The SCD Architecture and its Use in the Design of Story-Driven Interactive Spaces

Claudio S. Pinhanez

MIT Media Laboratory, 20 Ames Street
Cambridge, Massachusetts, MA 02139, USA
pinhanez@media.mit.edu

Abstract. In this paper we examine story-driven interactive spaces and argue that story development requires centralized control of the actions of the characters that inhabit such environments. This argument leads us to propose the *story-character-device (SCD)* architecture that, unlike previous approaches, separates the locus of story control in a distinct software module. To exemplify the use of the architecture we examine an art installation (called *"It"*) that we built according to the SCD model. The development process of *"It"* is then compared to the design of another story-driven interactive space that used similar technology and thematic, the computer theater play *"It / I"*. By looking into the design process of the two environments, we conclude that the SCD architecture allows much more flexibility in the design process of story lines besides bringing an overall simplification of the control system.

1 Introduction

There has been an increasing interest in creating interactive spaces for entertainment with a strong narrative structure or with an underlying story [4, 10, 16, 24]. These systems have employed many different control architectures, each of them with several shortcomings. The goal of this paper is to analyze the fundamental characteristics of story-based interactive spaces and how their characteristics influence the design of their architecture.

From our experience in developing story-driven interactive systems for entertainment (*"The KidsRoom"* [4], *"SingSong"* [22], and *"It / I"* [21]) we have concluded that centralized story control is essential to achieve successful story development. Such a claim is clearly contentious. Perlin and Goldberg [18] as well as Bates et al. [2] built (semi-) autonomous computer-actors used in systems where the story was distributed among characters or seen as the natural result of the interaction between the characters and the user. However, as noted by Langer [12] and Murray [15], well constructed stories require coordination and synchronicity of events and coincidences which we believe can only be achieved by centralized story control.

In this paper we propose a three-level architecture for story-driven interactive systems called *story-character-device* architecture, or *SCD*. The defining characteristic of the SCD architecture is the separation between character and story control. In the SCD architecture the characters are semi-autonomous, receiving commands from the story control module although possessing the ability to sense the on-going action and to coordinate the accomplishment of their goals according to the reality of the story world.

To exemplify the use of the SCD model we examine *"It"*, an interactive story-driven interactive space that recreates from a first-person point of view the story of the computer theater play *"It / I"* [21]. The original play featured an automatic character on stage interacting with a human actor through computer graphics, sound, and light, according to information gathered by cameras. In other words, the stage was transformed into an interactive space. *"It / I"* was originally produced using an almost flat system architecture and therefore, since *"It"* basically portrays the same story, we can compare the design and development process without and with the use of the SCD architecture.

2 Interactive Spaces

In this paper we adopt the term *interactive space* to refer to situations where human beings interact with a computer in a physical space through digital input/output devices such as cameras, microphones, video screens, and speakers. The classic example of an interactive space is the control room of spaceships as portrayed in science-fiction series and movies like *"Star Trek"* and *"2001:A Space Odyssey"*. In these two cases the computer that controls the spaceship is omnipresent, interacting with the ship's crew as an invisible, God-like figure that can either access the most vital data or entertain the occupants with a game of chess.

2.1 Characteristics of Interactive Spaces

Let us examine some common characteristics of an interactive space. First, it is a **physical space**. For instance, we do not consider either virtual reality or videogames as interactive spaces in our definition. In other words, we are interested in systems that are required to assume the existence of atoms, their physical interaction, and the difficulties in sensing real matter (as opposed to checking the state of virtual entities).

We also assume the existence of a **computer system** controlling the interaction, discarding human-controlled environments like a disco club (where the DJ manages the interaction between the patrons and the dance area). Moreover, interactive spaces must be **responsive**, that is, they must acknowledge the attempt of interaction by the users either by answering directly or by producing a detectable transformation in the space.

Another aspect is that agency should be attributed to the space itself and not to particular objects in the space. Perhaps the most characteristic feature of an interactive space is precisely this **omnipresence**. Notice that in the case of entertainment spaces inhabited by virtual or physically represented characters, the agency of the characters has clear locus, but the space must still be the non-localized source of the story, game, or narrative.

