Intelligent Narrative Technologies: Papers from the 2012 AIIDE Workshop
AAAI Technical Report WS-12-14

Punch and Judy AI Playset: A Generative Farce Manifesto Or: The Tragical Comedy or Comical Tragedy of Predicate Calculus

Ian Horswill
Northwestern University, 2133 Sheridan Road, Evanston IL 60208
ian@northwestern.edu

Abstract

Building complete interactive narrative systems is hard. Building systems that are satisfying for naïve users is especially hard since small deficiencies in component technologies can easily destroy the experience for a user. In this paper I argue that we can ameliorate some of these technical limitations through careful choice of genre and style, and discuss a number of properties of farce that make it a particularly attractive choice. Then I will describe work in progress on *Punch and Judy AI Playset*, a system that allows users to explore possible narratives in the Punch and Judy story world.

Introduction

Intelligent narrative technology is still in its infancy. Considerable progress has been made in component technologies such as story generation and drama management, and some complete playable systems have been developed, such as *Façade* (Mateas and Stern 2005), Cavazza et al.'s excerpts from Flaubert's *Madame Bovary* (Cavazza et al. 2007), and most recently, *Prom Week* (McCoy et al. 2011). These latter systems, while technical *tours de force*, are problematic for naïve users. *Madame Bovary* relies on synthesized speech and playback of fixed animations through a commercial game engine to convey character emotion. While no fault of the AI technology, this is problematic for a psychological novel where realistic depiction of emotion is central. *Façade* and *Prom Week* both require considerable understanding of the system on the part of the user before they can successfully craft inputs that do what they expect them to.

The problems with these systems lie not in their technology, which is marvelous, but in the choice of style and genres to which they apply it. *Madam Bovary* and *Who's Afraid of Virginia Woolf?* on which *Façade* is loosely based, are both told in highly realistic styles. Consequently, even small departures from realism can break the user's immersion in the story. An equally psychological but less realist style, such as that of David Cronenberg or Philip K. Dick, might be more appropriate since the uncanny and hallucinatory already feature prominently in their works. A synthesized voice might not seem out of place in Cronenberg's adaptation of *Naked Lunch*.

In this paper, I'll argue that farce is an attractive genre for generative AI systems. I'll begin by examining the prominence of a particular subspecies of farce, the slow burn, in a different electronic medium, Xtranormal[1], and argue that the slow burn as a genre is particularly well suited to Xtranormal's limitations. Then I'll discuss farce more broadly and argue that it has a specific set of tolerance properties that make it attractive for AI-based narrative systems. Finally, I'll describe work in progress on *Punch and Judy AI Playset*, a system for generative farce in which players can experiment with character situations and relationships and see how they play out.

Why do slow burns work in Xtranormal?

Xtranormal allows users to create simple animated dialogs given textual scripts. The software stages and renders the dialogs using simple 3D character models and text-to-speech voice synthesis. Because of Xtranormal's extreme limitations, users rarely use it for drama. Instead, stories are overwhelmingly skewed to the comedic. One particularly popular and effective genre on Xtranormal is the *slow burn*.[2]

The slow burn is a variant of the Vaudevillian double-act in which the "straight man" implacably repeats

[1] http://www.xtranormal.com

[2] See *Reading and Time: A dialectic between academic expectation and academic frustration*, and *iPhone4 vs HTC Evo* (authors unknown, both available on YouTube) for good examples of the genre.

variations of some unreasonable idea, while the "funny man" gradually moves from exasperation to anger, to fury. Monty Python's "dead parrot" sketch (Cleese and Chapman 1969) is a classic slow burn, but relies heavily on Cleese's superb acting.

