

Narración Digital e Interactiva: Dirección automática de entornos virtuales

Federico Peinado

Departamento de Sistemas Informáticos y Programación

Universidad Complutense de Madrid

c\ Profesor José García Santesmases s/n. 28040 Madrid, España

Tlf. 913947599 – Fax. 913947529

<http://federicopeinado.com>

Resumen: Los entornos virtuales son un medio digital donde cada vez se narran más historias con el propósito de enseñar, entrenar, divertir o comunicar algo a los usuarios. Los guiones predefinidos, la plena libertad de interacción e incluso la simulación de un mundo ficticio y de sus habitantes no son suficientes para crear historias de calidad. Desde finales de los años ochenta, una creciente comunidad científica ha estado desarrollando sistemas inteligentes capaces de dirigir automáticamente todo lo que ocurre en un entorno virtual, para adaptar la historia a los objetivos de la aplicación y a las características particulares de cada uno de los interactores. Este artículo presenta una revisión actualizada del progreso, los resultados en forma de sistemas implementados y las tareas aún pendientes en esta línea de investigación.

Palabras clave: Generación de Historias, Narratología Computacional, Sistemas Inteligentes, Inteligencia Artificial.

1. Introducción

La narración automática en entornos virtuales es un recurso cada vez más frecuente que, utilizado convenientemente en las aplicaciones aptas para ello, amplía enormemente las capacidades del software, otorgando una nueva dimensión al propósito educativo, lúdico o comunicativo de la aplicación. Esta dimensión permite obtener mayor rendimiento al esfuerzo de los autores al mismo tiempo que se mantiene la coherencia y el ritmo narrativo, adaptando además cada historia generada al perfil del usuario y a las decisiones que este vaya tomando durante la ejecución.

Con frecuencia la narración digital e interactiva (IDS de Interactive Digital Storytelling) en un entorno virtual se lleva a uno de estos dos extremos: el *guión predefinido* en forma de grafo y el *argumento emergente* en forma de simulación interactiva con personajes. El guión predefinido restringe las posibilidades de interacción del usuario mediante un determinado patrón impuesto por el autor de la aplicación. Habitualmente este patrón se corresponde con un grafo, donde cada nodo se

corresponde con la realización de una actividad y los arcos con las decisiones tomadas en ella y sus resultados, con los que se permite pasar de una a otra. Por otra parte, el argumento emergente otorga mayor libertad al usuario para interactuar con los personajes y los objetos del entorno de cualquier forma, pero no garantiza nada con respecto a la calidad de la historia que pueda surgir durante el desarrollo de dicha interacción.

Estos extremos no son adecuados para narrar de manera interactiva historias de calidad en entornos virtuales, ya que lo más conveniente es dar suficiente libertad de acción al interactor pero sin perder cierto grado de control sobre el contenido, la forma y la finalidad que autores e interactores buscan en dicha historia.

Para que el lector comprenda mejor la importancia de hallar este equilibrio, puede tomar el caso de *Carmen's Bright IDEAS* [7], una aplicación destinada a las madres de pacientes con cáncer pediátrico. Esta aplicación sumerge al usuario en una historia donde la protagonista asiste a una sesión de terapia psicológica en compañía de una doctora que le aconseja acerca de cómo mejorar su capacidad de resolver problemas y mantenerse optimista. La interactora en este caso debe sentirse libre para proyectar sus sentimientos en las decisiones de la protagonista pero al mismo tiempo es necesario supervisar el delicado desarrollo del diálogo entre esos personajes para evitar que las propiedades terapéuticas de la experiencia se pierdan.

No hay espacio suficiente en este artículo como para profundizar en todos los aspectos referentes al desarrollo de aplicaciones IDS (niveles de interacción, granularidad del control de los personajes, etc.) para lo cual se recomienda leer los primeros capítulos del trabajo de Peinado [13] o un artículo más extenso de Mateas [8] sobre el estado del arte a finales de los noventa.

La estructura de este artículo es la siguiente: en la sección 2 se resume la historia de la investigación en este campo, marcando los hitos más representativos de su evolución; en la sección 3 se describe una pequeña colección de sistemas completamente funcionales que sirven para mostrar los resultados de algunos proyectos de investigación; la sección 4 presenta una breve discusión acerca de lo logrado y los problemas pendientes por resolver en el área; la sección 5 concluye el artículo y al final se adjuntan las referencias bibliográficas utilizadas.