Part of the appeal of the idea of interactive spaces is exactly that, perhaps for the first time in history, we can see a space as dialoguing entity. Computers and sensing devices create spaces that are **machines**, that is, perceived as animated and active objects.

2.2 User-Driven Interactive Spaces

We employ the term *user-driven* to characterize interactive spaces where the user directs the direction and goal of the overall interaction. This category, in fact, comprises the majority of the work done in terms of creating actual interactive spaces. The pioneer of the idea is Mark Weiser who proposed the concept of *ubiquitous computing* about a decade ago [25]. In its simplest form, ubiquitous computing advocates the disappearance of the visible entity associated to computational power and its spread through the space and everyday objects.

Alex Pentland has promoted the idea of *Smart Rooms* as a domain of perceptual computing [17]. The most successful result was the *ALIVE* experience [14] where the user could interact with a CG-generated dog by watching herself in a *virtual mirror* — a large video screen showing a mirror-like image of the room with the user and superimposed computer graphics objects. Another example is the *Intelligent Room*, being developed at the MIT AI Laboratory [7], aiming to create a room for crisis management that interacts with its occupants and provides them critical information. It incorporates basic camera-based tracking, gesture detection, and reacts chiefly by answering verbal commands. The review of the innumerous other projects involving user-driven interactive spaces is beyond the scope of this paper.

2.3 Story-Driven Interactive Spaces

Narrative and story structures are very common in computer games. However, the number of physically interactive spaces with such elements is very restricted, although the interest in the area has considerably increased in the last years. In this paper, we use the term *story-driven* interactive spaces to refer to spaces that immerse the users in a story that develops partially in response to the users' actions and partially as a consequence of a narrative of events pre-determined by the designers of the space.

Interactive stories and narratives is still a genre to be born, as discussed by Janet Murray in [15]. Murray observes that there are particular narrative genres that seem to be more suitable for interaction such as journey narratives. In particular, Murray studies multi-threaded stories where the user has mechanisms to choose different paths in a web of events, creating an individual, personal story as a result of the interaction.

In the realm of interactive spaces the first experiments were created by Myron Krueger [11], although most of his "narrative" installations portrayed very simple stories. Larry Friedman and Glorianna Davemport's *Wheel of Life* was one of the first works to include complex and rich narrative elements in an interactive space [8]. Naoko Tosa and Ryohei Nakatsu have created two interactive pieces with strong narrative structure [16, 24]. In the *Interactive Poem*, the user dialogues with a stylized female face projected on a large computer screen [24]. In *Romeo and Juliet in Hades*, two users take the roles of Romeo and Juliet after their death and, through speech and gesture interaction, follow their journey towards rediscovering who they were and their love [16].

The work of Aaron Bobick at the MIT Media Laboratory has pushed the sensing envelope of interactive spaces in the recent years. In "*The KidsRoom*" [4], a children bedroom-like space takes a group of children through an adventure covering four different worlds inhabited by friendly monsters. The interaction is very physical, involving running, dancing, jumping, and rowing, completely based on information gathered by cameras.

Our experiments in computer theater (see [19] for a definition of the term), *"SingSong"* [22] and *"It / I"* [21], are also very compelling examples of the possibilities of story-driven interactive spaces. They also differ from most of the examples mentioned above in the emphasis in responsiveness as the key element of creating immersion in a story, instead of the more commonly used device of choice among different story paths. Although choosing the path of a story considerably empowers the user, it is very difficult to assure that all of the paths lead to equally satisfying experiences. In contrast, we have created powerful stories where the user is coerced, through mechanisms not explicitly visible, to remain inside the main path, while carefully designed interaction and good responsiveness keep the illusion that the story is in fact unfolding in response to the user's actions.

3 Centralized vs. Decentralized Control of Story

Most interactive spaces and virtual reality experiences (for instance, [13]) are based on the concept of *exploration*. That is, the user enters a world populated by characters, objects, and other users, and extracts most of its satisfaction from the encounter with the new. There is no narrative structure unfolding during the experience and therefore no need for story representation or control.

