8
Reading Digital Literature: Surface, Data, Interaction, and Expressive Processing

Noah Wardrip-Fruin

Introducing Digital Literature

Digital literature – also known as electronic literature – is a term for work with important literary aspects that requires the use of digital computation. Such literature has been produced for more than fifty years, with the first known example being Christopher Strachey's 1952 love letter generator for the Manchester Mark I computer (Strachey 1954). It ranges from some of the bestselling software of the 1980s (Montfort, Chapter 14, RIDDLE MACHINES: THE HISTORY AND NATURE OF INTERACTIVE FICTION, this volume) to a current diversity of work that includes experimental installations, performance-based pieces, and more.

Critical study of digital literature has a somewhat shorter history. For example, the first known PhD dissertation is Mary Ann Buckles's 1985 study of the seminal interactive fiction *Adventure* (Buckles 1985; Crowther and Woods 1976). In fact, digital literature maintained a remarkably low critical profile until the early 1990s, when a series of publications focused on hypertext literature garnered wider attention. Especially notable among these were Jay David Bolter's *Writing Space,* George Landow's *Hypertext,* and Robert Coover's essays for *The New York Times* (Bolter 1991; Landow 1992; Coover 1992; Coover 1993). The most widely discussed works of hypertext literature include Michael Joyce's *afternoon,* Stuart Moulthrop's *Victory Garden,* and Shelley Jackson's *Patchwork Girl* (Joyce 1987; Moulthrop 1991; Jackson 1995). All three of these were created within the Storyspace software environment, which organizes text into discrete nodes with links between them. The meaning of the term "hypertext" is, however, significantly broader than nodes and links – including, according to an early definition, all texts that "branch or perform on request" (Wardrip-Fruin 2004).

After hypertext criticism brought digital literature to the attention of a wider group, in the late 1990s a number of publications sought to expand the discussion.

Of these, two particularly notable books were Janet Murray's *Hamlet on the Holodeck* and Espen Aarseth's *Cybertext* (Murray 1997; Aarseth 1997). While vastly different in their critical approaches, both books employed examples from previously-neglected forms of digital literature such as interactive fiction, story generation systems, and interactive characters. These forms grow out of a heritage of computer science research, especially in artificial intelligence. The work of both Murray and Aarseth has since had an influence on interpretations of another once-neglected form of digital media that grows out of computer science, often employs artificial intelligence, and can have important literary aspects: computer games. In the years since these publications, significant work in digital literature has continued, ranging from the first book-length examination of the form of interactive fiction (Montfort 2003) to new theoretical approaches such as "media-specific analysis" (Hayles 2002).

Models for Reading Digital Literature

The starting point for this chapter is an observation: When studying a work of digital literature, as with any cultural artifact, we must choose where to focus our attention. To put this another way, we must operate with some model (explicit or implicit) of the work's elements and structures – one which foregrounds certain aspects while marginalizing others.

Most critical work in digital literature – whether focused on hypertext or other forms – proceeds from an implicit model that takes audience experience to be primary. The main components of the model are the surface of the work (what the audience sees) and the space of possible interactions with the work (ways the audience may change the state of the work, and how the work may respond).

The primary competitors to this implicit model are Aarseth's explicit models presented in *Cybertext*. The most important of these is a typology for discussing textual media, consisting of *scriptons* (text strings as they appear to readers), *textons* (text strings as they exist in the text), and *traversal functions* (the mechanism by which scriptons are revealed or generated from textons). This model, here referred to as the "traversal function" model, has been highly influential.

This chapter will consider one of the most famous works of digital literature: James Meehan's *Tale-Spin* (1976). Meehan's project is the first major story generation program. It made the leap from assembling stories out of pre-defined bits (like the pages of a Choose Your Own Adventure book) to generating stories via carefully-crafted processes that operate at a fine level on story data. In *Tale-Spin*'s case, the processes simulate character reasoning and behavior, while the data defines a virtual world inhabited by the characters. As a result, while altering one page of a Choose Your Own Adventure leaves most of its story material unchanged, altering one behavior rule or fact about the world can lead to wildly different *Tale-Spin* fictions. For this reason Meehan's project serves as an example in the books of Bolter, Murray, and Aarseth, among others.

This chapter argues that *Tale-Spin* is not just widely discussed – it is also widely misunderstood. This is demonstrated in several stages, beginning and ending with readings of *Tale-Spin* (and its companion text generator, *Mumble*) that employ the implicit audience model. Between these readings, the chapter will, first, attempt to apply Aarseth's traversal function model, second, trace the operations of *Tale-Spin*'s simulation, and, third, present a new model that helps clarify the issues important for a notion I call *expressive processing*. This new model differs from the audience model by, following Aarseth, including the work's mechanisms. It differs from Aarseth's model, however, by not assuming that the transition from textons to scriptons is either (a) readily identifiable or (b) the primary site of interest. Instead, after having identified the interplay between the work's *data, process, surface,* and possibilities for *interaction,* it is proposed that groupings of these may be considered as *operational logics* and explored both as authorial expressions (following Michael Mateas's notion of *Expressive AI* (2002)) and as expressing otherwise-hidden relationships with our larger society (particularly the cultures and materials of science and technology).

For *Tale-Spin* in particular, its primary operational logic is revealed to be that of the planning-based simulation. It is deeply connected to the cognitive science account of planning that has been extensively critiqued by scholars such as Lucy Suchman and Phil Agre (Suchman 1987; Agre 1997). It is also, more broadly, a "microworld" approach to AI of the sort that, by failing to scale up, resulted in the symbolic AI "winter" beginning in the 1980s. For the purposes of this chapter, it can be particularly valuable in two regards. First, it provides a legible example of the inevitably authored – inevitably fictional – nature of simulations of human behavior. Second, as we increasingly create human-authored microworlds as media (e.g., digital literature and computer games) it provides a fascinating example and cautionary tale.

Taking a step back, and realizing that none of this is visible from the audience's perspective, a new term is proposed. Just as the "*Eliza* effect" is used to describe systems that give the audience the impression of a much more complex process than is actually present, the "*Tale-Spin* effect" may be used to describe the obscuring of a complex process so that it cannot be perceived by the audience. The existence of these two effects helps demonstrate that the implicit audience model provides only a partial view of digital literature.

Reading *Tale-Spin*'s Outputs

From an audience perspective, a story generation program such as *Tale-Spin* is mostly experienced through its outputs – through the texts it produces. Many critics agree that this is *Tale-Spin*'s most famous output:

> Joe Bear was hungry. He asked Irving Bird where some honey was. Irving refused to tell him, so Joe offered to bring him a worm if he'd tell him where some honey was. Irving agreed. But Joe didn't know where any worms were, so he asked Irving, who refused to

> say. So Joe offered to bring him a worm if he'd tell him where a worm was. Irving agreed. But Joe didn't know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if he'd tell him where a worm was. . . . (129–30)

Two things are curious here. First, this text is rather unimpressive for being the most famous output of one of the most widely discussed works of digital literature. It raises the question: Why does this work open nearly every computer science treatment (and many critical treatments) of digital fiction?

