Capítulo 6

Esquema de Acción Selección

En la Sección 3.2 del Capítulo 3 se hizo una introducción sobre el paradigma de acción-selección para la animación de AVAs, presentando sus características y comentando algunos trabajos al respecto.

Dado que el problema de acción selección es resuelto mediante técnicas empleadas para seleccionar las acciones que ejecuta el agente en cada instante de tiempo, dichas técnicas son agrupadas en un modulo del Sistema de Control del Comportamiento (SCC).

En el capítulo anterior se presentó el SCC (Figura 5.1) y se describieron 3 de sus 4 módulos, dedicados a generar toda la información necesaria para la toma de decisiones del agente. El módulo restante (Motor de Conducta) es donde se resuelve el paradigma de acción selección para nuestro AVA, y es el tema que ocupa este capítulo.

El presente capítulo esta dedicado a describir los mecanismos que encierra el Motor de Conducta para el autocontrol del agente; es decir, aquellas técnicas que lo llevan a realizar unas acciones en lugar de otras, y de cómo esta elección de acciones toma lugar.

6.1 Motor de Conducta

El Motor de Conducta es un módulo del Sistema de Control del Comportamiento (Figura 5.1) encargado de definir el objetivo que el agente intentara alcanzar en cada instante de tiempo.

El problema asociado a la elección del objetivo también es conocido como *comportamiento arbitrario*, *elección del comportamiento*, o *acción selección*. Podemos encontrar interesantes aplicaciones a este problema en el campo de la robótica [ANDE90, MCFA93], así como en la animación de animales y humanos virtuales [CLIF99, FUNG99, THAL01, THAL04].

Es importante tener presente los estados por los que pasa un objetivo al ser seleccionado por un agente. Normalmente un objetivo estándar requiere varios pasos de una simulación para ser efectivamente alcanzado; por ejemplo, si en un parque virtual hay una banca disponible pero que se encuentra lejos de un agente cuyo objetivo es "descansar", algunas acciones previas deben tomar lugar: ubicar la banca, caminar hacia ella, evitar obstáculos (si los hay), y por último sentarse en la banca. Durante la caminata hacia la banca, el objetivo primario sigue siendo "descansar", pero el agente no es capaz de hacerlo aún (al no haber llegado a la banca). Al objetivo en esta situación lo llamaremos *objetivo candidato*. Una vez que el agente llega a la banca y se sienta en ella, comienza a recuperar sus energías, aunque todavía no este totalmente descansado. Nos referiremos a este estado como *objetivo en progreso*. Finalmente, cuando el agente ha descansado completamente, lo llamaremos *objetivo alcanzado*.

La elección de un objetivo determina la conducta del agente, y ésta a su vez, define el nivel de realismo de su simulación.

Todo agente comienza su simulación con un objetivo candidato inicial, el cual incluso puede ser elegido de manera aleatoria de una lista de objetivos factibles. Sin embargo, el tiempo que se debe mantener un objetivo como candidato, o la elección de un nuevo objetivo al haberse alcanzado el anterior (manteniendo la coherencia entre las acciones), no es una tarea trivial; por lo que el diseño de un mecanismo efectivo de selección de objetivos es uno de los mas importantes retos en la simulación de AVAs.

En este trabajo se presenta un nuevo esquema para resolver el paradigma de acción selección. El diseño propuesto es ilustrado en la Figura 6.1, al cual denominamos Motor de Conducta, y que es parte del Sistema de Control del Comportamiento de MaGeS.

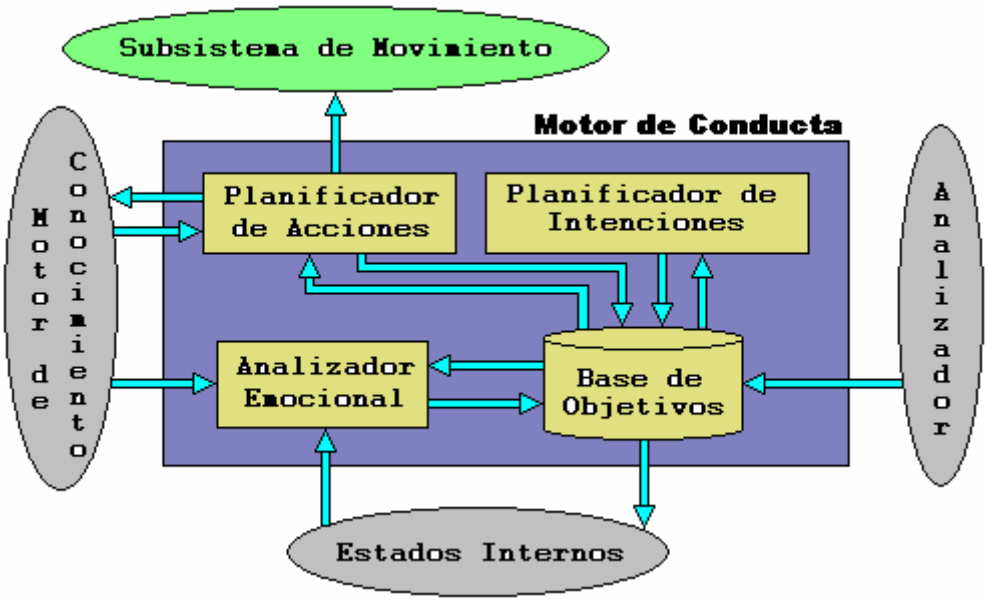


Figura 6.1. Motor de Conducta

El Motor de Conducta esta formado por tres módulos de procesamiento (el Planificador de Acciones, el Planificador de Intenciones y el Analizador Emocional) y por un registro de todos los objetivos que el agente puede pretender lograr (Base de Objetivos).

