Assumptions underlying the Erasmatron interactive storytelling engine

ChrisCrawford

2349 Sterling Creek Road, Jacksonville, OR 97530   chriscrawford@wave.net

Abstract

The Erasmatron engine presents an unconventional aproach to the problems of interactive storytelling design. The underlying cause of its peculiarity is the set of unconventional assumptions from which it was built: 1. While architecturally valid stories can be created by algorithm, humanly interesting stories can be created only by artists. 2. The battle between character-based stories and plot-based stories can be elegantly resolved by recourse to verb-based stories. 3. Interactivity is central to the design of computer storytelling systems. 4. A high threshold of complexity must be crossed to obtain reasonable results.

Prolegominon

Many fingers are creeping over the broad intersection of computers and stories. Seven years ago, I began fulltime efforts on the problem; my approach was dictated by my previous experience as a designer of computer games. My organizing concept was "a non-puerile computer game with strong story". As with any extrapolatory approach, this intellectual foundation boasted certain strengths and suffered from weaknesses. On the plus side were conflict and interactivity: my thinking was already focussed on these two notions. Computer games are obviously strong in conflict; we need merely increase the subtlety and indirection of the conflict to achieve the dramatic conflict so essential to good storytelling. My familiarity with interactivity focussed my thinking on the dynamics of the storytelling process as opposed to the static structure of an individual story.

Of course, there were also some penalties arising from my heritage as a computer game designer. Try as I might, I could not shake free from worries about the presentation of the story — its audiovisual values. Worse, my bias towards evaluatable results (winning or losing) poisoned my thinking.

Bumbling forward with these concepts, I built a software technology for interactive storytelling, consisting of an engine, a front end, and an editor. This technology, warts and all, can be found at www.erasmatazz.com. The Wintel versions of the technology are poor stepchildren of the Macintosh versions and perform poorly.

Artists, not algorithms

I begin by acknowledging the impressive work that has been done in creating stories solely by algorithm. I'm sure that this work will reveal much about both algorithm design and human communication. As far as storytelling itself goes, however, I don't think such efforts will take us far. Storytelling is not like chess inasmuch as there exists no rigorously objective criterion for its success. Suppose, for example, that some researcher somewhere were to create a software system powerful enough to replicate the story in Star Wars. I would still count such a system a failure, because its output is in no way new, and is therefore boring. Of course, if this software system were powerful enough to create dramatically interesting variations on Star Wars, then I'd genuflect before the designer — but no such system is on the horizon. It's not just a matter of inserting random variations at critical junctures in the story; if Luke Skywalker were to refuse Obi Wan Kenobe's offer to join him at the outset of the story, there wouldn't be much story to tell. Any variations must be dramatically interesting and consistent with the previous development of the storyline.

My arguement against purely algorithmic storytelling arises not from a simple extrapolation of past failures into the future. I think that there are more fundamental reasons for my pessimism. Algorithmic approaches are useful only to the extent that they get to the heart of the problem. A successful algorithm must express the fundamental abstractions that apply to the problem. The exception to this arises when the problem is so small that it can be addressed with high-level brute-force instantiation coupled with low-level algorithms. For example, the solutions to the chess problem are long on brute-force instantiation and short on deep abstraction. Yet I very much doubt that any such instantiations are available. The best we have here is Polti's The Thirty Six Dramatic Situations, and Propp's Morphology of the Folktale, and anybody who has read these works can agree that there is nothing in them easily reduceable to algorithms.

Moreover, storytelling is much like language in that it relies heavily on contextual knowledge of the culture in which it is told. While we have, after many decades of labor, managed to convert many of the underlying abstractions of language into algorithmic form, the overall goal of machine translation still eludes our grasp, because so many utterances use metaphors to particular details of the culture. Just as language confounds our efforts with its perverse references to the latest fads, so too will storytelling be tripped up by the storyteller's need to integrate the story into the warp and woof of the culture.

I therefore claim that interactive storytelling will always be the domain of the artist. Surely computers will play an important role in the creative process (and a necessary role in the performance), but the creative responsibility must lie with the artist, not the algorithm. This does not pre-

clude the possibility of the artist creating the algorithms; indeed, the Erasmatron requires the artist to supply a great many simple algorithms for character choices. But these are narrowly-applied algorithms; overall creative strategy remains a task for neurons, not transistors.