Second, and even more curiously, this is not actually an output from *Tale-Spin*. Rather, it is, as Meehan says, a "hand translation" (127) into English, presumably performed by Meehan, of an internal system state originally represented as a set of "conceptual dependency" (CD) expressions. Further, the output above was produced early in the creation of *Tale-Spin,* before the system was complete. To publish this as one of *Tale-Spin*'s tales is akin to printing a flawed photograph taken with a prototype camera, while it still has light leaks, and using this to judge the camera's function.

Let's begin with the second of these curiosities. Why would authors such as Aarseth and Murray present these hand-transcribed errors – what Meehan refers to as "mis-spun tales" – rather than actual *Mumble* outputs of *Tale-Spin* story structures? We can begin to get an idea by examining some of the system's actual output:

> George was very thirsty. George wanted to get near some water. George walked from his patch of ground across the meadow through the valley to a river bank. George fell into the water. George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. George couldn't get near the meadow. Wilma wanted George to get near the meadow. Wilma wanted to get near George. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow. George was devoted to Wilma. George owed everything to Wilma. Wilma let go of George. George fell to the meadow. The End. (227–8, original in all caps.)

Now, here are two mis-spun tales from similar scenarios:

> Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. He was unable to call for help. He drowned.
>
> Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned. (128–9, Meehan's parenthetical explanation removed.)

All three of these drowning ant stories are quite strange. But they're not strange in the same way. The first story – the successful story, from *Tale-Spin*'s perspective – might make one ask, "Why is this language so stilted?" or "Why are these details included?" or "What is the point of this story?" The second and third story – the mis-spun stories – on the other hand, elicit questions like, "Why didn't his 'good friend' save Henry?" or "How is it possible that 'gravity drowned'?"

To put it another way, the first example makes one wonder about the telling of the story, while the second and third make one wonder how such a story could come to be. These errors prompt readers to think about systems. And this, in turn, offers insight into the first curiosity mentioned after Joe Bear's story above. *Tale-Spin* begins so many computer science discussions of digital fiction because of the operations of its system – its processes – rather than any qualities of its output. Given this, it seems essential that an analysis of *Tale-Spin* at least consider these processes. This chapter will return, later, to the conclusions of those who read *Tale-Spin* employing an implicit audience model. The next section will attempt to employ Aarseth's traversal function model in studying *Tale-Spin* and *Mumble*.

Locating *Tale-Spin*'s Traversal Function

In order to employ Aarseth's traversal function model one must identify its three elements: *scriptons* (text strings as they appear to readers), *textons* (text strings as they exist in the text), and *traversal functions* (the mechanism by which scriptons are revealed or generated from textons). For example, when interacting with a simulated character such as *Eliza/Doctor,* the scriptons are the texts seen by audience members, the textons are the simple sentence templates stored within the system, and determining the traversal function provides a typological classification of how system operations and audience behavior combine to produce the final text (Weizenbaum 1966).

Aarseth, unfortunately, while he discusses *Tale-Spin,* does not analyze it employing his model. However, on a table on page 69 of *Cybertext* he does provide its traversal function. So to employ Aarseth's model one need only identify the scriptons and textons in *Tale-Spin/Mumble*. Finding the scriptons is easy. They are the sentences output by *Mumble* in stories such as those reproduced above.

Finding the textons is harder. *Tale-Spin* operates at the level of story structure, not story telling. In particular, *Tale-Spin* focuses on simulating a virtual world – its characters, its objects, and their histories and plans. As mentioned above, there are no English sentences inside *Tale-Spin*. Its virtual world, instead, is represented in the form of "conceptual dependency" (CD) expressions. These expressions were developed as a language-independent meaning representation in the "scruffy" branch of 1970s AI research, especially in efforts headed by linguist and computer scientist Roger Schank and psychologist Robert Abelson.

Schank was Meehan's dissertation advisor during the period of *Tale-Spin*'s completion. He outlines the basic form of CD expressions in *Conceptual Information Processing* (1975), presenting them as multidimensional diagrams. When used for projects such as *Tale-Spin,* however, CD expressions are generally represented as parenthesis-organized lists. So, for example, what *Mumble* outputs as "George was very thirsty" might be represented in a program like *Tale-Spin* as:

```
(WORLD '(THIRSTY (ACTOR GEORGE) (VAL (7))))
```

Similarly, what *Mumble* outputs as "George wanted to get near some water" might be represented in a manner such as this:

```
(GEORGE '(GOAL (ACTOR GEORGE) (AT GEORGE WATER)))
```

Are CD expressions the textons of *Tale-Spin*/*Mumble*? It seems unlikely. Aarseth's phrase "strings as they exist in the text" sounds more like *Eliza*/*Doctor*'s pre-existing sentence templates, rather than parenthesis-ordered lists created during *Tale-Spin*'s run.

Unfortunately, *Mumble* doesn't provide us with any better texton candidates. It contains no sentence templates or other recognizable texts. Instead, when presented with a CD expression for output, it first identifies the expression's main act, state, or connective. For example, in the expression above, that George has a goal. The first thing produced is a subject-verb pair, such as "George wanted . . . " Then nouns are inserted. If George wanted to go to particular water and the simulation indicated that, at that time, the water belonged to a particular character, then the possessive would be added (e.g., "get near Arthur's water"). If the simulation history indicated that George had already been there then *Mumble* would choose words to indicate this (e.g., "return to Arthur's water"). Once the main words are present in the correct order, *Mumble* goes through inserting articles and punctuation.

In other words, *Mumble* assembles sentences on the fly, using a body of knowledge about English and accessing information from *Tale-Spin*'s simulation. Each sentence is based on a CD expression, but not all CD expressions are employed, and most CD expressions used do not exist before the particular run for which they help form the output. Given this, it seems clear that *Tale-Spin*/*Mumble* does not provide us with a set of clear textons, of obvious "strings as they exist in the text." Nevertheless, we still have an idea of the elements that go into producing the system's scriptons, and this may be enough to allow a discussion of its traversal function.

Aarseth's presentation of traversal functions identifies a set of "variables" – the values of which define the function. These are a mixture of elements that include audience activity and system behavior. Specifically, drawing from Aarseth's pages 62–5: *Dynamics* describes whether the work's surface and data can change in particular ways – remaining static, with only surface variability, or also variability in the number of pieces of textual data in the system. *Determinability* describes whether the work's processes operate predictably in their production of textual output, or if the processes for producing surface texts can be influenced by unpredictable factors (e.g., randomness) and so yield different responses to the same audience actions. *Transiency* describes whether the work's processes cause surface texts to appear as time passes (e.g., as in textual animations). *Perspective* describes whether an audience member determines the strategic actions of a particular character. *Access* describes whether all possible surface texts are available to an audience member at any time (e.g., a book can be flipped through, so its surface texts are "random" access). *Linking* describes types of user-selectable connections that may be presented by the work's surface (such as links on World Wide web pages) which may be always available, available under certain conditions, or simply not present. *User functions* are Aarseth's last variable. Every text makes available the "interpretive" user function to its audience. Other possible functions include the "explorative" (selecting a path), "configurative" (selecting or creating scriptons), and "textonic" (adding textons or traversal functions).