En la Figura 6.1 se aprecia el flujo de información entre las componentes internas del Motor de Conducta, entre el Motor de Conducta y el resto de componentes del SCC (Analizador, Estados Internos, y Motor de Conocimiento), y entre el Motor de Conducta y el Subsistema de Movimiento del SCF.

Las siguientes secciones describen cada una de estas componentes.

6.1.1 Base de Objetivos

Básicamente, el Motor de Conducta tiene como función seleccionar una actividad para ser realizada por el agente. Tales actividades vienen dadas en la forma de objetivos, y determinan toda la gama de conductas que el agente puede exhibir durante una simulación dentro de un entorno virtual; siendo reunidas y mantenidas en la Base de Objetivos.

La Base de Objetivos es una lista que almacena todos los objetivos del agente. Cada entrada en la lista corresponde a un objetivo con sus atributos, y su estructura es mostrada en la Figura 6.2.

Objetivo	Factibilidad	Prioridad	Ganas	HUU	Logro

Figura 6.2. Estructura de la Base de Objetivos

El campo Objetivo, en la estructura de la Figura 6.2, corresponde al código identificador del objetivo. Los objetivos considerados para el humano virtual en la implementación del parque incluyen los siguientes: Descansar, Pasear, Conocer otros Agentes, Charlar con un Amigo, Leer un libro, Vigilar a los Hijos, Perseguir Aves, Jugar en la Rueda (solo y/o acompañado), Jugar en un Subibaja, Ejercitarse; y para el ave virtual: Comer, Escapar, Dormir, y Pasear. Todos estos objetivos se identifican a través de un número único asociado a cada uno de ellos.

El campo Factibilidad sirve para almacenar un valor asociado a la factibilidad "física" de que el agente pueda alcanzar el respectivo objetivo. Por ejemplo, en un parque donde no haya ruedas, el objetivo "Jugar en la Rueda" será 0% factible; por otro lado, si el parque tiene ruedas, y el agente observa un de ellas vacía, la factibilidad del objetivo será de 100%. Otros factores que pueden afectar esta probabilidad son, que en la rueda haya otros agentes jugando, lo cual disminuye la probabilidad de éxito del objetivo al haber menos asientos disponibles en la rueda; recordemos que en cada rueda pueden jugar hasta 4 agentes (Sección 4.2.1.4). O también que el agente no vea ninguna rueda, pero recuerde donde había visto una; esto hace que la factibilidad de lograr el objetivo sea poca, pero existente. Esta factibilidad es actualizada por el Subsistema Analizador (Sección 5.3), y de allí la conexión entre éste y la Base de Objetivos mostrada en la Figura 6.1.

El resto de los campos de la estructura de la Base de Objetivos son actualizados internamente por las otras componentes del Motor de Conducta. El campo "Prioridad" almacena un valor que corresponde al orden actual de intención del agente para realizar el objetivo, mientras mas bajo sea este valor, mayor es la prioridad. El objetivo con prioridad 1 corresponde al objetivo en curso. Cuando se vaya a seleccionar un nuevo objetivo, el objetivo actual será sustituido por aquel con mayor prioridad. Este atributo es actualizado por el Planificador de Intenciones.

Si el campo Prioridad define la intencionalidad del agente, el campo Ganas establece su nivel de deseo. Desear algo no significa que las condiciones estén dadas para conseguirlo. El nivel de deseo de realizar cada objetivo depende de factores internos al agente y es actualizado por el Analizador Emocional, mientras que la intención (definida por la Prioridad) considera factores tanto internos como externos al agente. Un valor en el campo Ganas de 100, indica el máximo deseo del agente de querer realizar el objetivo asociado a dicho valor, con lo que a la menor oportunidad en la

que estén dadas las condiciones (una alta factibilidad del objetivo), seguramente su prioridad se colocará a 1.

El campo HUU almacena el momento de la simulación (tiempo) en que el objetivo fue seleccionado. Cuando un objetivo pasa a prioridad uno; es decir, es elegido para ejecutarse, el tiempo asociado al ciclo de simulación es almacenado en este campo. Este valor permite tener noción del tiempo que lleva un objetivo en determinado estatus, además de ofrecer información sobre el tiempo que lleva sin ser seleccionado.

Y finalmente, en el campo Logro se almacena el estatus del objetivo actual antes de ser sustituido por uno nuevo. Tanto este campo, como HUU son actualizados por el Planificador de Acciones, aunque inicializados por el Planificador de Intenciones.

