Generating Natural Language Retellings from *Prom Week* Play Traces

Christopher Antoun[1], Matthew Antoun[1], James Owen Ryan[1,2], Ben Samuel[2], Reid Swanson[1], and Marilyn A. Walker[1]
[1] Natural Language and Dialogue Systems Lab
[2] Expressive Intelligence Studio
University of California, Santa Cruz
{cantoun, mantoun}@ucsc.edu; {jor, bsamuel, reid, maw}@soe.ucsc.edu

ABSTRACT

Because they have massive state spaces, authorial burden is especially pronounced in games that are underpinned by rich simulations; as such, this class of games represents perhaps the strongest use case for procedural content generation. In this paper, we outline ongoing work in which we procedurally generate natural language retellings of playthroughs of the social-simulation game *Prom Week*. We do this by parsing play trace files to create semantic encodings, which are then passed to an existing surface realizer that produces the retellings. Herein, we provide illustration and discussion of our methods as well as an evaluation of the retellings. While our work here concerns a specific game, we envision far-reaching implications: if a game engine can produce natural language describing what has happened in the world, then content such as in-game journals, quest logs, histories, and even character dialogue may be generated automatically.

Categories and Subject Descriptors

K.8.0 [**General**]: Games

General Terms

Algorithms, Design

Keywords

Authorial burden, natural language generation, procedural content generation, story recognition

1. INTRODUCTION

While for all games procedural content generation represents a promising way of reducing authorial burden, this is especially the case in games that can get into very many states. The more states a game can get into, the more content that must be authored so that the system can express these states to the player. These circumstances are perhaps most pronounced in *simulationist games*, which (due to combinatorial aspects of their underlying simulations) tend to exhibit huge state spaces. These games are celebrated for the complexity of their simulations, but when a system cannot express many or most of the states that its simulation yields, the latter will appear to the player much simpler than it in fact is—this has been called the *Tale-Spin effect* [30]—a symptom only alleviated by more content.

Most commonly, the content at issue is dialogue. Especially in role-playing games, which are known for their branching narratives, the quantity of dialogue required to cover the potential story configurations can be immense. Consider the case of *L.A. Noire* [28], a AAA title for which a team of authors produced a script exceeding 2,000 pages [9]. This accumulation of dialogue is commensurate to that of ten full-length motion pictures—all this for an interactive experience that is essentially on rails. The game, inasmuch as its narrative is concerned, cannot get into that many states—how else could it have a *script*?

Analogous to the problem of authoring dialogue for branching narratives is that of generating text from game state: for example, the production of retellings and summarizations. Such texts are likewise subject to the requirement that they reflect game state, whether it is the product of player choice or pure simulation. Further, they have the potential to express complexities of the underlying game state that may not otherwise even be visible. Simulationist games, with characteristically massive state spaces, make this authoring challenge particularly apparent. Games such as the seminal work of procedural generation and world building called *Dwarf Fortress* [1] and the social physics simulation *Prom Week* [18], have complex and sometimes hidden inner workings. In these settings, textual representations of game state could naturally be expressed as in-game artifacts, such as journals, histories, or recapitulations.

The crux of the matter, whether authoring dialogue or a report of events, is that the content must articulate the underlying game state. Manual authoring quickly becomes impractical in games with massive state spaces as the authorial burden explodes along with the possibility space. The fact of the matter is that, in a simulationist game, one's only hope for producing a base of content sufficient to give even marginal coverage to its state space is to author the majority of such content procedurally.

Beyond massive state spaces that command massive content bases, however, lies another serious challenge endemic to the simulationist genre. Games whose narratives emerge from simulations currently have no way of understanding the very narratives they support. Stories arise in many simulationist games only incidentally; they are remarkable streams of an otherwise overwhelming profusion of events. But while humans who play games are capable of recognizing which event sequences are storylike, the systems themselves are not. So as story generation in simulationist games works by a sort of event combinatorics, a major issue becomes *story recognition*. How does one make a system that can discern stories embedded in the morass of data that its simulation produces?

