



This is an author manuscript post-peer-reviewing (accepted version) of the original publication. The layout of the published version may differ .

Towards minimalism and expressiveness in interactive drama

Szilas, Nicolas; Wang Szilas, Jue; Axelrad, Monica

How to cite

SZILAS, Nicolas, WANG SZILAS, Jue, AXELRAD, Monica. Towards minimalism and expressiveness in interactive drama. In: DIMEA '08 Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts. Athens (Greece). New York, NY : ACM Press, 2008. p. 385–392. doi: 10.1145/1413634.1413703

This publication URL: <https://archive-ouverte.unige.ch/unige:48235>

Publication DOI: [10.1145/1413634.1413703](https://doi.org/10.1145/1413634.1413703)

Towards minimalism and expressiveness in Interactive Drama

Nicolas Szilas

TECFA, FPSE, University of Geneva
CH 1211 Genève 4,
Switzerland
+41 22 379 9307

Nicolas.Szilas@tecfa.unige.ch

Jue Wang

TECFA, FPSE, University of Geneva
CH 1211 Genève 4,
Switzerland

juewang@juewang.net

Monica Axelrad

TECFA, FPSE, University of Geneva
CH 1211 Genève 4,
Switzerland

Monica.Axelrad@pse.unige.ch

ABSTRACT

Having found it difficult for authors to be creative with current interactive drama systems, we propose an alternative author-centered approach allowing authors of nonlinear media to express themselves easily and smoothly.

At the level of character animation, a bottleneck in Interactive Drama, our idea is to build authoring tools and author-centered engines for interactive drama that make it possible for an author to create virtual worlds, characters and animation instantly. To achieve this goal, a minimalist design for each of those features is proposed. A Use Case is described to better illustrate the approach.

Categories and Subject Descriptors

I.2 [Artificial Intelligence]: Distributed Artificial Intelligence – Intelligent agents, Multiagent systems; Applications and Expert Systems – games. I.3.6 [Methodology and Techniques]: Ergonomics – Interaction techniques J.5 [Arts and Humanities]: Fine arts, Linguistics, Literature, Performing art.

General Terms

Algorithms, Design, Human Factors.

Keywords

Authoring, Author Centered Design, Minimalism, Animation Engine, Interactive Narrative, Interactive Drama.

1. THE ISSUE OF AUTHORIZING

Almost two decades have elapsed since S. Smith and J. Bates (Smith & Bates, 1989) proposed a new dramatic genre, now called Interactive Drama. The idea of such an artistic medium consists in enabling a player within a narrative game to play a main character, interact with other characters and deeply influence the course of events in a story. Since then, several systems have been proposed and developed (Aylett et al., 2006; Crawford, 1999; Magerko, 2002; Mateas & Stern, 2000; Spierling, Grasbon, Braun, & Iurgel, 2002; Szilas, 1999, 2007; Young et al., 2004).

For each of these systems, either the entire story or parts of the story need to be programmed/written in order to both use the system for real world applications and assess the qualities and drawbacks of the approach. Many researchers have observed that this authoring process is quite difficult (Mateas & Stern, 2005a; Skorupski, Jayapalan, Marquez, & Mateas, 2007; Spierling, 2007;

Spierling & Iurgel, 2003; Szilas, Marty, & Rety, 2003). These authoring difficulties are correlated to the fact that there exists very few published works using Interactive Drama technologies (Portugal, 2006).

There is one clear exception with *Façade*, a fully complete Interactive Drama released in 2005 (Mateas & Stern, 2005b). *Façade* constitutes an artistic outcome of Interactive Drama research. It involves a 20 minute long story with free text input and cartoon-like rendered, emotional 3D characters. We believe however, that this case also illustrates the above mentioned difficulties:

- *Façade* was created by researchers/developers who designed and programmed the algorithms of narrative management and user interaction. This situation illustrates the fact that authoring involves the highest computer skill, which constitutes an issue if the goal is to create more pieces based on Interactive Drama technology.
- *Façade* required a lot of authoring work (4 years for two people, including authoring). This is striking because it is always possible to make an amateur short movie in a few weeks. Similarly a simple adventure game can be written rather quickly. Historically, these media start small, while for interactive storytelling, this seems different.

The point we make here might seem highly subjective and indeed, we cannot easily assess each demo and « prove » that the result is insufficient because of an authoring issue. It could be argued that this is only a question of maturity of the field. However, we are making the conjecture that authoring should be considered as the cornerstone of Interactive Drama and that research is needed to improve authoring tools and methodologies. This idea is shared by several researchers in the field (Mateas & Stern, 2005a; Portugal, 2006; Skorupski et al., 2007; Spierling, 2007; Spierling & Iurgel, 2003).

To support the conjecture above, we first describe two approaches that were followed in two previous research projects carried out by one of the authors. These two approaches lead to a third albeit not yet implemented approach applied to the design of an animation engine. Finally a global author centered architecture for Interactive Drama is proposed, in conjunction with methodological guidelines.