So the traversal function here is not simply the means by which *Tale-Spin* triggers English-language output from *Mumble*. Rather, the traversal function encompasses the work's operations in a number of manners. Given this, in order to go further one must investigate the operations of the *Tale-Spin* simulation.

Tale-Spin's Simulation

Tale-Spin was intended to be a storytelling system built on a veridical simulation of human behavior. As Meehan puts it:

> Tale-Spin includes a simulator of the real world: Turn it on and watch all the people. The purpose of the simulator is to model rational behavior; the people are supposed to act like real people. (107)

The basis of this simulation was the work being done Schank, Abelson, and the researchers working with them. For example, each *Tale-Spin* story begins with a character with a problem, what the group called a "sigma-state." A problem might be solved by a simple act (e.g., if a hungry character has food then she can eat it). But if a problem can't be solved by a basic act, then the character must plan to change the state of the world so that it can be. In the group's terminology such a plan was called a "delta-act." For example, if the character does not have food then she may plan to change the world so that she does have food.

However, things don't stop there. Any delta-act may, itself, have pre-conditions that aren't present in the current state of the world. For example, getting food may require knowing where some food is located, and a character may not know. Or a delta-act may include several "planboxes" that represent different approaches to a problem, which are considered serially.

Each time something is made true about the world ("asserted") *Tale-Spin* automatically works through many inferences from it. For example, if it is asserted that a character is thirsty (i.e., if a CD expression is added to the simulation that expresses this fact) then the inference mechanisms result in the character knowing she is thirsty, forming the goal of not being thirsty, forming a plan (a chain of delta-acts) for reaching her goal, etc. Crafting these inferences was an important part of authoring *Tale-Spin*. For example, in an early version of the system, when a character traveled to a place other characters nearby did not "notice." (The inference mechanism from the act of travel didn't give knowledge of the character's new location to those nearby.) This lack resulted in the mis-spun tale, quoted above, in which Henry Ant drowns while his friend Bill Bird sits nearby, unresponsive.

Take, for example, the beginning of an example story, drawn from Meehan's chapter 11. Initially, *Tale-Spin* asks what characters to use for the story, and the audience chooses a bear and a bird. *Tale-Spin* gives the characters names (Arthur Bear and George Bird), homes, basic physical characteristics, etc. Next the audience is given a list of possible miscellaneous items to create in the world, and chooses a worm. The audience is asked who knows about the worm, and chooses Arthur Bear. *Tale-Spin* then asks who the story will be about (the audience chooses Arthur) and what his problem is ("hunger" is chosen).

Through inference, Arthur forms the goal "Arthur has some food." *Tale-Spin* checks to see if it's already true, and it's not. Then *Tale-Spin* checks to see if it's already a goal. It's not, so it is added to Arthur's goal structure. This second check is performed for two reasons. Most obviously, because if it is already his goal (or a subgoal toward a higher goal) then it makes little sense to add it. But another reason to check for the goal's existence is that *Tale-Spin* also keeps failed goals, and the reasons for their failure, as part of a character's goal structure. Before this was added to the system it was easy to create mis-spun tales like the one quoted earlier in this chapter – *Tale-Spin*'s best-known product: Joe Bear forming the goal of bringing Irving Bird a worm over and over.

The first step in the plan Arthur forms, since he doesn't know where to find any honey, is to ask someone else where there is honey. He knows the location of George Bird, so the audience is asked how Arthur conceives of his relationship with George (e.g., does Arthur think that George likes him?). The answers are encouraging, so Arthur travels to ask George (after *Tale-Spin* creates the parts of the world that lie between them). Oddly enough, the CD expressions for this sort of travel seem to be sent to *Mumble* for output in full. The example quoted above, when the ant "walked from his patch of ground across the meadow through the valley to a river bank," is actually one of the less egregious.

When Arthur reaches George's tree he asks George to tell him the location of some honey. Again, the audience is asked for information about George and Arthur's relationship, this time from George's perspective. The answers lead to George believing that Arthur will believe whatever he says. Given this, George starts to speculate – the *Tale-Spin* inference mechanisms are used not to change the state of the world but for one character to "imagine" other possible worlds. George draws four inferences from Arthur believing there is honey somewhere, and then he follows the inferences from each of those inferences, but he doesn't find what he's after. In none of the possible worlds about which he's speculated is he any happier or less happy than he is now. Seeing no advantage in the situation for himself, he decides, relatively arbitrarily, to answer. Specifically, he decides to lie.

So George creates, in *Tale-Spin*'s memory, a set of CD expressions that aren't yet believed by anyone – including himself. These describe Ivan Bee, and Ivan's honey, which exist at a particular location (a location which *Tale-Spin* creates in support of George's plan). Of course, it must be a lie, because honey is not among the miscellaneous items that the audience chose to create at the outset of the story.

Observations on the Simulation

Arthur's saga continues, but enough has been said to allow a few observations. The first is that this *Tale-Spin* story contains quite a bit of psychological "action." Characters are weighing how they view their relationships with each other, spinning out many possible worlds to look for ones in which they achieve benefit, making multi-stage plans, telling elaborate lies, and so on. This is the material from which fiction is made.

However, in contrast to the detailed description of travel itineraries, the CD expressions that describe psychological action are almost never sent to *Mumble*. While Meehan doesn't provide the *Mumble* output for the story of Arthur and George, here is an excerpt from a story containing similar events:

> Tom asked Wilma whether Wilma would tell Tom where there were some berries if Tom gave Wilma a worm. Wilma was inclined to lie to Tom. (232)

All the psychological action described above between George and Arthur, in some version, took place for Wilma and Tom as well. But one would never know it from the output. Instead, by far the most interesting events of Wilma and Tom's *Tale-Spin* story take place in the gap between these two *Mumble* sentences. From an audience perspective Wilma's decision to lie might as well have been determined by a random number.

On the subject of randomness, it is also worth noting that George's decision to answer Arthur was not really arbitrary. Rather, seeing no advantage to any world about which he can speculate, *Tale-Spin* has George decide whether to answer based on his kindness. The audience is asked, and decides that George is "somewhat" kind. So, as Meehan puts it, he decides to answer "out of the goodness of his little heart" (183). But then the answer that George chooses to give, out of kindness, is a lie about honey that doesn't exist. This isn't a simulation of George thinking Arthur should diet, but a breakdown in the simulation. The component of *Tale-Spin* that determines what answer to give doesn't have any knowledge of the fact that the answer is being provided out of kindness.

There is much more that could be discussed about *Tale-Spin*'s simulation, but, for now, this is enough to return to this chapter's attempt to employ Aarseth's model.