Además de la estructura de lista para llevar registro de todos los objetivos del agente, la Base de Objetivos cuenta con una estructura de pila, la cual permite interrumpir momentáneamente la ejecución de un objetivo por otro cuya prioridad amerite su inmediata ejecución, y guardar en ella la información sobre el objetivo interrumpido. Por ejemplo, supongamos el caso del ave virtual cuyo objetivo actual es Comer. Mientras el ave come, un humano virtual puede acercarse peligrosamente al ave, por lo que ésta, aun sin haber alcanzado su objetivo, lo interrumpe por el objetivo Escapar cuya prioridad pasa a 1, por la proximidad del humano. La información sobre el objetivo interrumpido es almacenada en la lista, para que una vez que el ave alcance el objetivo Escapar, pueda retomar la actividad que estaba haciendo antes de ser abruptamente interrumpida. Esta estrategia del uso de la pila, para casos como el expuesto, fue presentada en [TERZ94] y permite mantener la coherencia entre las acciones, así como evitar saltar entre objetivos de máxima prioridad sin dedicarse a ninguno de ellos. Siendo el Planificador de Intenciones el encargado de actualizar la pila, en la Sección 6.1.3 se ahonda más al respecto.

6.1.2 Analizador Emocional

El módulo Analizador Emocional tiene como única función actualizar el valor del campo Ganas para cada objetivo en la Base de Objetivos. Como se comentara anteriormente, el valor del campo Ganas determina el nivel de deseo que tiene el agente de realizar el objetivo respectivo, sea este factible o no; ya que, por ejemplo, el deseo de conversar con alguien no implica la factibilidad de la acción, pudiendo la otra persona estar presente o no y el deseo de conversar permanecer igual.

Es usual encontrar en los trabajos de esta índole solo variables equivalentes al campo Ganas; es decir, variables representando un estado interno del agente y directamente relacionadas a un objetivo. Esto se debe a que normalmente se busca tener un indicador (asociado a un estado físico o mental) que mida la necesidad de ejecutar un objetivo particular. Por ejemplo, en [TERZ94] se manejaban variables como hambre o libido, llamadas estados emocionales (correspondiendo a estados internos) para activar objetivos específicos como alimentarse o aparearse, respectivamente, cuando ellas alcanzaban un cierto valor umbral. Del mismo modo sucede en [CAIC00], donde estados internos como hambre o felicidad marcan la pauta para la acción de comer o bailar.

En este trabajo, esas variables se han dejado como atributos de los objetivos (en el campo Ganas) y nuevas variables asociadas a estados físicos y mentales del agente, que pueden repercutir en el deseo de uno o varios objetivos, se mantienen en el Subsistema de Estados Internos para una más fácil actualización y mejor modelado de sus procesos.

El valor del campo Ganas se encuentra dentro del rango 0 y 100, y es actualizado de acuerdo a funciones matemáticas asociadas a cada objetivo. Tal como se aprecia en la Figura 6.1, las funciones del Analizador Emocional pueden utilizar parámetros procedentes del Motor de Conocimiento (rasgos característicos de personalidad como Sociabilidad, Dinamismo, etc), del Subsistema de Estados Internos (como el nivel de Energía, Timidez, etc) y de la Base de Objetivos (como el valor del campo Ganas de otros objetivos).

Cuando un objetivo ha sido seleccionado, es decir, que se convierte en objetivo activo, su campo Ganas deja de ser actualizado por la función respectiva del Analizador Emocional; siendo sustituido por el valor del estado interno correspondiente a la Ansiedad (Sección 5.4.4) mientras el objetivo se encuentra en estado Candidato (Sección 6.1), o con el valor del estado interno Aburrimiento (Sección 5.4.3) si el objetivo se encuentra en estado En Progreso.

De esta forma, Ganas se convierte en un indicador de cuando un objetivo debería ser seleccionado (actualizado por un modelo asociado), o de cuando debería de dejar de estar activo (actualizado con el valor del estado interno ansiedad/aburrimiento).

Denotaremos con $ganas_k$ la función que actualiza el valor del campo Ganas del objetivo k. Ahora, para definir cada una de ellas, se utilizan los siguientes parámetros,

Parámetro	Significado	Procedencia
S	Sociabilidad: constante que determina si el agente prefiere estar en compañía de otros agentes (valor alto), o solo (valor bajo).	Motor de Conocimiento
D	Dinamismo: constante que determina si el agente es adepto a las actividades físicas como hacer ejercicio o jugar (valores altos).	Motor de Conocimiento
H	Hora: tiempo de simulación del agente.	Motor de Conocimiento
r	Energía: variable que mide el nivel de energía del agente (valores altos = mucha energía, valores bajos = poca energía).	Estados Internos
t	Timidez: variable que mide el deseo de sociabilizar (valores bajos) del agente, o de estar solo (valores altos).	Estados Internos
L_k	Campo Logro del objetivo k	Base de Objetivos
HUU_k	Campo HUU del objetivo k	Base de Objetivos

Y las siguientes definiciones,

$$MH = \min_{k \neq j}\{HUU_k\}$$

donde j es el identificador del objetivo en curso, y MH es el valor del campo HUU del último objetivo en ser activado; y

$$\Omega_k = [\delta_k * ((H - HUU_k)/(H - MH))]$$

donde $\delta_k = \begin{cases} 1 & \text{Si } L_k \text{ es objetivo alcanzado} \\ 1.2 & \text{En otro caso} \end{cases}$, y Ω_k es un factor de peso para aumentar las ganas del objetivo mientras no sea activado.