2. EXPERIENCE FEEDBACK

2.1 *IDtension*: Algorithm-centered design

Authoring-related approach: Interactive Drama requires new algorithms in order to be able to overcome the conflict between narrative and interactivity (Crawford, 1996; Louchart & Aylett, 2003; Portugal, 2000).

IDtension is an Interactive Drama system which includes a narrative engine able to combine both local and global agency. Users' actions influence immediate and future events in the story while narrative interest and consistency is maintained (Szilas, 2007).

Departing from the pure agent-based approach, *IDtension* is based on a general model of narrative. The system generates narrative events according to this general model. It contains the following four main distinctive features:

1. Narrative is described at the level of actions (not scenes or plot points)
2. The system includes a user model aimed at estimating the impact of each possible action on the user according to several narrative criteria (Szilas, 2001). Some of these criteria focus on believability. Other criteria such as complexity and conflict are only guided by narrative concerns.
3. The articulation of actions is twofold. *IDtension* considers generic actions and specific tasks. Generic actions stem from narratology (Bremond, 1973; Greimas, 1983; Todorov, 1970). They are, for instance, inform, encourage/dissuade, accept/refuse, perform, felicitate/condemn. Tasks are specific to a story: kiss, hug, slap (in a romance story) or threaten, torture, kill (in a roman noir), etc. This makes it possible to handle complex actions like "John tells Mary that Bill has robbed her jewels" without requiring the author to explicitly enter them into the system. The narrative engine handles logical forms such as *Inform(John, Mary, have_finished(Bill, rob , [jewels,Mary])*. The author only specifies the task (rob), the characters (Mary, John, Bill) and the objects (the jewels) in the story.
4. *IDtension* explicitly processes the notion of (ethical) values. Values are thematic axes according to which each task is evaluated. Such values include honesty, friendship, family, etc. This mechanism adds another dimension to the story that goes beyond the pure performative dimension. The User Model processes those values to evaluate some narrative criteria, in particular conflict.

Action selection is performed in three steps:

1. The Narrative Logic generates the set of all possible actions at a given time in the narrative via a set of narrative rules.
2. The User Model assesses each of these actions according to its estimated impact on the user. The actions can be ranked accordingly.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

3. The Theatre displays the selected action by generating the text with a template based approach and by sending it to the virtual characters.

The system alternates actions chosen by the engine and actions chosen by the user.

While the current demo of *IDtension* exhibits an interesting combination of narrative and interactivity, it lacks artistic expressiveness.

This difficulty could be reduced to the absence of an authoring tool for entering narrative content. Although, the difficulties likely stem from the highly procedural nature of the underlying narrative model combined with an algorithmic-centered design.

2.2 *BEcool*: Authoring-informed design

Authoring-related approach: the algorithm design of an engine must favor easy to author choices over complex ones, while preserving essential expressive power of the engine.

The Behavior Engine is a module that integrates with the *IDtension* narrative engine for representing actions in real-time 3D environments. It is responsible for grouping simple animations into larger units called behaviors and scheduling the animations in real time. Several systems have been developed to carry out this task (Donikian, 2001; Loyall & Bates, 1991; Mateas & Stern, 2004) but these systems tended to sacrifice usability for descriptive power. Therefore we have opted for an easy to author behavior engine. This method uses the simple and visual notion of a graph to structure a behavior. Note that a similar approach is proposed in (Wages, Grützmacher, & Conrad, 2004), but for managing the entire narrative.

BEcool enables sequencing animations, event-based parallel animations, intra- and inter-character coordination. Therefore it is more powerful than script-based behavior engines. Contrary to other advanced behavior engines such as *HTPS* (Donikian, 2001) or *HAP/ABL* (Mateas & Stern, 2004), it does not use a programming language. However, it lacks some features such as hierarchical organization of behaviors. This illustrates the design compromise that was made.

BEcool has been successfully integrated with both the *IDtension* narrative engine and a 3D game engine (Szilas, Barles, & Kavakli, 2007). However, graphs tended to become slightly complex, and not very easy to design. The next phase is to implement an authoring tool for the engine, and then test with authors in order to both detect authoring difficulties and improve the engine and the authoring tool.

Despite the fact that this engine was designed with the author in mind, the design process still remained linear. The steps were organized sequentially as follows:

1. Writing the functional specifications of the engine
2. Development of the engine.
3. Initial testing
4. Development of an authoring tool.
5. Use of the authoring tool with authors along with the engine to produce a more convincing work.

Authors intervene too late in the process, increasing the risk that the engine might lack usability. Therefore, the approach that will

be developed in the next section will incorporate the author right at the beginning of the design.

