

## Parte II

# Estado del Conocimiento



# Capítulo 2

---

## Conceptos Básicos

Aunque la Vida Artificial es un campo joven en las ciencias de la computación, la presencia cada vez mayor de películas para el cine y la televisión con abundancia de efectos especiales, en el que aparecen seres y escenarios generados por ordenador, así como la proliferación de los video juegos, nos tienen cada vez más familiarizado con esta tecnología, sus avances, y su terminología; aunque sea de manera general.

Conceptos como Realidad Virtual, Animaciones Generadas por Ordenador, Mundos y Seres Virtuales, ya no son del todo desconocidos para muchos, no importando su edad u ocupación. Sin embargo, para entender apropiadamente el verdadero significado detrás de estos conceptos, es necesario estar familiarizado con sus técnicas, funciones, y objetivos.

En ese sentido se orienta el presente capítulo, en el que se hace una introducción formal a los principales conceptos manejados en este trabajo.

### 2.1 Agentes Virtuales Autónomos

Un Agente Virtual Autónomo (AVA) es un programa de software con las siguientes características:

- Su hábitat natural (entorno de ejecución) es un mundo gráfico simulado 3D.

- Posee una representación gráfica (cuerpo modelado geométricamente) dentro del mundo que habita, y es capaz de percibir, adaptarse, y reaccionar a su entorno; exhibiendo gráficamente un comportamiento como el de un ser vivo.
- Es una entidad independiente de software que, aunque solo existe y funciona en un entorno específico, es “conciente” de los cambios que se producen a su alrededor y es capaz de responder a ellos de manera autónoma, o sea, sin necesidad de instrucciones o control externo.

Es común confundir el concepto de AVA con el de Avatar. Un Avatar es un Agente Virtual cuyo comportamiento es explícitamente controlado por un animador externo, mediante comandos de control; por lo tanto, no es autónomo.

Algunos autores se refieren a los AVAs como Agentes Virtuales Inteligentes, o simplemente como Agentes Inteligentes. Por otro lado, según a quién o a qué pretendan simular, suele referirse a ellos como Criaturas o Actores Virtuales (de manera genérica o haciendo referencia a un animal o humanoide), o Humanos Virtuales (refiriéndose específicamente a personas), y omitiendo la característica de “autónomos” cuando queda implícito en el contexto en que aparecen, o cuando el AVA recibe alguna clase de ayuda externa para su animación, no siendo completamente autónomo.

Claro está, en el desarrollo de AVAs el mayor reto es construir humanos virtuales autónomos inteligentes. Se hace hincapié en “inteligente”, ya que su conducta debe ir acorde al objeto a imitar (el ser humano); es decir, un humano virtual debe ser capaz de actuar libre y emocionalmente, ser conciente e impredecible, y ser coherente consigo mismo y su entorno.

La creación de humanos virtuales autónomos (HVAs) que resulten creíbles en todos sus aspectos, es una tarea ardua que ha requerido (y requiere) la colaboración de varios dominios del conocimiento, y de diversidad de investigadores dedicando sus esfuerzos a problemas específicos asociados al modelado, animación, y comportamiento de los HVAs; y aun así, hay una gran cantidad de trabajo por hacer (ver [NEDE98]).

En las siguientes subsecciones se hace un repaso por las técnicas empleadas en los diversos aspectos involucrados en la creación de AVAs, particularmente el de los humanos virtuales.

## 2.1.1 Modelado

El modelado de un agente virtual autónomo determina la apariencia visual final que tendrá el actor virtual. Dicha apariencia persigue la credibilidad fisonómica del agente, sin dejar de lado las consideraciones del modelo que facilitan su movilidad para posteriores efectos de animación; es decir, un buen modelado logra proporcionarle al personaje virtual una apariencia lo mas creible posible, a las vez que le otorga al animador las facilidades para su animación.

Los modelos 3D desarrollados hasta ahora pueden clasificarse en cuatro categorías: Modelo de Segmentos, Modelo de Superficie, Modelo de Volumen, y Modelo Multi-Capas.