This claim has big implications for the way we should proceed. Our task shifts from designing the storytelling iself to designing the infrastructures in which the artists will work, and the tools they will need to do so. It's rather like the use of word processors. The designer of the word processor makes no attempt to design the content of the documents to be created with the word processor; instead, the designer defines the mental image that organizes the user's thinking while writing the document. For example, the introduction of multiple fonts and typestyles has changed the way the average user creates documents. People now freely use boldface, underlining, italics, and multiple fonts to endow their words with additional meaning. We must do the same thing on a much larger and more abstract scale — but we must refrain from intruding into the artistic side of storytelling. This decision demands more than intelligence: it requires wisdom.

I am too conceited to permit the requirement of wisdom to deter me from attempting a solution. With the Erasmatron, I segregated artistic functions from algorithmic functions based on my assessment of the artistic content of each function. Character movement, for example, is directly controlled by algorithm and indirectly controlled by the artist. To wit, if the artist sets a dramatic goal for the character, such as "beat up Joe", the algorithms will move the character to the stage occupied by Joe. If the character has no assignments, then the algorithms will move the character randomly, based on artist-assigned values of "Wanderlust" for each character and "Territorialities" for each stage and character.

Similarly, I decided that gossip-functions are too mundane to be directly controlled by the artist. Therefore, the artist declares the "Loquacity" of each character, who then gossips with other characters based on that loquacity and algorithmic decisions as to the import of the gossip to the recipient, the desirability of the recipient's likely reaction, the intrinsic credibility of the gossip, and the possibly secretive nature of the information in question.

The dramatically most critical decisions are the verb choices made by the characters: how they react to the events they experience. These decisions are all under the direct control of the artist. I have learned, however, that few artists are willing to endure the mathematical tedium required by this control, and am currently modifying the Erasmatron to reduce the tedium without compromising artistic control. More details on these modifications can be found at www.erasmatazz.com/Version 2.0/index.html.

Plot-based stories vs character-based stories

This is an old battle amongst writers. Some argue that "character drives story" while others insist that "plot IS story". Each side acknowledges the importance and necessity of the other's favored element; the dispute revolves around which should be accorded primacy. This problem has fascinating correlates in many other fields such as physics (waves vs particles), economics (services vs goods), computer science (cycles vs bytes), and linguistics (verb versus noun). For more on this topic, see www.erasmatazz.com/library/miscellania/OperationalDefinition.html

My own opinion is that the plot-based approach is truly more fundamental, but the character-based approach is often more useful. Writers can use character-based approaches to address storytelling problems that simply cannot be resolved using plot-based approaches.

However, we need not waste energy debating this problem, for we can neatly sidestep it in the limited domain of interactive storytelling by thinking in terms of verbs and sentences. Thus, our fundamental design atom is not {character possessed of certain attributes, motivations, hang-ups, etc}; nor is it {plot composed of certain critical points, twists, and junctions}; it is instead {sentence consisting of subject, verb, direct object, and other elements}.

This schema allows us to analyze the problem along cleavage lines that are easier to work with. The two warring concepts of character and plot are merged at an atomic level in the sentence. Our task then becomes to define the verbs available during storytelling and the character-istics that will lead to one verb-choice rather than another.

This sentence-based approach is the core of the Erasmatron technology. The central element of each sentence is the verb; the editor is therefore organized around the master list of verbs for a storyworld. The artist declares, for each verb, a set of roles defined by the relationships with the actors in the sentence. For example, the verb "beat up" would likely be accompanied by roles for the victim, a disinterested bystander, a sympathetic bystander, and so on. Each role includes a set of options (verbs) that constitute dramatically reasonable reactions to the predicating event. The meaning of verbs is extensible by means of the inclusion of "secondary objects". These take the place of indirect objects, prepositional objects, adverbial phrases, or the components of dependent clauses; their meaning is intrinsic to the definition of the verb. This crude mechanism vastly extends the expressive power of verbs without requiring complex linguistic parsing.