The existence of a story in an interactive system requires the management of multiple characters and events in orchestrated ways. A good image of this distinction is to observe that while exploratory worlds have *creatures* living in them, story-based interaction requires *actors.* Actors know that a story must start, develop, reach a climax, and finish.

An important question is where the story is represented and how it is controlled. For example, in most story-based interactive systems created until now, like Perlin and Goldberg [18], Bates et al. [2], and Blumberg and Galyean [3], the story is carried by (semi-) autonomous computer-actors with partial knowledge of the story contents and development. The story is seen as the natural result of the interaction between the characters, the user, and the story traces in the character's brains.

We believe that centralized story control is fundamental for an interactive system to achieve successful story development. As pointed by Langer [12] and Murray [15], well-constructed stories require coordination and synchronicity of events and coincidences. However, the coordination required to achieve coincidence is, in our view, only possible with centralized story control. Also, as brilliantly pointed by Langer in her study of theater ([12], chapter 17), it is essential for dramatic structure to **forecast the future**:

> *"In actual life we usually recognize a distinct situation only when it has reached, or nearly reached, a crisis; but in the theater we see the whole setup of human relationships and conflicting interests long before any abnormal event has occurred…(…) This creates the peculiar tension between the given present and its yet unrealized consequent, "form in suspense", the essential dramatic illusion." (Suzanne Langer,[12] pg. 311).*

If we agree with Langer, it is necessary for an interactive system with dramatic structure to have, in some form, the ability to look ahead and sketch in the present the conflict of the future. To leave this job for each separate character and to expect that they will all come up with the same story structure is pure nonsense.

4 The Story-Character-Device (SCD) Architecture

Interactive spaces are normally complex structures involving multiple inputs, outputs, and control sources. In the interactive space *"It"* described in this paper we have used a novel software architecture called the *story-character-device* architecture, or simply, the *SCD* architecture. Figure 1 shows the basic elements of the SCD architecture that is composed of five levels. The *world level* corresponds to the actual sensors and output generators of the system. The *device level* corresponds to low-level hardware and software to track and recognize gestures and speech from users, and to control the output devices and applications (including low-level control of the movement of computer graphics characters). The *character level* contains software modules that control the actions of the main characters — including here the users and the virtual crew (e.g. cameraman, editor, and light designer) — by considering both the story goals and constraints and the actual events in the real space. The *story level* is responsible for coordinating characters and environment factors, aiming to produce the interactive structure contained in an *interaction script*. Finally, it is possible to have a *meta-story level* that dynamically changes the story as a result of the on-going interaction; such a module should encompass knowledge about how stories are structured and told [5].

The fundamental distinction with previous architecture models [2, 18] is the existence of a story level separated from the character level. As discussed above, we do not believe that story development is achievable simply by character interaction but rather that central control is necessary. On the other hand, preserving the distinction between story and characters simplifies considerably the management of the story. Notice that in the SCD architecture the characters are semi-autonomous since they receive commands from the story control module. However, they coordinate the accomplishment of their goals (as set by the story module) according to the actions happening in the interactive space.

The best analogy to the role of the *story level* in the SCD architecture is the old theatrical figure of a *whisperer* or *prompter*, the person who used to sit underneath a small canopy in the front of stage, facing the actors, and whispering them the next lines to be said. In this model, the actors are responsible for determining **how** the lines are uttered and for making up the accompanying action; the whisperer's job is to assure that story proceeds according to the script, that is, to determine **which** and **when** the lines are said. In the SCD architecture, the story level module watches the user actions (regarding the user as just another character) and tries to adjust the story to match his actions, according to a script that describes how the interaction is supposed to happen. If the meta-story level is present, the script itself can change — as if the printed text in front of an old-timer prompter suddenly shuffled its words into a new plot.