3. AUTHORED CENTERED DESIGN OF AN ANIMATION ENGINE

Based on our experience described above, we want to design an animation engine in a more radical authored centered approach. Of course it could be argued that we need to improve our methodology on the same engine instead of switching to a third type of engine. But animation constitutes a bottleneck in Interactive Drama systems, in terms of authoring. Let us take the example of 3D modeling and character animation. Firstly, modelling characters requires highly specialized graphic designers, and sufficient time resources. It is sometimes possible, depending of the 3D format used by the animation and rendering engines, to use pre-existing characters (using for example a game engine such as NeverWinterNight) , but it comes at the expense of the author's expressivity. Secondly, animating 3D characters requires either intensive work from a graphic designer or advanced algorithms for generating movements using procedural animation (Pelachaud & Bilvi, 2003). As a result, designing an Interactive Drama requires a team of authors to yield a finalised work. This situation not only constitutes a practical burden to the development of convincing works but also is incompatible with the very nature of the interactive medium, as will be explained below.

3.1. Interactive sketching

Authoring nonlinear media is fundamentally different. As a procedural art (Murray, 1997), it requires thinking in terms of function rather than data. That is, thinking in a more abstract way (Szilas et al., 2003). Let us make a comparison with other media. When writing a book, a novelist uses his pen and paper or typewriter or word processor to create pieces of his work. His is able to re-read his production instantaneously and to edit the text accordingly. This immediate feedback and correcting loop is essential in the design process. In the case of a movie, such a loop is not available. However, the industry has invented some intermediate products that allow immediate feedback and a correcting loop. In particular authors use the script. A script is a text document particularly used to share the basic narrative properties of the whole movie, in terms of narrative perception. By reading a script, one has a good perception of what the movie will be. The script can be considered analogical to the whole movie.

Other linear narrative media also have such analogical intermediate products: The scenario in theater, the libretto in opera, the sketch in cartoon, etc. Nonlinear media doesn't. There is no intermediate product that the nonlinear media designer can interact with in order to get a good perception of what the final work will be.

We believe that even the videogame industry suffers from this problem, although it is usually not visible from the outside. Most innovative videogames are released after a considerably unpredictable and painful process, not to mention the high percentage of products that are cancelled before being finished. We hypothesize that the deep reason for these difficulties is a lack of vision during the design process of the whole product, which generates a large set of difficulties. To solve this issue, the video

game industry has developed some storyboarding tools, sometimes specific to a given company.

It is interesting to hypothesize that the field of Interactive Drama lacks an intermediate artifact that will be easy to manipulate and interact with and yet would be analogical to full scale interactive pieces. Pen, papers and word processors are not suitable tools because they do not produce interactive products. Existing storyboarding tools for multimedia products, such as INSCAPE (Zagalo, et al., 2006) are dedicated to branching stories, which are fundamentally different from Interactive Drama. Thus a reasonable research goal is to design and develop a tool or a series of tools, which would allow an author of interactive drama to almost immediately implement his/her ideas in order to enter a creative process of trial and error with his/her work. This work could be an intermediate product different from the final work, which would play the same role as the script in the movie industry.

3.2. The design starting point for an animation engine in Interactive Drama

Given the above considerations the engine and authoring tool targeted by this research will enable an author to: create a world in a minute, create a character in a minute, create an animation in a minute, test a world with animated characters, connect to other engines and authoring tools for the higher levels (behaviors, reasoning, narrative, pedagogical, etc.).

These constraints would seem unfeasible to anyone involved in the creation of virtual 3D worlds, because traditionally the priority is the quality of the rendering. By putting this authoring priority first, some other characteristics need to be downsized, considering realism, quality of rendering, range of behaviors, etc. In fact, as will be proposed in the next sections, the resulting engine may not be a 3D engine at the end.

There have been several attempts to make 3D worlds and characters available to a larger public, for example the Virtools commercial product or the Alice project at Carnegie Mellon University (Pausch et al., 1995). While these products choose a reasonable way to solve the issue of authorability, this paper explores a more radical approach, a minimalist approach.

3.3 The different usage of the engine

Before describing the minimalist animation engine itself, it is most important to describe its usage.

The sketchpad: Following the discussion above, the animation engine could be used to foster the creativity of the author. It would be a kind of sketchpad where the author could experiment with his/her ideas and get immediate feedback. The outcome of the engine would consist in authors' innovative ideas for new virtual agent based pieces.

The prototype: prototyping is a common practice in both research and industry involving virtual agents. In research, initial tests and publications are done on limited initial versions of the advanced product. In videogame industry, prototypes are huge and costly, which means that "prototypes of prototypes" are needed. Prototypes are simplified versions of the products that are used to test and demonstrate a product or part of a product. Contrary to the sketchpad usage, the outcome of the creative engine would be used to communicate around the product with other people, but it

would not be reused directly in the final product (or next stage prototype).

The final product: the target of most interactive drama attempts are realistic 3D worlds as largely illustrated in the videogame industry that is itself greatly influenced by the movie industry. This strong influence of old media on new one has already been criticized as a limiting factor for creativity. Furthermore, it raises the issue of believability, with the problem known as “uncanny valley”: virtual characters become odd when their realism reaches a certain threshold while not being fully realistic, especially when dynamic behavior is involved. Our investigation might be the occasion to definitively get rid of the movie model (which of course leads to realism) and start from constraints which are proper to the medium. In other words, the creative animation engine envisioned in this paper could be used to produce final works, quite remote from typical virtual agent-based works. But these works would still be interesting and innovative if this engine enables an author to express his or her creativity. To support this hypothesis, one should recall that before the Renaissance, the activity of drawing was only considered as a tool for sketching future paintings or sculptures. Then it became an art in and of itself.

