

Evocación de hábitos en personajes virtuales mediante Mapas Cognitivos Difusos y técnicas de videojuegos

J.Carlos Conde-Ramírez, Abraham Sánchez-López

Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla, México

{juanc.conde, asanchez}@cs.buap.mx

Resumen. Los últimos avances tecnológicos en realismo de personajes virtuales y sus ambientes han vuelto necesaria la adaptación autónoma de dichos personajes a situaciones particulares en tiempo real. Esta investigación propone un proceso simple para crear agentes virtuales que aparenten mayor inteligencia a través de sus comportamientos. Se verifica como el modelado cognitivo de personajes permite alcanzar respuestas apropiadas a estímulos externos y deseos. Como prueba se implementó un mundo submarino donde los comportamientos reportados están basados en una arquitectura cognitiva propia (MoCAMG) combinada con técnicas comunes de videojuegos y Mapas Cognitivos Difusos no aumentados. La definición e identificación de patrones de comportamiento, reportados al final, justifican la importancia de la arquitectura cognitiva utilizada. Ésto sirve como base para investigaciones futuras en fenómenos detallados relacionados con la memoria.

Palabras clave: Modelado cognitivo, comportamiento, adaptación autónoma, personajes virtuales, agentes autónomos, videojuegos.

1. Introducción

Para que las aplicaciones gráficas modernas puedan generar procesos mecánicos realistas en personajes, basados en elementos cognitivos, existen estudios en seres vivos e información (psicológica, anatómica, etológica, etc.) de sobra que puede ser de gran utilidad. Como se menciona en [19], un modelo cognitivo adecuado puede proporcionar los datos necesarios para generar simulaciones validadas por modelos teóricos. Hoy en día aplicaciones de este estilo, como auto-animaciones y videojuegos, necesitan implementar modelos cognitivos que le permita a un personaje reflejar un comportamiento congruente con su entorno y situación particular.

En específico, este trabajo muestra cómo a partir de un conjunto de requerimientos obtenidos en tiempo real (percepción), es posible *pre-diseñar* agentes o personajes autónomos sin tener que especificar cada uno de los detalles de su comportamiento (control de alto nivel). Como evidencia se reproduce el comportamiento peces inmersos en un ambiente submarino. Se enfatiza la sencillez

de utilizar Mapas Cognitivos Difusos (FCM's) independientes o no aumentados, ni anidados, para obtener comportamientos congruentes con patrones de comportamiento o hábitos. Estos patrones se obtienen combinando técnicas de sensado usadas en videojuegos y la definición *off-line* de un di-grafo tratado de forma difusa. Se destaca el hecho de que el uso de FCM's no excluye a los personajes de enfrentarse a situaciones imprevistas, pero integra acciones primarias o primitivas en acciones motivadas por factores dinámicos.

A continuación se describe cómo está estructurado este documento. La *Sección 2* describe cuáles son los componentes de comportamiento en los seres vivos más relevantes para este trabajo. Algunos de los enfoques más utilizados para representación del conocimiento en este tipo de trabajos se presentan en la *Sección 3*. La *Sección 4* resume algunos de los trabajos relacionados con el área de animación y videojuegos. Por su parte, la *Sección 5* contiene las especificaciones de la arquitectura utilizada y una metodología para el modelado cognitivo de personajes virtuales. En la *Sección 6*, se muestra la implementación de un sistema que simula un comportamiento básico (supervivencia) en peces, con el objetivo de probar la utilidad de la arquitectura y la metodología propuesta, utilizando FCM's simples. Como resultado, la *Sección 7* especifica todos los posibles patrones de comportamiento obtenidos de la ejecución del sistema implementado. Para concluir, la *Sección 8* destaca la importancia de este trabajo y la dirección que tomará el trabajo futuro.

2. Elementos comportamentales

Las prioridades u objetivos primordiales para todos los seres vivos, tal como sobrevivir o reproducirse, son objetivos que pueden descomponerse en otros más inmediatos.

Además, es sabido que los seres vivos enfocan su atención de dos maneras: utilizando órganos de percepción especializados y enfocando su atención de forma cognitiva. Son características importantes que mejoran el procesamiento de la información sensorial. Por lo tanto, cualquier sistema que intente simular tales características debe considerar los siguientes factores [20]:

- El *ambiente*, los *estímulos externos* y los *deseos*.
- El proceso requerido para seleccionar una acción.
- La animación comportamental derivada.

Aquí el problema crítico es seleccionar una de todas las posibles acciones. Ésto conduce a un problema de diseño que puede ser resuelto de la siguiente manera:

1. Identificando los principios por los cuales un ser vivo selecciona sus acciones.
2. Representando y organizando el conocimiento de manera eficiente.