2. Historia y evolución de esta investigación

El interés por la Narratología Computacional hunde sus raíces en las década de los

setenta, cuando se buscaba conocer la estructura formal común de los artefactos narrativos, así como los procesos de comprensión y generación de historias, para poder implementarlos en el ordenador. Por entonces Roger Schank, en la universidad de Yale, dirigía un grupo dedicado a estudiar la Lingüística y la Narratología desde la perspectiva de las Ciencias Cognitivas. Este grupo desarrollo varias aplicaciones informáticas que hoy se consideran clásicas en Inteligencia Artificial (IA), entre las que se encuentra Tale-Spin [10], probablemente el generador de cuentos cortos más popular.

Durante los años ochenta la investigación en IA cambió de rumbo, dirigiéndose a obtener resultados a más corto plazo y aplicables en la industria. Se dejaron aparcados muchos trabajos relacionados con la narrativa y otras disciplinas propias de las Humanidades, aunque existen notables excepciones a esta regla. La tesis doctoral de Brenda Laurel [6], considerada hoy día como la primera propuesta para la construcción de un sistema IDS, es una de ellas. Laurel aplica teorías aristotélicas propias del arte dramático al diseño de un sistema IDS, una idea tan importante que posteriormente ha sido generalizada y llevada al diseño de interfaces software en general. Otro ejemplo interesante es el de Minstrel [16], un generador de historias secuenciales con bastante calidad que utiliza redes semánticas para la representación narrativa y razonamiento basado en casos (CBR, Case-Based Reasoning) con heurísticas creativas para el procesamiento. A pesar de ser un proyecto antiguo que no se centra en los problemas de la narración interactiva, Minstrel es un sistema interesante y bien documentado que construye historias aplicando ideas innovadoras. El problema de continuar la narración de la historia desde un punto concreto de la misma y siguiendo un orden causal se resuelve utilizando historias anteriores. La situación problemática se transforma a otra similar pero en un dominio distinto, donde será más fácil resolverla, continuando la historia de manera que al hacer la transformación inversa se obtiene una posible continuación de la historia pero en el dominio original.

En realidad hasta finales de los noventa no se crea una verdadera comunidad científica especializada en los problemas que plantea el desarrollo de aplicaciones IDS. El taller *AAAI Workshop on Interactive Fiction and Synthetic Realities* celebrado en 1990 fue el primer evento que reunió a los pioneros en el área con intención de averiguar cuales eran las cuestiones fundamentales a las que la comunidad debía responder. Durante esta década, Joseph Bates dirigió un proyecto en la universidad Carnegie Mellon (CMU) [8] llamado Oz. Este proyecto, uno de los más influyentes e importantes en el área, realizó importantes progresos en tres grandes frentes: definición

de una primera *teoría computacional* para los sistemas IDS genéricos con arquitectura de dirección automática sobre entorno virtual, diseño de *personajes creíbles* dotados de autonomía, capacidad de percepción, razonamiento, emoción y comunicación en lenguaje natural, diseño de *interfaces* para presentar el entorno al usuario y permitirle interactuar con él. A las puertas del siglo XXI se celebró el simposio *Narrative Intelligence: AAI Symposium on Computer Games and Artificial Intelligence* (1999) donde se dieron cita muchos de los que hoy lideran la investigación en Narratología Computacional.

En la actualidad existen varios congresos internacionales especializados en el área, siendo los europeos *International Conference on Virtual Storytelling* (ICVS) y *Technologies for Interactive Digital Storytelling and Entertainment* (TIDSE) anteriores al estadounidense *Artificial Intelligence and Interactive Entertainment Conference* (AIIDE), aunque no han faltado toda clase de talleres y seminarios independientes o vinculados a otros congresos de temática similar o más general, como los de Agentes Software, IA, Interacción Hombre-Máquina, etc. Salvo notables excepciones en el campo de la pedagogía, el entrenamiento militar o la ficción interactiva., son pocos los proyectos que han desembocado en aplicaciones de uso real, aunque la comunidad no desiste en su empeño. En la siguiente sección se describen los sistemas IDS más representativos de las distintas líneas de trabajo que están en marcha y que han dado resultados bastante esperanzadores.