3.4 Description of the animation engine

3.4.1. Minimalist character modeling

Modeling a 3D character is clearly a process that is not feasible “in a minute” given current standards. To reach the goal mentioned in the previous section, it is necessary to drastically reduce the realism of the characters.

Cartoon based 3D characters, such as those used in the *Victor* project (Aylett et al., 2006) are certainly less costly than realistic characters but still require a considerable production effort (see Figure 1). A more extreme approach has been followed by K. Perlin for demonstrating procedural animation (Perlin). A character is made by articulated geometric shapes (parallelepipeds). Expressivity of the character does not lie in the 3D model itself but in the animation of its subparts. Another interesting direction has been taken in the *Oz* project in Carnegie Mellon University where characters are “Woggles”, simple spheres with facial expressions drawn on them. Further commercial development of the project led to more cartoon-like characters. Finally, the most extreme representation of characters

is... a pixel! Indeed, in “Le pixel blanc” (Schmitt), a single pixel moves inside of a rectangle and seems to be alive thanks to specific dynamics of the pixel, which seem to hesitate, take decision, etc. (see Figure1).

This brief overview shows that there exist a variety of options for representing characters in a non realistic way, which may still provide believable characters. The simplest ones satisfy our constraint for enabling one minute character creation. For example, creating a Woggle would require choosing a name, a color and the two dimensions of the ovoid representing the character. Note however that the cartoon type of facial expression in Woggles is already a design choice that might not suit every author or every piece. Would an author use Woggles for a Shakespeare-like tragedy? Thus the simplification involved in the modeling yields to culturally connotated characters, namely comic characters. Therefore, we suggest starting with a radical approach for character modeling, that we denote “symbolic virtual characters”. Screen representations on the screen do not share any physical analogies with the characters they represent. In a 3D world, a character could be represented by a single non articulated shape (a sphere, a stick, etc.), with no facial emotion on it. An author, while designing a character, would propose quite neutral shapes such as a vertical cylinder or a pyramid. The authoring tool would enable simple customization of these shapes to add some connotation and analogical meaning (the “fatty boy” would be a sphere, while the “nerd” a thin pyramid, etc.) but only as an option. All characters could have the same basic shape and their meaning would come only from what happens to them.

From a semiotic point of view, such characters are symbols because they arbitrarily refer to a character, like the word “apple” refers to a certain fruit. But contrary to a book, this symbol is spatially located, like a point in radar that represents a plane.

A more iconic representation of a character could also be used, such as changing the shapes for each character (see above) or adding icons or pictures to character description. It would be interesting to explore these variants as long as they satisfy the three following constraints: 1) Authoring work is minimal. 2) The creative dimension is not suppressed by imposing a limited set of expressive options, like choosing among a limited set of pictures drawn by someone else. 3) Realism is maintained low enough so that characters do not seem badly drawn but rather come across as signs suggesting characters (the determination of such a threshold would require further research on believability).

3.4.2. Minimalist animation

Given the simplistic models proposed in the previous section, the animation can be drastically reduced. Indeed, characters modeled with a single geometric shape do not need any limb animation and coordination. Animation is then reduced to the displacement of characters from one place to the other. Given this reduction, it is now interesting to think of how authors would still be able to express themselves. The following propositions can be suggested.

First, in order to express what is related to limb movements, the author can mostly use text. For example, if a character is waving at another, a bubble displaying “waving towards X” will be attached to the character; with X being the character towards whom the waving is addressed. If the characters’ shapes are not volumes of revolution (Cylinder, sphere, etc.), the orientation of the characters can also be used. In our example, the character

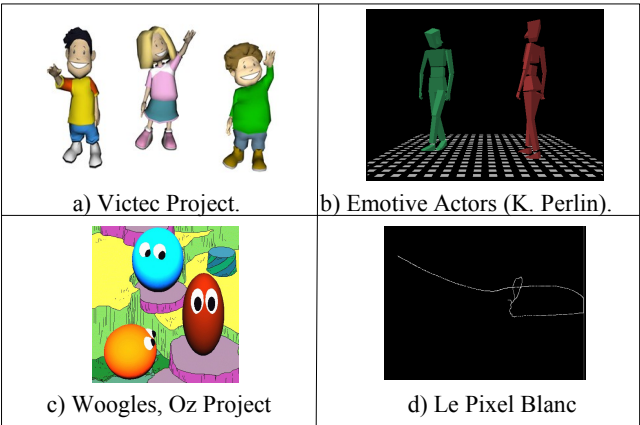


Figure 1. Four examples of non realistic character rendering.

would be oriented towards X. Another parameter that can be easily authored is the distance between characters or objects involved in the animation. For a “kissing” or “fighting” animation, the distance would be minimal, while for a “greeting” or “laughing at” animation, the distance would be larger.