Interactivity

This is more a matter of defining ends rather than means. If we were able to get computers to think success-

fully about storytelling, what would we use this new-found talent for? While agent technology has obvious and important applications in improving user interface and creating proactive software resources, the computer as storyteller seems rather lost without the element of interactivity. After all, what entertainment, explanatory, or educational value would a computer storyteller have in the absence of interactivity? Interactivity lies at the heart of the computer revolution; we must set our compass by its poles. We must factor interactivity into our design effort at a fundamental level. Thus, our task is not to create computer stories and then tack some kind of interactivity onto them; our task is to design interactive computer storytelling from the ground up. The operational expression of this goal is the question "What does the user do?" Note that this key question is almost impossible to answer when one thinks in terms of characters or plots. The character-based designer faces utter nullity, for the character of the user is necessarily unknown. The plot-based designer finds the free will of the user diametrically opposed to the imposition of plot. The apparent conflict between interactivity and narrative has confounded people for too long; detailed explanations of the chimerical nature of this conundrum can be found at www.erasmatazz.com/library/ lilan/inimical.html.

Complexity Threshold

The challenge of interactive storytelling is made more difficult by the high threshold of complexity required by the audience. A great many successes in computer science involve, in some way, "faking it": providing a solution not quite the same as that desired by the user, but hardly distinguishable from the desideratum. Storytelling, however, crawls with complexity expectations on the part of the users. Stories must reflect the complexities of our personalities and cultures; if they don't, audiences disdain them. Such high expectations render useless the traditional analysis and synthesis approach that serves so well elsewhere. A story that runs afoul of some well-known detail of human behavior will be dismissed out of hand.

I first learned this lesson some years ago while building my first interactive storyworld, Le Morte D'Arthur. My interactive version of the Arthurian tale did not fail to include the requisite affair between Lancelot and Gwenevere. Unfortunately, my algorithm for gossip was mostly but not completely successful in representing human behavior. Thus, after Lancelot bedded Gwenevere, he was so excited that he felt a need to share this great news with a close friend, a friend whom he could trust — who turned out to be Arthur!

All of us can smile at so egregious a blunder, yet consider this: if we were to compile a list of human behaviors required to successfully model characters in our story-telling, the rule "Don't tell your best friend about your sexual exploits if he's married to your partner" would surely fall far down on the list.

Any storytelling software must be equipped with algorithms of great complexity before it can yield interesting results. Building from the ground up, one feature at a time, will yield results so far short of genuine storytelling as to obviate any useful feedback on the design. We can't build living systems by stitching together body parts and zapping them with high voltage a la Frankenstein. Nor can we succeed by playing chemistry games with amino acids. The minimum threshold of complexity is the gene of DNA — a very large and complicated structure. Many of the early attempts at interactive storytelling have a Frankensteinian air about their designs.

I believe that the Erasmatron bristles with enough behavioral complexity to inch its way over the threshold of storytelling complexity. Aside from the obvious reaction, plan-making and plan-execution facilities, it offers a capability for anticipating likely reactions to a contemplated move; the ability to tell lies and reveal secrets, and later to track them down to their source; defer intended behaviors if they might be witnessed by the wrong people, and so forth. Indeed, I have been hoisted by my own petard in the matter of complexity: as yet, largely because of its complexity, not a single artist has completed creation of a viable storyworld using the Erasmatron!

I am therefore re-writing the Erasmatron with the primary purpose of easing the artist's workload. This involves some simplification of the engine, but most of the changes are means of assisting the artist in handling the complexity.

Assisted Storytelling

Recognition of the complexity threshold does not necessitate abandonment of our goal. Pursuit of a lesser goal now might put us in a position to tackle the ultimate objective later. The Erasmatron demonstrates that interactive storytelling can be achieved, but only with the application of Herculean efforts. This suggests that a junior form of interactive storytelling might be better suited to the current inexperience of the artistic community. One such junior form I will call "assisted storytelling". It uses the Erasmatron technology in every way but one: the user, not the artist's algorithms, makes the choices for the characters. This transforms the user from protagonist to director. The artist's role is altered concomitantly: s/he now determines dramatically viable options and defines the roles for them, but no longer provides the inclination equations that govern character choices. This greatly reduces overall effort, making the technology more accessible to the artist community.