A continuación pasamos a describir algunas funciones utilizadas para actualizar el nivel de deseo de cada objetivo.

Descansar: $ganas_1 = \begin{cases} 0 & \text{Si } e \geq 15 \\ 100 - r & \text{en otro caso} \end{cases}$

Pasear: $ganas_2 = [0.3r + 0.7t]\frac{S}{100}\Omega_2$

Conocer otros agentes (Conversar): $ganas_3 = (100 - t)\frac{S}{100}\Omega_3$

Leer: $ganas_4 = t\frac{S}{100}\Omega_4$

Perseguir Aves: $ganas_5 = \min\left\{100, \max\left\{0, e^{[r*8.82*D/\ 10000]}\right\}\right\} * \Omega_5$

Jugar: $ganas_6 = \left[0.5r + \frac{1}{2}(100 - t)\right]\frac{(D+S)}{200}\Omega_6$

Estas funciones solo son muestras de algunos tipos (condicionales, lineales, exponenciales) que pueden ser utilizadas para modelar el deseo para cada objetivo; sin embargo, consideramos ventajoso el uso de aquellas que son construidas como combinación lineal de los parámetros empleados, a fin de lograr un mejor control sobre el efecto de un mismo parámetro en múltiples funciones.

Ningún modelo es definitivo. Las funciones solo intentan cuantificar un nivel de deseo, e irán depurándose hasta lograr resultados que se consideren aceptables en las simulaciones.

6.1.3 Planificador de Intenciones

El módulo Planificador de Intenciones debe su nombre a que su única función es establecer la prioridad de ejecución de los objetivos, determinando, de esta forma, la intención del agente en cada instante de tiempo. Para ello se vale solo de tres atributos de los objetivos: la factibilidad, las ganas, y la hora de su última activación.

El Planificador de Intenciones es equivalente al Generador de Intenciones de [TERZ94], solo que ellos utilizan estrictamente un sistema basado en reglas, seleccionando aquel objetivo que satisface un conjunto de condiciones. En nuestro caso, la prioridad es directamente proporcional al producto entre la factibilidad y las ganas del objetivo. Entre mayor sea el producto, mayor será la prioridad (entendiendo por la prioridad mas alta aquella con valor de 1). En el caso de que el producto resultase igual para más de un objetivo, el empate es resuelto utilizando el campo HUU, aumentando la prioridad del objetivo cuanto más tiempo tenga éste sin ser activado. De esta forma se actualiza el campo Prioridad de todos los objetivos; salvo para el activo.

El objetivo activo tiene siempre prioridad 1, y cuando su campo Ganas alcanza un máximo (recordemos que para el objetivo activo, este campo almacena el estado interno ansiedad o aburrimiento, según el estatus del objetivo), es recalculado nuevamente utilizando su función asociada, para luego volver a actualizar su campo Prioridad; en este punto todos los objetivos compiten para ser seleccionados, y aquel de mayor prioridad es activado.

Podría ocurrir también que el agente necesitara ejecutar un objetivo sin haber alcanzado el anterior. Por ejemplo, un niño virtual que este jugando, y sin haberse aburrido de jugar, le da mucha sed; entonces interrumpe el juego para satisfacer el nuevo objetivo (beber agua), para luego volver a jugar. Esto ocurre porque mientras que el objetivo "jugar" tiene prioridad 1, las Ganas del objetivo "beber agua" alcanza su máximo, y al ser este un objetivo primario, su prioridad debería ser de 1 también. En esta situación, el Planificador de Intenciones se vale de la estructura de pila de la Base de Objetivos para almacenar allí el objetivo "jugar" y hacer activo el de "beber agua", ya que solo puede haber un objetivo activo a la vez. Alcanzado el objetivo "beber agua", se saca de la pila el objetivo "jugar", reactivándolo.

Cuando otro u otros objetivos, además del activo, alcanzan la prioridad 1, el Planificador de Intenciones se enfrenta a dos problemas, la persistencia y la intermitencia de objetivos. La persistencia se refiere a que se debe intentar mantener un objetivo activo el tiempo suficiente para ser alcanzado, a menos que las condiciones del entorno no sean las adecuadas para culminarla con éxito. La intermitencia ocurre cuando varios objetivos tienen prioridad 1 y el sistema trata de ejecutarlos todos saltando de un objetivo a otro en cada instante de tiempo, sin lograr alcanzar ninguno efectivamente.

Ambos problemas son resueltos utilizando la pila de la Base de Objetivos. Cuando más de un objetivo inactivo califica para prioridad 1, el Planificador de Intenciones determina cual de ellos es mas prioritario; si resulta ser el activo, el resto es colocado a prioridad 2, y todo sigue su curso. Si de los objetivos que califican a prioridad 1, el activo no es el más urgente entonces pasa a la pila de la Base de Objetivos, donde se almacena junto con el valor de sus campos Ganas, y HUU. Luego, de los objetivos inactivos que califican a prioridad 1 se selecciona el mas urgente para ser activado y la prioridad de los además se coloca a 2. Una vez alcanzado el nuevo objetivo, se revisa la pila y se restaura el objetivo pendiente. Cuando un objetivo es restaurado de la pila, tanto su factibilidad como su estatus pudieron haber cambiado, por lo que son ajustados para seguir con la ejecución del mismo. De esta forma se resuelven los problemas de persistencia e intermitencia de objetivos.