La selección de acciones basada en el razonamiento involucra el uso de técnicas de Inteligencia Artificial llamadas *planificación de movimiento a nivel de*

tarea. Por su parte, las secuencia de acciones que le permiten a un personaje artificial ser autónomo y “sobrevivir” en ambientes dinámicos corresponde a un comportamiento básico conocido como comportamiento reactivo o adaptativo [16,17].

3. Representación del conocimiento

La representación del conocimiento y el razonamiento es un área de la inteligencia artificial cuyo objetivo fundamental es representar el conocimiento de una manera que facilite la inferencia (sacar conclusiones) a partir de dicho conocimiento. Junto a una teoría de interpretación, dan significado a las frases en la lógica.

En este caso, un personaje realiza una acción de “percepción” (sensado) para actualizar su información y entonces hacer posible la replaneación de sus acciones. Por eso es importante que la arquitectura utilizada en la implementación de modelos cognitivos esté armada con un enfoque viable para representar la incertidumbre.

3.1. Árboles de decisión

Según la literatura, usando *axiomas de precondition* es posible definir sentencias que especifiquen cuál es el estado del mundo antes de realizar alguna acción. Los efectos de una acción están dados por los *axiomas de efecto*. *Acciones*, *axiomas de efecto* y *axiomas de precondition* pueden ser expresados en forma de árbol, donde:

- Los nodos representan las situaciones.
- Los axiomas de efecto describen las características de cada situación (la raíz del árbol corresponde a la situación inicial s_0).
- Los axiomas de precondition permiten saber cuáles son las secuencias de acciones permitidas.

Además cada ruta sobre el árbol puede representar una posible secuencia de acciones. Definiendo algunas situaciones (nodos) como “objetivos” es posible utilizar programación basada en lógica convencional y realizar búsquedas para hallar una secuencia de acciones que conduzcan al personaje hacia su objetivo [9].

3.2. Mapas Cognitivos Difusos

Un ambiente virtual cambia con el tiempo conforme el actor se mueve, por lo que existe la posibilidad de que el actor modifique su entorno o que su entorno lo cambie a él. De acuerdo a Julie A. Dickerson y Bart Kosko en [6], los cambios en un mundo virtual son causales y a su vez esta causalidad retroalimenta otros cambios.

Modelos matemáticos tales como las ecuaciones de Navier-Stokes, las ecuaciones de cinemática inversa o las ecuaciones diferenciales acopladas, son modelos

que pueden ayudar a resolver cambios en un mundo virtual. La desventaja es que son difíciles de encontrar, difíciles de resolver y difíciles de ejecutar en tiempo real. Usualmente los árboles de decisión “facilitan” la elección de acciones en agentes artificiales a través de búsquedas que retroalimentan el conocimiento del personaje. Sin embargo, cada inferencia utiliza sólo una pequeña parte del conocimiento almacenado, sin mencionar que el diseñador debe anticipar qué acción podría ser seleccionada bajo todas las posibles condiciones.

En contraste, los Mapas Cognitivos Difusos (FCM's) [14] son redes causales representados por un grafo dirigido difuso, donde los nodos representan conceptos, acciones o deseos, y los ejes causales establecen las reglas difusas entre dichos conceptos. Además cada entrada dispara todas las reglas, en cierto grado, para modelar una “causalidad circular” en mundos virtuales realistas.

Por si mismos, los FCM's actúan como un sistema dinámico no lineal (tal como en una Red Neuronal) donde las entradas son mapeadas con estados de equilibrio como salida. Por lo tanto, en un FCM simple la ruta concluye en un punto fijo o en un ciclo limitado.

4. Trabajo relacionado

Los problemas relacionados con la creación de personajes virtuales con mejores esqueletos y cuerpos deformables más realistas son considerados en la mayoría de trabajos recientes en el campo de gráficos y animación por computadora. No obstante, la problemática de brindar un nivel de comportamiento creíble recae en el reto de generar movimientos libres y suaves dentro de ambientes utilizando capacidades de “percepción” realistas. Hasta ahora, un área de investigación poco abordada en comparación con otras.

Uno de los trabajos importantes en el área es el de N. Magnenant-Thalman y D. Thalman, quienes presentan los resultados de crear humanos virtuales realistas. La interacción y comportamientos grupales son parámetros importantes, por lo que se concluye que la *percepción realista* y el *estado interno* del “actor” debe verse reflejado en su comportamiento [15].

Por su parte, estudios en *emociones artificiales* demuestran que la implementación de *modelos cognitivos* en aplicaciones gráficas, como videojuegos, generan una interacción con los usuarios más notable. En [13], los autores diseñan y prueban modelos de emoción artificial en un personaje. El trabajo consiste en una herramienta de pre-visualización que puede ser utilizada para planear niveles en videojuegos. En el mismo sentido, los autores de [21] proponen un sistema basado en modelos de comportamiento que revelan la interrelación jugador-ambiente.

