activity, then the mechanism for behavior selection carries out the selection of the highest activation level register for its execution.
- (3) Executing of the behavior associated with the selected behavior state of activity. When the behavior with the highest activation level register has been selected for its execution, the mechanism for behavior selection invokes the behavior responsible for the creation of that register so that it executes its final action: the creation or modification of the corresponding solution element on the Domain Working Memory.

In this way, the three activities iterate in a three-step cycle, determining in each execution cycle which registered state of activity is to be executed and carrying out its execution.

5. THE REVISION OF BELIEFS

One of the principal characteristics that make ECN-MAES a consistent problem-solving system is its process of revision of beliefs in the spirit and justification argued by Shoham (1993). The revision of beliefs is the process of adjusting the belief values for the accepted hypotheses, created on the Domain Working Memory. If the belief value of a hypothesis on the Domain Working Memory is modified, then the belief values of the hypotheses found on all the lines of inference, with forward chaining starting with this hypothesis, will also be modified in accordance with this new value.

When the modification of a belief value of a hypothesis on the Domain Working Memory has taken place, the system consults whether this hypothesis has contributed to the creation of any other hypothesis at that same level or at higher levels of the blackboard. If this is the case, then a forward processing is executed, through which the rules whose conditions have already been fulfilled given the previous belief value of the hypothesis are activated again.

The process of revision of beliefs requires the consultation of the belief revision type elements, registered on the Meta Working Memory. This elements are the following:

- L-BI: List of identifiers of the local behaviors in a node that have contributed to support hypothesis.
- L-CC: List of certainty values associated to the L-BI identifiers.
- L-NI: List of node identifiers that have contributed to the support hypothesis.
- L-NC: List of the certainty values associated to the L-

NI identifiers.

For each behavior that has been fired a second time the following actions are executed:

- (1) Calculate the new matching strength of the behavior (antecedent certainty).
- (2) Calculate the new value of efficiency of the behavior (the contribution of the behavior).
- (3) Check in list L-BI, if at least one new identifier has been recorded, different from the behavior under analysis.

—If this is the case, calculate the global effect between the new efficiency value and the corresponding values again of uncertainty in the identifiers in L-BI. These certainty values are already recorded in list L-CC.

Check in list L-NI if at least one node identifier has been recorded of some other node of the net.

—If this is the case, calculate the global effect between the last uncertainty value corresponding to the identifiers recorded in L-NI again. These values are already recorded in L-NC.

The checking of the process of the beliefs continues until the belief values corresponding to all the hypothesis to be modified in the inference pathways have been modified. (See González & Negrete (1996a) for more details.)

6. DISCUSSION

6.1. The Behavior Selection in an ECN-MAES Node

- (1) The behaviors of our nodes are autonomous agents with characteristics referred (in Maes' terminology) as: type of behavior, preconditions and activation level.
- (2) The executable code is not encapsulated within the behavior and no add list and delete list are present. The Meta Working Memory plays their role.
- (3) It is worth noting that the world for some of the behaviors is internal and declarative: DWM and MWM.
- (4) Opportunistic forward reasoning is inherited from the blackboard architecture.
- (5) For the present stage of the prototype, the question of opportunistic backward reasoning is reduced to a default condition (see Table 2).
- (6) From the point of view of their implicit goals, forward reasoning and communicating behaviors are mainly 'appetitive' and backward reasoning is 'consummatory' (Maes, 1991a).
- (7) The internal control of each expert system node is unpredictable for the user, and in this sense emergent

and bottom-up. Experimentally the control converges to solve a problem, but for the moment the system behaves as expedient but not optimal (see Maes, 1991a).