By contrast, Xtranormal videos have no real acting whatsoever. Characters perform little or no gesture and the text to speech algorithm delivers lines in near deadpan. While disastrous for a love story, this fits surprisingly well with the needs of the slow burn; the straight man's whole purpose is to be deadpan, and while the funny man's anger is most often played as "hot" (yelling and gesticulating), the audience can read deadpan in this context as cold anger and irony. That said, the slow burn requires some kind a gradual escalation in affect, and that is difficult to achieve with a speech synthesizer. Consequently, slow burns in Xtranormal videos rely on escalation of the content of the funny man's lines rather than their delivery, often in the form of escalating profanity. The dead parrot sketch doesn't work on Xtranormal precisely because Cleese's part was written for yelling and mugging for the camera, not deadpan.[3]

This is not to say we should drop what we're doing and build AI systems to make bobble-headed funny animals spout deadpan obscenities through voice synthesizers (amusing as that might be). But we need to take heed of the subtle interactions between genre and style on the one hand, and the limitations of our technologies on the other. In the short term, we need to limit ourselves to genres that are well adapted to the capabilities of our systems. Farce is a good candidate.

Farce

Farce is a comedic genre that involves improbable situations and behaviors. Monty Python's sketches, Oscar Wilde's *The Importance of Being Earnest*, and Shakespeare's *The Comedy of Errors* are all farces. Farce is also a common genre for animated film and television, both in children's animation (e.g. *Looney Tunes*, *Tom and Jerry*, etc.) and modern adult-oriented animation (*The Simpsons*, *South Park*, *Family Guy*, *Moral Orel*). By focusing on the audacious and transgressive, farce involves the audience in the narrative, keeping them wondering oh-my-God-what-will-he-do-next, without necessarily producing closure in the traditional narratological sense (Abbott 2008), or even the desire for it.

The looseness and versatility of farce make it a forgiving genre for AI. It can tolerate a number of departures from classical storytelling, allowing designers to focus development on individual technologies while getting by with simple "stubs" for other technologies. In effect, farce is a genre with training-wheels.

Plot-tolerance

Although farce can be very tightly plotted, it is notable for its ability to sustain very loosely-plotted stories. The term farce derives from the French word for stuffing and refers to the medieval and early renaissance Christian Church's practice of adding humorous scenes to ecclesiastic plays to help maintain the audience's attention (Davis 2002). Similar to C-3PO and R2D2's ongoing bickering in *Star Wars*, these scenes did little to advance the plot of the underlying story, but instead provided comedic relief from dramatic tension or simple boredom.

The *commedia dell'arte*, an important predecessor of modern improv acting, developed this historical sense of farce. Each actor developed a stock of dozens or hundreds of character-specific stock gags (*lazzi*) that could be inserted into any point in story as needed (Gordon 2001). Commedia plays were largely improvised, with only a rough sequence of plot points having been agreed upon in advance (Duchartre 1996). Its Anglophone descendant, *Punch and Judy*, is extremely loosely plotted, consisting of a picaresque series of episodes with only very loose causal ties, such that they can be selectively omitted or rearranged as necessary (Collier and Cruikshank 2006).

While modern farce is often very tightly plotted, it need not be. Monty Python was known for abruptly ending sketches whenever they ran out of ideas for them, abandoning narrative closure entirely (see *Monty Python and the Holy Grail* for one notorious example).

This is not to say that tight plotting is bad (far from it!), but rather that there are advantages to choosing a genre for which it is not *mandatory*. You don't need to worry nearly so much about drama management if your audience will accept a piece ending with a giant foot descending from the sky and squishing the entire set, or the sudden appearance of an extradiegetic character who stops the story because he deems it too silly.

Character-tolerance

Farce is also forgiving of flat characters. Whereas audiences expect major characters in dramas to have complex personalities and inner conflicts that are changed by the story, in farce the story is often a pretext to allow the characters to manifest their own personal forms of dysfunction. When Eric Cartman of *South Park* leads a genocidal campaign against ginger kids that goes horribly awry, no one expects him to learn from the experience; they expect him to return the following week and lead a genocidal campaign against some other random group, which he often does.

[3] See http://www.xtranormal.com/search/?q=dead+parrot for a number of attempts.

