# *Interactive Digital Storytelling: Automatic direction of virtual environments*

**Federico Peinado**

Departamento de Sistemas Informáticos y Programación

Universidad Complutense de Madrid

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

Tel. 913947599 – Fax. 913947529

http://federicopeinado.com

**Abstract:** Virtual environments are a digital media in which each time more stories are told with the purpose of teaching, training, entertaining or communicating something to the users. Predefined scripts, plenty of interaction freedom and even the simulation of a fictional world and its inhabitants are not enough to create quality stories. Since late eighties, a growing scientific community has being developing intelligent systems able to direct automatically everything that happens in a virtual environment, in order to adapt the story to the application goals and the particular characteristics of each interactor. This paper presents an updated review of the steps forward, results in the form of implemented systems and pending tasks of this research line.

**Keywords:** Story Generation, Computational Narratology, Intelligent Systems, Artificial Intelligence.

## 1. Introduction

Automatic storytelling in virtual environments is becoming a common resource that, used in the right way for the appropriated applications, expands the software capacities, giving a new dimension to the pedagogical, recreational or communicative purpose of the application. This dimension allows getting more performance from the efforts of the authors at the same time coherence and narrative rhythm are maintained, also adapting each generated story to the user profile and the decisions she makes during the execution.

Frequently, Interactive Digital Storytelling (IDS) in a virtual environment is taken to one of these two extremes: the *predefined script* represented as a graph and the *emergent plot* represented as an interactive simulation involving characters. The predefined script restricts the possibilities of interaction of the user by means of an specific pattern imposed by the author of the application. Usually this pattern is represented as a graph where each node is related with the achievement of an activity and the arches are related to the decisions taken within it and the results obtained. These arches let the user go from one activity to another. On the contrary, the emergent plot grants more freedom to the user for the interaction with characters and objects in the environment in any way, but it does not guarantee anything about the quality of the story that would emerge during the interaction.

These extremes are not suitable to narrate interactively good stories in virtual environments, because the most advisable thing to do is giving enough freedom to act to the interactor but not losing control over the content, the form and the purpose of the story in which authors and interactors agree.

For the reader to understand better the importance of balancing these elements, the case of *Carmen's Bright IDEAS* [7] is presented, an application oriented to the mothers of pediatric cancer patients. This application immerses the user in a story which protagonist visits a doctor for a session of psychological therapy. The doctor gives advices to her about how to improve her problem-solving ability and to be optimistic all the time. In this case the interactor must feel free to project her feelings in the decisions of the protagonist, but at the same time it is needed to supervise the delicate development of the dialog between the characters in order to avoid the therapeutic properties of the experience getting lost.

There is no room enough in this paper to study every aspect about IDS application development in depth (levels of interaction, granularity of the characters control, etc.), so it is recommended to read the first chapters of the work of Peinado [13] or a longer paper by Mateas [8] about the state of art at the end of the nineties.

The structure of this papers is as follows: section 2 summarizes the history of research in this field, marking the most representative milestones in its evolution; section 3 describes a small collection of fully-implemented systems that show the results of some research projects; section 4 presents a brief discussion about what has been achieved and what are the unsolved problems in the field; section 5 is the conclusion of the paper and finally the bibliographic references that have been used are enclosed.

## 2. History and evolution of this research

The interest in Computational Narratology sinks roots in the seventies, when researchers were trying to find out what is the formal structure shared by every narrative artefact and how the processes of comprehension and generation of stories work, in order to implement them in the computer. Those days Roger Schank, at Yale University, was leading a group dedicated to the study of Linguistic and Narratology from the perspective of Cognitive Sciences. This group developed several computer applications that today are considered classics on Artificial Intelligence (AI), including Tale-Spin [10], probably the most popular generator of short tales.