Second, because displacement in space is automated in 3D virtual worlds, using path planning algorithms, there is usually a limited authoring control at this level. However there exists a feature that is not commonly used in virtual worlds: the velocity profile of the character on the trajectory. In a standard walk, velocity quickly reaches its maximum value and quickly returns to zero when arriving at destination. In addition to the choice of this max value (the character’s speed), other velocity curves or profiles could be defined by an author, similar to the profiles used in the Adobe Flash animation software. These profiles define the velocity of the character on the trajectory according to the time in the animation. The profile could be selected by choosing from a list and modifying it directly. An oscillating curve could for example express a hesitating character, while a saw tooth curve could express a sneaky character.

To sum up, an author would express himself in animation by simple bubble’s texts and a few parameters such as:

- orientation during animation
- inter-characters distances
- velocity profiles

What is interesting here is that by starting with a strong constraint on authoring, we came to some interesting expressive features that are not used with 3D visualization. This fully illustrates that starting with strong authoring constraints does not necessarily limit the expressive power of the medium.

Note that these parameters are examples only. More valuable parameters could be added, as long as they provide expressive power to an author, without adding significant authoring effort.

3.4.3 The virtual space

Following our process of authoring simplification, it is natural to propose that virtual worlds should be kept very simple in order to be easily authored. Simply textured horizontal and vertical rectangles should be enough to represent floors, roofs and walls. The author could for example draw the 2D map of a room, which would be automatically converted into a 3D room. Contrary to character modeling, simple space modeling is feasible, as demonstrated by level modelers in game engines or the Google SketchUp program. Furthermore, this approach is in line with some forms of theatrical representations where the set is kept minimalist.

Having drastically reduced modeling and animation, we should take into consideration the question of the need for a three dimensional space in such virtual worlds. Characters, objects and the environment are only symbols, and therefore could be 2D symbols as well. Therefore, one version of the system could be developed as a 2D engine. Characters would simply be seen in a top view, similar to a map in a 3D videogame. Of course, some 3D specific spatial configurations could not be used, such as a character walking on a bridge over another road. It would be rich enough however to represent the majority of events in a usual drama. The choice of a 2D or a 3D world should be motivated by the predominance of 3D specific space configurations that justify

the use of 3D. 2D environments are always easier to use in practice, consider for example, the difference in time required for launching a 2D or 3D engine. This might seem insignificant from a research perspective but it can be critical in the creative process of iteratively designing an interactive piece.

3.5 Use case description

In order to better illustrate our approach and prepare the detailed design phase of the project, a use case is described. A use case is a now popular technique in software engineering for defining the functionalities of software. It consists in describing the sequence of actions that a user of the to-be-developed system would undertake to achieve a certain goal. Let us thus consider the situation of an author creating a scene with five characters in a dance room. These characters can sit, stand, walk, dance, and talk. There are chairs in the room as well. Note that the current paper is limited to the animations, which does not include behaviors. In this use case, a 2D variant of the engine is considered, without audio.

The initial design of the environment would require the following steps:

- World Design: create a new world; enter its description (for designers only – optional); give it a screen name; give X and Y dimensions - we assume that worlds contain a single room with rectangular shape.
- Character Design: create a new character; enter its description (for designers only – optional); give it an internal name: ricardo; give it a screen name: Ricardo; load an icon for it; give it X and Y coordinates for the initial position: 10, 10; give it initial orientation: 0. Character design is repeated for each of the four other characters.
- Animation Design: create a new animation; give it an internal name: danceAlone; select a type of animation: gesture; define additional parameters (other than the actor): none; give it a textual expression: dancing; Select duration: 30 sec.; draw an orientation profile: a non regular oscillating curve.

The last parameters depend on the type of animation, gesture in this case. Other types of animations are: displacement and speech. For a displacement, the steps are: create a new animation; give it an internal name: shyWalkTo; select a type of animation: displacement; define additional parameters (other than the actor): character \$target; give it a textual expression: walking to \$target; select a velocity profile: curve with a low level slightly oscillating plateau; select an orientation: towards \$target; choose final distance: 1m. Other animations are created in the same way.

In a testing mode, commands to the engine can be sent manually. In this use case, we use a predicative form for the commands, which are freely typed by the user. This might not be the most ergonomic way to send commands; variants of the system could also use a syntactically constrained interface, as in software such as *Agentsheet* or *Alice* (Pausch et al., 1995). For each command that follows, the corresponding events occurring on the screen are described:

- shyWalkTo(Ricardo, Cristina): the icon of Ricardo moves towards the icon of Cristina, with a slightly oscillating speed, giving the impression that the character is not sure of himself. A small bubble follows the icon, with the text “walking to Cristina”.