Dialog-tolerance

Although farce can often feature very witty dialog (e.g. *The Importance of Being Earnest*), it need not do so. Much farce relies on physical comedy, including slapstick; and early cinematic farces, such as the Keystone Cops films, were entirely silent. When dialog is used, it can be simpler and, like the Xtranormal slow burns, rely on implied irony or even heavy profanity to communicate affect.

Stupidity-tolerance

Finally, there is the fact that farce, whose characters often do inappropriate or counterproductive things anyway, will be inherently more forgiving of characters doing such things because of bugs in the AI. If a hungry character eats their spouse's dinner (or their spouse) rather than their own dinner, it looks like a clever quotation of the Marx Bros., rather than an inference failure.

This is important since, until such time as we have true, broad-coverage common-sense knowledge bases covering social norms, any knowledge-based story system will necessarily have the brittleness typical of knowledge-based systems: it will perform very well within its domains of expertise but degrade as it reaches the boundaries of that expertise. The user will be much more accepting if the genre already leads them to expect such behavior.

Punch and Judy

The *Tragical Comedy or Comical Tragedy of Punch and Judy* is one of the oldest and best known puppet shows in the English language. Derived in part from the *commedia dell'arte* (Punch's name is generally believed to be derived from Pulchinella of the *commedia*), its plot is not fixed, but can be readily adapted by traveling performers to suit the needs of a particular audience.

Although, largely a children's show today, the historical Punch and Judy shows were considerably more adult-oriented, and even political. As with Cartman, much of the pleasure of Mr. Punch derives from his ruthless pursuit of his own appetites, combined with society's utter inability to make him pay for his deplorable behavior. In Collier's historical script (Collier and Cruikshank 2006), Mr. Punch successively beats to death his friend's dog, his own baby, his own wife, his horse, the doctor who tries to treat him after he's injured by the horse, a policeman (beaten but not killed), his own executioner, and the Devil himself. When his wife confronts him over his murder of his own child, Mr. Punch, who wants to have sex with her, replies that she'll soon have another one.

Punch and Judy AI Playset: a Generative Farce

Punch and Judy AI Playset is a narrative God-game in which the player chooses characters and props from the Punch and Judy story world, adjusts their beliefs, desires, and intentions to suit the player's taste, and sets them running to see what happens. The player can also inject new beliefs, desires, and intentions into characters during the action. *Playset* is a work in progress. Its goal is to implement a sufficient system to be able to generate all the interactions found in the Collier's historical version of the story, then use the generativity of the simulation to allow the player to explore other possible variations of the scenes. Instead of killing the baby, Punch might eat the baby, sell the baby for drug money, or use the poor child as a club to bludgeon someone else.

The system is implemented using the [REDACTED] engine, which supports procedural animation and simple physical simulation for when characters hit one another with blunt instruments. Characters are controlled using a custom, in-engine Prolog interpreter.

Each character has both a goal for the scene and an immediate goal. The executive runs steps from the current plan for the immediate goal, replanning when an action fails, and choosing a new goal when the current goal is achieved. The executive also responds to external events, usually actions of other characters. Events are handled by a set of reaction rules (see below) that propose responses along with a priority for the response. The executive runs the highest priority response, if any. Common plans, facts, and reaction rules are stored in a shared knowledge base and can be overridden by rules in the individual characters' knowledge bases.

Detailed example

To give a sense of the operation of the system, its limitations, and its generativity, I will work through an example interaction between Punch and Judy based on the Collier script. This interaction runs in the current version of the system.

The play begins with Mr. Punch wanting to have sex with his wife, Judy. He calls out to her, and while other plot events happen in the meantime, she eventually arrives and he tries to persuade her to kiss him. In the simulated Punch and Judy, Mr. Punch begins with the scene goal:

```
make_whoopee(mrpunch, X)
```

i.e. that Mr. Punch wants to have sex with *someone*. The executive chooses the scene goal as the immediate goal, and calls the planner, an HTN planner based on SHOP (Nau et al. 1999). The planner chooses the method:

```
make_whoopee(Me, Person) ==>
  {fancy(Person)},
  persuade(Person,
           make_whoopee(Person, Me)),
  actually_make_whoopee(Person).
```