Although the system of our computer theater play *"It / I"* [21] was designed with the concept of SCD architectures in mind (the idea predates the project), our first

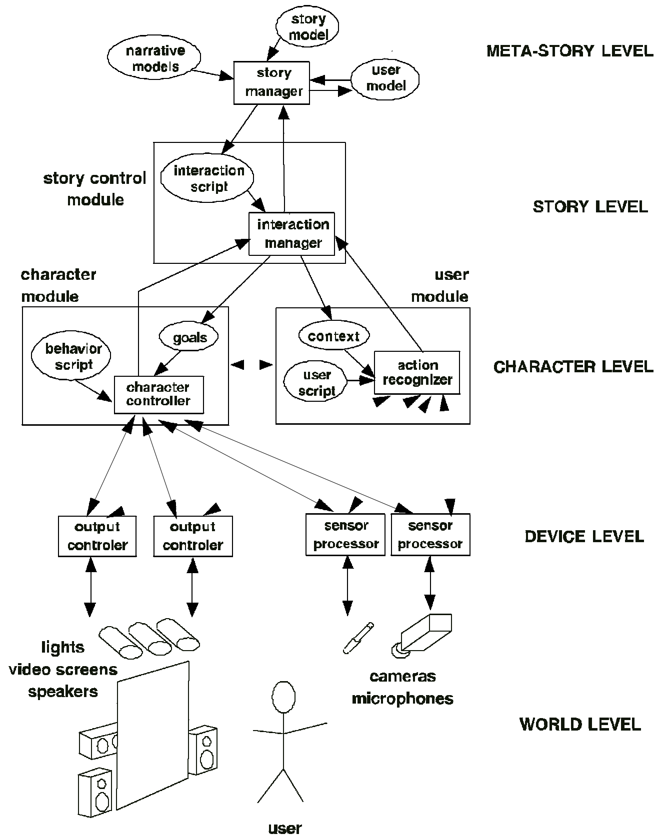

Figure 1. The SCD architecture for interactive story-based systems.

work that fully followed the model is the art installation *"It"* described in the next section. Since the story behind *"It"* is basically the same as the story in *"It / I"*, we can compare the impact of the different architectures in the process of building the systems.

Like most of the work in software architecture, it is hard to characterize the success and even appropriateness of a particular model. We do not claim that SCD is the right or best architecture for a story-based interactive system. However, the comparative experience of building *"It"* and *"It / I"* shows that the SCD model addresses issues that have been neglected by most of previous proposals.

5 *"It"*: An Interactive Space based on the SCD Architecture

In November of 1997 we premiered the computer theater play *"It / I"* at the MIT Media Laboratory, portraying the story of a human character, called ***I***, who was taunted and played by an autonomous computer-graphics character, ***It***. After the performances the audience was invited to go up on stage and re-enact the play. This

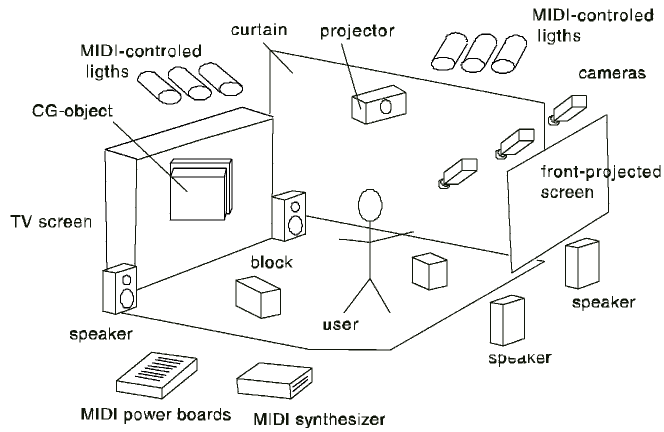

Figure 2. Physical setup of *"It"*.

was possible because the stage and the computer character were automatically controlled and therefore the story of the play could be repeated as many times as needed, for each individual member of the audience.

However, we were not totally pleased with the final result of the audience interaction with the computerized story. First we felt that it was necessary to develop further the automatic control system to allow full dramatic immersion of a non-actor. Second, the open stage proved to be a hard place for a member of the audience to fully experience the story of the play, mostly due to the discomfort of being watched.