In this paper, we present a project that works to meet both of these simulationist challenges. By treating play trace data from the social-simulation game *Prom Week* as an encoding of the story of gameplay, we are able to automatically generate a deeper, computable semantic representation of this narrative—and thereby a natural language retelling of it. These procedurally generated retellings are intended as game content that, by virtue of their ability to express arbitrary underlying states, could work to substantially reduce authorial burden in this game. Further, our treatment of gameplay as story brings to light pivotal questions surrounding the challenge of recognizing stories from among simulation data. While our work here concerns a specific game, *Prom Week*, we envision far-reaching implications: if a game engine can produce natural language describing what has happened in the world, then content such as in-game journals, quest logs, histories, and even character dialogue may be generated automatically.

The rest of this paper is organized as follows. In the next section, we discuss related work. Section 3 outlines our general method and system pipeline. In Section 4, we attempt to evaluate the generated retellings, but end up (for better and for worse) learning more about nuances surrounding the evaluation of generated game artifacts than we do about the quality of our system's output. Finally, we conclude in Section 5.

2. RELATED WORK

Several projects have explored generating narrative prose outside the context of a digital game or interactive story (e.g., [3]). Work that has used expressive natural language generation (NLG) for these media has typically targeted the generation of character dialogue [4, 25, 23, 29, 14, 11]. In a recent project of our own, we annotated human-authored *Prom Week* dialogue to specify which lines express which aspects of game state and then recombined the marked-up dialogue to produce new exchanges that express new state combinations [26]. While our task there was to explicitly map pieces of human-authored content to the aspects of underlying state that they express, here we are attempting to generate totally new content that is tailored to active game state. As an effort in expressive NLG, this project is related to the above work, but we more specifically situate it among a handful of other projects that have tackled the problem of constructing a story representation from game state or other structured data. In this section, we recount these, as well as other prior work that is foundational to the method we introduce in this paper.

Hoek et al. develop a system, similar in overall concept to our own, for generating textual narratives of the events of a simulation-based serious game [10]. The authors propose a pipeline that begins with the automatic generation of a semantic representation from game state (that they call fabula) and results in a narrative retelling of events (sjuzhet). The focus of their work is on the production of the sjuzhet from the fabula, rather than the automatic generation of the fabula from game state. In fact, they take the fabula as a given and create it manually.

Our work automatically encodes game state fabula using a formalism called the Story Intention Graph, or SIG [6]. The SIG is a powerful and flexible semantic narrative representation with applications from summarization to question answering. It might also be useful for the computational identification of interesting stories from large sets of events: by defining a set of SIG subgraphs similar to Lehnert's plot units [13], Elson is able to reliably detect analogies between fables [7].

And of course, the SIG is also anchored in language: Scheherazade, the SIG encoding tool, is able to render surface text from encodings, a feature we make use of in this work. But this is not the only path from SIG to natural language: Rishes et al. describe the EST [24], a system for producing different tellings of a story from its SIG representation. They are able to introduce lexical and syntactic variations in the natural language realization of a SIG-encoded story by bridging to PERSONAGE [15], an NLG engine that can generate multiple variations of the same utterance based on changes to a personality model. In follow-up work, they explore using this system specifically to automatically generate variations in character dialogue for simulationist games [14]. Such a system might naturally find a place in our pipeline in future versions of this work.

In the interactive fiction domain, Montfort conceives of the underlying world simulation as fabula, distinct from the sjuzhet, the expression of it that is the game [20]. He discusses possible narratological variations that resonate with our future plans to employ the EST.

Previous work has also examined using structured data resulting from sports games, such as box scores and chess moves. Allen et al. present a system for transforming sports recaps, like box scores and play-by-play accounts, into narratives [2]. Their system can select certain events to highlight, but it does not operate over or expose the salient details of a rich underlying representation. Gervás addresses the problem of finding interesting retellings for chess replays by focalizing on the individual pieces and evaluating the subset of events visible to each piece [8].

In this work, we make use of *Prom Week*, a game about high-school students and their social interactions that is underpinned by a social artificial intelligence system called *Comme il Faut* [16, 19, 17]; it is freely available online and can be played in a web browser. *Prom Week*'s basic unit of interaction is the *social exchange*, which encapsulates things like asking someone on a date or trying to annoy them. Players are tasked with effecting some change to the state of the world—for instance, getting two characters to begin dating—by having characters engage in social exchanges. In the game's simulation, characters' traits and volitions, among other factors, affect the way social exchanges play out.