Un área de oportunidad puede ser la mejora en motores de juegos. Un enfoque como éste es propuesto por E. Hudlicka en [12], en donde se establecen los requerimientos a satisfacer por parte de los trabajos y aplicaciones de este tipo. El objetivo es mejorar el realismo social y afectivo de los personajes identificando “emociones” en los personajes.

Finalmente, un análisis crítico realizado por T. Giang en [11] revela los enfoques de alto nivel empeñados en identificar las áreas clave de interés para in-

vestigaciones futuras relacionadas con las *Técnicas de Animación de Personajes en Tiempo-Real*.

5. Propuesta teórica

La arquitectura MoCAMG (Movis's Cognitive Architecture for Modeling Game-characters) mostrada en la Figura 1, combina algunos componentes de dos arquitecturas bastante conocidas; ACT-R and Soar [1,18].

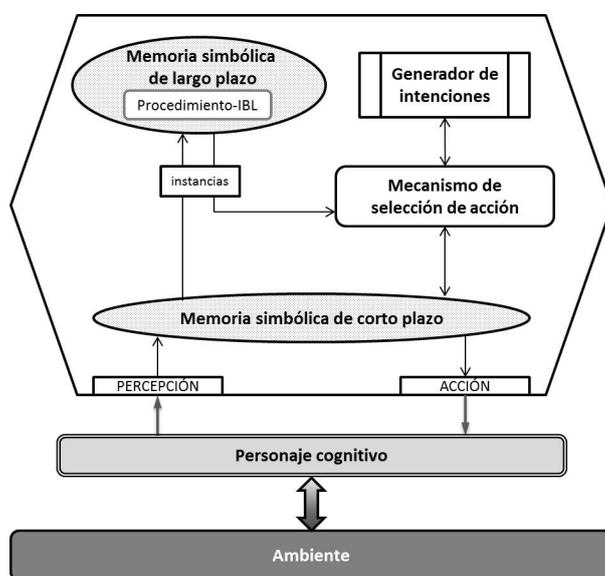


Fig. 1. Vista de alto nivel de la arquitectura MoCAMG.

La arquitectura propuesta posee dos tipos de memoria: declarativa (datos) y procedural (reglas). La memoria declarativa está asociada con objetivos y es removible, por lo que es considerada como una *memoria de corto plazo*. Por su parte, en una memoria procedural las “reglas” pueden ser actualizadas o agregadas, pero no removidas en tiempo de ejecución tal como en una *memoria de largo plazo*.

En este trabajo nos enfocaremos en el módulo nombrado Generador de Intenciones, que es el componente clave para obtener patrones de comportamientos. Funciona como arbitrador de comportamientos cuya activación depende del estado interno y externo de la entidad cognitiva. En este caso, el Mecanismo de Selección de Acciones combina comportamientos primitivos en comportamientos motivados.

Los detalles relativos a los demás componentes se describen en [4]. La única observación importante de este trabajo es que se actualizó el tipo de conexión

entre el Generador de Intenciones y el Mecanismo de Selección de Acciones en el diagrama de la arquitectura, ya que en la práctica se comprobó que existe un flujo de trabajo bidireccional entre estos dos componentes (Figura 1).

5.1. Metodología para el modelado cognitivo en personajes virtuales

Esta metodología (Figura 2) está basada en la jerarquía de modelado en animación comportamental, la cual ha dado como resultado personajes virtuales auto-animados dotados de comportamientos realistas [10,9,8,5].

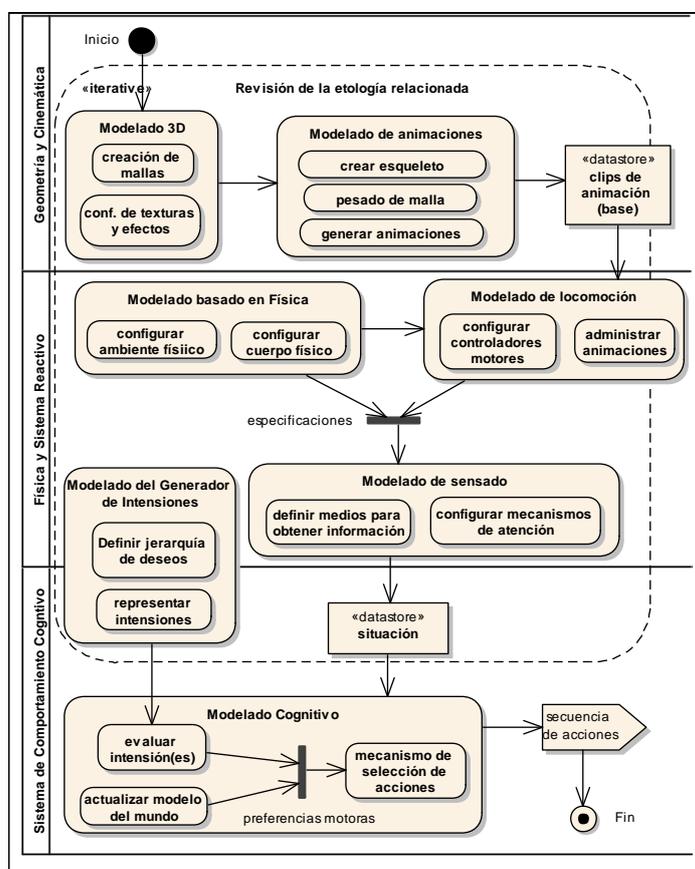


Fig. 2. Descripción de las actividades para el modelado cognitivo de personajes virtuales.