Which states that to make whoopee with a person, one must fancy[4] the person, persuade them to make whoopee with you, and then do the actual making of whoopee[5]. This method is matched against Mr. Punch's knowledge base:

```
fancy(judy).
```

to create the binding `Person=judy`, and the planner eventually generates the plan:

```
wait_condition(nobody_speaking)
say_immediately(summon(judy), judy)
wait_condition(nearby(judy))
wait_condition(nobody_speaking)
say_immediately(
   propose(
       make_whoopeee(judy, mrpunch)
   ),
   Judy
)
wait_event(
   accept(make_whoopee(judy, mrpunch))
)
actually_make_whoopee(judy)
```

The executive checks that no other characters are speaking and sends the speech act `summon(mrpunch, judy)` to Judy. The natural language generator converts this to the English text "Judy!", renders it on the screen, and plays it through the voice synthesizer. The Judy character, however, receives the raw logical form of the speech act, not the English text. She chooses the response rule stating that when someone summons you, you go up to them:

```
general_response(
    summon(Summoner, Me),
    goal(answer_summons(Summoner)),
    0
):-
    me(Me).
```

The 0 at the end is the priority of this response. Judy changes her immediate goal to `answer_summons(mrpunch)` and her planner selects the method:

```
answer_summons(Who) ==>
  ensure(nearby(Who)),
  say(query(current_goal(X)), Who).
```

Meaning that to answer someone's summons, walk up to them and ask them what they want, yielding the plan:

```
goto(mrpunch)
wait_condition(nearby(mrpunch))
wait_condition(nobody_speaking)
say_immediately(query(current_goal(X))
                mrpunch).
```

She walks up to Mr. Punch and sends him the speech act:

```
query(judy, mrpunch, current_goal(X))
```

which is rendered by the NL generator as "what do you want?" Mr. Punch chooses the response rule:

```
general_response(
  query(Querier, Me, Query),
  do(say(answer(Query), Querier),
  0
) :-
  me(Me), Query.
```

which runs `Query`, whose value is the expression `current_goal(X)`, as a Prolog goal, resulting in the binding:

```
X=make_whoopee(judy).
```

And sends the the speech act:

```
answer(
   current_goal(make_whoopee(mrpunch,
                             judy))
)
```

back to Judy. This is rendered by the NL generator as "I want to make whoopee with you." When Judy reaches Mr. Punch, he continues his plan and sends her the speech act:

```
  propose(make_whoopee(judy, mrpunch))
```

(rendered "Let's make whoopee!") and she responds either:

```
agree(make_whoopee(judy, mrpunch))
```

[4] The British English term for being attracted to someone.

[5] This presently, involves the two characters jumping up and down, yelling "whoopee!"; we're keeping it clean for the kids.

(rendered as “okay”), or:

```
refuse(make_whoopee(judy, mrpunch))
```

(rendered as “no!”). If she agrees, both characters jump up and down, yelling "whoopee!"

Preliminary implementation

Preliminary versions of the core AI system (planner, executive, event handler, and natural language generator) are running, but implementing the props, set, and procedural animations for character actions are expected to take the rest of the summer. Sound effects, e.g. for collisions, would also be useful. An appraisal system, (Ortony et al. 1990) and some sort of system to track inter-character relationships will be added, as well as a graphical front-end to allow players to choose character goals and beliefs without having to write Prolog code. The current plan is to let users specify goals and beliefs in natural language, since the existing parser-generator supports auto-completion of partial sentences, continually showing users what types of sentences they can type.

The planner is based on the SHOP (Nau et al. 1999). As its authors point out, SHOP already uses a Prolog control strategy, so our implementation simply macro-expands methods into Prolog rules. This speeds execution and allows the inclusion of arbitrary Prolog code in methods, when necessary.