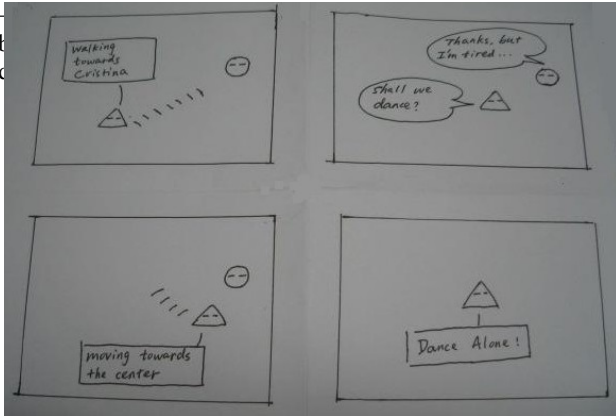


Figure 2. Artistic view of the sequence of animation between two characters (see text). The dash lines are not represented on the screen but used here to represent the movement. Note that the two bubbles represented at the top right are not necessarily represented simultaneously on the screen.

- talkTo(Cristina, Ricardo, "Thanks you, but I am tired..."): similarly, a speech bubble associated to Cristina's icon displays the text during a few seconds.
- yawn(Cristina): a bubble displays the text "yawning" during 2 seconds.

- sadWalkTo(Ricardo, RoomCenter): Ricardo's icon moves towards the centre of the room, with a slow movement.
- danceAlone(Ricardo): a bubble displays the text "dancing". during 30 seconds.

These commands are presented in a sequence but the user does not have to wait for an animation to finish before sending another command. Furthermore, any animation can be interrupted, which erases any bubble associated to the animation and stops the displacement of the icon, if any.

4. AN AUTHORIZING ARCHITECTURE

The process that has been described for the animation engine should be extended to other modules involved in Interactive Drama. Each engine that constitutes an Interactive Drama system (narrative engine, text generation, behavior engine, animation engine, etc.) should follow the same approach: The design of the authoring tool should occur at the same time as the design of the corresponding engine itself. Following this approach, the authoring tools are not components that are added afterwards to improve the usability of already existing engines. The authoring tools are constitutive of the Interactive Drama system.

Furthermore, an additional function must be explicitly incorporated into the system to ease the "interactive sketchpad" approach described earlier: monitoring. Monitoring consists in controlling the execution of one or several modules in the system in real-time, as the author/researcher/programmer, not as an end user who "plays" with the work.