Tale-Spin's Traversal Function

As noted above, Aarseth provides *Tale-Spin*'s traversal function to readers of *Cybertext*. Specifically, Aarseth reports that *Tale-Spin* has "textonic dynamics" (the number of textons is variable), is "indeterminable" (perhaps Aarseth identifies a random element in *Tale-Spin*), is "intransient" (it does nothing if not activated by the user), has an "impersonal perspective" (the user is not playing a strategic role as a character), has "controlled access" (not all the possible scriptons are readily available), has no linking, and has a "configurative user function."

This seems largely accurate. But it may also clarify why Aarseth doesn't employ his model in his discussion of *Tale-Spin*. This model doesn't turn attention to *Tale-Spin*'s most salient features. As seen above, the action in *Tale-Spin*'s simulation – its most telling operations – are in the formation, evaluation, and execution of plans. Most of this action is never output by *Mumble* – it never becomes scriptons. So a model that focuses on the traversal function "by which scriptons are revealed or generated from textons" is going to miss the primary site of action as well.

Further, the basic components of *Tale-Spin* and *Mumble* are hard to think about using Aarseth's model. We've already discussed the difficulty in recognizing textons within *Tale-Spin*/*Mumble*. The difficulty also runs the other direction. How should we think about *Tale-Spin*'s processes for inference making or character planning? They aren't mechanisms for turning textons into scriptons, except extremely indirectly. They aren't pointed at by "textonic dynamics" or a "configurative user function" or any of Aarseth's other variables.

Given this, I believe it is time to consider alternative models. As these models are developed it seems important to retain Aarseth's focus on system operations, such as the mechanisms by which surface text is produced. At the same time, it also seems necessary to abandon the particular focus and specific elements of his model in favor of concepts that will support consideration of a wider variety of digital literature.

A New Model

In developing a new model it may prove useful to diagram some of the alternatives, as in the diagram of an implicit audience model in Figure 8.1.

In this model the audience(s) can see the media object and engage with it through interaction. The interaction may produce some change visible to the audience(s), but what happens inside the object is unknown, as is the object's internal structure. Also, while it is known that the work is authored, the "author function" is represented in gray – author studies having been explicitly set aside by most critics. The focus is on the object as it is visible to (and can be acted upon) by the audience(s).

We might diagram Aarseth's traversal function model somewhat differently. Figure 8.2 shows an attempt.

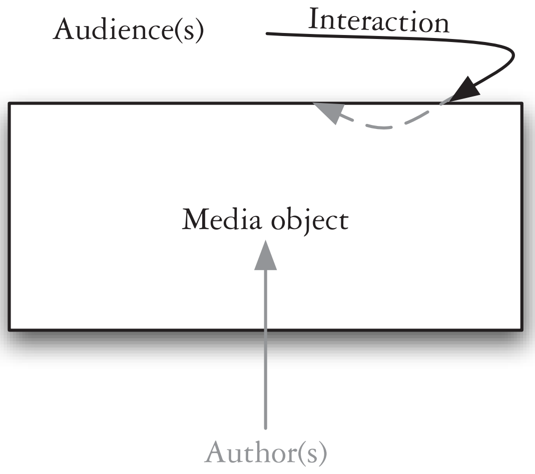


Figure 8.1 An implicit audience model of digital media.

In this model the audience(s) can see the scriptons and also work through the scripton surface to provide some of the variables of the traversal function that generates/reveals scriptons from textons (as well as, in some cases, contribute scriptons and/or textons). The textons, in most cases, along with some traversal function variables, are provided by the grayed out author(s).

Neither of these make a very good fit with the elements we've discussed of *Tale-Spin* and *Mumble,* which can be diagram in a manner such as in Figure 8.3.

This diagram represents the audience reading *Mumble*'s output on a teletype or terminal and typing replies to questions at the same point, creating a combined text. Audience responses to questions feed into *Tale-Spin*'s processes, leading to CD expressions being asserted, developing the facts and history of the simulated world.

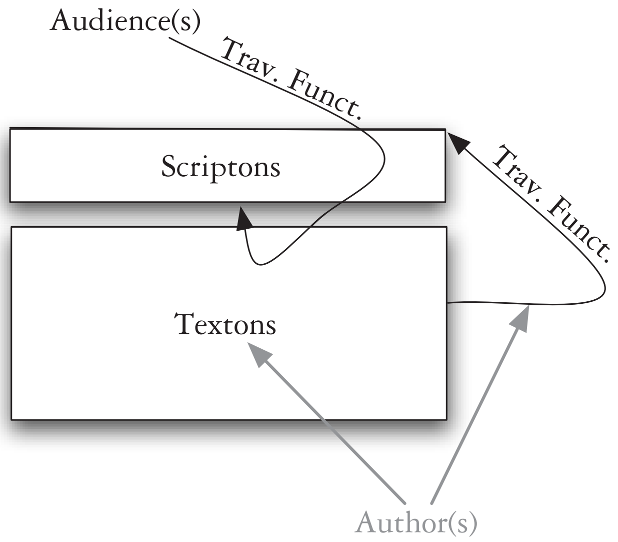


Figure 8.2 An attempt at diagramming Aarseth's traversal function model.

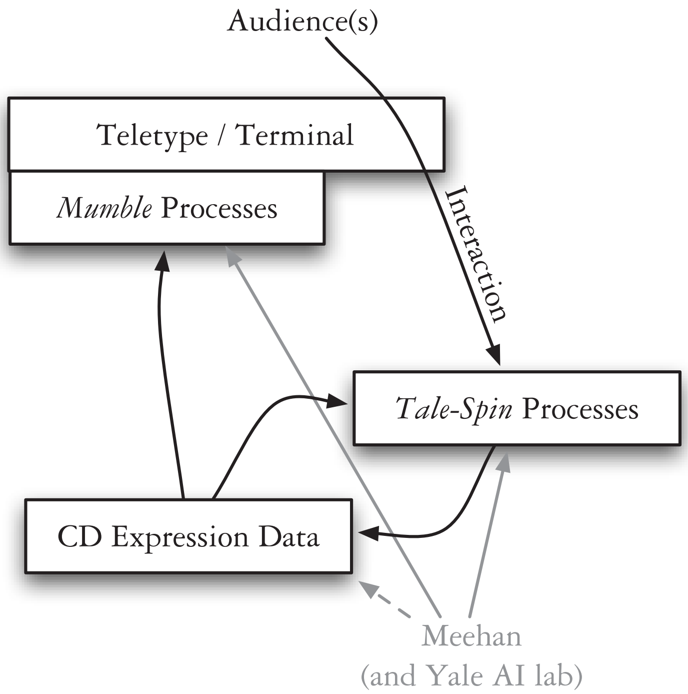


Figure 8.3 The elements of *Tale-Spin* and *Mumble*.

Inferences are drawn from assertions, using *Tale-Spin* processes, resulting in the assertion of further CD expressions (and possibly triggering character planning operations, world building operations, travel operations, etc.). A subset of CD expressions are sent to *Mumble*'s natural language generation processes, resulting in English-language output at the teletype or terminal. The *Tale-Spin* and *Mumble* processes, along with the structure of the CD data, were authored by Meehan (building on concepts then current at the Yale AI lab).