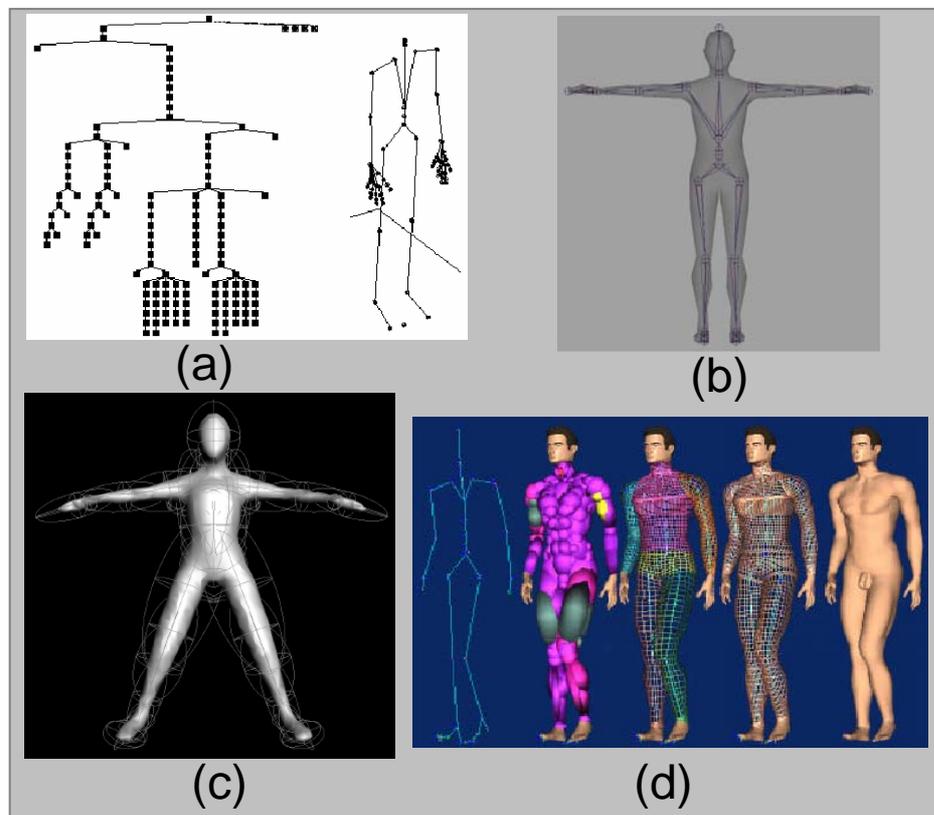


Figura 2.1. Estilos de Modelado de Agentes Virtuales: a) Segmentos [BOUL93] © EPLF Computer Graphics Lab-LIG, b) Superficie [LEWI00] ©Ohio State University, c) Volumen [LEWI00a] ©Ohio State University, y d) Multi-Capas [SHEN96] © EPLF Computer Graphics Lab-LIG.

El Modelo de Segmentos consiste de un conjunto jerárquico de segmentos conectados por uniones. Esos modelos son llamados Cuerpos Articulados y pueden ser más o menos complejos, dependiendo de la cantidad de segmentos y uniones que posea. Cada unión puede tener hasta 3 grados de libertad, asociados a la rotación alrededor de cualquiera de los 3

ejes. La principal ventaja de este modelo es que la especificación del movimiento es muy sencilla; solo es necesario proporcionar una matriz 3D para cada unión. La desventaja es que produce simulaciones poco realistas. La Figura 2.1(a) muestra un ejemplo de este modelo.

El Modelo de Superficie esta constituido por un esqueleto rodeado por una superficie compuesta de parches planos o curvos, simulando la piel. Este modelo es básicamente un extensión del de segmentos, en el que se ha añadido una capa adicional (un poliedro representando la superficie) que recubre el esqueleto (modelo de segmentos) del actor virtual, tal como se muestra en la Figura 2.1(b); y en la que también se puede incluir información antropométrica.

Los Modelos de Volumen aproximan la estructura y la forma de un cuerpo mediante una colección de primitivas de volumen, tales como el cilindro, la elipsoide, la esfera, etc. Unos pocos cilindros o elipsoides pueden capturar la superficie y las propiedades de muchas partes del cuerpo; sin embargo, no pueden producir mejores resultados que los modelos de superficie, y su mecanismo de control se vuelve inadecuado para grandes conjuntos de estas primitivas. El uso de metabolas, que consideran el volumen como una función potencial, produce mejores y mas naturales resultados. La Figura 2.1(c) presenta un modelo formado por 35 metabolas; los segmentos del cuerpo (huesos entre uniones) son representado por, de una a tres metabolas, y colocadas en forma jerárquica, partiendo de la pelvis como raíz.

El Modelo Multi-Capas es mucho más reciente que los anteriores, y ha sido desarrollado especialmente para figuras humanas. En este enfoque, un esqueleto es usado para soportar capas intermedias que dan volumen al cuerpo, simulando huesos, músculos, grasa, etc; y finalmente, recubriendo todo por una capa de piel. En algunos casos, la ropa que viste al actor virtual puede ser considerada también una capa. Cada capa es un modelo simulado conceptual, el cual hace un mapeo de parámetros de entrada de alto nivel, a salidas de bajo nivel. El animador especifica varias restricciones sobre la relación entre las capas, y puede controlar el movimiento global desde comandos de alto nivel. La Figura 2.1(d) ilustra los estados en la construcción de un modelo de cuerpo humano.

En adición a las técnicas de modelado descritas, los animadores buscan que los personajes virtuales incluyan un modelado basado en física ([SIMS94, TU96, GRZE98]), y reconstrucción de deformaciones de forma libre ([WILH94, NEDE98a, AUBE01]), para lograr un mayor realismo en su apariencia y movilidad. Además, prestan especial interés en el modelado de partes específicas del cuerpo humano, por sus características particulares,

como lo son: la cara (por las expresiones faciales), las manos (por su tipo de comportamiento), el cabello (por el reto que implica animarlo), y la ropa del actor ([KALR98, MAGN96]).

## 2.1.2 Animación

La animación, en relación a los Agentes Virtuales Autónomos, se refiere a la capacidad de los agentes de desenvolverse por si mismos dentro del ambiente en el que se encuentran inmersos, actuando y reaccionando a su entorno. El caso de los humanos virtuales autónomos (HVAs) es más complicado, ya que se pretende que exhiban conductas más complejas, basadas en estados emocionales y razonamientos; tal como se esperaría de un humano real.