World states are represented using STRIPS-style add and delete lists, but are coded as situations in the situation calculus to permit more general representations if needed. The initial situation for the planner is the situation constant 'now', specified by the axiom:

```
holds(F, now) :- F.
```

which states that to determine if fluent F is true now, run F as a normal Prolog predicate. Prolog rules for fluents can then call directly into the game engine to access the simulation state. For example, the fluent `nearby(X)` used above is defined by the rule:

```
nearby(Obj) :-
   me(Me),
   distance(property(Me, 'Position'),
            property(Obj, 'Position'))
      < 2.
```

The natural language parser-generator uses a definite clause grammar with Montague semantics (Pereira and Shieber 1987). This is simplistic, limited in scope, and not at all fluent. But again, the idea is to use player expectations produced by the genre to paper over limitations in the technology. Speech synthesis is performed by translating into SSML and rendering using the Microsoft Speech SDK. This has the major disadvantage that the SDK ships with only 1 voice under Windows 7, so character voices are distinguished only by pitch.

Related work

The core argument of this paper follows in the tradition of writers such as Montfort (Montfort 2005) and Wardrip-Fruin (Wardrip-Fruin 2011) who have analyzed the aesthetic affordances of particular technologies. This is a similar project, but is focused on guiding the design of new artifacts rather than critiquing existing ones.

As a system, *Punch and Judy AI Playset* is similar to Mateas et al.’s *Terminal Time* (Mateas et al. 1999), both in its deliberate farcical qualities and its exploration of the generative qualities of symbolic AI systems. However, *Playset* is fundamentally an emergent narrative system in the tradition of the Woggles (Bates 1994), The Sims 3 (Evans 2009), and FearNot! (Aylett et al. 2005).

Storytelling RPGs, a sub-genre of tabletop RPGs, are arguably the most successful emergent narrative systems to date, albeit non-electronic ones. In these games, players create a set of characters and relationships, then create a story through collaborative gameplay. For example, in *Fiasco* (Morningstar 2009), players begin by incrementally selecting player relationships, locations, and props, then improve a farcical drama in the style of the films of the Cohen Brothers. In *Dread* (Barmore et al. 2005), players answer questionnaires in character as a way of creating their character’s personality and back story, then players improvise a horror story based on a predetermine set of major plot points.

Technologically, *Playset* uses character-centered planning, similar to TALE-SPIN (Meehan 1977). More sophisticated story planning systems (Ware and Young 2011; Porteous and Cavazza 2009; Young and Riedl 2005; Reidl 2010) would likely produce better plotting, however, as argued above, plotting is tangential to *Punch and Judy*. Its character architecture is similar to Shakey’s Planex (Fikes et al. 1972) and Gat’s Atlantis (1992), although it also shares some features with Golog (Levesque et al. 1997). Reactive planners such as Hap (Loyall and Bates 1991) or ABL (Mateas and Stern 2002) are more common in interactive narrative systems, and more sophisticated than *Playset’s* executive. These are substantially similar to the HTN planner in used in *Playset*; however they rely on guard conditions rather than full forward simulation to choose between methods.

Conclusion

The long-term development of interactive narrative technology will require us to build systems that real users can play. While our ultimate goal as a community remains the development of technologies to allow sophisticated characters and plots in a wide range of genres, this is not a realistic target in the short run given how easily a player's sense of immersion in the story world can be broken when the system makes even small mistakes.

The goal of *Punch and Judy AI Playset* is to show that through careful choice of genre and tone, even 1980s AI technology can produce satisfying experiences involving generative planning and natural language. Such systems can then form a "beachhead" from which component technologies can be improved while still being aesthetically satisfying for users. As component technologies improve, we can gradually branch out to other styles and genres.

References

Abbott, H. Porter. 2008. *The Cambridge Introduction to Narrative*. 2nd ed. Cambridge, UK: Cambridge University Press.

Aylett, Ruth et al. 2005. Fearnot!: An experiment in emergent narrative. In *Intelligent Virtual Agents (LNCS, vol. 3661)*, 305-316. Heidelberg: Springer.

Barmore, Nat et al. 2005. *Dread*. The Impossible Dream.

Bates, Joseph. 1994. "The Role of Emotion in Believable Agents." *Communications of the ACM* 37 (7): 122-125.