Esta metodología considera tres niveles de modelado (1) modelado de apariencia realista, (2) modelado de movimientos realistas, suaves y flexibles, (3) modelado de comportamientos realistas de alto nivel.

De acuerdo a lo que le atañe a este trabajo, por ahora sólo es necesario definir que un deseo es una influencia potencial para una intención. Un personaje puede tener múltiples deseos, pero solo aquel que sea más fuerte ó jerárquicamente más importante será considerado como una intención. La información obtenida del ambiente se representa como *recomendaciones cualitativas* para que una acción se realice. Dichas recomendaciones son conocidas como *preferencias motoras*.

5.2. Detalles de funcionamiento

El sistema reactivo es responsable de ejecutar acciones primarias (incluyendo acciones de sensado), pero este debe trabajar con el Generador de Intenciones y el Mecanismo de Selección de Acción para generar comportamientos inducidos. Con el objetivo de caracterizar conceptos difusos (deseos e intenciones) de forma que no se obtengan comportamientos 100 % deterministas, es recomendable que el generador de intenciones esté implementado con técnicas no clásicas para la toma de decisiones.

6. Modelado cognitivo de personajes

Se seleccionó un mundo submarino con el objetivo de analizar comportamientos elaborados relacionados con el paradigma de *vida artificial*. La implementación consiste de presas (peces) y un depredador (tiburón), plantas y diferentes obstáculos para probar las capacidades cognitivas inherentes a un pez. Este mundo fue implementado con la ayuda del motor de juegos ShiVa3D.

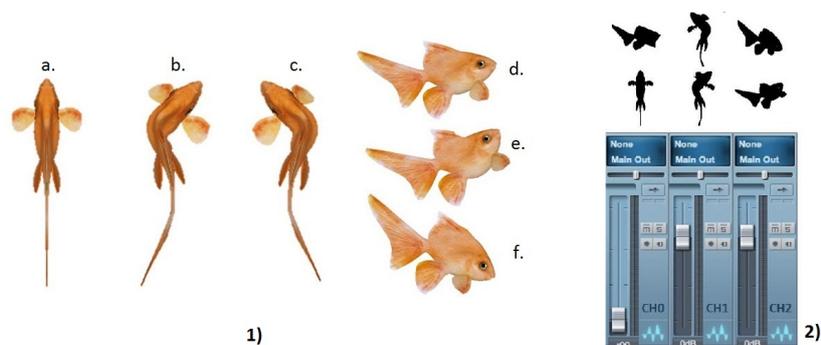


Fig. 3. 1) Movimientos atómicos (idle, derecha, izquierda, frenar, arriba, abajo). 2) Esquema propuesto para programar un administrador de animaciones efectivo.

En base al enfoque de H. Barthel en [2], esta implementación enfatiza la combinación de movimientos atómicos para generar secuencias de movimientos realistas. Se almacenan 6 animaciones simples (Figura 3, inciso 1) dentro de

una base de datos de movimientos donde más de uno puede ser ejecutado simultáneamente para generar movimientos compuestos.

Se define un enfoque factible que obtiene una combinación efectiva de animaciones. La idea es tener dos tipos de acciones: *exclusivas* y *no-exclusivas* (Figura 3, inciso 2). En este caso las animaciones mutuamente exclusivas no pueden reproducirse simultáneamente, por lo que se reproducen en el mismo canal de animación.

El sistema de percepción o sentido propuesto considera cuatro modelos:

1. Evadir colisión con el terreno.
2. Evadir colisión con obstáculos fijos.
3. Evadir colisión con obstáculos dinámicos.
4. Contacto físico.

La capacidad para *evadir de colisiones con el terreno* fue diseñada para ser activada sólo cuando la entidad se está moviendo. Debido a las características intrínsecas del terreno submarino, la implementación utiliza un método que traza rayos. Este método toma como referencia las coordenadas inferiores del *bounding-box* correspondiente al modelo 3D del personaje. Cuando el personaje avanza los rayos son trazados con respecto a las coordenadas delanteras y cuando retrocede los rayos son trazados con respecto a las coordenadas traseras.