3. Tres sistemas IDS actuales

Las historias pueden estar predefinidas por el autor, emerger como resultado de la interacción o ser el fruto de una combinación entre un guión multiforme y una interacción controlada de manera automática. En ese caso puede distinguirse dos clases de sistemas, en unos recae el mayor peso de la historia sobre el autor del guión y en otros son el diseño del entorno virtual y la simulación los que cobran mayor importancia. Además, utilizando como criterio el sistema de dirección automática utilizado o la forma en la que se representa y procesa el guión multiforme para generar las historias, pueden obtenerse otra clasificaciones distintas. Gordon et al. [4] presentan una clasificación basada en el tipo de mediación y de arquitectura software que posee el sistema, que se amplía en este artículo de la siguiente forma:

- Pura narrativa emergente, sistemas sin dirección explícita donde la historia

surge como resultado de la interacción, en un entorno completamente simulado, de personajes interactivos con objetivos dramáticos [1].

- Procesamiento de un lenguaje específico orientado a la autoría de guiones multiformes, como es el caso de Erasmatron [2] debido a que Crawford no considera a las técnicas de IA actuales lo suficientemente potentes como para utilizarse en aplicaciones IDS y por ello es necesario que el guión multiforme sea procesable sin necesidad de razonamiento.
- Razonamiento centralizado, ya esté basado en reglas como la mayoría de los sistemas expertos, en casos [3], en algoritmos de planificación [17] o de propósito específico para mediar en el dilema de la narración interactiva. Estos sistemas pueden comportarse como interactores omnipotentes [15] que a veces ni siquiera requieren que exista realmente un entorno virtual, ya que lo inventan sobre la marcha en función de las acciones de los interactores [4].
- Razonamiento distribuido, generalmente mediante el uso de agentes software para implementar los personajes de la historia, controlados a su vez por un planificador [9] que actúa sobre una simulación, o un personaje único sobre el que se centra toda la historia y que actúa también como director [7, 14]. El sistema Story Engine Architecture [11] es una excepción donde la arquitectura multiagente sirve para implementar la dirección automática, haciendo que cada agente se ocupe de un aspecto distinto de la narración.

A continuación se exponen los aspectos más interesantes de cada uno de estos sistemas. En la Tabla 1 se extrae un resumen de la información que aparece en esta sección relativa a los tres sistemas IDS analizados.

	<i>Descripción</i>	<i>Conceptos</i>	<i>Tecnología</i>	<i>Aplicación</i>
Façade	Personajes creíbles controlados por un gestor dramático	Programación ABL, golpe de efecto, arco dramático	Java y razonador en JESS	Una obra de teatro interactivo de libre distribución
http://www.interactivestory.net/				
I-Storytelling	Actores virtuales con objetivos dramáticos	Planificación HTN	Unreal y planificador en C++	Televisión interactiva
http://www-scm.tees.ac.uk/users/f.charles/				
Zocalo	Servicios de razonamiento y planificación para dirección automática	Planificación DPOCL, servicios web	Entornos 3D como Unreal y razonadores y planificadores en C# para .NET	Entornos educativos
http://zocalo.csc.ncsu.edu/				

Tabla 1. Resumen de la información relativa a tres sistemas IDS.

3.1. Façade

Mateas y Stern son antiguos miembros del proyecto Oz que continuaron sus investigaciones con la tarea de construir Façade [9], un sistema que reconciliase la aproximación basada en agentes software con la del razonador centralizado. La autonomía de los personajes se reduce sólo a aspectos menores del argumento, siendo necesario que un director automático gestione los golpes de efecto narrativos y los saltos entre distintas escenas de la historia, de manera que se mantenga el interés y el conflicto entre personajes en todo momento. El director automático está implementado como un planificador que selecciona, ordena y ejecuta los golpes de efecto, que son los eventos atómicos que permiten que avance la historia. Estos golpes de efecto se representan como pares acción-reacción, formando una unidad como lo hacen una pregunta y una respuesta dentro de un diálogo entre dos personajes. El director procura mantener un arco dramático adecuado, es decir, escoger aquellos golpes de efecto que conducen cada vez a una mayor tensión en la historia, produciendo altibajos en determinados momentos para marcar el final de una escena y el comienzo de la siguiente. Uno de los inconvenientes del sistema es que el autor debe aprender a programar en ABL, un lenguaje específico para describir el comportamiento reactivo de los personajes. La ventaja es que este lenguaje es muy potente ya que permite expresar acciones concurrentes de varios personajes de manera que estos pueden coordinarse sin necesidad de monitorizarse mutuamente de forma permanente. En ABL pueden especificarse objetivos comunes relacionados con la progresión de la historia y no con los objetivos particulares que persigue cada personaje.