En la actualidad, la tendencia en la animación del comportamiento va dirigida hacia la creación de simulaciones de ambientes virtuales inteligentes donde los actores sintéticos exhiben conductas, acorde a la naturaleza del ser que emulan, de manera autónoma; por ejemplo, en el caso de un humano virtual, éste debería ser capaz, por si mismo, de tomar decisiones como resultado de un proceso de razonamiento y en consonancia con su situación dentro del entorno. Para llegar a este punto, la animación del comportamiento ha evolucionado a partir de las técnicas de animación por ordenador, que a su vez comparten la esencia de la animación tradicional.

La animación tradicional es definida como la técnica con la cual se crea la ilusión de movimiento; y esto se logra generando y presentándole al ojo humano una secuencia de imágenes a una velocidad adecuada. En el caso de la animación por ordenador, la técnica es la misma, la diferencia radica en la forma en que cada imagen es generada.

Tal como sucede en la animación tradicional, en una secuencia animada por ordenador las imágenes son producidas según las directrices de un animador; el cual ha venido contando con modelos computacionales cada vez más elaborados que facilitan enormemente su labor.

La Figura 2.2, presentada originalmente por Terzopoulos [TERZ99], muestra una pirámide ilustrando la jerarquía en el modelado de la animación de gráficos por ordenador, la cual esta coronada por modelos de comportamiento y cognitivos que sellan el compromiso entre los gráficos por ordenador y la inteligencia artificial, en el campo de la vida artificial.

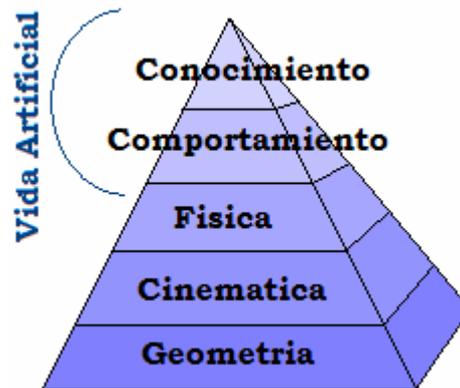


Figura 2.2. Jerarquía en el modelado de la animación de gráficos por ordenador y vida artificial [TERZ99]

Los primeros modelos usados por los animadores fueron los modelos geométricos, a partir de los cuales se producen las imágenes. En conjunto con estos modelos, métodos como keyframing [BURT71] facilitaban la labor del animador, donde el ordenador es usado para interpolar entre dos estados de la misma escena (generando la secuencia de imágenes intermedia), y completar de ese modo la secuencia de animación ([LASS87]). Una forma de producir mejores imágenes es interpolar parámetros del modelo en vez de los objetos en sí. Con esta técnica, llamada Keyframing Paramétrico [PARK74], el animador genera una imagen estableciendo los parámetros de los modelos que componen la escena; entonces, dichos parámetros son interpolados a un nuevo conjunto de parámetros establecidos por el animador, de manera que el proceso de interpolación vaya generando las consecutivas imágenes que compondrán la animación.

Los modelos geométricos proporcionan una mayor versatilidad para generar diversidad de escenas. Los objetos modelados presentan estructuras cada vez más elaboradas, con miembros móviles y partes articuladas, lo que dificulta su animación utilizando los métodos tradicionales. Una técnica heredada del campo de la robótica, conocida como Cinemática Inversa, aborda este problema. En un sistema típico basado en cinemática inversa, el animador especifica posiciones discretas y movimientos en el espacio para las extremidades del objeto, entonces el sistema calcula los ángulos de las uniones y las orientaciones necesarias para las otras partes del cuerpo, y colocar las partes especificadas en las posiciones deseadas a través de los movimientos requeridos, considerando las restricciones intrínsecas de movilidad del objeto modelado. Este enfoque ha funcionado bien para uniones simples; sin embargo, las soluciones para posiciones particulares se convierten en complicadas y numerosas cuando el número de uniones aumenta. Badler y colegas [BADL87] presentaron un algoritmo iterativo para resolver múltiples restricciones usando cinemática inversa.

Una de las más populares técnicas de animación es la Captura de Movimiento, la cual involucra medir la posición y orientación de un objeto en el espacio físico, y grabar esa información en un formato que el ordenador pueda usar. El movimiento es capturado mediante sensores magnéticos o basados en visión, colocados en el animador, y grabando las posiciones de las partes de su cuerpo y los ángulos de las uniones, mientras éste ejecuta la acción deseada. Con dispositivos de captura de movimiento en tiempo real, los datos recabados pueden ser usados interactivamente con un mínimo de retardo ([BOUL97, BROG98]). En sistemas de captura de movimiento que no trabajan en tiempo real, los datos pueden requerir un pos-procesamiento antes de ser usados. Si bien, la captura de movimiento esta creciendo en popularidad gracias a su forma relativamente fácil de grabar muchas acciones humanas, presenta problemas que lo alejan de ser la solución ideal para todas las aplicaciones. Al igual que con keyframing, la secuencias de captura de movimiento grabadas deben ser combinadas con habilidad para crear movimientos realistas que cambien en respuesta a las acciones del usuario; además, las discrepancias entre las formas o dimensiones del sujeto del que se captura el movimiento y el personaje gráfico, también puede producir problemas.