6.1.4 Planificador de Acciones

El módulo Planificador de Acciones es el encargado de definir la secuencia de acciones, paso a paso, que deberá ir ejecutando el agente, y comunicárselas al Subsistema de Movimiento para alcanzar el objetivo propuesto.

Las funciones del Planificador de Acciones por cada ciclo de animación, son las siguientes:

1. Determinar la siguiente acción que el agente debe ejecutar.
2. Comunicar al Subsistema de Movimiento dicha acción para ser ejecutada.
3. Actualizar el progreso del objetivo en la Base de Objetivos.
4. Actualizar el estatus del agente en el Motor de Conocimiento.

Para cada objetivo, el Planificador de Acciones cuenta con un elaborado Sistema Experto que le indica las acciones que debe realizar para alcanzar el objetivo, tomando en consideración cualquier factor que pudiese obstaculizar su logro. También cuenta con una variable llamada *estatus_accion*, que indica en qué situación (en cuanto a acciones) se encuentra el agente.

Lo primero que hace el Planificador de Acciones es chequear en la Base de Objetivos aquel con prioridad 1. Si es el mismo que se viene ejecutando, continua con el sistema experto para dicho objetivo en el punto donde lo dejo (indicado por *estatus_objetivo*). Si no es el mismo, ejecuta la secuencia de acciones dictaminada por el sistema experto para cambios de objetivo, el cual considera transiciones coherentes, como cambios de postura, para iniciar una nueva acción. Luego de determinada la acción, se actualiza el campo Logro en la Base de Objetivos (Sección 6.1.1), y la variable *status* en el Motor de Conocimiento, la cual, junto con la variable *comando* permiten la comunicación de la acción al Subsistema de Movimiento (Sección 5.1.1).

Este módulo es uno de los mas elaborados de todo el sistema, ya que por cada objetivo, son muchas las consideraciones que se deben tomar; tal como se ilustra en el siguiente ejemplo (cuya simulación aparece el en CD incluido en este trabajo en la sección Videos – Interacciones, número 8).

La Figura 6.3 muestra distintas escenas de una misma simulación, donde 5 agentes virtuales (todos niños) comparten el mismo objetivo, “jugar en el subibaja”. Como solo hay dos subibajas, los niños deben competir por

ellos. En la parte inferior de la imagen de la Figura 6.3a, un niño espera en uno de los puesto de un subibaja a que venga otro niño a jugar con él. Sin embargo, al observar que es el otro subibaja el que comienza a ser ocupado, decide irse hacia allá (Figura 6.3b). Pero otro niño se le adelanta, y al ver que ya no puede jugar en el segundo subibaja, se devuelve al primero en el que estaba (Figura 6.3c). Al devolverse, ve que un niño ocupa un asiento del único subibaja que queda disponible, por lo que se apresura a ocupar el otro asiento (Figura 6.3d). Finalmente, llega y se pone a jugar en el subibaja (Figura 6.3e). Todos estos cambios de acciones fueron producto de las distintas condiciones que se dieron en el entorno, y que influían en la estrategia del agente para lograr su objetivo; todas ellas llevadas a cabo por el Planificador de Acciones. Para el caso del niño que quedo sin asiento en los subibajas, la factibilidad de su objetivo ahora es cero, por lo que el Planificador de Intenciones cambia de objetivo a "jugar en la rueda"; lo que finalmente termina haciendo el agente (Figura 6.3f).