3. METHOD

Here we describe the implementation of a pipeline to perform the conversion from *Prom Week* play trace data into a semantic encoding and then into generated natural language. A *Prom Week* play trace is a detailed record of the sequence of events and changes to game state that occurred in a playthrough. We consider these data as a fabula and map them to a SIG, described in detail in Section 3.4. For this work, we use Scheherazade, the SIG encoding tool, to automatically generate surface text from this representation.

In broad terms, the pipeline accepts *Prom Week* play trace files as input and generates a natural language document as output. The following example shows the output of our system for one of the traces in our collection.

> *A sexy student named Oswald expressed concern and wooed a second student named Cassandra. Cassandra allowed Oswald. Cassandra smiled and wooed Oswald. Oswald discouraged Cassandra and blamed her. Oswald smiled and wooed Cassandra. Cassandra allowed Oswald and blushed. Cassandra expressed concern and asked Oswald to date Cassandra. Oswald didn't agree. Cassandra attempted to plan to meet a third student named Nicholas. Nicholas agreed and congratulated Cassandra. Cassandra expressed anxiety and confided feelings to Oswald. Oswald listened to Cassandra and hugged Cassandra.*

The pipeline, depicted in Figure 1, is structured as follows. The first stage is an XML parser which generates an intermediate representation of the play trace data. This is simply a data structure containing a list of characters and the social interactions that occurred between them.

The second stage augments the intermediate data structure by cross-referencing *Prom Week*'s data files in order to flesh out details of the interactions. The third stage renders the intermediate data structure as a valid VGL file (this is the SIG file format) by consulting hand-authored mappings. In the final stage of the pipeline, Scheherazade generates a natural language realization of the SIG.

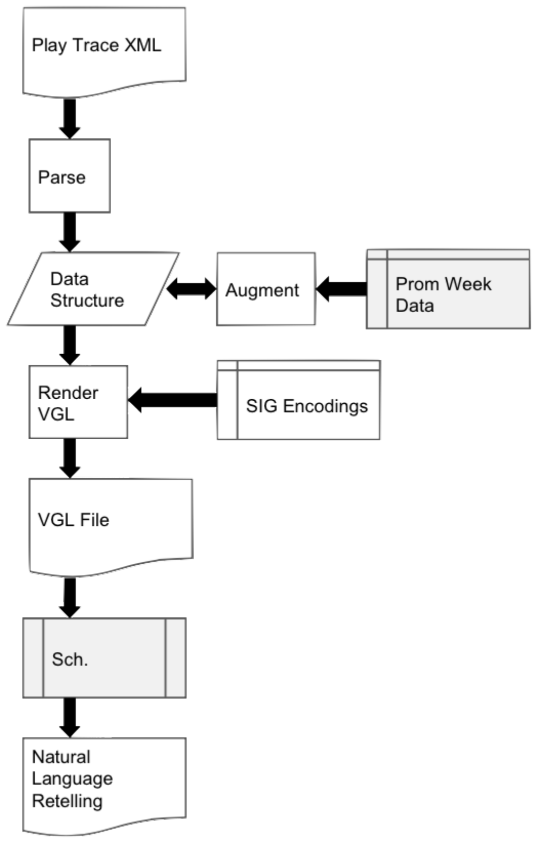

Figure 1: The pipeline for translating *Prom Week* play trace XML into natural language via SIG encoding.

3.1 *Prom Week*

As described previously, *Prom Week* games can be seen as a sequence of social exchanges designed to manipulate the game's underlying social state. Crucially, for any given playthrough, *Prom Week* records the set of exchanges enacted by the player in a structured play trace format.

These play traces are perfect for reconstructing the high-level events of a playthrough. They are standard XML files that document the sequence of who tried to do what with whom and whether or not they were successful [27, 21].

While we didn't have access to the game's engine, we obtained the data files that define each of the social exchanges. The file AskOut.xml, for example, encodes 43 different ways in which a character might ask another out. Each of these instantiations of the Ask Out interaction has its own set of pre-conditions, results, animations, and dialogue.

We also had access to a vast collection of play traces. We drew our development data from a collection of 11,323 *Prom Week* play traces in XML format, hand-selecting 10 traces for relative coherence (*Prom Week* games are often chaotic) and the limited set of social exchanges they included.