This is a quite complicated picture – and one may be inclined to think twice about it as the starting point for a model. And yet, as a work of digital literature, the structure of this system is in some ways quite simple. A brief comparison with two of my collaborative works as an author of digital literature may help to demonstrate this point.

First, in *Tale-Spin/Mumble,* audience display and interaction happen through a single, text-only device. But in a work like *Screen* (Wardrip-Fruin et al, 2003–5), created with collaborators at Brown University, the site of display and interaction includes a room-sized virtual reality display (the Cave), shutter glasses synchronized with the display via infrared pulses, and magnetic motion trackers attached to the audience member's body (Figure 8.4). This allows words (of short fictions exploring memory as a virtual experience) to appear to peel from the walls, fly around the reader, be struck with the hand, split apart, and return to the walls in new places to create altered versions of the original texts.

To take another example, the structure of *Tale-Spin/Mumble* is relatively simple because all of its processes and data can exist, self-contained, on one computer.

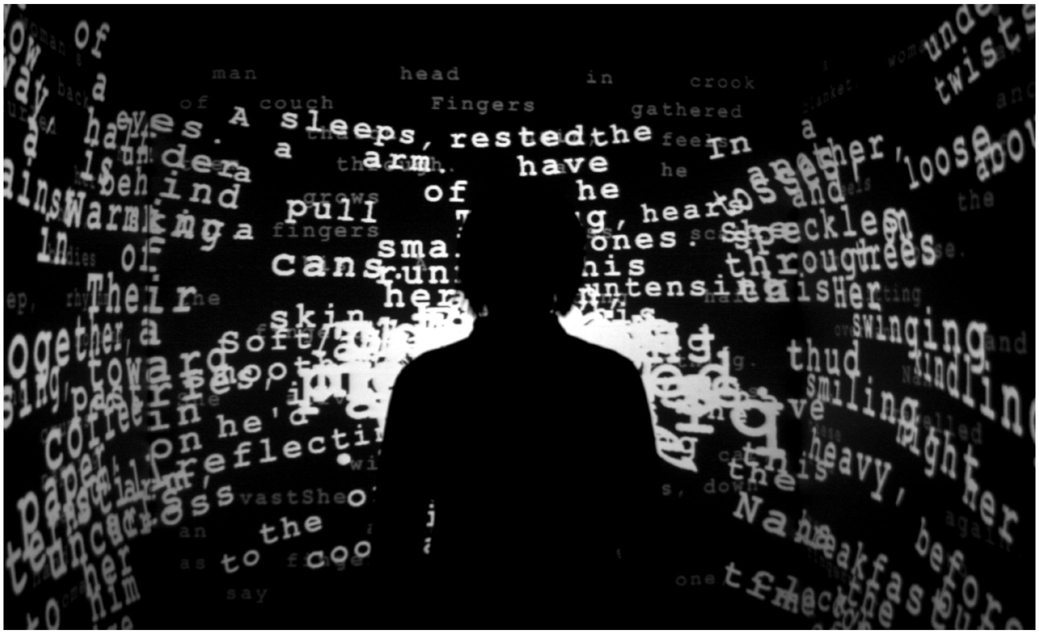
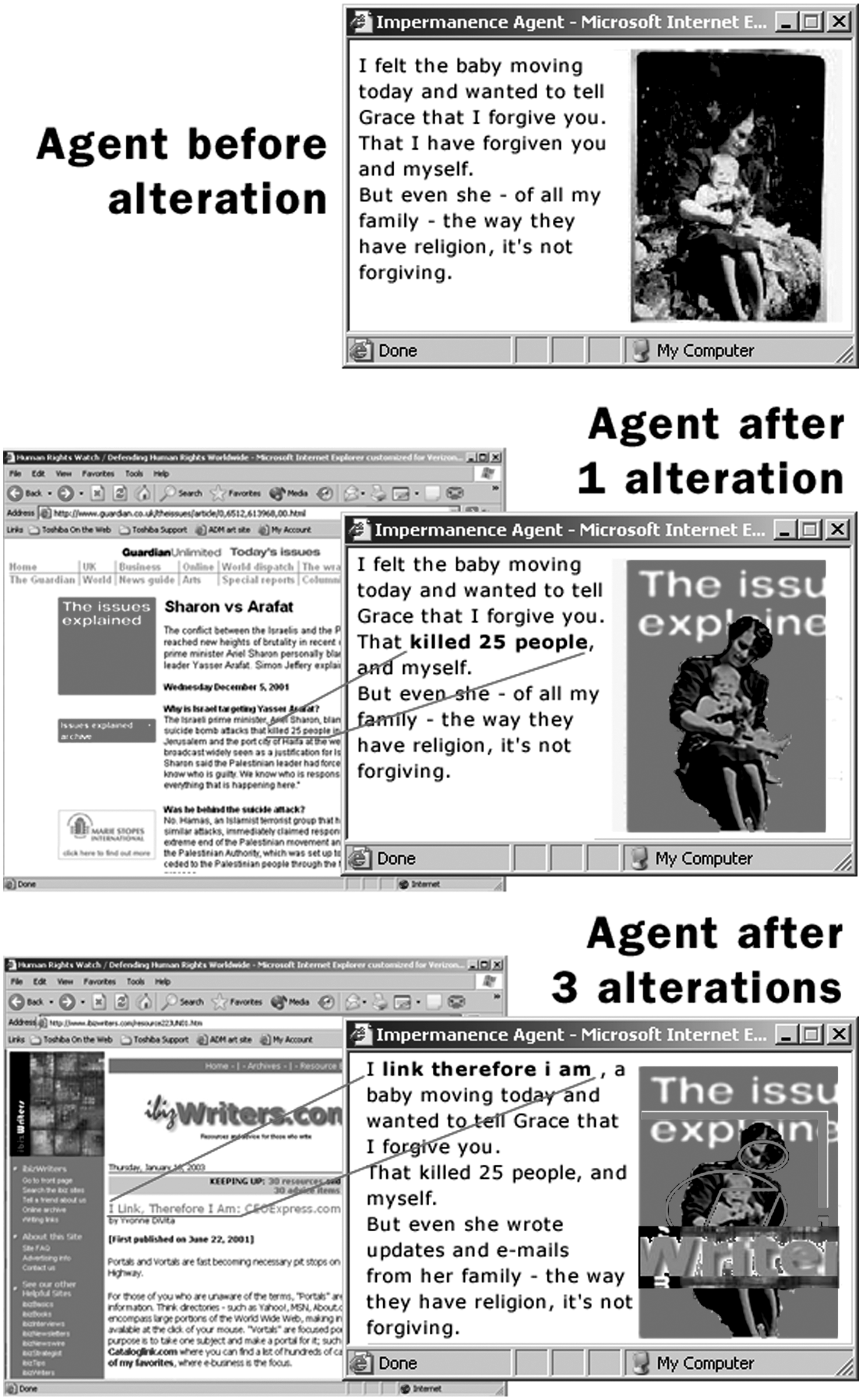


Figure 8.4 *Screen*, a digital fiction for the virtual reality Cave.