La *evasión de colisión con obstáculos fijos* es utilizada cuando no se tiene una idea precisa de la forma, altura y posición de los mismos. Formas difíciles de reconocer son difíciles de procesar. Un pez real utiliza *puntos de referencia* para generar mapas mentales de relaciones geométricas [3]. Así que el cálculo del campo potencial atractivo define un objetivo en el espacio 2D definido por los ejes X y Z. Por su parte, el cálculo del campo potencial repulsivo es una opción viable para evadir colisiones con obstáculos fijos conocidos y que tienen formas demasiado irregulares (árboles, ramas, etc.).

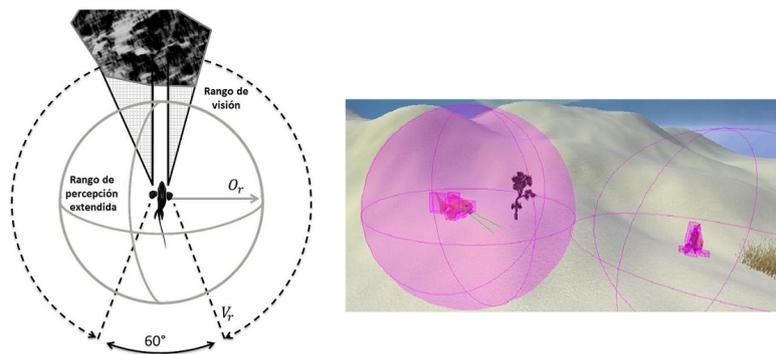


Fig. 4. Modelo de sentido que intenta simular habilidades especiales de percepción en un pez; rango de visión y rango de percepción extendida.

No obstante, la mayoría de los peces poseen órganos de percepción altamente desarrollados, como los *quimiorreceptores* o los receptores que forman el sistema de *línea lateral*, mediante los cuales detectan comida, suaves corrientes y vibraciones. De aquí la necesidad de modelar capacidades sensoriales similares. Capacidades como estas son útiles para *evadir la colisión con obstáculos dinámicos* e incluso para mantener el seguimiento de otros objetos (comida, peces, etc.). En la Figura 4 se muestra la propuesta para esta implementación, donde el rango de percepción extendida es un sensor esférico que detecta sensores similares.

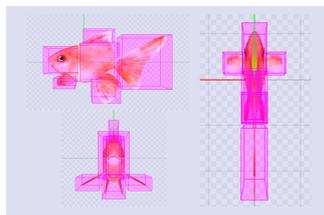
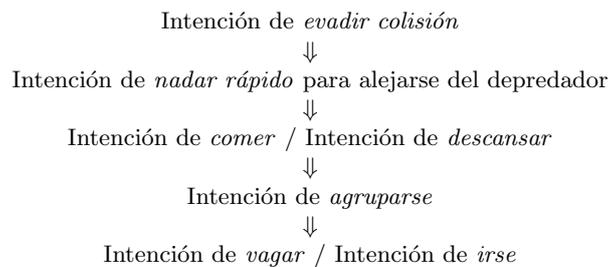


Fig. 5. Ragdoll del pez para simular la percepción del contacto físico.

Por consiguiente es necesario simular el contacto físico con el objetivo de definir comportamientos más complejos o incluso reacciones simples. En este caso se utiliza un *ragdoll* (Figura 5) para poder simular la captura de comida y de peces, y para simular un comportamiento grupal libre de colisiones.

La jerarquía de intenciones considerada en la literatura [20] proporciona los fundamentos teóricos para proponer el siguiente esquema de esta implementación:



Sin embargo, existen otras variables relacionadas con estas intenciones consideradas como deseos o *influencias potenciales de intención* tales como: el *hambre*, la *fatiga* o la *amenaza de supervivencia*, los cuales cambian de manera impredecible con respecto al tiempo. Por lo tanto el generador de intenciones está implementado con una técnica no clásica como lo es los Mapas Cognitivos Difusos (FCM). El FCM para modelar el generador de intenciones del pez es tomado del trabajo de Julie A. Dickerson y Bart Kosko en [6], el cual es mostrado en la Fig. 6. La red causal del Mapa Cognitivo Difuso está formado por las reglas o aristas

que conectan a los nodos o conceptos causales dentro de una matriz de conexión. Esta matriz puede contener valores en $\{-1, 0, 1\}$.

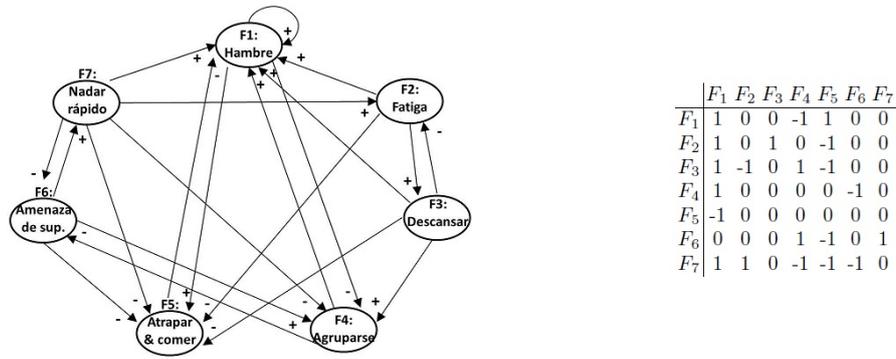


Fig. 6. Mapa Cognitivo Difuso trivalente del generador de intenciones para un pez.