During the decade of the eighties, AI research changed its course, making its own way toward more short-term results and industrial applications. Many projects related to narrative and other topics associated with the Humanities were abandoned, but there are remarkable exceptions to this rule. Doctoral thesis of Brenda Laurel [6], considered today as the first proposal for the construction of an IDS system, is one of them. Laurel apply Aristotelian theories typical for the dramatic art to the design of an IDS system, such an important idea that later on, it has been generalized and applied to the design of software interfaces in general. Another interesting example is found in Minstrel [16], a generator of sequencial stories with enough quality that uses semantic networks for the narrative representation and Case-Based Reasoning (CBR) with creative heuristics for the processing. Although it is an old project that does not focus on the problems of interactive storytelling, Minstrel is an interesting system, well-documented, that builds stories applying innovative ideas. The problem of continuing the narration of a story from an specific situation and following a causal order is solved using previous stories. The problematic situation is transformed to a similar one but in a different domain, where it will be easy to solve, continuing the story in a way that, applying the inverse transformation a possible solution is obtained for the original domain.

In fact, until the end of the nineties there is no truly scientific community specialized in the problems that arise in the IDS application development. The *AAAI Workshop on Interactive Fiction and Synthetic Realities* hold in 1990 was the first event that gathered the pioneers in the field together with the purpose of finding out what are the fundamental questions that the community had to answer. During this decade, Joseph Bates leaded a project at the Carnegie Mellon University (CMU) [8] called Oz. This project, one of the most influential and relevant in the field, carried out important improvements in three great fronts: definition of a first *computational theory* for the IDS generic systems with architectures involving automatic direction of virtual environments, design of *credible characters* showing autonomy, capacity of perception, reasoning, emotion and communication in natural language, design of *interfaces* to present the environment to the user and let him interact with it. Almost at the beginning of the XXI century a symposium was hold called *Narrative Intelligence: AAAI Symposium on Computer Games and Artificial Intelligence* (1999) where many of the nowadays leaders on Computational Narratology research published their contributions.

Currently there is a number of international conferences specialized in the field, being the Europeans *International Conference on Virtual Storytelling* (ICVS) and *Technologies for Interactive Digital Storytelling and Entertainment* (TIDSE) older than the north american *Artificial Intelligence and Interactive Entertainment Conference* (AIIDE), but there are also many others workshops and seminars independent or associated with other conferences on similar or more general topics such as Software Agents, AI, Human-Computer Interaction, etc. Except outstanding exceptions in the fields of pedagogy, military training or interactive fiction, there are few projects that produce applications for the real life, but the community is still working on that. Next section describes the most representative IDS systems on the different research lines that are currently active and offering encouraging results.

## 3. Three up-to-date IDS systems

Stories could be predefined by the author, emerge as result of the interaction or as a combination of a multiform plot and an automatic directed interaction. In this case two types of systems could be distinguished: in the first one the story is up to the author of the plot and in the second one the design of the virtual environment and the simulations are the key factors. Moreover, using the automatic direction mechanism or the representation and processing of the multiform plot as criterions, many different classifications are possible. Gordon et al. [4] presents a classification based on the type of mediation and software architecture that systems are using. This classification is extended here in this paper:

- Pure emergent narrative, systems without explicit direction where the story appears as a result of the interaction, in a completely simulated environment with interactive characters provided with dramatic goals [1].
- Processing of an specific language oriented to the authors of multiform plots, as in the case of Erasmatron [2] because Crawford does not consider the current AI techniques powerful enough to be used in IDS applications, that is the reason for the multiform plot to be processable without reasoning.
- Centralized reasoning based on rules as many expert systems, based on cases [3], on planning algorithms [17] or specific-purpose algorithms for mediating in the dilemma of interactive storytelling. These systems may act as

omnipotent interactors [15] that usually do not require a truly virtual environment to work, because it is created on-demand depending on the actions taken by the interactors [4].

- Distributed reasoning, generally using software agent to implement the characters of the story, controlled by a planner [9] acting over a simulation, or a stand-alone protagonist that controls the whole story as a director [7, 14]. The Story Engine Architecture [11] is an exception because its multiagent architecture is used to implement a complex automatic direction system in which each agent has a different responsibility in the storytelling process.