Siguiendo otro peldaño en la pirámide de la Figura 2.2, encontramos los modelos que incorporan el uso de leyes físicas, manejando conceptos como masa, gravedad, inercia, entre otros. Los modelos físicos permiten la automatización de la animación de los objetos pasivos, tales como cadenas cayendo, u objetos chocando. Para objetos activos (de anatomía autoanimada), los investigadores enfocan sus esfuerzos a la construcción de Modelos Biomecánicos. Hasta ahora ha sido posible utilizar modelos biomecánicos simplificados para automatizar el proceso de locomoción en una variedad de criaturas virtuales ([TERZ87, TERZ94, ZAJA90]). Trabajos como los de Radek Grzeszczuk y colegas [GRZE98] logran reducir la fuerte demanda computacional asociada a la solución de las ecuaciones matemáticas implícitas en los modelos biomecánicos, utilizando redes neuronales. Todos estos métodos caen en lo que se conoce como Animación Procedimental, donde el movimiento es descrito algorítmicamente. La ventaja de usar leyes (físicas) algorítmicas para animación, es que logran una animación exacta (según la calidad del modelado de las mismas); y ser considerada como simulación pura.

En 1987, Craig Reynolds publicó un estudio sobre trayectorias en grupos: bandadas de aves, manadas de animales, y cardúmenes de peces [REYN87]. En él, cada boid (como los llama el autor) decide su trayectoria sin intervención del animador, mediante un modelo de comportamiento distribuido para simular las bandadas de aves, las manadas de animales, y los cardúmenes de peces. La simulación de bandadas esta basada en un

sistema de partículas, donde los boids son las partículas. Una bandada es el resultado de la interacción de los comportamientos de boids individuales. El trabajo de Reynolds muestra cómo complejas animaciones pueden surgir con un mínimo de esfuerzo por parte del animador, a través de modelos que controlan cómo los actores sintéticos deberían comportarse. Esos modelos toman la forma de reglas de comportamiento que gobiernan la interacción de múltiples agentes autónomos capaces de locomoción y percepción dentro del mundo que habitan. El Modelado del Comportamiento marca el nacimiento de la autonomía de los agentes virtuales. Esta clase de animación es casi imposible de lograr usando un enfoque tradicional (como keyframing o procedimental). Sin embargo, este nivel de comportamiento solo incluye evasión de obstáculos y comportamiento grupal; conductas comunes a muchas criaturas y situaciones ([HAUM88, REYN92, NOSE93, ZHAN97, REYN99, THAL99]).

Aunque Reynolds tuvo éxito al lograr producir comportamiento realista en actores virtuales, su modelo de comportamiento está limitado por su simplicidad, ya que solo está pensado para comportamiento en grupo. Trabajos como [TERZ94, BLUM95] extendieron este enfoque para intentar algunas conductas más complejas, desarrollando un modelo de comportamiento reflexivo instintivo, que incluye evasión de obstáculos mediante el esquema percepción-acción; además, a través de un efectivo mecanismo de selección-acción, el comportamiento instintivo es combinado con comportamiento emocional cuya activación depende de variables que representan el estado mental del actor virtual. El principal problema presentado aquí es modelar satisfactoriamente el proceso cognitivo de los personajes; es decir, la forma como ellos manejan su conocimiento para conducirse libre y emocionalmente por el entorno. Los Modelos Cognitivos son el siguiente paso lógico en la jerarquía de modelos que han sido usados en la animación por ordenador. Su construcción es el principal tópico de investigación en el campo de la Inteligencia Artificial, cuya área de aplicación original fue la robótica, y donde su teoría está siendo adaptada a los problemas de modelado cognitivo para la animación por ordenador. Es usual que estos modelos se nutran de la etología para lograr mayor credibilidad en el comportamiento de los actores virtuales. Los aportes más interesantes y significativos en la evolución de los modelos cognitivos los encontramos en aquellos dirigidos a humanos virtuales autónomos, los cuales intentan dotarles de personalidad mediante la representación de conceptos como comportamiento, inteligencia, autonomía, adaptación, percepción, memoria, libertad, sentimientos, conciencia, presencia, y ser impredecibles ([THAL94]). La complejidad de estos modelos ha propiciado el desarrollo de arquitecturas de animación del comportamiento [DONN94, BECH98, MILL99, THAL01, THAL04], que facilitan el desarrollo y mantenimiento de las implementaciones hechas de los módulos de conducta

para agentes virtuales autónomos. De esta forma el animador es el responsable del comportamiento de los actores virtuales, del mismo modo que un director de teatro lo es de la actuación de sus actores; el director solo marca las directrices de la actuación y el actor las ejecuta aportando su personalidad.

Es así como la Vida Artificial ha evolucionado en el campo de la animación del comportamiento, desde la planificación de caminos hasta complejas interacciones emocionales entre actores virtuales [THAL99, THAL00, MUSS01].

## 2.1.3 Caminos y Colisiones

Las secciones anteriores se han dedicado a hacer una revisión por todos los aspectos involucrados en la animación de agentes virtuales autónomos, y en las que se han podido apreciar la complejidad y el gran trabajo involucrado en este tópico. Sin embargo, algunas técnicas han podido pasar desapercibidas, no siendo por ello menos importantes.

Esta sección pretende hacer una breve mención a dos técnicas que son indispensables para la animación de AVAs, no profundizadas aquí, y que de por sí constituyen un área de investigación y producción científica por sí mismas.