El FCM para modelar el generador de intenciones de un tiburón, está basado en el FCM propuesto en [6]. Sin embargo, cuando se dispara algún estado que hace referencia a los conceptos *buscar comida*, *cazar pez* o *atrapar & comer*, el flujo causal se estanca o se cicla sobre estos tres nodos. En base a los fundamentos teóricos, se modificó este modelo para obtener puntos de atracción más equilibrados sobre el FCM del tiburón. La Fig. 7 muestra el FCM propuesto y la matriz que representa la red causal del FCM de un tiburón.

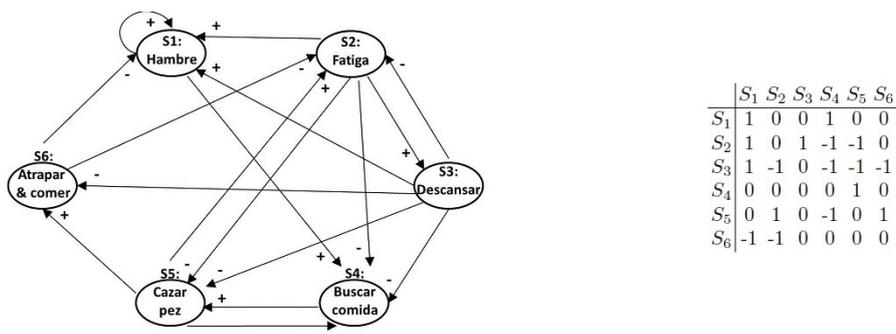


Fig. 7. Mapa Cognitivo Difuso trivalente propuesto para un tiburón.

A diferencia del uso que se le da a los FCM en [6], donde se utilizan *FCM's aumentados*, esta implementación propone utilizar *FCM's simples*, es decir, mapas cognitivos que no estén vinculados en una sola red causal de forma explícita.

Esto es posible gracias al soporte de *percepción-acción* que brinda la arquitectura MoCAMG. Además de que esto es congruente con el objetivo de alcanzar la autonomía. De lo contrario se estaría dotando a los personajes con la capacidad de saber explícitamente el estado mental de otro personaje. En general, el sistema de comportamiento está implementado como muestra el Algoritmo 1. Este procedimiento lo ejecuta tanto un pez como un tiburón, haciendo diferencias relativas dentro de cada uno de los métodos implementados. Este algoritmo sólo mencionan aquellos procesos vitales para el funcionamiento.

Algorithm 1 Comportamiento Artificial

```

1: Inicializar variables globales
2: Cargar animaciones
3: Cargar Mapa Cognitivo Difuso
4: Cargar campo potencial repulsivo
5: Inicializar y activar sensores ragdoll
6: Inicializar sensor de percepción extendida
7: Obtener datos del Bounding-Box del modelo 3D
8: Obtener datos de rotación actual (ángulo en Y)
9: Inicializar variable Fatiga aleatoriamente
10: Inicializar variable Momento con el tiempo (frame/s)
11: for Cada frame do
12:   if Locomoción activa then
13:     Activar percepción extendida
14:     Actualizar información sensorial
15:     Evadir colisión con obstáculos
16:     Evadir colisión con terreno
17:     Actualizar movimiento                                ▷ frenado, giro, desplazamiento
18:     Dirigirse hacia meta
19:   else
20:     Activar flotado
21:     Desactivar percepción extendida
22:     Decrementar variable Fatiga                        ▷ factor alto de recuperación
23:   end if
24:   Generador de Intenciones                               ▷ arbitrar comportamientos
25: end for

```

7. Patrones de comportamiento obtenidos

Como resultado se observa cómo es que los Mapas Cognitivos Difusos de un pez y un tiburón se comportan. Se trató de evaluar de manera no formal el comportamiento de cada rol dentro del mundo virtual submarino implementado. Para un pez los deseos que evocan un comportamiento particular son *hambre*, *fatiga* y *amenaza de supervivencia*. Si un pez tiene *hambre* se dispara:

$$C_1 = [1 0 0 0 0 0] \text{ F1:hambre}$$

J.Carlos Conde-Ramírez, Abraham Sánchez-López

- $C_2 = [1\ 0\ 0\ 0\ 1\ 0\ 0]$ F1:hambre, F5:atrapar & comer
- $C_3 = [0\ 0\ 0\ 0\ 1\ 0\ 0]$ F5:atrapar & comer
- $C_4 = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ vagar