Now the most interesting aspects of each system is presented. In the Table 1 there is a summary of the information that appears in this section in relation to three analyzed systems.

| | *Description* | *Concepts* | *Technology* | *Application* |
|---|---|---|---|---|
| **Façade** | Credible characters controlled by a dramatic manager | ABL programming, beat, dramatic arch | Java and JESS reasoning | An interactive drama, freely distributed software |
| | http://www.interactivestory.net/ | | | |
| **I-Storytelling** | Virtual actors with dramatic goals | HTN planning | Unreal and C++ planning | Interactive television |
| | http://www-scm.tees.ac.uk/users/f.charles/ | | | |
| **Zocalo** | Services of reasoning and planning for automatic direction | DPOCL planning, web services | 3D environments as Unreal and C# reasoning and planning for .NET | Pedagogical environments |
| | http://zocalo.csc.ncsu.edu/ | | | |

**Table 1.** Summary of the information related to three IDS systems.

### 3.1. Façade

Mateas y Stern are old members of the Oz project that follow their research with the purpose of finishing Façade [9], a system that reconciles the character-centred and the director-centred approaches. The autonomy of the characters plays only a minor role in the development of the plot, because it is necessary an automatic director managing narrative beats and changes between different scenes of the story, maintaining the interest and the conflict between main characters all the time. The automatic director is implemented as a planner that selects, orders and executes beats, the atomic events that unfold the story. These beats are represented as action-reaction pairs, working as a single unit as a question and its answer within a dialog between two characters. The director tries to maintain an appropriated dramatic arch, that means to choose those beats that will guide the story to more tension and creating contrast at the end of one scene and the beginning of the next one. One of the inconvenients of this system is that the author has to know the basis of ABL programming, an specific language to describe reactive behaviour of the characters. The good point is that ABL is a powerful language that allows the author to express concurrent actions of several characters, avoiding a continuous monitorization between characters. ABL can be used to specify common goals related to the development of the plot and not only related to the particular goals of a single character.

Façade is used in an application with the same name, a short interactive drama play. In the story the interactor plays the role of a friend of a just-married couple that is visiting them in their flat in the evening. The couple cannot hide their resentments and the interactor will be obliged to take part in the conflict between them. The 3D virtual environment is simple: a number of rooms with some objects to take and drop. But the inhabitants of this world presents a very rich and complex behaviour, interacting autonomously with the user avatar and the elements of the stage, establishing a dialog in natural language between them and the interactor, expressing on-the-fly emotions with their faces, gestures, etc. Façade can be freely distributed, but there is no source files included in the distribution. Java and JESS have been used to code the system; JESS is the inference engine that fires all the rules about character behaviour and automatic direction.

### 3.2. I-Storytelling

I-Storytelling is an IDS system which author has to edit a multiform plot assigning dramatic goals to a cast of characters. The interaction between these characters and the user in the virtual environment generate new stories. These actors follow their goals using plans created by an external and centralized planner based on the theory on Hierarchical Task Networks (HTN).

HTNs are AND/OR graphs that represent tasks assigned to each character. The root node is the main task that a character has to solve, that is decomposed in several subtasks, linked nodes in two types of groups: *AND nodes* group, that means for solving the parent task it is necessary to solve every subtask of the group, and *OR nodes* group, that means for solving the parent task it is enough to solve any of the subtask.

The HTN planner follows the hierarchy of nodes from the root node to the sheet nodes in the order imposed by the priority that the author uses to tag each node in the editor. Sheet nodes are basic actions of the characters that could be considered as trivial tasks with immediate success or failure, depending of the availability of the resources involved in the action. When an action cannot be carried out because some implicated resource is missing, the algorithm changes the current plan using *backtracking*.

Some versions of the system can be requested from the authors. In these versions the interactor only establish the initial situation of the virtual environment for each execution, or modify the objects state and the information that the characters own, giving “warnings” and “advices” in natural language using the microphone.

The system has been implemented in C++, but it has files in proprietary format for transferring data. The planner uses sockets for the communication with the Unreal game engine, in which there is another module with code to manage the characters, objects and stage at the most lower level.

### 3.3. Zocalo

Zocalo is an open architecture oriented to web services in order to control IDS systems in virtual environments. Its architecture follows the classic organization of three layers: one layer for the web services, another layer for the execution manager and a third layer for a specific execution environment in what each step of the simulation is represented and the interface and the user interaction are managed.