Façade se utiliza en una aplicación del mismo nombre que consiste en una obra breve de teatro interactivo. En esta obra el interactor interpreta el papel del amigo de una pareja de recién casados que les visita una noche en su apartamento. La pareja no podrá ocultar su resentimiento mutuo y el interactor estará obligado a tomar partido cuando estalle el conflicto. El entorno virtual 3D es sencillo y consiste en un par de habitaciones con algunos objetos para manipular, pero los personajes que lo habitan poseen un comportamiento rico y complejo: interactúan con el avatar del usuario y con el escenario de manera autónoma, mantienen un diálogo en lenguaje natural entre ellos y con el interactor, sus rostros y sus gestos expresan con precisión las emociones que sienten en cada momento, etc. Façade es de libre distribución, aunque no incluye el código fuente. Ha sido programada en Java y JESS es el motor de inferencia que se

encarga de ejecutar todas las reglas de comportamiento de los personajes y del director automático.

3.2. I-Storytelling

I-Storytelling es un sistema IDS en el cual el autor edita el guión multiforme otorgando objetivos dramáticos a un reparto de actores virtuales cuya interacción en un entorno virtual, junto con la participación del usuario, hace emerger nuevas historias. Estos actores persiguen sus objetivos mediante planes que elabora un planificador externo y centralizado que se basa en Redes Jerárquicas de Tareas (HTN, Hierarchical Task Networks).

Las HTN son grafos Y/O que representan las tareas asignadas a cada uno de los personajes. El nodo raíz de uno de estos grafos representa la tarea principal que debe resolver el personaje, que a su vez se descompondrá en subtareas, nodos conectados en dos clases de grupos distintos: grupo de *nodos Y*, lo que significa que para resolver la tarea con éxito es necesario resolver todas y cada una de las subtareas con éxito, y grupo de *nodos O*, donde bastará con que una de las subtareas se resuelva con éxito para que la tarea del nodo padre se considere resuelta con éxito.

El planificador HTN va recorriendo la jerarquía de nodos desde la raíz hasta las hojas, en un orden marcado por la prioridad que el autor ha asignado a cada nodo. Los nodos hoja son acciones elementales del personaje, de manera que pueden considerarse tareas cuyo éxito o fracaso se puede conocer trivialmente, ya que depende de si hay o no recursos disponibles de entre los necesarios para realizar con éxito esa acción. Cuando una acción no se puede realizar por falta del recurso apropiado para ello el algoritmo replanifica aplicando *vuelta atrás*.

Algunas versiones del sistema pueden conseguirse estableciendo contacto con los autores. En ellas el interactor se limita a establecer la situación inicial del entorno virtual para cada ejecución, o a modificar el estado de los objetos y la información que poseen los personajes en cada momento, para lo que cuenta además con la posibilidad de darles “avisos” y “consejos” en lenguaje natural a través del micrófono.

El sistema ha sido implementado en C++, aunque posee ficheros en formato propietario para la transmisión de datos. El planificador se comunica mediante sockets con el motor de juego Unreal, donde se ha implementado otro módulo cuyo código gestiona los personajes, los objetos y el entorno a más bajo nivel.

3.3. Zocalo

Zocalo es una arquitectura abierta orientada a servicios web para controlar sistemas IDS en entornos virtuales. Su arquitectura sigue la clásica organización en tres capas: una capa de servicios web, otra capa con el gestor de ejecución y una tercera capa con el entorno de ejecución correspondiente donde se representa cada paso de la ejecución y se gestiona el interfaz y la interacción con los usuarios.

Este sistema proviene de otro anterior llamado Mimesis, aunque ahora sus elementos han sido distribuidos en subsistemas y herramientas separadas para facilitar su extensión e integración en distintas plataformas y entornos. Esta es la descripción de dichos elementos:

- Servicios Web. Su funcionalidad es la de estar disponibles en Internet mediante APIs abiertas para resolver problemas de planificación y mediación de narración interactiva a nivel conceptual. Estas tareas son generalmente muy costosas en términos de computo, por lo que su distribución facilita que cualquier máquina con conexión a Internet tenga la posibilidad de completar su capacidad con la de servidores más potentes y dedicados exclusivamente a esas tareas. Algunos de los servicios que actualmente incluyen Zocalo son:
 - Fletcher, que ofrece generación de planes de orden parcial. Utiliza Crossbow, un planificador de enlace causal jerárquico implementado en C# y basado en el algoritmo DPOCL. Crossbow descende de un planificador anterior llamado Longbow e incluye aspectos complejos como la representación de las creencias de los personajes.
 - Crosswind, que ofrece mediación reactiva. Esto implica crear y mantener los árboles con políticas de mediación que utiliza el gestor de ejecución para monitorizar la actividad del entorno virtual, mantener la coherencia de la historia y adaptarla según las acciones del interactor, aunque estas hagan que el argumento difiera mucho de lo esperado. Crosswind está implementado en ASP.NET para .NET 1.1, puede crear un plan de contingencia llamando a Fletcher o replanificar localmente. El árbol con políticas sufre podas y su raíz variará según sean las decisiones de mediación tomadas; su profundidad puede limitarse para no disparar el coste en eficiencia.
 - Kyudo, que ofrece mediación proactiva. Comparte muchas

capacidades con Crosswind, además de ser capaz de elaborar soluciones incrementales. También está implementado como una librería C#.

- Gestor de Ejecución. Recibe los operadores conceptuales que devuelven los servicios web tras una petición HTTP y los transforma en operadores concretos para su envío y posterior ejecución en un entorno de ejecución concreto, cuyas características obviamente debe conocer el gestor de ejecución. Actualmente existen tres implementaciones de este gestor, para facilitar la tarea de integrar Zocalo en diferentes tipos de aplicación: una DLL para .NET, otra DLL en C que utiliza .NET internamente y un ejecutable aparte que se comunica por sockets.
- Entorno de Ejecución. Los desarrolladores de cada aplicación han de desarrollar clientes que se comuniquen con el gestor de ejecución de Zocalo. Actualmente se ha desarrollado un cliente que utiliza el motor Unreal Tournament 2004 y hay otro para Half-Life 2 en preparación, aunque ninguno de ellos está disponibles al público.

El sistema además cuenta con herramientas visuales como Bowman, para crear y editar librerías de planificación para Crossbow. Esta herramienta en principio pretende abstraer los detalles de bajo nivel para que pueda usarse por autores de guiones multiformes sin conocimientos de programación, aunque la versión actual requiere dominar ciertos conceptos de planificación. Es una aplicación cliente programada en C# para .NET 1.1 compuesta por un editor y un navegador del dominio de la planificación, el entorno virtual, y los planes individuales que allí se ejecutan. Bowman invoca planificadores basados en Zocalo comunicándose a través de Fletcher.

Toda la información relativa a planificación se envía y almacena en ficheros XML. La complejidad de estos ficheros con problemas de planificación, planes, conjuntos de acciones, etc. hace que sea conveniente usar un estándar para implementarlos y un API común para manejarlos de manera que se facilite la tarea a los desarrolladores que utilicen Zocalo. Los esquemas de los ficheros se han implementado con XML Schema Definition Language y están embebidos en un ensamblaje de .NET. Las clases de este ensamblaje incluyen lectores para validar documentos XML, analizadores y escritores de XML que extraen y almacenan información utilizando métodos fuertemente tipados.

4. Discusión sobre el trabajo futuro

Aunque sistemas como Zocalo aseguran ser reusables y extensibles para adaptarse a distintos dominios y aplicaciones, lo cierto es que muy pocos han conseguido ser implementados en más de una aplicación. Façade directamente se plantea como un sistema pensado específicamente para una aplicación muy concreta e I-Storytelling es aún un proyecto experimental en busca de aplicación comercial. La arquitectura en capas y componentes de Zocalo permite extender, añadir o reemplazar unas implementaciones por otras, e incluso ampliar la funcionalidad de los subsistemas y comunicarlos entre sí. Pero muchas de esas posibilidades de los sistemas IDS están aún por explotar. Existen muchas otras aproximaciones, otros sistemas IDS, aún en desarrollo como es el caso de KIIDS [12], que prometen alcanzar la reusabilidad con la formalización del conocimiento en ontologías y la utilización de componentes software asociados. KIIDS utiliza una arquitectura CBR como la de Minstrel u OPIATE [3], aunque el conocimiento se hace extensible a dominio y aplicación y se implementa haciendo uso de tecnologías más actuales vinculadas a la Web Semántica. El formalismo de representación basado en Lógicas Descriptivas, OWL DL, se está convirtiendo poco a poco en un estándar para estas aplicaciones y además permite razonamiento automático para dirección IDS como muestran Koller et al. [5].