To overcome these problems we decided to re-create the environment of the play *"It / I"* in a version for users called *"It"*. Inhabited by the ***It*** character, the space tries to trap the user inside it, under the disguise of a game of taking and showing pictures. *"It"* uses the same visual (images, computer graphics) and sound elements as *"It / I"*, but it was designed to be a self contained, stand-alone piece that can be enjoyed by users completely unfamiliar with the play.

Figure 2 shows the basic physical setup of *"It"*. It consists of two facing screens inside an enclosed, dark room. Three cameras, in a stereo vision configuration, monitor the space detecting the user's presence and position. The installation only admits one user each time. Two independent sets of audio speakers are associated to each screen, reproducing the sound generated by a MIDI synthesizer. Theatrical fixtures controlled by MIDI power devices illuminate the space.

The theme of *"It"* is the same of *"It / I"*, the entrapment of people by technology. However, in *"It"*, we wanted the user to be actually attracted by narrative devices, seduced to play with the machine, gradually start to feel uncomfortable about the situation, and suddenly discover that she has no way to escape. The interactive story is summarized by the following script.

> *When there is nobody in the space,* ***It*** *(the computer character that inhabits the space) is dormant: green lights, soft machine sound in the background. When* ***I*** *(an individual user) enters the space,* ***It*** *wakes up, switching the lights on and stopping the music.* ***It*** *then brings a camera-like object to the right screen. If at any moment* ***I*** *leaves the space,* ***It*** *tries to bring her back*

*by playing loud sound and fast switching color lights. The camera on the right screen follows **I** around the space and when she stops moving, zooms in and "takes a picture". Immediately after, **It** moves a TV-like object to the left screen displaying the silhouette of the picture taken by the camera. This silhouette is shown for some moments. The goal of **It** is to involve **I** in this game of taking and seeing pictures in a crescendo frenzy of action. Each iteration of the taking-viewing cycle is shorter than the previous. After the game reaches a very fast pace, **It** looses interest in **I**, removes the camera and the TV from the screens, and starts trying to push **I** out of the space by flickering the lights and simulating on the screen the action of throwing blocks into **I**'s direction. When **I** leaves the space, **It** goes back to the dormant state, ready for the next victim.*

5.1 The Technology

Most of the technical structure of *"It"* is based on the technology built for *"It / I"* (more details in [20]). To simplify the understanding of the final architecture of *"It"*, we describe briefly the different modules that compose the installation in the following paragraphs.

The vision system module basically tracks the position of the user. In the setup we employed a frontal 3-camera stereo system able to segment the user and to compute the silhouette image that is used to track the position. The stereo system, based on the work of Ivanov et al. [9], constructs off-line a depth map of the background — floor, curtains, and screens. Based on the depth map, it is possible to determine in real-time whether a pixel in the central camera image belongs to the background or to the foreground, in spite of lighting or background screen changes.

The computer graphics modules control the generation and movement of the different objects that appear on the two screens. All the sounds and music in *"It"* are produced by a Korg X5DR MIDI-synthesizer. To control the production of the sounds, the sound module sends the MIDI events corresponding to the different MIDI files associated to each sound and piece of music. To control the lights we employed two 4-channel TOPAZ MIDI-controlled power bricks in a daisy configuration, providing 8 independent dimmer channels. The different channels are controlled by the light module through MIDI signals.

To implement the *"It"* installation we employed a scripting language for interactive spaces called *interval scripts* [20, 22] that is based on the description of the interaction by encapsulating the individual actions in units called *intervals*. Basically, an interval contains descriptions of how to activate and stop the actions of the characters. To describe the narrative flow, the designer can both use procedural methods (like in *Lingo* [1]) but also he can include constraints about what should happen or what can not occur.