Si un pez entra en un estado de *fatiga* el generador de intenciones dispara:

$$C_1 = [0\ 1\ 0\ 0\ 0\ 0\ 0] \text{ F2:fatiga}$$

- $C_2 = [1\ 0\ 1\ 0\ 0\ 0\ 0]$ F1:hambre, F3:descansar
- $C_3 = [1\ 0\ 0\ 0\ 0\ 0\ 0]$ F1:hambre
- $C_4 = [1\ 0\ 0\ 0\ 1\ 0\ 0]$ F1:hambre, F5:atrapar&comer
- $C_5 = [0\ 0\ 0\ 0\ 1\ 0\ 0]$ F5:atrapar & comer
- $C_6 = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ vagar

Si un pez entra en un estado de alerta por *amenaza de supervivencia* el generador de intenciones dispara:

$$C_1 = [0\ 0\ 0\ 0\ 0\ 1\ 0] \text{ F6:amenaza de supervivencia}$$

- $C_2 = [0\ 0\ 0\ 1\ 0\ 0\ 1]$ F4:agruparse, F7:nadar rápido
- $C_3 = [1\ 1\ 0\ 0\ 0\ 0\ 0]$ F1:hambre, F2:fatiga
- $C_4 = [1\ 0\ 1\ 0\ 0\ 0\ 0]$ F1:hambre, F3:descansar
- $C_5 = [1\ 0\ 0\ 0\ 0\ 0\ 0]$ F1:hambre
- $C_6 = [1\ 0\ 0\ 0\ 1\ 0\ 0]$ F1:hambre, F5:atrapar & comer
- $C_7 = [0\ 0\ 0\ 0\ 1\ 0\ 0]$ F5:atrapar & comer
- $C_8 = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ vagar

Tal como lo menciona la teoría, es posible observar que existen ciclos limitados que muestran patrones ocultos de comportamiento. Incluso algunos ciclos están contenidos dentro de otros al disparar un estado diferente. A simple vista se puede asegurar que estos patrones son congruentes con el comportamiento de un pez. Es importante decir que la intención de *vagar* esta representado por un estado vacío (sólo ceros) y este determina el punto de paro del ciclo.

Para el caso de un tiburón los deseos que evocan un comportamiento particular solamente son *hambre* y *fatiga*. Si un tiburón tiene *hambre* el generador de intenciones dispara:

$$C_1 = [1\ 0\ 0\ 0\ 0\ 0] \text{ S1:hambre}$$

- $C_2 = [1\ 0\ 0\ 1\ 0\ 0]$ S1:hambre, S4:buscar comida
- $C_3 = [1\ 0\ 0\ 1\ 1\ 0]$ S1:hambre, S4:buscar comida, S5:casar pez
- $C_4 = [1\ 1\ 0\ 0\ 1\ 1]$ S1:hambre, S2:fatiga, S5:casar pez, S6:atrapar & comer
- $C_5 = [1\ 0\ 1\ 0\ 0\ 1]$ S1:hambre, S3:descansar, S6:atrapar & comer
- $C_6 = [1\ 0\ 0\ 0\ 0\ 0]$ S1:hambre

Si el tiburón entra en estado de *fatiga* el generador de intenciones dispara:

$$C_1 = [0\ 1\ 0\ 0\ 0\ 0] \text{ S2:fatiga}$$

- $C_2 = [1 0 1 0 0 0]$ S1:hambre, S3:descansar
- $C_3 = [1 0 0 0 0 0]$ S1:hambre
- $C_4 = [1 0 0 1 0 0]$ S1:hambre, S4:buscar comida
- $C_5 = [1 0 0 1 1 0]$ S1:hambre, S4:buscar comida, S5:casar pez
- $C_6 = [1 1 0 0 1 1]$ S1:hambre, S2:fatiga, S5:casar pez, S6:atrapar & comer
- $C_7 = [1 0 1 0 0 1]$ S1:hambre, S3:descansar, S6:atrapar & comer
- $C_8 = [1 0 0 0 0 0]$ S1:hambre

A diferencia de un pez este ciclo no está limitado por un estado vacío. De hecho el deseo *hambre* siempre está presente. Por lo tanto se puede inferir que para un tiburón el estado [100100] (S1:hambre, S4:buscar comida) es equivalente a la intención de vagar de un pez. Además cada patrón de comportamiento lo describe un ciclo limitado por el estado simple [100100] (S1:hambre). Esto es comprensible hasta cierto punto, ya que algo que caracteriza a un tiburón es mantener siempre un comportamiento agresivo. Nótese que la *jerarquía de intenciones* definida previamente es un factor clave para generar un mecanismo de selección de acción efectivo. Principalmente cuando se genera un estado con 2 o más intenciones simultaneas.