Cavazza, Marc et al. 2007. Madame Bovary on the Holodeck: Immersive Interactive Storytelling. *ACM Multimedia 2007*. Augsburg, Germany: ACM Press.

Cleese, John, and Graham Chapman. 1969. "Full-Frontal Nudity", Monty Python's Flying Circus, ep. 8. British Broadcasting Corporation.

Collier, John Payne, and George Cruikshank. 2006. *Punch and Judy: A Short History with the Original Dialogue*. Dover.

Davis, Jessica Milner. 2002. *Farce*. Updated. Piscataway, NJ: Transaction Publishers.

Duchartre, Pierre Louis. 1996. *The Italian Comedy*. Dover.

Evans, Richard. 2009. AI Challenges in Sims 3. In *Artificial Intelligence and Interactive Digital Entertainment*. Stanford, CA: AAAI Press.

Fikes, Richard et al. 1972. "Learning and Executing Generalized Robot Plans." *Artificial Intelligence* 3: 251-288.

Gat, Erann. 1992. Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 809-815. San Jose, CA: AAAI Press.

Gordon, Mel. 2001. *Lazzi: The Comic Routines of the Commedia dell'Arte*. PAJ Publications.

Levesque, Hector J. et al. 1997. "GOLOG: A Logic Programming Language for Dynamic Domains." *Journal of Logic Programming* 31 (1-3): 59-83.

Loyall, Bryan A, and Joseph Bates. 1991. *HAP: A Reactive, Adaptive Architecture for Agents*. Pittsburgh: Carnegie Mellon University School of Computer Science.

Mateas, Michael et al. 1999. Terminal Time: An Ideologically-biased History Machine. In *Proceedings of the AISB'99 Symposium on Creative Language: Humor and Stories*, 69-75.

Mateas, Michael, and Andrew Stern. 2002. "A Behavior Language for Story-Based Agents." *IEEE Intelligent Systems* 17 (4): 39-47.

Mateas, Michael, and Andrew Stern. 2005. Façade.

McCoy, Joshua et al. 2011. Comme il Faut: A System for Authoring Playable Social Models. In *Proceedings of the 7th AI and Interactive Digital Entertainment*, ed. Vadim Bulitko and Mark O. Riedl. Stanford, CA: AAAI Press.

Meehan, James R. 1977. TALE-SPIN, an interactive program that writes stories. In *Proceedings of the 5th international joint conference on Artificial intelligence*, 91-98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Montfort, Nick. 2005. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge, MA, USA: MIT Press.

Morningstar, Jason. 2009. *Fiasco*. Durham, NC: Bully Pulpit Games.

Nau, Dana et al. 1999. SHOP: simple hierarchical ordered planner. In *Proceedings of the 16th international joint conference on Artificial intelligence*, 968-973. Stockholm, Sweden: Morgan Kaufmann Publishers Inc.

Ortony, Andrew et al. 1990. *The Cognitive Structure of Emotions*. Cambridge, UK: Cambridge University Press.

Pereira, Fernando C. N., and Stuart Shieber. 1987. *Prolog and Natural Language Analysis*. Brookline, MA: Microtome Publishing.

Porteous, Julie, and Marc Cavazza. 2009. Controlling nar- rative generation with planning trajectories: the role of constraints. In *Proc. of 2nd Int. Conf. on Interactive Digital Storytelling*.

Reidl, Mark O. 2010. "Story Planning: Creativity Through Exploration, Retrieval, and Analogical Transformation." *Minds and Machines* 20 (4).

Wardrip-Fruin, Noah. 2011. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Cambridge, MA, USA: MIT Press.

Ware, Stephen G., and R. Michael Young. 2011. CPOCL: A Narrative Planner Supporting Conflict. In *The Seventh Annual International Conference on Artificial Intelligence in Interactive Digital Entertainment*. Stanford, CA: AAAI Press.

Young, R Michael, and Mark O Riedl. 2005. Integrating plan-based behavior generation with game environments. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. Valencia, Spain: ACM.