Another piece on which I collaborated, *The Impermanence Agent* (Wardrip-Fruin et al, 1998–2002) is significantly more complex in this regard (Figure 8.5). First, because it is a work for the World Wide Web that includes operations both on the server and on audience members' computers, its processes and data are split across two computers even for a single audience member. Further, while the work's processes were all defined by the authors, the work's data was different for each audience member. *The Impermanence Agent* monitored each reader's web browsing (presumably across many far-flung web servers) and incorporated parts of images and sentences from each individual's browsing into the version of its story (of documents preserved and destroyed) being performed for that reader. And it is important to realize that neither this work's split across multiple computers nor its indeterminacy of data is uncommon. They are also present for many other web works, as well as for other digital forms such as virtual worlds (e.g., massively multiplayer online games).

These two examples are only the proverbial tip of the iceberg in the complex and rapidly developing field of digital literature. Given this, how can one construct a model that will accommodate the variety of work being done in the field, provide a vocabulary useful for talking comparatively about different works, and help turn attention to the aspects significant to individual works? I offer a proposal here, but recognize that any such proposal should be preliminary, open to rejection or refinement by others, and perhaps most useful in helping us see how individual works differ from the generic. My model might be visualized as shown in Figure 8.6.

All works of digital literature are somehow presented to their *audiences* – whether on teletypes, in web browser windows, through immersive installations, or by other means. If the audience is able to interact with the work, the means for this are also


Agent before alteration
Impermanence Agent - Microsoft Internet E...
Done
My Computer
Agent after 1 alteration
Human Rights Watch / Defending Human Rights Worldwide - Microsoft Internet Explorer customized for Verizon...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.guardian.co.uk/theissues/article/0,6512,613968,00.html
Links Toshiba On the Web Toshiba Support ADM art site My Account
GuardianUnlimited Today's issues
Home UK Business Online World dispatch The wra
The Guardian World News guide Arts Special reports Columni
The issues explained
Issues explained archive
MARIE STOPES INTERNATIONAL click here to find out more
Done
Internet
Impermanence Agent - Microsoft Internet E...
The issu expl ine
Done
My Computer
Agent after 3 alterations
Human Rights Watch / Defending Human Rights Worldwide - Microsoft Internet Explorer customized for Verizon...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.ibizwriters.com/resource223JUN01.htm
Links Toshiba On the Web Toshiba Support ADM art site My Account
ibizWriters
Home - | - Archives - | - Resource
ibizWriters.com
Resources and advice for those who write
IbizWriters
Go to front page
Search the ibiz sites
Tell a friend about us
Online archive
Writing links
About this Site
Site FAQ
Advertising info
Contact us
See our other Helpful Sites
ibizbasics
ibizbooks
ibizinterviews
ibizNewsletters
ibizNewswire
ibizStrategist
ibizTips
ibizWriters
Done
Internet
Impermanence Agent - Microsoft Internet E...
The issu expl ine
Writer
Done
My Computer

I felt the baby moving
today and wanted to tell
Grace that I forgive you.
That I have forgiven you
and myself.
But even she - of all my
family - the way they
have religion, it's not
forgiving.

Sharon vs Arafat

The conflict between the Israelis and the P
reached new heights of brutality in recent
prime minister Ariel Sharon personally bla
leader Yasser Arafat. Simon Jeffery explai

Wednesday December 5, 2001

Why is Israel targeting Yasser Arafat?
The Israeli prime minister, Ariel Sharon, blan
suicide bomb attacks that killed 25 people in
Jerusalem and the port city of Haifa at the we
broadcast widely seen as a justification for Is
Sharon said the Palestinian leader had force
know who is guilty. We know who is respons
everything that is happening here."

Was he behind the suicide attack?
No. Hamas, an Islamist terrorist group that h
similar attacks, immediately claimed respon
extreme end of the Palestinian movement an
the Palestinian Authority, which was set up to
ceded to the Palestinian people through the

I felt the baby moving
today and wanted to tell
Grace that I forgive you.
That **killed 25 people**,
and myself.
But even she - of all my
family - the way they
have religion, it's not
forgiving.

Thursday, January 16, 2003

KEEPING UP: 30 resources and
30 advice items

I Link, Therefore I Am: CEOExpress.com
by Yvonne DiVita

[First published on June 22, 2001]

Portals and Vortals are fast becoming necessary pit stops on
Highway.

For those of you who are unaware of the terms, "Portals" are
information. Think directories - such as Yahoo!, MSN, About.c
encompass large portions of the World Wide Web, making in
available at the click of your mouse. "Vortals" are focused po
purpose is to take one subject and make a portal for it; such
Cataloglink.com where you can find a list of hundreds of ca
of my favorites, where e-business is the focus.

I **link therefore i am** , a
baby moving today and
wanted to tell Grace that
I forgive you.
That killed 25 people, and
myself.
But even she wrote
updates and e-mails
from her family - the way
they have religion, it's not
forgiving.

Figure 8.5 *The Impermanence Agent* customizes its story of impermanence for each reader, using material from that reader's web browsing.

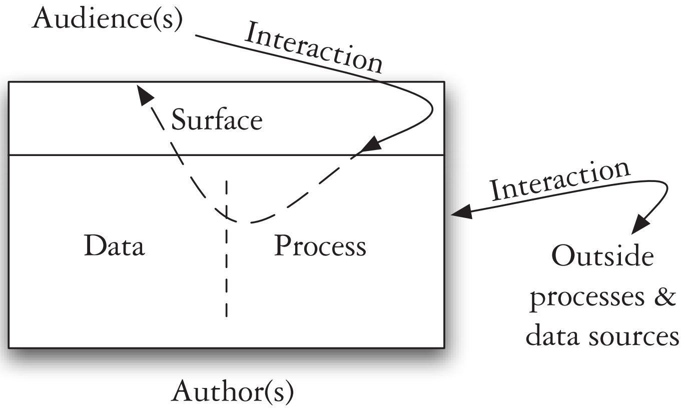


Figure 8.6 The proposed model of digital literature.

part of the work. I will call this site of presentation and (possible) interaction the work's *surface*. It may be as simple as a generic personal computer, consist of a large space or dizzying number of devices, or even take unexpected form (e.g., *The Impermanence Agent* makes all web browsing part of its interaction surface).

Works of digital literature also – whether they are organized as one or more systems, and whether they exist across one or more computers – operate via *processes* that employ *data*. This is not always obvious. For example, an email narrative may appear to consist entirely of data: the text of the email messages. But an email narrative cannot be email without the processes of the audience's email readers and at least one email server. Here the model follows Chris Crawford's vocabulary in his discussion of *process intensity* (1987) rather than the vocabulary of computer science (which might substitute a word such as *algorithm* for *process*).