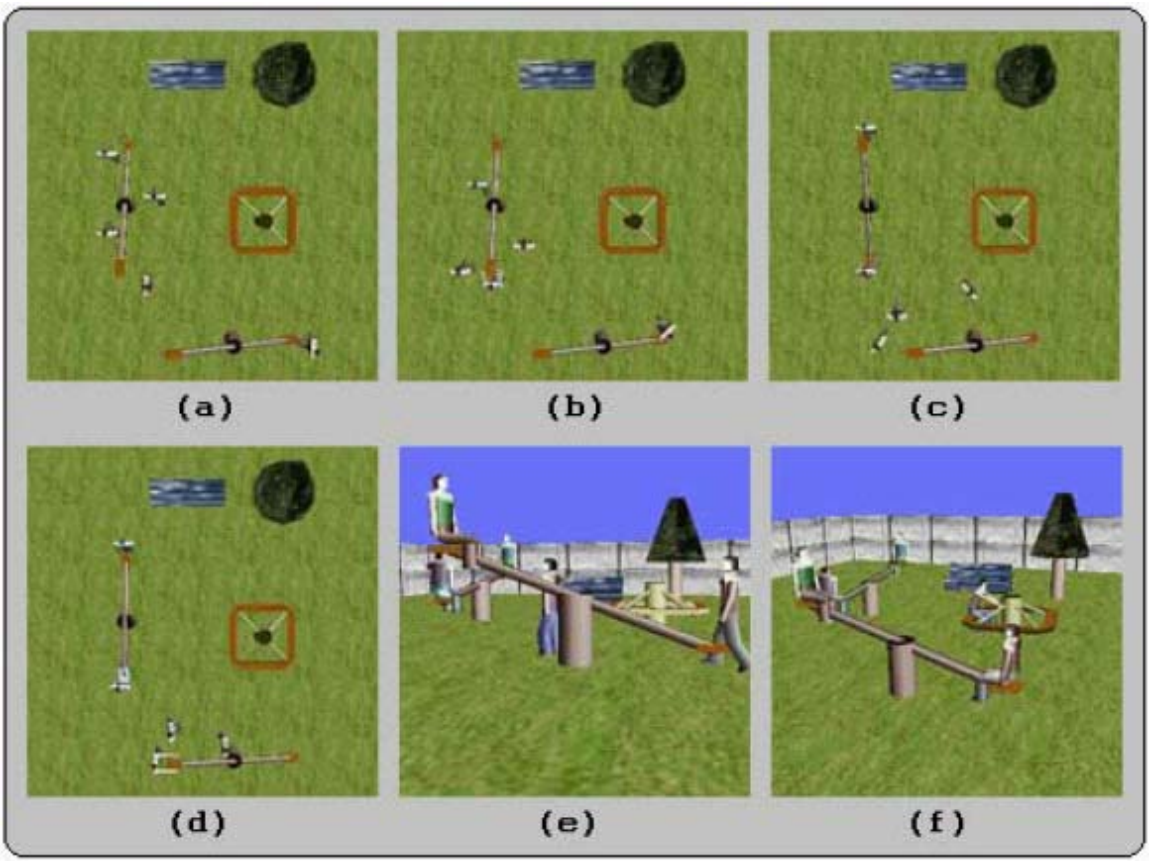


Figura 6.3. Ejemplo de los diferentes cambios de condiciones con los que se enfrenta el Planificador de Acciones para determinar las acciones de cada objetivo

A pesar de su complejidad, el Planificador de Acciones esta estructurado por objetivos, es decir, se desarrolla un sistema experto para el manejo y control de cada objetivo, sin que se afecten entre sí; por lo que se podría

sacar completamente un sistema experto asociado a un objetivo, sin afectar al resto, y luego sustituirlo con otro. Esta estructura permite un desarrollo incremental de objetivos, añadiendo nuevos sistemas expertos para darle mayor capacidad al agente.

Por otro lado, estos sistemas expertos cuentan con rutinas para búsquedas de caminos, detección de colisiones, y evasión de obstáculos; así como comandos relacionados a las rutinas del Subsistema de Movimiento, para simplificar su trabajo.

La principal rutina para la animación de todo agente virtual autónomo es el Algoritmo de Búsqueda de Caminos, ya que como se explicara en la Sección 2.1.3, es el que le permite al AVA desplazarse por el entorno, explorarlo, e interactuar con él.

La siguiente sección esta dedicada al algoritmo de búsqueda de caminos implementada en nuestro sistema.

6.1.4.1 Rutina de Búsqueda de Caminos

La búsqueda de caminos es probablemente uno de los más importantes problemas en el campo de la animación de agentes virtuales, aunque también es estudiado por muchas otras disciplinas (teoría de grafos, investigación de operaciones, etc); y si bien, muchos enfoques han sido propuestos para resolverlo, la gran variedad de consideraciones particulares para cada contexto de aplicación lo mantienen como un problema abierto.

Un algoritmo de búsqueda de caminos (también conocido como planificador de caminos) define la ruta óptima desde un punto de origen a un punto de destino, evitando los obstáculos que puedan aparecer en el trayecto. La “ruta óptima” dependerá del propósito que se persiga, por ejemplo, puede referirse a la ruta mas corta entre los dos puntos, o a la primera que se consiga, o a la de menor costo, etc.

Desde los trabajos de Dijkstra [DIJK59] hasta la actualidad, son muchos los algoritmos propuestos y las estrategias utilizadas para resolver el problema de la búsqueda de caminos en sus diferentes variantes.

Ya que no es objetivo de esta investigación resolver dicho problema con una nueva técnica de búsqueda, y dadas las características propias de nuestro trabajo, se implementa el muy conocido algoritmo A*, introducido originalmente en [NILS82]. Este algoritmo es la elección mas común para los

diseñadores de video juegos porque es muy flexible y puede ser utilizado en una gran variedad de contextos.

El algoritmo A* es esencialmente el algoritmo de Dijkstra para encontrar el camino mas corto entre dos puntos, pero mejorado, ya que incorpora conocimiento heurístico para guiar la búsqueda de manera de llegar mas rápido al destino. Detalles sobre la implementación de este algoritmo para 2 y 3D puede encontrarse en [DELO00].

Sin embargo, la implementación básica del algoritmo no es suficiente, ya que las características propias del AVA y del entorno dinámico de simulación no garantizan la evasión efectiva de obstáculos, pudiendo ocurrir colisiones. Esto es debido, en primer lugar, a que el AVA tiene un rango de visión limitado, y cuando se plantea una ruta, el algoritmo de búsqueda de caminos puede no considerar todos los obstáculos, al quedar algunos de estos fuera de su campo de visibilidad o escondidos por otros objetos. El otro caso se presenta cuando la ruta de dos agentes se intercepta provocando una colisión, o un agente se detiene sobre la ruta de otro, ofreciendo un obstáculo que antes no existía.