5.2 The Architecture of *"It"*

One of the reasons for building *"It"* was to experiment with the idea of implementing an interactive space using the SCD architecture. Figure 3 diagrams the architecture of *"It"*. The device level contains the software modules responsible for

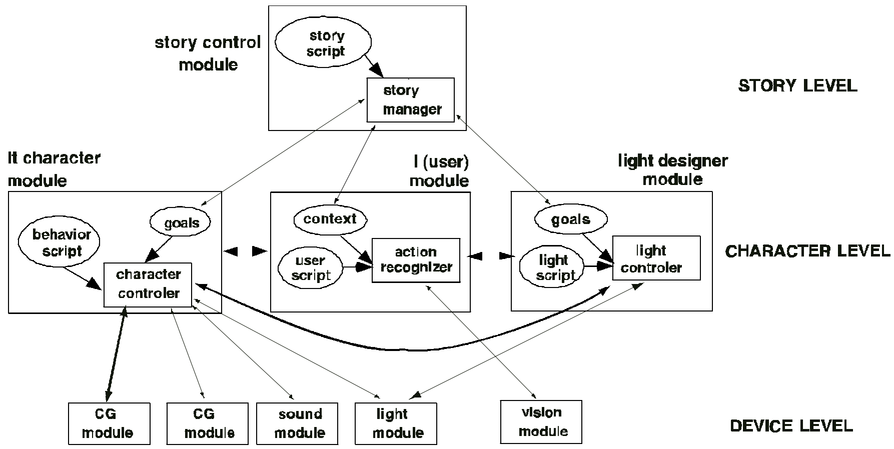

Figure 3. System architecture of *"It"*.

the direct control of the computer graphics, sound, lights, and vision system. The character level implements three modules: one corresponding to the ***It*** character, one for the user (or ***I*** character), and a third that corresponds to a "virtual" crewmember, a light designer.

There are two reasons for singling out light control at the character level. First, unlike other actuators such as the computer graphics modules, light is controlled both by the character ***It*** and by the top-level story module. Second, the user module changes its behavior according to the light conditions in the space. For instance, when strong, dark color lighting is used, the positional data becomes unreliable. We found it conceptually easier to have the user module sending a single query just to the light designer module, asking whether the light conditions are normal, instead of checking every dimmer of the low-level dimmer control module.

The basic distinction between the story-level module and the character and crew modules is that the latter contain a great deal of information about how to perform an action while the former is mostly concerned about when and why to do it. For example, the story module has to decide when it is the time for the ***It*** character module to attract ***I***'s attention, when to play ***I*** with pictures, and when to expel him from the space. In this last situation, a request for expelling ***I*** from the space is translated by the ***It*** module into a request to the light designer module to flash the lights from red to white, continuously, and into a sequence of commands to the computer graphics modules to generate CG-explosions of objects on the screens.

5.3 The Experience

We finished the construction of the installation *"It"* in March 1999 in a laboratory space at the MIT Media Laboratory. Since then there have been dozens of users experiencing the feeling of being trapped by ***It***. The current site is far from ideal, since one of the walls is made of glass and there is a considerable amount of surrounding noise. Nevertheless, we are very pleased with the reaction of the users

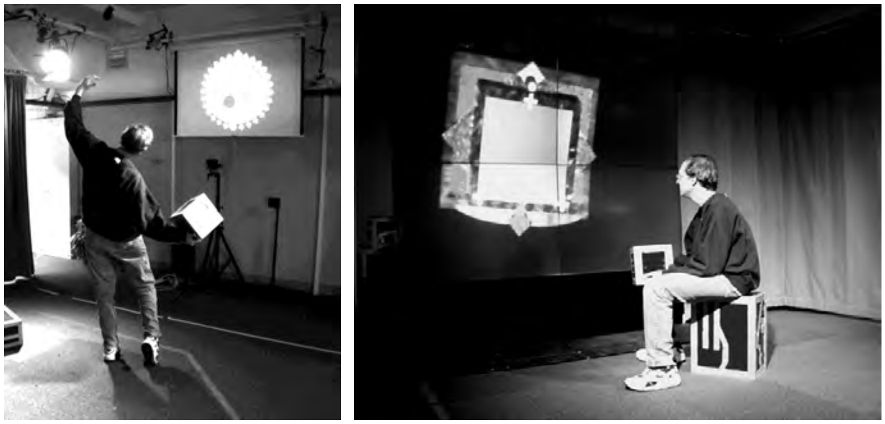

Figure 4. Scenes from a run of *"It"*.

and with the easiness that the interaction is understood. In particular, there had been some very interesting runs with children. Figure 4 shows pictures of a user playing in *"It"*.