8. Conclusiones y trabajo futuro

Apoyados en la arquitectura MoCAMG, una representación del conocimiento sencilla pero útil (FCM's simples), y las herramientas de sensado y física que proporciona un motor de juegos se logró comprobar que [a menudo] *es posible descomponer patrones complejos de comportamiento en unidades más pequeñas, algunas de las cuales son inmediatamente comparables con reflejos. Y que frecuentemente el comportamiento es organizado de una forma jerárquica; así que los sistemas de control superior deben competir por el control de los reflejos más que los reflejos competir por el control de los músculos.* Tal como lo dice A. Manning en [16].

Cabe mencionar que hay pocas arquitecturas cognitivas enfocadas en videojuegos, la mayoría están enfocadas en procesos detallados relacionados con la psicología humana. Sin embargo, la arquitectura y la metodología utilizadas probaron ser útiles para este enfoque. Además, el ambiente virtual implementado es una base importante para probar procesos cognitivos más complejos.

Cómo trabajo futuro, se planea que la situación de una entidad cognitiva (estado interno y externo) y una respuesta apropiada (acción) sea almacenada como *instancia* en la memoria de largo plazo, procesada por el módulo nombrado Procedimiento-IBL (Figura 1). De forma similar a como lo realiza la IBLTool de V. Dutt y C. González en [7]. La correcta descripción de estas instancias tiene el objetivo de lograr un mejor rendimiento y respuestas más ponderadas por parte del mecanismo de selección de acción. En pocas palabras, evolucionar hacia un sistema de razonamiento.

Referencias

1. Anderson, J.: The Architecture of Cognition. Cognitive science series, Lawrence Erlbaum Associates (1996)
2. Barthel, H., Dannenmann, P., Hagen, H.: Towards a General Framework for Animating Cognitive Characters. In: Proceedings of the 3rd IASTED International Conference on Visualization, Imaging, and Image Processing. VIIP'03, Benalmádena, Spain (2003)
3. Chung, S.: Appropriate maze methodology to study learning in fish. University of Toronto Journal of Undergraduate Life Sciences 2(1) (2009)
4. Conde Ramírez, J.C., Sánchez López, A., Sánchez Flores, A.: An architecture for cognitive modeling to support real-time adaptation and motivational responses in video games. In: MICAI (1). Lecture Notes in Computer Science, vol. 8265, pp. 144–156. Springer, Mexico City, Mexico (2013)
5. Dannenmann, P., Barthel, H., Hagen, H.: Multi level Control of Cognitive Characters in Virtual Environments. In: Proceedings of the 14th IEEE Visualization 2003 (VIS'03). pp. 92–. VIS '03, IEEE Computer Society, Washington, DC, USA (2003)
6. Dickerson, J.A., Kosko, B.: Virtual Worlds as Fuzzy Cognitive Maps. Presence 3(2), 173–189 (1994)
7. Dutt, V., Gonzalez, C.: Making Instance-based Learning Theory usable and understandable: The Instance-based Learning Tool. Comput. Hum. Behavior 28(4), 1227–1240 (Jul 2012)
8. Funge, J.: Cognitive modeling for games and animation. Commun. ACM 43(7), 40–48 (Jul 2000)
9. Funge, J., Tu, X., Terzopoulos, D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 29–38. SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., New York, USA (1999)
10. Funge, J.D.: Making them behave: cognitive models for computer animation. Ph.D. thesis, University of Toronto, Toronto, Ont., Canada (1998)
11. Giang, T., Mooney, R., Peters, C.: Real-time character animation techniques. Tech. rep., Image Synthesis Group Trinity College Dublin (2000)
12. Hudlicka, E.: Affective game engines: motivation and requirements. In: Proceedings of the 4th International Conference on Foundations of Digital Games. pp. 299–306. FDG '09, ACM, New York, NY, USA (2009)
13. Kim, M.: The artificial emotion model of game character through analysis of cognitive situation. In: Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology. pp. 489–493. ICCIT '09, IEEE Computer Society, Washington, DC, USA (2009)
14. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies (24), 65–75 (1986)
15. Magnenat-Thalmann, N., Thalmann, D.: Virtual humans: thirty years of research, what next? The Visual Computer 21(12), 997–1015 (2005)
16. Manning, A., Dawkins, M.: An Introduction to Animal Behaviour. Cambridge University Press (1998)
17. McFarland, D.: Animal behaviour: psychobiology, ethology, and evolution. Longman Scientific and Technical (1993)
18. Ritter, F.E.: Two cognitive modeling frontiers Emotions and usability. Transactions of the Japanese Society for Artificial Intelligence 24(2), 241–249 (2009)

19. Sun, R.: Theoretical status of computational cognitive modeling. *Cogn. Syst. Res.* 10(2), 124–140 (Jun 2009)
20. Tu, X.: *Artificial animals for computer animation: biomechanics, locomotion, perception, and behavior*. Springer-Verlag, Berlin, Heidelberg (1999)
21. Yan, W., Kalay, Y.: Geometric, cognitive and behavioral modeling of environmental users. In: GERO, J. (ed.) *Design Computing and Cognition '06*, pp. 61–79. Springer Netherlands (2006)