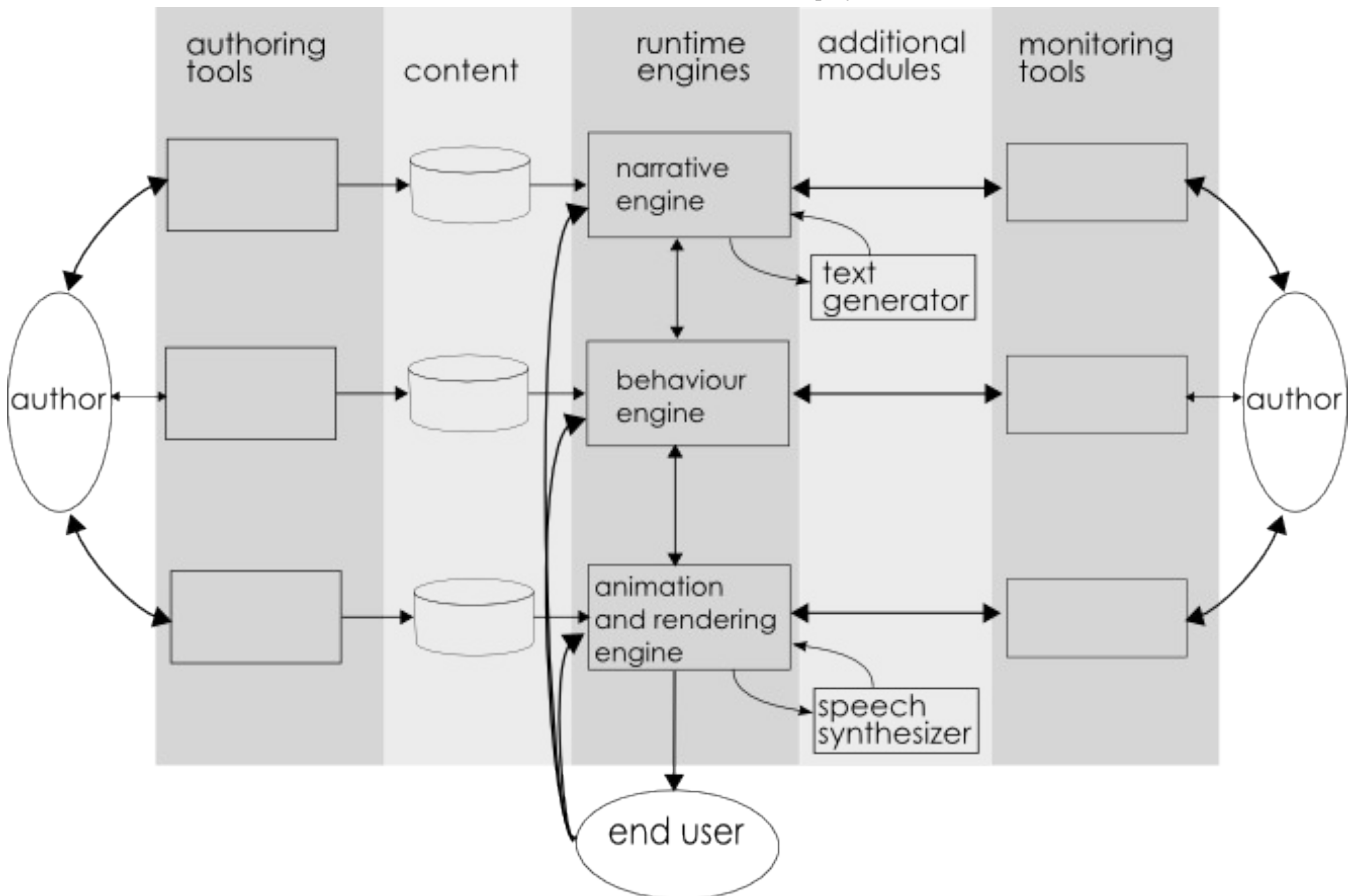


Figure 3. General architecture of an author-centred Interactive Drama system.

An architecture that includes all levels of Interactive Drama (narrative, behaviours, animation) as well as all functions (execution, authoring, monitoring) is depicted in Figure 3.

The main types of modules are the following:

1. Runtime engines: with additional modules they ensure the execution of the digital story for the end user. They are structured hierarchically, following a convenient hierarchical decomposition of story management.
2. Additional modules: they work as sub-modules of the main modules, as they process some data for a main module. The speech synthesis module is optional.
3. Content: all information describing one specific story is stored here. It includes higher level narrative content (e.g. characters' goals, possible actions, etc.), behavior specification (decomposition of high level actions into smaller animations) and characters' animations and models.
4. Authoring tools: they are used to enter content into the system, with author-oriented Graphical User Interfaces. The architecture will be designed such that, whenever possible, content can be added/modified during execution.
5. Monitoring tools: they serve two main functions. First they allow viewing « behind the scene » of a given execution, that is, visualizing in the most readable way the various variables involved in the calculation of the computer's actions. The choice of which variables are visible and which ones are not is essential at this level (Szilas, 2006a). Second, the monitoring can consist in sending information to the different modules in order to test certain behaviours. Typically, the monitoring tools should enable a user to input an action to one of the modules, shortcutting another module normally responsible for this input in the architecture.

All these modules will exchange information as messages written in XML. XML provides an intermediate level of readability that is not suitable for authoring but still enables an advanced user to debug problems, if needed.

5. CONCLUSION

We have suggested in this paper a design approach of an Interactive Drama system that is primarily guided by authorability. Applied to the animation level, this approach generates interesting alternative features that serve authors' expressivity and has inspired the design of a global Interactive Drama architecture that includes authoring and monitoring facilities.

Not focused on the synthesis of always more realistic behaviors, the minimalism adopted in this paper is in contrast with the main research trend in animation. We are not arguing that minimalist animation is better but that in the context of scientific and artistic research on Interactive Drama, in which the focus is on the higher narrative level, minimalist animation is certainly a necessary step. Paradoxically, minimalism might be a huge step in the field, opening the way to more medium-specific artistic creation.

In order to take this step, the development of the minimalist animation engine will be undertaken in conjunction with the development of the architecture. The process will be driven by three "target applications", to ensure that authoring constraints from authors themselves guide the design of the different modules in the architecture.

6. REFERENCES

- [1] Aylett, R., Louchart, S., Dias, J., Paiva, A., Vala, M., Woods, S., et al. 2006. Unscripted narrative for affectively driven characters. *IEEE Computer Graphics and Applications*, 26(3), 42-52.
- [2] Bremond, C. 1973. *Logique du récit*. Paris: Seuil.
- [3] Crawford, C. 1996. Is interactivity inimical to storytelling. from www.erasmatazz.com/library/Lilan/inimical.html
- [4] Crawford, C. 1999. *Assumptions underlying the Erasmatron interactive storytelling engine*. Paper presented at the AAAI Fall Symposium on Narrative Intelligence, Technical Report FS-99-01.
- [5] Donikian, S. 2001. *HPTS: a behaviour modelling language for autonomous agents*. In Proceedings of the fifth international conference on Autonomous agents (Montreal, Canada, May 2001).
- [6] Greimas, A. J. 1983. *Du sens*. Paris: Seuil.
- [7] Louchart, S., & Aylett, R. 2003. *Solving the Narrative Paradox in VEs-Lessons from RPGs*. Paper presented at the Intelligent Virtual Agents: 4th International Workshop, IVA 2003.
- [8] Loyall, A. B., & Bates, J. 1991. *Hap: A Reactive, Adaptive Architecture for Agents* (No. CMU-CS-91-147). Pittsburgh, PA: School of Computer Science, Carnegie Mellon University. Document Number)
- [9] Magerko, B. 2002. *A Proposal for an Interactive Drama Architecture*. Paper presented at the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment.
- [10] Mateas, M., & Stern, A. 2000. *Towards Integrating Plot and Character for Interactive Drama*. Paper presented at the Working notes of the Social Intelligent Agents: The Human in the Loop Symposium.
- [11] Mateas, M., & Stern, A. 2004. A Behavior Language: Joint action and behavioral idioms. In H. Prendinger & M. Ishizuka (Eds.), *Life-like Characters. Tools, Affective Functions and Applications*: Springer Verlag.
- [12] Mateas, M., & Stern, A. 2005a. *Procedural Authorship: A Case-Study Of the Interactive Drama Façade*. Paper presented at the Digital Arts and Culture (DAC).
- [13] Mateas, M., & Stern, A. 2005b. *Structuring content in the Façade interactive drama architecture*. Paper presented at the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005).
- [14] Murray, J. 1997. *Hamlet on the Holodeck. The future of narrative in the cyberspace*. New York: Free Press.
- [15] Pausch, R., Burnette, T., Capeheart, A. C., Conway, M., Cosgrove, D., DeLine, R., et al. 1995. Alice: Rapid prototyping system for virtual reality. *IEEE Computer Graphics and Applications*, 15(3), 8-11.
- [16] Pelachaud, C., & Bilvi, M. 2003. Computational Model of Believable Conversational Agents. In M.-P. Huguet (Ed.), *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies, LNCS 2650* (pp. 300-317). Berlin/Heidelberg: Springer Verlag.