6 Using the SCD Architecture

In order to evaluate the appropriateness of the SCD architecture, we can compare our experience in developing *"It / I"*, where we did not use the paradigm, to the construction of *"It"* which was structured according the SCD model.

The main advantage observed was a reduction in the conceptual complexity of the system. Since the character structure is independent of story control, we could develop each module of *"It"* separately, resulting in more robust and interesting automatic characters. Also, making the ***light designer*** distinct from the other modules allowed the programming of very complex light effects while preserving simple ways to inform the other modules of the lighting conditions. In other words, the SCD architecture allows for easy hiding of complex character or crewmember behaviors.

For instance, there were about 10 different lighting conditions, some of them involving patterned changes of lighting. However, from the standpoint of the ***I*** character, any state of the lights was simply labeled either as "normal" (where lighting conditions allow tracking of detailed features of the body) or "not normal" (when only the presence or absence of the user could be detected). This is, of course, the old concept of data abstraction seen in the context of a story-based system. In many ways, what the SCD model proposes is a good way to structure the abstraction, that is, along the lines between characters and crewmembers and with a supervising story control module.

One of the advantages of a separated story control module felt during the development of *"It"* was the easiness in making changes in the story line. In *"It / I"* such kind of changes many times led to low-level conflicts inside the characters' behavior. In *"It"*, changing the story had no side effects on the characters, allowing much more time and freedom of experimentation with the story. For instance, it becomes trivial to experiment with different ordering for the story events since the character level would respond to the change without the need for alterations.

From an artistic/directorial point of view, the SCD model has the positive benefit of forcing the designer of thinking the interaction as actions being performed by characters. As pointed by, among others, Clurman [6], dramatic structure is best approached if scenes are seen as the result of the conflicting actions and goals of their characters and not as a sequence of emotional states. By isolating the actors from the story and among themselves, the designer has to structure the story as a sequence of commands to each character to act upon the other characters. Although subtle, this distinction benefited tremendously the implementation of *"It"* and helped to design an experience with a stronger dramatic structure.

The major drawback of using the SCD architecture according to our experience with *"It"* is the increase in the number and frequency of messages between modules. Also, it requires a communication protocol where the concepts of "action", "intention", and "goal" can be expressed. In our case this did not become a problem because we employed a very expressive language, ACTSCRIPT, as our communication protocol. ACTSCRIPT was developed by Pinhanez and Bobick (see [20]) based on Roger Schank's *conceptualizations* [23].

7 Conclusion

The main argument of this paper is that centralized control is fundamental in story-driven interactive environments. We do not believe that good story or narrative development can be achieved by dispersing the story through the different characters and actors as proposed in previous works [2, 3, 18]. The SCD architecture is, in this view, just an implementation of the concept that story is the commander of the high-level behavior of the characters.

Our experience using the SCD model in the design and construction of *"It"* has been extremely positive. Although it increases the amount of communication between modules in a system, it reduces the conceptual complexity of designing a story-driven interactive space and increases the flexibility of making changes. We could observe that, compared to "*It / I*", we did much more experimentation with different developments for the story. Also, as pointed above, the SCD model forces the designer to think of interaction independently of the low-level programming and designing of the characters, facilitating the use of dramatic elements as the key element of the interaction.

Acknowledgements

The ideas presented in this paper benefited immensely from the comments of my Ph.D. advisor, Prof. Aaron Bobick. John Liu implemented the vision system used in *"It"*. The computer graphics objects were created by Raquel Coelho and the sound track was composed by Freedom Baird (originally for "*It / I*"). The production of *"It"* and "*It / I*" was sponsored by the Digital Life Consortium of the MIT Media Laboratory.