Los Algoritmos de Planificación de Caminos son aquellos que resuelven el problema de encontrar una secuencia de estados de transición desde algún estado inicial (punto de partida) a un estado final (punto destino), o averiguar si dicha secuencia no existe. La solución ‘óptima’ a este problema depende del propósito que se persiga, ya que no siempre es mas conveniente el camino mas corto entre dos puntos. Sin este tipo de algoritmos, a los agentes virtuales se les haría imposible desplazarse por el entorno ([STEN95, BAND98, UNDE01]).

Otra técnica fundamental para la simulación con agentes virtuales autónomos es la de Algoritmos de Detección de Colisiones. Usados para dar mayor realismo a las animaciones, evitan el efecto ‘fantasma’, el cual se produce cuando dos objetos (que aparentan ser sólidos) se atraviesan como si no se produjese contacto entre ellos. El costo computacional de un algoritmo de detección de colisión depende de varios factores (la forma de los objetos, las pruebas y las veces que se aplican), aunque como de costumbre, el equilibrio entre rapidez y robustez dependerán de las exigencias de la aplicación. Para un recuento sobre estas técnicas se recomienda [JIME01].

## 2.2 Objetos Inteligentes

Un Objeto Inteligente es un objeto virtual que es modelado con sus características de interacción; o sea, con todas las partes, movimientos y descripciones del objeto que son importantes cuando ocurre una interacción con un agente. Los Objetos inteligentes proveen los parámetros necesarios para la generación de movimiento, así como sus propiedades intrínsecas (posición, masa y apariencia).

La Figura 2.3 muestra dos objetos inteligentes que pudieran ser encontrados en un parque virtual. Tales objetos son considerados inteligentes porque encapsulan la información necesaria para permitir interactuar con ellos, tal como se aprecia en la parte derecha de la figura, donde un humano virtual juega dando vueltas en la rueda, y otros dos se columpian en el subibaja.

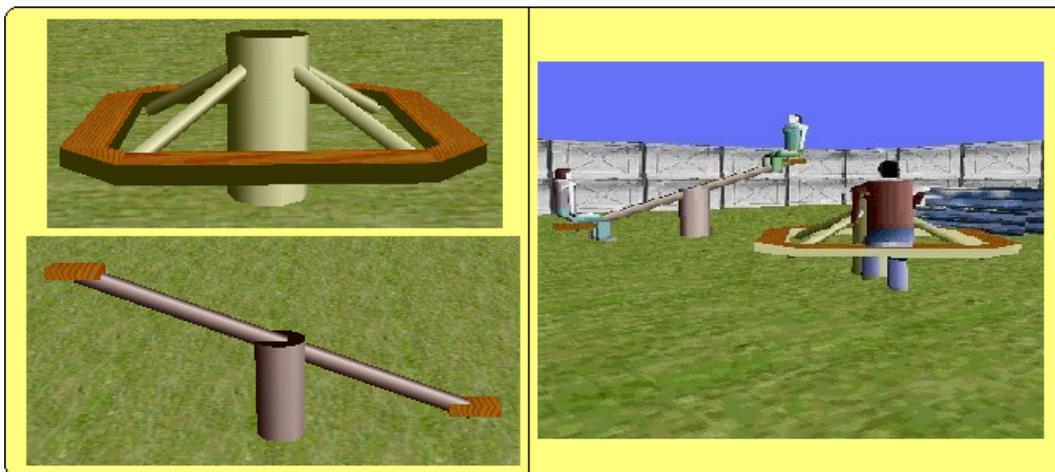


Figura 2.3. Objetos Inteligentes. Izquierda: Rueda (arriba) y Subibaja (abajo) Virtuales. Derecha: Agentes interactuando con los objetos inteligentes

Las aplicaciones con objetos inteligentes proporcionan un número de ventajas superior a otros enfoques (tales como el Razonamiento Específico de Objeto [LEVI96], que solo utiliza las propiedades intrínsecas al objeto), ya que descentraliza el control de la animación, separando la animación de nivel alto de la de nivel bajo, y permitiendo al mismo objeto ser usado en múltiples aplicaciones; además de promover el diseño orientado a objetos, ya que cada objeto encapsula sus propios datos.

Un problema particular de este enfoque consiste en el diseño efectivo de objetos inteligentes. Se pueden encontrar propuestas de marcos de desarrollo para modelar interacciones generales agente-objeto basadas en la información de interacción contenida en el objeto, la cual puede ser

intrínseca al mismo, información de cómo interactuar con él, del comportamiento del objeto, y del comportamiento esperado del agente.

Como ejemplo del diseño de objetos inteligentes comentaremos brevemente dos enfoques distintos.

En el primer trabajo que se discute, presentado por Marcelo Kallmann y Daniel Thalmann [KALL98], los autores proponen un enfoque para modelar interacciones entre humanos y objetos virtuales, en el cual plantean incluir con la descripción del objeto, toda la información necesaria para describir como interactuar con él (partes móviles, instrucciones de funcionalidad, localización de la interacción, etc). Para ello utilizan un programa modelador de objetos inteligentes.

La Figura 2.4 muestra el uso del programa mencionado para modelar tres objetos inteligentes: un escritorio (con tres compartimientos que se abren de manera distinta), un libro (que se puede abrir para ojearlo), y una lámpara (que se puede encender para una iluminación de lectura).