This system is a descendent of a previous one called Mimesis, now its elements have been distributed in subsystems and stand-alone tools to easily integrate Zocalo in different environments and applications. These are the descriptions of those elements:

- Web Services. The functionality is to be ready on the Internet via open APIs to solve problems about planning and mediation in interactive storytelling at a conceptual level. These tasks are generally very expensive in computational terms, that is why their distribution make easy to give any connected machine the opportunity of taking advantage of the capacity of more powerful servers dedicated exclusively to these tasks. Some services included in Zocalo are:
    - Fletcher, it offers partial order generation of plans. It uses Crossbow, a hierarchical causal planning system implemented in C#, following the old algorithm DPOCL. Crossbow is a descendant of a planner called Longbow and it includes complex aspects as the representation of belief of characters.
    - Crosswind, it offers reactive mediation. This means creating and

maintaining policy trees for mediation that are used by the execution manager to monitorize the activity of the virtual environment, preserving story coherence and adapting the content to the actions of the interactor, no matter how far from the original plot are they. Crosswind is implemented in ASP.NET for .NET 1.1, it can create a contingency plan calling Fletcher or replanning locally. The policy tree is pruned and the root node changes depending on the mediation decisions taken; the depth of the tree can be limited to avoid too much computational cost.

  - Kyudo, it offers proactive mediation. It shares functionality with Crosswind, it is implemented as a library in C# and it can also create incremental solutions.

- Execution Manager. It takes the conceptual operators that web services return after the HTTP request and transform them in specific operators to send an execute them in the specific execution environment. The characteristics of the environments have to be known by the execution manager. Nowadays there are three implementations of this manager to make easy the integration of Zocalo in different types of applications: there is one DLL for .NET, another one for C (that uses .NET internally) and a stand-alone executable file that uses sockets for communication.
- Execution Environment. The developers of each application will have to develop clients that communicate with the Zocalo execution manager. Now there is only one client that uses the game engine of Unreal Tournament 2004, and there is another one for Half-Life 2 under development, none of them are publicly available yet.

The system also has visual tools as Bowman, for creating and editing planning libraries for Crossbow. This tool initially tries to abstract low-level details to be used by multiform plot authors with no programming experience, although to use the current version some knowledge about planning are required. It is a client application coded in C# for NET 1.1, it includes an editor and a navigator for the planning domain and the particular plans. Bowman calls Zocalo planner using Fletcher as a communication channel.

All the information related to planning is send and saved in XML files. Complexity of these files with problems of planning, plans, action sets, etc. makes important to use a standard to implement them and a common API to control everything, simplifying the life of future Zocalo developers. Files schemas are implemented using XML Schema Definition Language and they are included in an assembly for .NET. Classes of this assembly includes readers to validate XML documents, XML parsers and writers that extract and save the information using strong-typed methods.

## 4. Discussion about future work

Although systems as Zocalo declare being reusable and extensible to be easily adapted to different domains and applications, the truth is few systems have been used to implement more than one application. Façade is designed directly as an application-oriented system and I-Storytelling is an experimental project yet, searching for a good commercial application. The layers and components based architecture of Zocalo allows to extend, add or replace some implementations by others, even coding more functionality for a subsystem and establishing communication between them. But many of these possibilities of IDS systems have not been exploited yet. There are many other approaches, other IDS systems, still under development as in the situation of KIIDS [12], that promises more reusability using formalization of knowledge in ontologies and associated software components. KIIDS uses a CBR architecture as Minstrel or OPIATE [3], but the knowledge is made extensible to domain and application, and it is implemented using up-to-date Semantic Web technologies. The representation formalism based on Description Logics, OWL DL, is becoming a standard for these applications and it also allows automatic reasoning for IDS as Koller et al. show [5].

Commercial applications as narrative videogames do not use these technologies yet, but they are searching for efficient mechanisms for the quality and variability of their stories to increase, going far beyond the simple processing of predefined scripts or realistic simulation of environments and characters.

## 5. Conclusion

In this paper the course of research on IDS systems with automatic directions has been briefly described, focusing the discussion around three of the most representative up-to-date systems: Façade, I-Storytelling y Zocalo. Nowadays few real applications are using as ambitious systems as these, but each day the industrial implantation seems to be closer. In one hand, the first step to offer quality narrative virtual environments, where contents are presented depending on the action rhythm and the interests of the users, is to know very well what the scientific community is offering. In the other hand, it is necessary the collaboration of that community with private individuals and companies to bring the technology far beyond the laboratories and apply it to real situations where its truly utility was shown.

## 6. References


[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).