References

1. Director's User Manual. MacroMind Inc. (1990)
2. Bates, J. A., Loyall, B., Reilly, W. S.: An Architecture for Action, Emotion, and Social Behavior. Proceedings of the Fourth European Workshop on Modeling Autonomous Agents in a Multi-Agent World, S. Martino al Cimino, Italy (1992)
3. Blumberg, B. M., Galyean, T. A.: Multi-Level Direction of Autonomous Agents for Real-Time Virtual Environments. Proc. of SIGGRAPH'95 (1995)
4. Bobick, A. et. alli: The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. Presence: Teleoperators and Virtual Environments (1999)
5. Brooks, K. M.: Do Story Agents Use Rocking Chairs? The Theory and Implementation of One Model for Computational Narrative. Proc. of the ACM Multimedia'96 (1996) 1-12
6. Clurman, H.: On Directing. Collier Books, New York, New York (1972)
7. Coen, M. H.: Building Brains for Rooms: Designing Distributed Software Agents. Proc. of IAAI'97, Providence, Connecticut (1997) 971-977
8. Davenport, G., Friedlander, L: Interactive Transformational Environments: Wheel of Life. In: Barrett, E., Redmond, M. (eds): Contextual Media: Multimedia and Interpretation. The MIT Press, Cambridge, Massachusetts (1995) 1-25
9. Ivanov, Y., Bobick, A., Liu, J.: Fast Lighting Independent Background subtraction. Proc. of the IEEE Workshop on Visual Surveillance (VS'98), Bombay, India (1998) 49-55
10. Johnson, M., Wilson, A., Kline, C., Blumberg, B., Bobick, A.: Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters. Proc. of CHI'99, Pittsburgh, Pennsylvania (1999)
11. Krueger, M. W.: Artificial Reality II. Addison-Wesley (1990)
12. Langer, S. K: Feeling and Form. Charles Scribner's Sons, New York, New York (1953)
13. Laurel, B., Strickland, R., Tow, R.: Placeholder: Landscape and Narrative in Virtual Environments. ACM Computer Graphics Quarterly, vol. 28 (2) (1994)
14. Maes, P., Darrell, T., Blumberg, B., Pentland , A.: The ALIVE System: Full-Body Interaction with Autonomous Agents. Proc. of the Computer Animation'95 Conference, Geneva, Switzerland (1995)
15. Murray, J.: Hamlet on the Holodeck: the Future of Narrative in Cyberspace. The Free Press, Simon & Schuster, New York, New York (1997)
16. Nakatsu, R., Tosa, N., Ochi, T.: Interactive Movie System with Multi-person Participation and Anytime Interaction Capabilities. Proc. of ACM Multimedia'98, Bristol, England (1998) 129-137
17. Pentland, A.: Smart Rooms. Scientific American, vol. 274 (4) (1996) 68-76
18. Perlin, K., Goldberg, A.: Improv: A System for Scripting Interactive Actors in Virtual Worlds. Proc. of SIGGRAPH'96 (1996)
19. Pinhanez, C. S.: Computer Theater. Proc. of the Eighth International Symposium on Electronic Arts (ISEA'97), Chicago, Illinois (1997)
20. Pinhanez, C. S.: Representation and Recognition of Action in Interactive Spaces. Ph.D. Thesis. Media Arts and Sciences Program. Massachusetts Institute of Technology (1999)
21. Pinhanez, C. S., Bobick, A. F.: "It / I": A Theater Play Featuring an Autonomous Computer Graphics Character. Proc. of the ACM Multimedia'98 Workshop on Technologies for Interactive Movies, Bristol, England (1999) 22-29
22. Pinhanez, C. S., Mase, K., Bobick, A. F.: Interval Scripts: A Design Paradigm for Story-Based Interactive Systems. Proc. of CHI'97, Atlanta, Georgia (1997) 287-294
23. Schank, R. C., Goldman, N. M., Rieger III, C. J., Riesbeck, C. K: Conceptual Information Processing. North-Holland (1975)
24. Tosa, N., Nakatsu, R.: Interactive Poem System. Proc. of ACM Multimedia'98, Bristol, England (1998) 115-118
25. Weiser, M.: The Computer for the Twenty-First Century. Scientific American (1991) 94-100