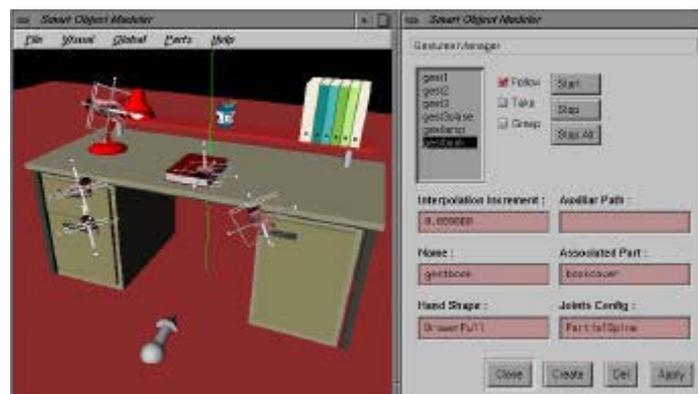


Figura 2.4. Modelador de Objetos Inteligentes [KALL98]  
© Eurographics

La aplicación presenta una interfase gráfica que le permite al usuario, interactivamente, especificar diferentes características para el objeto, generando un archivo de texto, que incluye comandos que detallan las partes móviles, las características de interacción (tales como las propiedades intrínsecas del objeto, información de cómo interactuar con él, el comportamiento del objeto, y el comportamiento esperado del agente), y toda la información deseada sobre el objeto, mediante un lenguaje de guión. Con este diseño para los objetos inteligentes, los autores buscan descentralizar el control de la animación, es decir, encapsulando en el objeto toda la información sobre la interacción objeto-agente, el control principal de animación es liberado del cómputo asociado a dicha animación.

Una continuación de este trabajo se halla en [KALL99].

Un enfoque más reciente para el diseño de objetos inteligentes lo encontramos en el trabajo de Christopher Peters y colegas [PETE03]. Basado en el trabajo de Kallmann y Thalmann (anteriormente explicado), el modelo de Peters y colegas presenta objetos inteligentes que son construidos de manera de hacerlos el centro de la interacción entre agentes, permitiendo simulaciones de múltiples agentes que deben cooperar para interactuar con un simple objeto.

Del mismo modo que en [KALL98], los autores utilizaron un editor de objetos inteligentes para el diseño de los mismos. Pero a diferencia de Kallmann y Thalmann, cuyo programa fue implementado usando la librería Open Inventor<sup>1</sup> como herramienta gráfica, y una capa de interfase de usuario simplificada sobre Motif<sup>2</sup>; Peters y sus colegas crearon un plug-in editor en 3D Studio Max<sup>3</sup>, el cual fue escrito en su lenguaje guión, y que permite construir herramientas personalizadas utilizando los magníficos recursos de 3D Studio Max.

## 2.3 Mundos Virtuales

Los Mundos Virtuales (o Ambientes Virtuales) son representaciones gráficas, generadas por ordenador, de hábitat, bien sea imaginarios o inspirados en escenarios reales.

Aunque dichos mundos pueden ser usados como escaparate para exhibir cualquier tipo de escenario 3D y poder examinar y estudiar sus contenidos; cuando en ellos habitan agentes virtuales autónomos, suele llamárseles “Entornos Virtuales Inteligentes”, ya que en las simulaciones generadas, el Mundo Virtual le ofrece a los AVAs un entorno dinámico (cambiante y con amplias posibilidades de interacción), permitiéndoles exhibir conductas adaptativas a los cambios que experimentan a su alrededor.

El uso de Mundos Virtuales como interfaz de usuario no debería estar limitado a pasear por hermosos escenarios 3D. Un Entorno Virtual estático, vacío de cambios y comportamientos, limita la experiencia de inmersión y, por consiguiente, el interés del usuario; por ejemplo, una ciudad que no muestre signos de “vida” puede rápidamente acabar con la sensación de estar en presencia de un ambiente “real”.

---

<sup>1</sup> Open Inventor es un conjunto de herramientas orientada a objetos para el desarrollo interactivo de aplicaciones gráficas 3D.

<sup>2</sup> Motif es un conjunto de especificaciones para la construcción de interfases de usuario, y se ha convertido en un estándar para las interfaces de usuario gráficas; propuesta por la Fundación de Software Abierto ([www.opengroup.org](http://www.opengroup.org)).

<sup>3</sup> 3D Studio Max es una herramienta para el modelado y animación 3D.

Paralelamente al modelado de estos escenarios virtuales, muchos investigadores han sido capaces de crear plantas ([MECH96], [PRUS96]), animales ([TERZ94], [TERZ96], [TU96], [BLUM95]), personajes ([KALR98]), y cualquier criatura ([SIMS94]) sintética con sus propios comportamientos, otorgándoles a los mundos virtuales mayor dinamismo y credibilidad.

En la actualidad, los Entornos (o Ambientes) Virtuales Inteligentes son empleados en una variedad de áreas, principalmente relacionadas a simulación, entretenimiento, y educación. Ejemplo de esas aplicaciones las encontramos en elaborados ambientes simulados relacionados a espacios urbanos, calles, construcciones interiores, diseño arquitectónico, ingeniería civil, control de tráfico y multitudes.

## 2.3.1 Herramientas de Desarrollo

Como podríamos deducir, el desarrollo de los ambientes virtuales inteligentes no solo incluye el modelado de los escenarios propios del entorno, sino también la correcta integración de los actores inteligentes, y de los objetos inteligentes, en el mundo creado. Además, no existe una única metodología y/o arquitectura para el desarrollo de estos ambientes interactivos 3D, debido a que las características propias del objetivo que se persigue orientan el diseño y las consideraciones finales; por ejemplo, hay casos donde es requerida la exactitud del modelado de las leyes físicas, y otros donde el énfasis es colocado en la apariencia visual.