Las aplicaciones comerciales como los videojuegos narrativos todavía no incorporan estas tecnologías, aunque están buscando mecanismos eficientes que eleven la calidad y la variabilidad de sus historias y que vayan más allá del procesamiento básico de guiones predefinidos y la simulación realista de entornos y personajes.

5. Conclusión

En este artículo se ha descrito brevemente la trayectoria de la investigación en sistemas IDS de dirección automática, centrándose la discusión en tres de los más representativos actualmente: Façade, I-Storytelling y Zocalo. Actualmente pocas aplicaciones reales utilizan sistemas tan ambiciosos, aunque cada día parece estar más cerca su implantación a nivel industrial. El primer paso para ofrecer entornos virtuales narrativos de calidad, donde los contenidos se presenten según el ritmo de acción y el interés de sus usuarios, es conocer lo mejor posible lo que nos ofrece la comunidad científica. De la misma forma, es necesario que dicha comunidad se preste a colaborar con particulares y empresas para llevar su tecnología más allá de los laboratorios y aplicarla

en situaciones reales donde se ponga de manifiesto su verdadera utilidad.

6. Referencias

- [1] M. Cavazza, F. Charles, S.J. Mead. Character-Based Interactive Storytelling. *IEEE Intelligent Systems: Special Issue on AI in Interactive Entertainment* 17, 17-24 (2002).
- [2] C. Crawford. Erasmatron & Erasmaganza. In. (Erasmatazz), 2003. <http://www.erasmatazz.com/>
- [3] C.R. Fairclough. Story Games and the OPIATE System. Doctoral Thesis, University of Dublin - Trinity College (2004).
- [4] A.S. Gordon, N.V. Iuppa. Experience Management Using Storyline Adaptation Strategies. International Conference on Technologies for Digital Storytelling and Entertainment, Darmstadt, Germany. (2003).
- [5] A. Koller, R. Debusmann, M. Gabsdil, K. Striegnitz. Put my Galakmid Coin into the Dispenser and Kick It: Computational Linguistics and Theorem Proving in a Computer Game. *Journal of Language & Computation. Hermes Science Publishing LTD* 1, 1-19 (2001).
- [6] B.K. Laurel. Toward the Design of a Computer-Based Interactive Fantasy System. Ohio State University (1986).
- [7] S.C. Marsella, W.L. Johnson, L. Catherine. Interactive Pedagogical Drama. International Conference on Autonomous Agents, Barcelona, Spain. (2000), 301-308.
- [8] M. Mateas. An Oz-Centric Review of Interactive Drama and Believable Agents. *AI Today: Recent Trends and Developments. Lecture Notes in Artificial Intelligence* 1600, 297-328 (1997).
- [9] M. Mateas, A. Stern. Façade: An Experiment in Building a Fully-Realized Interactive Drama. Game Developers Conference, Game Design track. (2003).
- [10] J.R. Meehan. TALE-SPIN, An Interactive Program That Writes Stories. *IJCAI*. (1977), 91-98.
- [11] B.A. Osborn. An Agent-Based Architecture for Generating Interactive Stories. The MOdeling, Virtual Environments and Simulation (MOVES) Institute (2002).
- [12] F. Peinado. Knowledge-Intensive Interactive Digital Storytelling system. In., 2006. <http://federicopeinado.com/projects/kiids/>
- [13] F. Peinado. "Mediación Inteligente entre Autores e Interactores para Sistemas de Narración Digital Interactiva" *Tech. Report* (Departamento de Sistemas Informáticos y Programacion. Facultad de Informática. Universidad Complutense de Madrid, 2004).
- [14] J. Piesk, G. Trogemman. Animated Interactive Fiction: Storytelling by a Conversational Virtual Actor. International Conference on Virtual Systems and MultiMedia, Geneva, Switzerland. (1997).
- [15] B.J. Rhodes, P. Maes. The Stage as a Character: Automatic Creation of Acts of God for Dramatic Effect. *AAAI Spring Symposium on Interactive Story Systems*, Stanford. (1995).
- [16] S.R. Turner. "Minstrel: A Computer Model of Creativity and Storytelling" *Tech. Report UCLA-AI-92-04* (Computer Science Department, University of California, 1992).
- [17] R.M. Young. "Notes on the Use of Plan Structure in the Creation of Interactive Plot" *Tech. Report FS-99-01* (AAAI Fall Symposium on Narrative Intelligence, AAAI Press, Menlo Park, 1999).