- [17] Perlin, K. Emotive Virtual Actors. 2008, from <http://mrl.nyu.edu/~perlin/experiments/emotive-actors/>
- [18] Portugal, J.-N. 2000. La fiction interactive: contenu cardinal des médias digitaux?, *Les Dossiers de l'Audiovisuel*, 92, 9-11.
- [19] Portugal, J.-N. 2006. L'avenir dira ce que nous avons créé. In N. Szilas & J.-H. Réty (Eds.), *Création de récits pour la fiction interactive* (pp. 133-165). Paris: Lavoisier.
- [20] Schmitt, A. Le Pixel Blanc. 2008, from <http://www.grafin.org/as/texts/lepixelblanc.html>
- [21] Skorupski, J., Jayapalan, L., Marquez, S., & Mateas, M. 2007. *Wide Ruled: A Friendly Interface to Author-Goal Based Story Generation*. Paper presented at the Proceedings of the Fourth International Conference on Virtual Storytelling - ICVS 2003, Lecture Notes in Computer Science.
- [22] Smith, S., & Bates, J. 1989. *Towards a Theory of Narrative for Interactive Fiction* (No. CMU-CS-89-121). Pittsburgh, PA: Department of Computer Science, Carnegie-Mellon University.
- [23] Spierling, U. 2007. *Adding Aspects of "Implicit Creation" to the Authoring Process in Interactive Storytelling*. Paper presented at the Proceedings of the Fourth International Conference on Virtual Storytelling - ICVS 2003, Lecture Notes in Computer Science.
- [24] Spierling, U., Grasbon, D., Braun, N., & Iurgel, I. 2002. Setting the scene: playing digital director in interactive storytelling and creation. *Computers & Graphics*, 26(1), 31-44.
- [25] Spierling, U., & Iurgel, I. 2003. 'Just Talking About Art' - *Creating Virtual Storytelling Experiences in Mixed Reality*. Paper presented at the Proceedings of the Second International Conference on Virtual Storytelling - ICVS 2003, Lecture Notes in Computer Science.
- [26] Szilas, N. 1999. *Interactive Drama on Computer: Beyond Linear Narrative*. Paper presented at the AAAI 1999 Fall Symposium on Narrative Intelligence, Technical Report FS-01-02.
- [27] Szilas, N. 2001. *A New Approach to Interactive Drama: From Intelligent Characters to an Intelligent Virtual Narrator*. Paper presented at the Spring Symposium on Artificial Intelligence and Interactive Entertainment.
- [28] Szilas, N. 2007. A Computational Model of an Intelligent Narrator for Interactive Narratives. *Applied Artificial Intelligence*, 21(8), 753-801.
- [29] Szilas, N., Barles, J., & Kavakli, M. 2007. An implementation of real-time 3D interactive drama. *Computers in Entertainment (CIE)*, 5(1).
- [30] Szilas, N., Marty, O., & Réty, J.-H. 2003. *Authoring Highly Generative Interactive Drama*. Paper presented at the Virtual Storytelling: Using Virtual Reality Technologies for Storytelling: Second International Conference, ICVS 2003.
- [31] Todorov, T. 1970. Les transformations narratives. *Poétiques*, 3, 322-333.
- [32] Wages, R., Grützmacher, B., & Conrad, S. 2004. *Learning from the Movie Industry: Adapting Production Processes for Storytelling in VR*. Paper presented at the Technologies for Interactive Digital Storytelling and Entertainment: Second International Conference, TIDSE.
- [33] Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R. J., & Saretto, C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1(1), 51-70.
- [34] Zagalo, N., Göbel, S., Torres, A., Malkewitz, R., Branco, V. 2006. INScape: Emotion Expression and Experience in an Authoring Environment. TIDSE 2006: 219-230.