A continuación se presenta una breve descripción de algunos trabajos orientados en este sentido; suficientes para darnos cuenta de la gran complejidad intrínseca en el desarrollo de ambientes virtuales inteligentes, la variedad de técnicas involucradas, y en el especial énfasis que se coloca en la animación del comportamiento de los agentes virtuales para dotar de mayor credibilidad a las simulaciones. Además de proporcionarnos una mejor perspectiva de los recursos y esfuerzos necesarios para estos desarrollos.

### 2.3.1.1 ACE

ACE [KALL00] es la abreviación de Agent Common Environment, y consiste en una plataforma para la simulación de agentes humanos virtuales en ambientes tridimensionales. Provee comandos pre-construidos para la percepción y actuación, mientras que el procesamiento para el razonamiento

y la conducción es definido mediante una colección módulos de comportamiento externos, extensibles, y parametrizados. El núcleo del sistema ACE interpreta un conjunto de comandos para controlar una simulación; como por ejemplo, comandos para crear y colocar en escena diferentes humanos virtuales, objetos, y objetos inteligentes, aplicar un motor de movimientos a un humano virtual, ejecutar la interacción de un objeto inteligente con un humano virtual, y consultas al sistema de percepción. Todos los comandos descritos son disponibles a través de guiones Python [LUTZ96], un poderoso lenguaje de programación interpretado. Los guiones Python están organizados en módulos, los cuales son cargados dinámicamente desde otro guión, y cumplen diferentes funciones, tales como, generación de la interfase gráfica de usuario, creación de hilos de ejecución para cada agente, módulos de comportamiento, conexión TCP/IP, etc. La arquitectura del sistema incluye un controlador centralizado de agentes, el cual se encarga de ofrecer una coherente simulación multi-agente y facilitar la comunicación entre ellos, ya que cada agente es ejecutado en un hilo (thread) independiente en el que se establece las acción que el agente realiza, pudiendo ser manipulada por el usuario en tiempo de ejecución. La Figura 2.5 muestra la ventana de salida gráfica del sistema ACE con una subventana para la interacción vía Python.

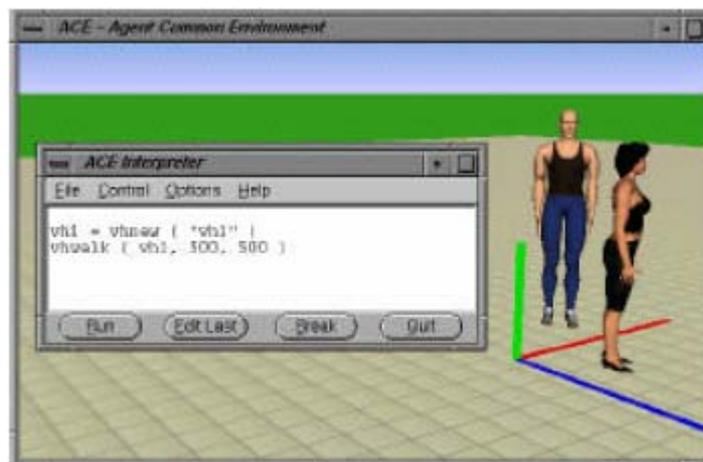


Figura 2.5. Sistema ACE [KALL00] © EPLF Computer Graphics Lab-LIG

### 2.3.1.2 mVITAL

mVITAL [ANAS01] es un ambiente para aplicaciones autor de multi-agentes inteligentes fácilmente aplicables en varios dominios, que van desde sistemas de control distribuido hasta ambientes virtuales para entretenimiento y educación. El sistema esta basado en una arquitectura modular con tres partes principales: el mundo-servidor, el agente-cliente, y

el observador-cliente; todas implementadas como aplicaciones de software separadas de acuerdo al enfoque cliente-servidor. El mundo-servidor representa un mundo virtual dentro del cual todas las actividades de los agentes toman lugar. Los agentes-clientes representan los actores dentro de un entorno, percibiendo el ambiente y actuando según su comportamiento orientado a objetivos. Los observadores-clientes ofrecen la posibilidad a los usuarios supervisores de observar el entorno y todas las actividades que en él ocurren. El sistema opera asincrónicamente sobre un ciclo percibir-decidir-actuar. El proceso de razonamiento de los agentes-clientes es implementado en SICStus Prolog [SICS99], basado en un enfoque orientado a objetivos y una representación simbólica de creencias. El software de visualización está implementado en OpenGL. Los avatares, escenas y objetos son cargados desde archivos VRML, los cuales pudieron ser exportados de paquetes de diseño 3D como Kinetix 3D Studio y Macromedia Poser.