6.2. The Inter-Node Cooperation in ECN-MAES

Some interesting characteristics of the internode cooperation in ECN-MAES are the following:

- (1) The cooperation at this level is also emergent, that is to say, it is a non-programmed property that arises dynamically (for example, unexpected but acceptable consultation of an inactive node) in the process of interconsultation among nodes. It could be also said that ECN-MAES exhibits a global emergent control partially inherited from the emergent control at node level. A case is when a node switches unexpectedly from a forward search to an acceptable backward search, and in this new process the communication with other nodes can also switch to an unexpected domain of expertise in other nodes.
- (2) The cooperation is facultative. In ECN-MAES, cooperation is a faculty possessed by each node in the network, and does not constitute a permanent necessity. This means that a node can solve a complete problem, if this node possesses the necessary capacity and resources to arrive at the solution. When this is not the case, then the node that is responsible for the problem can ask for cooperation from other nodes in the network. The ECN-MAES architecture does not foresee an administrator node with faculties for the breaking down of a task into sub-tasks so as to distribute them among different nodes in the network for their execution. Instead of this, the task is requested from a particular node in the network (which is presumed to possess the best abilities, knowledge and resources for carrying this out), making it responsible for the task and deciding if in any part of the problem-solving process the help from any other node in the network is required. Therefore, the ECN-MAES system can be seen as a collection of Expert Systems 'facultatively' associated in a cooperative network.
- (3) Cooperation can be executed in real time or deferred time. At any given time each node may find itself working on the solution of a different problem from those that are being solved by the other nodes in the network. However, even under these conditions it is possible for the nodes to establish cooperative links among themselves. This is achieved through two cooperation mechanisms defined in ECN-MAES: cooperation in real time and cooperation in deferred time. When the cooperation takes place in real time, the node that requests the commitment receives the reply within a time lapse that the node has dedicated to the processes of problem solving. Deferred

cooperation occurs when the reply to the requested commitment cannot be elaborated and sent within a given time lapse. This may occur because the node that has received the request for commitment is busy with the solution of another problem which for it has greater priority, and decides to conclude some process already begun before attending to the request received. When the node that requested the commitment receives the reply, it may be working on another problem and then it will decide whether to continue the processes leading to the solution of its present problem, or to take up again the unconcluded problem for which it had requested the commitment.

- (4) The process of revision of belief values guarantees the consistency of the hypotheses created on the Domain Working Memory. The new pieces of evidence obtained in favor of or against a hypothesis already created on the Domain Working Memory modify the belief value associated with this hypothesis. In turn, this causes the belief values of all the hypotheses created directly or indirectly from the modified hypothesis to be modified also in virtue of the new belief value calculated for the hypothesis.

7. DEVELOPMENT

ECN-MAES has been developed into a practical application as a cooperative network of expert systems for medical consultation and interconsultation: INTERMED. The INTERMED system (González, 1995; González & Negrete, 1996b) is a cooperative network of expert systems (nodes) that carries out the work of a group of medical specialists in medical consultation and interconsultation. We have developed INTERMED within ECN-MAES architecture.

In INTERMED each node possesses knowledge, reasoning methods and expertise to make diagnoses within a limited range of diseases belonging to a specific specialty. The INTERMED system covers four nodes which have been defined following the subdomains of interest that make up the global domain of a system: diagnosis and treatment of diseases of the digestive system, respiratory system and cardiovascular system, and analysis and interpretation of clinical laboratory tests.

A node is capable of carrying out alone a complete task of diagnosis, whenever it possesses all the knowledge, skill and resources that are necessary to find the solution to the problem in question. When this is not the case, the node carrying out the medical consultation can request help from the other nodes in the network, to complete the problem-solving processes and thus 'arrive at' an accurate diagnosis. In this way, each node or medical consultant in INTERMED is capable of executing diagnostic tasks within its domain or area of interest, and at the same time possesses faculties that permit it to

establish cooperative processes leading to the solution of its problem or to problems that concern other specialists in the network.

7.1. The Domain Working Memory

The INTERMED system defines structural and functional characteristics which are common to the domain working memory of the specialist nodes in the diagnosis and treatment of diseases, whereas the characteristics of the domain working memory of the specialist node in the analysis and interpretation of clinical laboratory tests are unique.

For specialist nodes in the diagnosis and treatment of diseases, the domain working memory organizes the solution elements or hypotheses in five levels of abstraction:

- 1st level: Signs and Symptoms, Coarse Grain (SSCG);
- 2nd level: Signs and Symptoms, Fine Grain (SSFG);
- 3rd level: Physiopathological Entities (PE);
- 4th level: Nosological Entities (NE);
- 5th level: Therapeutic (T).