Para resolver estas situaciones, nuestra implementación del algoritmo procede de la siguiente manera:

1. Se aplica el algoritmo A* para establecer una ruta desde la posición del agente a la posición de destino, considerando como obstáculos solo aquellos objetos u otros agentes que se encuentran dentro de su campo de visión o que son recordados por él.
2. Chequeando solo los nuevos obstáculos que van apareciendo dentro de su campo de visión, se verifica que ninguno este sobre el trayecto trazado por el algoritmo. De ser así, la ruta se mantiene. Si por el contrario, aparece un objeto obstaculizando el camino, se descarta la ruta y se vuelve a aplicar el algoritmo A* desde la nueva posición del agente hasta el punto de destino.
3. Si el obstáculo es presentado por otro agente, las consideraciones son distintas antes de descartar la ruta establecida. Cuando un agente encuentra a otro sobre su ruta, comprueba si ese otro agente esta atravesando su trayectoria o simplemente esta solo parado sobre ella. Para el primer caso, el Planificador de Acciones detiene al agente para esperar que el otro termine de pasar, y así poder seguir su ruta. Si por el contrario, el otro agente se encuentra quieto obstruyendo el camino, la ruta es descartada y vuelta a definir con el algoritmo A*; siempre desde la actual posición del agente.

4. El Planificador de Acciones siempre comprueba que en su siguiente paso el agente no va a encontrar obstáculo, como por ejemplo, otro agente cruzándose en su camino. Con información visual de otros agentes (como su posición, estatus, etc) el Planificador de Acciones anticipa una posible colisión, deteniendo al agente que controla para que el otro pase, o continuando su camino si no hay peligro de choque.

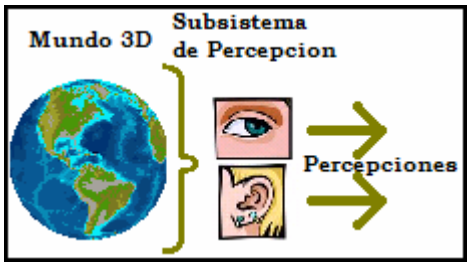
Ya que el costo computacional asociado a una búsqueda de camino no suele ser bajo, siempre es preferible buscar los medios para no tener que recalcular dicha ruta.

6.2 Ciclo de Simulación

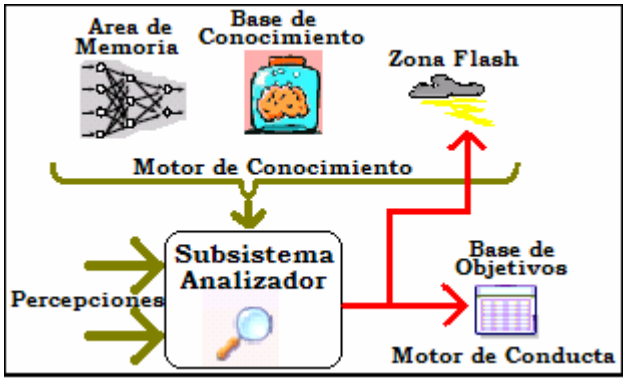
Hasta este punto se han explicado todas las componentes que conforman una implementación de la arquitectura MaGeS para la simulación de agentes virtuales autónomos.

Esta sección esta dedicada a hacer un breve repaso de todas ellas, ilustrando todo el proceso llevado a cabo por MaGeS durante un ciclo de simulación.

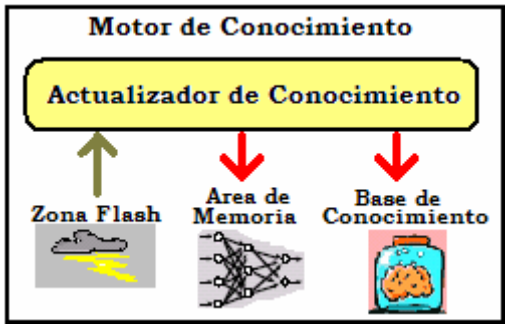
El proceso de animación del AVA durante cada ciclo es el siguiente:



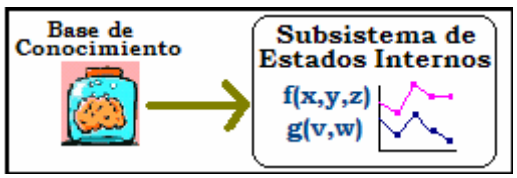
Al inicio de cada ciclo de simulación, la primera rutina en activarse es la asociada al sentido de la vista (Sección 4.3.1.2), la cual identifica los objetos del entorno 3D que entran dentro del campo de visión del agente, tomando su información y transformándola en formato de percepción. Del mismo modo, la rutina asociada al sentido de la audición (Sección 4.3.1.3) busca en el buffer_oido (usado para almacenar todo los sonidos que se producen en el entorno) frases dirigidas al agente, o sonidos que pueda escuchar, y transforma esa información en formato de percepción. Ambos conjuntos de percepciones (los de la vista y del oído) son enviados al Sistema de Control del Comportamiento cuyo Subsistema Analizador es quien los recibe.