While there are many definitions of *interaction,* for the purposes of this model I define it as a change to the state of the work, for which the work was designed, that comes from outside the work. Given this, the audience is not the only possible source of interaction. It is also worth noting that, in many cases, some trace of interaction is immediately apparent on the surface (e.g., an audience member types and the letters appear as they are typed, or an audience member moves her hand and a video image of her hand moves simultaneously) but this is not required. Interaction, while it always changes the state of the work, can be approached with the primary goal of communication between audience members.

The *author* is still present in this model, but the arrows representing attribution of different elements are gone. This is because we cannot predict which portions of a work will be created by authors. An installation-based work may present a physical surface almost entirely constructed by one or more authors, while an email narrative may be presented on a physical device, and using an email reading program, individually selected by each audience member. The data employed in a piece may be created by the author, contributed by the audience, generated through processes, or selected from pre-existing sources by the author, audience, or processes. Or it may be a mixture. The same is true of processes. An author may simply select among those provided by a tool such as Flash, or authors may write or modify programming language code, or the audience may be involved in process definition. The authorial status of a work may even be unknowable.

Employing the Model

As the above discussion suggests, this model can help us consider work comparatively. Just as Aarseth's model points us toward comparisons along variables such as linking, this model points us toward comparisons along variables such as the form the work's surface takes, the sources of the data employed, the state changes produced by interaction, and so on.

It also provides a structure for thinking about the operations of a work's processes – and the relationship of processes to data, surface, and interaction – more broadly than in terms of texton/scripton traversal functions. We can ask, "What is the work doing?" without expecting that we already know the answer ("turning textons into scriptons"). That is to say, in addition to comparison, I believe this model also helps us consider individual works specifically and give appropriate weight to their internal operations.

As readers have likely noticed, the work of viewing *Tale-Spin* through this model is already underway in this chapter. There has already been discussion of its processes, data, surface, and interaction. We've already seen – e.g., in the example of George Bird incoherently lying out of kindness – the way the specifics of the system operations can present revealing gaps between what the system is presented as doing (acting as "a simulator of the real world," in Meehan's words) and what it actually does. We've also seen how the most interesting operations of a work – e.g., George Bird imagining the many worlds in which Arthur believes there is honey somewhere, searching for his own advantage – may never be visible on the work's surface. Hopefully these are convincing demonstrations that tracing an algorithm's steps (watching the interplay between process and data over time) can be an important type of critical reading for digital literature. Putting this type of reading on a more equal footing with audience-perspective readings is a primary goal of this model.

What hasn't yet been explored is the wider view that this sort of examination can help develop. *Tale-Spin* has many fascinating processes, from its inference mechanisms to its simulation of interpersonal dynamics to its creation of virtual geography. I would argue that – by tracing the interplay between *Tale-Spin*'s surface, data, and process – one may be able to abstract from these to characterize the *logic* of operations such as inference. One can then discuss this operational logic itself, as well as identify this logic, and different approaches to this logic, in works with implementations (surface, data, and process) that differ from *Tale-Spin*'s. And, crucially, one can also go further, because there is a central operational logic that can be identified in *Tale-Spin*.

This central logic is *planning*. One can identify it by looking carefully at the contexts in which the other logics come into play. Consider those that create the geography of the virtual world. *Tale-Spin* does not begin by modeling a virtual world that includes all its characters and objects. Rather, many spaces, and the connections between spaces, only come into existence once one character begins to plan to travel from one place to another. The same is true of the logics that simulate interpersonal relationships. *Tale-Spin* does not create a world and then determine how all the characters feel about each other. Rather, none of the feelings that characters have about each other are determined until one of them begins to plan for interaction with the other. And characters have no feelings or other characteristics that are not those needed for *Tale-Spin*'s simulation of rational planning. Consider the following statement from Meehan's dissertation in terms of our culture's stories of love – for example, any Hepburn and Tracy movie:

> "John loves Mary" is actually shorthand for "John believes that he loves Mary." . . . I'm not sure it means anything – in the technical sense – to say that John loves Mary but he doesn't believe that he does. If it does, it's very subtle. (64)

In fact, it is not subtle at all. It is a significant plot element of the majority ofromantic novels, television shows, and movies produced each year. But from within *Tale-Spin*'s central logic his conclusion is perfectly rational. If John doesn't know that he loves Mary, then he cannot use that knowledge in formulating any conscious plans – and in *Tale-Spin* anything that isn't part of conscious planning might as well not exist.

This blindness to all but planning – this assumption that planning is at the center of life – was far from unique to Meehan. Within this wider AI and cognitive science community, at the time of Meehan's work, the understanding and generation of plans was essentially the sole focus of work on intelligent action. Debate centered on what kind of planning to pursue, how to organize it, and so on – not on whether planning deserved its central place as a topic for attention. This was in part due to the field's technical commitments, and in part the legacy of a long tradition in the human sciences. Lucy Suchman, writing a decade later in her book *Plans and Situated Actions* (1987), puts it this way:

> The view, that purposeful action is determined by plans, is deeply rooted in the Western human sciences as the correct model of the rational actor. The logical form of plans makes them attractive for the purpose of constructing a computational model of action, to the extent that for those fields devoted to what is now called cognitive science, the analysis and synthesis of plans effectively constitute the study of action. (ix–x)

This view has, over the last few decades, come under widespread attack from both outside and within AI. As Suchman puts it, "Just as it would seem absurd to claim that a map in some strong sense controlled the traveler's movements through the world, it is wrong to imagine plans as controlling action" (189). As this has happened – and particularly as the mid-1970s theories of Schank, Abelson, and Meehan have moved into AI's disciplinary history – *Tale-Spin* has in some sense lost its status as a simulation. There's no one left who believes that it represents a simulation of how actual people behave the in the world.

As this has taken place, *Tale-Spin* has become, I would argue, *more* interesting as a fiction. Its logics can no longer be regarded as an accurate simulation of human planning behavior, with a layer of semi-successful storytelling on top of it. Rather, its entire set of operations is now revealed as an authored artifact – as an *expression,* through process and data, of the particular and idiosyncratic view of humanity that its author and a group of his compatriots once held. Once we see it this way, it becomes a new kind of fiction, particularly appreciable in two ways. First, it provides us a two-sided pleasure that we might name "alterity in the exaggerated familiar" – one that recalls fictions such as Calvino's *Invisible Cities* – which presents us with the both strange and recognizable image of life driven only by plans within plans. At the same time, it also provides an insight, and cautionary tale, that helps us see the very act of simulation-building in a new light. A simulation of human behavior is always an encoding of the beliefs and biases of its authors – it is never objective, it is always a fiction.

Resurfacing

Having spent most of this chapter on an examination of *Tale-Spin*'s internal operations (and on proposing a model of digital literature that provides space for such examinations) this chapter will now return to *Tale-Spin*'s surface by considering two examples of what has been written by those noted earlier in this chapter as operating via an implicit audience model.