For specialist node in the analysis and interpretation of clinical laboratory, the solution elements are organized at four levels of abstraction:

- 1st level: Laboratory Tests (LT);
- 2nd level: Laboratory Data (LD);
- 3rd level: Nosological (N);
- 4th level: Etiological (E).

7.2. The Behaviors

For each node in INTERMED two types of behaviors are defined: domain-dependent behaviors and behaviors independent of the domain. The domain-dependent behaviors (reasoning behaviors) possess specific knowledge about the domain system and their function consists of creating or modifying solution elements on the domain working memory. The behaviors independent of the domain are defined by ECN-MAES's own architecture.

For the specialist nodes in the diagnosis and treatment of diseases, INTERMED defines the following domain-dependent behaviors: refine-behavior, physiopath-behavior, nosology-behavior, and therapy-behavior.

The behaviors independent of domain includes the communicating behaviors, the backward reasoning behavior, and the stop behavior. The main role of the receiver and transmitter behaviors (receiver-behavior and transmitter-behavior) in the process of medical diagnosis is the establishment of communication channels among the expert nodes in the network, to carry out the processes of request and the sending of results from

laboratory tests, consulting room and medical inter-consultation.

An expert node uses the transmitter-behavior to request commitments from other nodes in the network so as to clarify a probable, inconclusive or doubtful diagnosis; to answer commitments for diagnosis that have been requested and to offer any relevant information (physiopathological entities or nosological entities that are present) which it has ascertained in its process of medical diagnosis and knows that this information could be useful for some other expert node in the network.

For its part, the receiver-behavior is in charge of receiving, classifying and selecting the information that has arrived at the node from other nodes in the network in reply to a commitment to interconsultation previously requested by the node, as a request for commitment to interconsultation made to the node, or as an offer made to it by some other node in the network.

The backward-behavior executes the final stage of the process of medical diagnosis: the differential diagnosis. Their function is to discern, from among all the diagnostic possibilities suggested as probable causes of a certain group of clinical manifestations, which of them is the most likely cause of the morbid process.

The stop-behavior possesses knowledge which allows it to decide when to stop or avoid stopping processes leading to the solution of a diagnosis problem under consideration. A reliable solution for the diagnosis problem under consideration has been reached when within a group of nosological entities, causes of a specific syndrome, or clinical manifestations, at least one disease has been verified with a high level of certainty and the rest have been discarded. A pending diagnosis problem corresponds to the state of a consultation already begun, for which the available resources have temporarily run out without a solution to the problem being reached. A new diagnosis problem is the beginning of a new consultation, for which there exists sufficient information from the external world. This information might correspond to a request for interconsultation made by any other expert in the network or simply the introduction of a new problem introduced by the expert node user.

When the information at the lower levels of the domain working memory (signs and symptoms) is insufficient, then the generation of new solution elements at the higher levels of the blackboard (syndromes and nosological entities) cannot take place. In such a situation, stop-behavior is triggered, recording a new register on the meta working memory. The creation of this register sparks off the activity of backward-behavior, which executes local plans leading to the creation of the most promising solution elements on the domain work-

ing memory. If the wait time for a request commitment (laboratory tests, consulting-room or interconsultation) has been exceeded or there are no resources to continue with the problem-solving processes, then stop-behavior decides to take up again a pending diagnosis problem for which it already possesses pertinent information (results of laboratory tests, consulting-room, or interconsultation) or initiate a new diagnosis problem based on a given formulation.

8. CONCLUSIONS

We think that the architecture we proposed in ECN-MAES opens an interesting window to search for expertise as a social outcome both in agent societies within Expert Systems and within societies of Expert Systems. The addition of more behaviors per node, refinements at the bottom-up emergent control, such as the introduction of 'motivations' at node level and implementation of a more structured emergent control at internode level (nodes can also be autonomous agents with emergent control), is our future research.