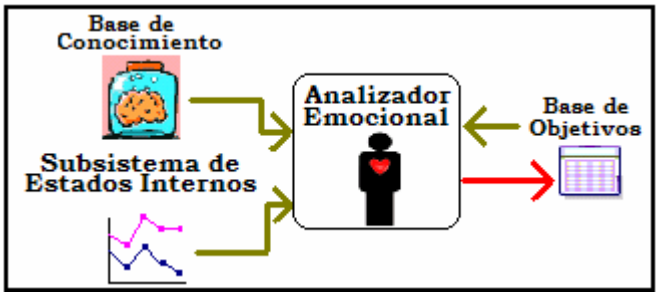
Las percepciones llegan al Subsistema Analizador (Sección 5.3), quien utilizando el contenido del Motor de Conocimiento, establece el nivel de impacto de cada percepción, para almacenarlas luego en la Zona Flash (la cual retiene solo lo que el agente esta percibiendo). Por otro lado, y empleando toda la información disponible por el Motor de Conocimiento, un sistema experto basado en lógica difusa del Subsistema Analizador, determina la probabilidad de éxito para cada objetivo y la actualiza en la Base de Objetivos del Motor de Conducta.



Luego, el conocimiento del agente es actualizado en el Motor de Conocimiento (Sección 5.2). Lo primero es activar el mecanismo de aprendizaje de la red neuronal que mantiene los recuerdos de los elementos virtuales que ha visto el agente (Área de Memoria). Esto lo realiza el Actualizador de Conocimiento pasándole como entrada, a la red neuronal, la información de la Zona Flash. El Actualizador de Conocimiento también activa el sistema experto de la Base de Conocimiento, el cual comprueba si nuevo conocimiento puede ser inferido de la actual información que maneja el agente.



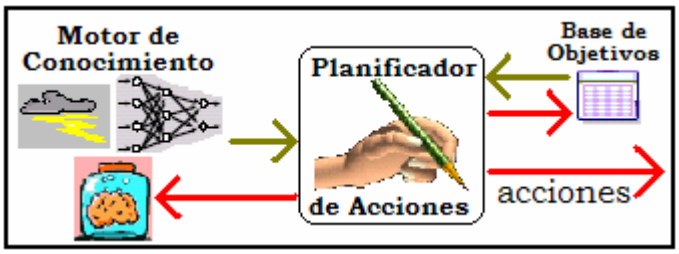
El siguiente paso es actualizar los estados internos (Sección 5.4) del agente, ajustando su nivel de energía (según el esfuerzo hecho), y los niveles de soledad, hambre, dinamismo, etc, según sus características propias y su estado actual, almacenados en la Base de Conocimiento.



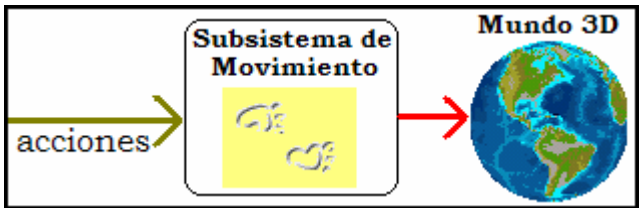
Siguiendo con la actualización de parámetros internos, pero esta vez en el Motor de Conducta, el Analizador Emocional (Sección 6.1.2) calcula el nivel de deseo del agente para

cada objetivo empleando información sobre ellos mismos (tomados de la Base de Objetivos) y sobre el agente (tomados de la Base de Conocimiento). Pero, el nivel de deseo del objetivo activo es actualizado con los valores de los estados internos Ansiedad o Aburrimiento, según el estatus del objetivo.

Una vez actualizadas la Ganas del agente; es decir, su nivel de deseo para cada objetivo, el Planificador de Intenciones (Sección 6.1.3) determina si mantener el objetivo actual o cambiarlo por otro. Esta decisión dependerá principalmente de los campos Ganas y Factibilidad, tanto del objetivo actual, como del resto de los objetivos en la Base de Objetivos, para establecer la prioridad de cada uno de ellos.



Luego, el Planificador de Acciones (Sección 6.1.4) consulta en la Base de Objetivos cuál es el objetivo actual, selecciona el sistema experto que maneja las acciones para dicho objetivo (como caminar evitando obstáculos, activar algún objeto inteligente, etc), y una vez determinadas, las envía al Subsistema de Movimiento del Sistema de Control Físico. Para definir la secuencia de acciones que se deben ejecutar se toma en cuenta el conocimiento del agente (lo que ve, recuerda, y sabe). Además de esto, el Planificador de Acciones también actualiza el estatus del agente en la Base de Conocimiento, y su progreso en la Base de Objetivos.



Finalmente, el Subsistema de Movimiento (Sección 4.3.2), ejecuta las acciones especificadas por el Planificador de Acciones, reflejándose físicamente dentro del entorno 3D y animando al agente.

De esta forma se lleva a cabo cada ciclo de simulación de un AVA. En el Capítulo 8 se comentan varias secuencias de simulación, en las que nos centramos en explicar aspectos particulares de la ejecución de algunos de los módulos de MaGeS, analizándolos desde un punto de vista más práctico.