### 2.3.1.3 SimHuman

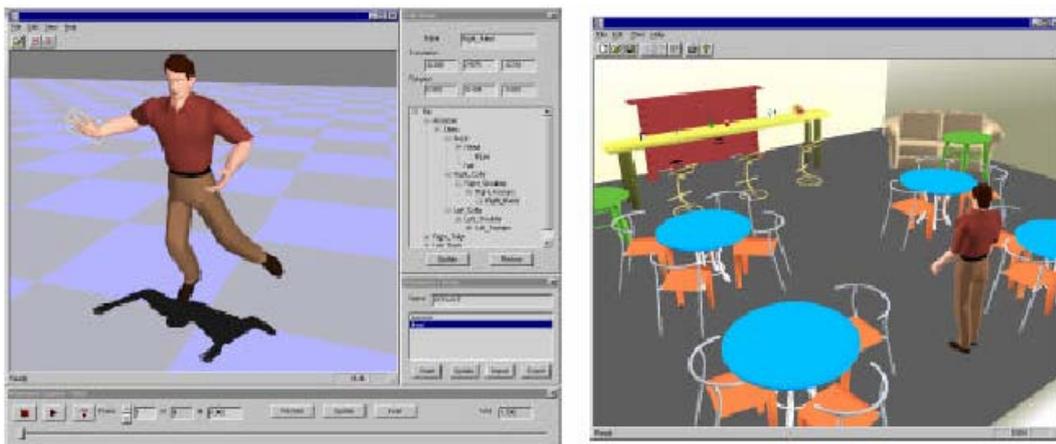


Figura 2.6. Agent Designer (izq.) y Scene Designer (der.), las dos aplicaciones de SimHuman [VOSI03] © Kluwer Academic Publishers

SimHuman [VOSI01][VOSI03] es una herramienta que asiste al programador en el diseño, representación, y animación de ambientes virtuales inteligentes con uno o múltiples agentes y avatares habitando en ellos. Esos mundos sintéticos tienen la característica de incluir cinemática inversa, modelado basado en física, detección de colisiones, visión artificial, y respuesta (acciones sobre otros objetos o agentes) a cambios en el estado de la escena. Esta herramienta consta de una librería para programadores y dos aplicaciones. La librería está implementada en C++ y le permite a los usuarios definir y animar escenas 3D con un número variable de objetos, humanos virtuales, y avatares. Una de las aplicaciones permite cargar modelos de agentes y ayudar al usuario a diseñar el esqueleto y a configurar

las librerías de animación. La otra aplicación le permite al usuario colocar objetos en la escena y visualizar secuencias de acciones de los agentes. La Figura 2.6 muestra las dos aplicaciones.

## 2.3.2 Aplicaciones de Entornos Inteligentes

Casi desde sus comienzos, tanto las aplicaciones de gráficos por ordenador, como de inteligencia artificial, han sido utilizadas para entrenamiento y entretenimiento. Los avances en multimedia ofrecen experiencias cada vez más cercanas a la realidad, haciendo de esta tecnología una herramienta cada vez más presente en nuestras vidas, no importando a que nos dediquemos.

En la actualidad podemos encontrar diversidad de áreas en las que aplicaciones basadas en agentes autónomos virtuales juegan un rol protagónico. Algunos ejemplos de dichas áreas y aplicaciones son las siguientes.

En el mundo del entretenimiento encontramos una inmensa cantidad de aplicaciones; tanto en la industria del video juego, como la del cine. Como ejemplo de esas aplicaciones tenemos *Creatures* [CLIF99], un software de entretenimiento basado en las técnicas de Vida Artificial, que fue lanzado en Europa a finales de 1996, y en Estados Unidos y Japón a mediados de 1997. Instalado en un ordenador domestico, el CD-ROM de *Creatures* crea un mundo virtual en el cual agentes autónomos coexisten. Mucho mas reciente, encontramos el software de relaciones interpersonales de humanos virtuales *The Sims* ([www.thesims.com](http://www.thesims.com)), en el cual el usuario tiene que tratar de mantener el equilibrio emocional de, por lo menos, el personaje que controla, a través de la cotidianidad de los días y de sus relaciones con otros humanos virtuales.

Para terminar con el ámbito del entretenimiento, una aplicación para la producción en tiempo real de productos multimedia y shows de televisión, es el sistema VHD (Virtual Human Director) [SANN99], el cual permite la simulación de movimientos humanos y comportamientos, en sustitución de la animación tradicional de caracteres que usan sistemas de animación de propósito general como Softimage, Alias-Wavefront, Maya, o 3D Studio Max.

Cambiándonos al área del entrenamiento, podemos volver a los tradicionales simuladores de vuelo, pero esta vez para entrenar pilotos de combate en tácticas de vuelo; poniéndolos a prueba en combates simulados con enemigos virtuales inteligentes que miden sus habilidades [JONE93]. En

---

el ámbito deportivo, podemos encontrar aplicaciones especialmente diseñadas para atletas del ciclismo, en la que el usuario puede competir con otros 16 ciclistas simulados a través de ambientes que modelan las rutas reales usando mapas topográficos; consiguiendo que el atleta pueda experimentar la sensación del trayecto, tanto visual como físicamente [BROG98]. Como ejemplo final podemos mencionar STEVE (Soar Training Expert for Virtual Environments) [RICK97], un agente pedagógico que puede físicamente colaborar con estudiantes, adiestrándolos en procedimientos y tareas, orientadas a la operación y reparación de equipos.

Resulta claro que las aplicaciones no se limitan solo a las áreas mencionadas aquí. La inteligencia artificial invade hasta los más elementales electrodomésticos; y bien sea, generados gráficamente o con cuerpos físicos (robots), cada vez nos servimos más, e interactuamos más con agentes virtuales autónomos.