REFERENCES

- Engelmore, R., Morgan, T., & Nii, H. P. (1988a). Blackboard systems. In R. Engelmore & T. Morgan T. (Eds) *Blackboard Systems*. Reading, MA: Addison Wesley.
- Engelmore, R. Morgan, A. J., & Nii, H. P. (1988b). 'HEARSAY II'. In R. Engelmore & T. Morgan (Eds) *Blackboard Systems*. Reading, MA: Addison Wesley.
- Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1988). The Hearsay II speech-understanding system: integrating knowledge to resolve uncertainty. In R. Engelmore & T. Morgan (Eds) *Blackboard Systems*. Reading, MA: Addison Wesley.
- González, P. P. (1995). Sistemas expertos facultativamente asociados en red cooperativa con arquitecturas de Pizarrón: una aplicación en la consulta e interconsulta médica. M.Sc. thesis, UNAM, Mexico.
- González, P. P., & Negrete, J. (1996a). REDSIEX: a cooperative network of expert systems with blackboard architectures. In *Proceedings ISAI/IFIS*. Cancun, Mexico: ITESM.
- González, P. P., & Negrete, J. (1996b). INTERMED: una red cooperativa de sistemas expertos para la consulta e interconsulta médica. *Memorias IBERAMIA*. Puebla, Mexico: SMIA.
- Maes, P. (1990). Situated agents can have goals. In P. Maes (Ed.) *Designing Autonomous Agents. Theory and Practice from Biology to Engineering and Back*. Cambridge, MA: MIT Press.
- Maes, P. (1991a). Bottom-up mechanism for behavior selection in an artificial creature. In J. A. Meyer & S. W. Wilson (Eds) *Proceedings of the First International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.
- Maes, P. (1991b). Adaptive action selection. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum.
- Nii, H. P. (1989). Blackboard systems. In A. Barr, P. R. Cohen, & E. A. Feigenbaum (Eds) *The Handbook of Artificial Intelligence, Volume IV*. Reading, MA: Addison Wesley.
- Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1), 51–92.

A Model for Combination of External and Internal Stimuli in the Action Selection of an Autonomous Agent

Pedro Pablo González Pérez¹
Instituto de Investigaciones Biomédicas/UNAM
Instituto de Química/UNAM
ppgp@servidor.unam.mx

José Negrete Martínez
Instituto de Investigaciones Biomédicas/UNAM
Maestría en Inteligencia Artificial/UV
jnegrete@mia.uv.mx

Ariel Barreiro García
Centro de Cibernética Aplicada a la Medicina/ISCM-H
ariel@cecam.cu

Carlos Gershenson García
Instituto de Química/UNAM
Fundación Arturo Rosenblueth
Facultad de Filosofía y Letras/UNAM
carlos@jlagunez.iquimica.unam.mx

Abstract

This paper proposes a model for combination of external and internal stimuli for the action selection in an autonomous agent, based in an action selection mechanism previously proposed by the authors. This combination model includes additive and multiplicative elements, which allows to incorporate new properties, which enhance the action selection. A given parameter α , which is part of the proposed model, allows to regulate the degree of dependence of the observed external behaviour from the internal states of the entity.

Este artículo ha sido aceptado para su publicación en Proceedings MICAP'2000 / Springer Verlag

Action Selection Properties in a Software Simulated Agent

Carlos Gershenson García¹
Instituto de Química/UNAM
Fundación Arturo Rosenblueth
Facultad de Filosofía y Letras/UNAM
carlos@jlagunez.iquimica.unam.mx

Pedro Pablo González Pérez
Instituto de Investigaciones Biomédicas/UNAM
Instituto de Química/UNAM
ppgp@servidor.unam.mx

José Negrete Martínez
Instituto de Investigaciones Biomédicas/UNAM
Maestría en Inteligencia Artificial/UV
jnegrete@mia.uv.mx

Abstract

This article analyses the properties of the Internal Behaviour network, an action selection mechanism previously proposed by the authors, with the aid of a simulation developed for such ends. A brief review of the Internal Behaviour network is followed by the explanation of the implementation of the simulation. Then, experiments are presented and discussed analysing the properties of the action selection in the proposed model.

Este artículo ha sido aceptado para su publicación en Proceedings MICAI'2000 / Springer Verlag