Janet Murray, in her book *Hamlet on the Holodeck* (1997), writes of *Tale-Spin* in the context of her argument that writers "need a concrete way to structure a coherent story not as a single sequence of events but as a multiform plot" (185). Murray reprints the famous mis-spun tale of Joe Bear forming the failed goal, over and over, of bringing Irving Bird a worm, and then writes:

> The program goes into a loop because it does not know enough about the world to give Joe Bear any better alternatives. The plot structure is too abstract to limit Joe Bear's actions to sequences that make sense. (200)

In fact, as discussed earlier, *Tale-Spin* looped because – in its partially-completed state at the time this mis-spun tale was generated – its characters could reassert a goal that had already failed. Further, Joe Bear's problem had to happen at the character level – it could not happen at the level of "plot structure" – because *Tale-Spin* has no representation of plot at all. Murray's failure to understand *Tale-Spin/Mumble*'s operations leads to a missed opportunity. As the next chapter of her book demonstrates, she is very interested in systems that model the interior operations of fictional characters. And characters like Joe Bear and George Bird have quite complex interior operations, if one looks beyond the anemic events output by *Mumble,* making them a good potential example for arguments like Murray's.

In *Cybertext,* on the other hand, *Tale-Spin* is one of Aarseth's three primary examples for the argument that machine narrators should not be "forced to simulate" human narrators (129). *Tale-Spin* is presented as a failed example of such simulation, with its mis-spun tales its only claim to interest. From the viewpoint of AI, Aarseth's is an exceedingly strange argument. The primary critique of *Tale-Spin* in AI circles is precisely that it *does not* attempt to simulate a human narrator. *Tale-Spin* simulates characters – not narrators, not authors. We can overlook this, however, because Aarseth is arguing against simulating human narrators only as a proxy for their assumed poetics. He writes:

> To achieve interesting and worthwhile computer-generated literature, it is necessary to dispose of the poetics of narrative literature and to use the computer's potential for combination and world simulation in order to develop new genres that can be valued and used on their own terms. (141)

Of course, as our examination of its operations shows, *Tale-Spin* can be seen as precisely the sort of literature for which Aarseth is calling. The story structures it produces are almost never like those that a human storyteller would produce. Instead, it uses "combination and world simulation" to produce strange branching structures of plans within plans within plans. From this it is possible to see that, for Aarseth, too, *Tale-Spin* could serve as a strong positive (rather than misleading negative) example.

Aarseth's missed opportunity, combined with Murray's missed opportunity, helps to reveal something interesting. *Tale-Spin,* early as it was, stands at an important crossroads. If we choose to emphasize its continuities with traditional fiction and drama, via its characters, then it becomes a useful touchstone for views such as Murray's. If we choose to emphasize its complicated strangeness, its computational specificity, then it becomes an important early example for views such as Aarseth's. In either case, a close examination of the system's operations reveals something much more intriguing than either author assumed.

And there is also something further that can be learned from considering the readings of these two generally insightful scholars. Even dedicated, careful researchers were unable to see what might interest them about *Tale-Spin* by looking at the *Mumble* output. Its fascinating operations were completely hidden, when viewed from the surface perspective.

There is, in fact, a term for works that, when viewed from the surface, seem (at least at first) much more complex and interesting than they actually are: the "*Eliza* effect," in reference to Joseph Weizenbaum's early interactive character. I would like to propose a companion term, inspired by what we see here about *Tale-Spin*/*Mumble*'s surface: the "*Tale-Spin* effect" which describes works that have complex and interesting internal processes that are hidden when the work is viewed from the surface.

I believe the *Tale-Spin* effect is important to consider for two reasons. First, scholars of digital literature must be aware that the surface may not reveal the aspects of a work that will be most telling for analysis (a case in which scholars may miss what a work's processes express). Second, and just as importantly, authors of digital literature must realize that an interesting, successful, *hidden* process will offer less to an audience even than the *visible* errors produced by a broken process, as can be seen with *Tale-Spin*'s mis-spun tales (a case in which authors are not effectively employing processes in their expression through the work). In both cases, I believe that a model organized around the relations of surface, data, process, and interaction – and their interplay in operational logics – may provide fruitful insights into expressive processing.

References and Further Reading

Aarseth, E. J. (1997). *Cybertext: Perspectives on Ergodic Literature*. Baltimore: Johns Hopkins University Press.

Agre, P. E. (1997). *Computation and Human Experience*. Cambridge: Cambridge University Press.

Bolter, J. D. (1991). *Writing Space: The Computer, Hypertext, and the History of Writing*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Buckles, M. A. (1985). *Interactive Fiction: The Computer Storygame Adventure*. PhD thesis, University of California, San Diego.

Coover, R. (1992). "The end of books." *The New York Times Book Review* June 21: 1, 23–5.

—— (1993). "Hyperfiction: novels for the computer." *The New York Times Book Review*, August 29: 1, 8–12.

Crawford, C. (1987). "Process Intensity." *Journal of Computer Game Design* 1.5.

Crowther, W., and D. Woods (1976). *Adventure*. [Computer game].

Hayles, N. K. (2002). *Writing Machines*. Cambridge, MA: MIT Press.

Jackson, S. (1995). *Patchwork Girl*. Watertown, MA: Eastgate Systems Inc.

Joyce, M. (1987). *afternoon: a story*. Watertown, MA: Eastgate Systems Inc.

Landow, G. P. (1991). *Hypertext: The Convergence of Contemporary Critical Theory and Technology*. Baltimore: Johns Hopkins University Press.

Mateas, M. (2002). *Interactive Drama, Art and Artificial Intelligence*. PhD thesis, Carnegie Mellon University.

Meehan, J. R. (1976). *The Metanovel: Writing Stories by Computer*. PhD thesis, Yale University.

Montfort, N. (2003). *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge, MA: MIT Press.

Moulthrop, S. (1991). *Victory Garden*. Watertown, MA: Eastgate Systems Inc.

Murray, J. H. (1997). *Hamlet on the Holodeck*. New York: The Free Press.

Schank, R. C. (1975). "Conceptual Dependency Theory." In R. C. Schank (Ed.). *Conceptual Information Processing*. New York: Elsevier Science Inc., pp. 22–82.

Strachey, C. (1954). "The 'Thinking' Machine." *Encounter* III.4: 25–31.

Suchman, L. A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge: Cambridge University Press.

Wardrip-Fruin, N. (2004). "What Hypertext Is." In J. Whitehead and D. De Roure (Eds). *HYPERTEXT '04: Proceedings of the Fifteenth ACM Conference on Hypertext & Hypermedia*. New York: ACM Press, pp. 126–7.

——, A. Chapman, B. Moss, and D. Whitehurst (1998–2002). *The Impermanence Agent*. <http://www.impermanenceagent.com/>.

——, S. Becker, J. Carroll, R. Coover, S. Greenlee, and A. McClain (2003–5). *Screen*. Providence, RI: Brown University Center for Computation and Visualization.

Weizenbaum, J. (1966). "Eliza: A Computer Program for the Study of Natural Language Communication between Man and Machine." *Communications of the ACM* 9.1: 36–45. New York: ACM Press.