3.2 Baseline Encoding

The first stage of the pipeline parses a *Prom Week* play trace file and creates a data structure containing a list of characters and social interactions that occurred in the game. For each social interaction, we capture the initiating and responding characters, the time-step at which it occurred, and whether or not the interaction was successful.

Considering the input described by Figure 2, for example, we can see that two major events happened: at time 0, Doug tried to Make Plans with Cassandra; and at time 1, Doug tried to Confide In Chloe.

We can determine if each of these was successful by looking at the `responderScore` attribute. If the value is negative, the responder has rejected the action. Otherwise, the action succeeded.

We define a baseline encoding as a rendering of each of the social exchanges present in the play trace. Setting aside the question of actually producing the semantic encoding, the input play trace XML file is essentially all that's required for a baseline encoding.

> *A student named Doug attempted to plan to meet a second student named Cassandra. Cassandra didn't agree. Doug confided feelings to a third student named Chloe. Chloe listened to Doug.*

A baseline encoding tells us what happened, but doesn't provide much detail. We can do better by augmenting the encoding to bring to light other details of the exchange. In a play trace, each instance of a social exchange contains an

```
<LevelTraces startTime="2" endTime="2"
     name="Friday: The Prom" storyName="Doug"
     juice="56" type="endOfStory">
  <SFDB>
    <SocialGameContext gameName="Confide In"
         initiator="Doug" responder="Chloe"
         initiatorScore="10" responderScore="19"
         time="1" effectID="2"
         chosenItemCKB="retro phone">
    </SocialGameContext>
    <SocialGameContext gameName="Make Plans"
         initiator="Doug" responder="Cassandra"
         initiatorScore="-1" responderScore="-3"
         time="0" effectID="10">
    </SocialGameContext>
  </SFDB>
</LevelTraces>
```

Figure 2: An excerpted play trace. The `SocialGameContext` elements describe the social exchanges that occurred (a MAKE PLANS between Doug and Cassandra, followed by a CONFIDE IN between Doug and Chloe). The `effectID` attribute serves as a key for looking up the details of each exchange: character traits, animations, and result statuses.

`effectID` attribute, which allows us to gather just such information.

3.3 Augmentation

Recall that each social exchange has a corresponding data file that describes a number of concrete ways in which an exchange might play out. The `effectID` attribute ties a social exchange to a particular instantiation as described by *Prom Week*'s data files. By cross-referencing the `effectID` of a social exchange with its associated data file, we can extract all manner of information with which to enrich our retelling. For example, we might consult the sequence of animations played during the interaction, or look at its effects on the initiator or responder.

In Figure 2, Doug's attempt to MAKE PLANS with Cassandra was instantiated with `effectID` 10. By consulting the corresponding instantiation in the MAKE PLANS data file, we are able to determine that Doug feels lonely as a result of Cassandra's rejection and that he's concerned when he goes to tell Chloe what happened. After augmentation, our output becomes:

> *A* ***shy*** *student named Doug* ***smiled and*** *attempted to plan to meet a second student named Cassandra.* ***Doug felt loneliness,*** *and Cassandra didn't agree. Doug* ***expressed concern and*** *confided feelings to a third student named Chloe. Chloe listened to Doug* ***and smiled.***

Note briefly that Doug seems to feel lonely even before Cassandra turns him down. This is a quirk of Scheherazade's sentence aggregation which we will discuss in more detail later. In terms of the fabula, Doug is lonely *after* Cassandra turns him down.

We pursued three strategies for augmenting our representation: collecting animations, status results, and traits.

To augment our representation with animations, we consulted the particular `Instantiation` node used by the social exchange in question. `Instantiation` nodes contain the specific lines of dialogue exchanged between two characters as well as the animations accompanying each line. In particular, the body animation, facial animation, and facial state are specified for both the initiator and responder.

To select animations to describe in our retelling, we used a simple algorithm. First, we ignored all animations like *talking* and *idle*, reasoning that they did not add anything to the retelling. For the initiator, we consulted only the first line of dialogue and attempted to choose the body animation. If that was not present, we fell back to the facial state. For the responder's animation, we used the same process on the last line of dialogue. In our example, this leads to Doug smiling as he attempts to make plans with Cassandra.

Status effects are results of the social exchange. They are encoded as `ChangeRule` predicate elements in the XML. Each instantiation has a number of `Predicate` nodes that describe changing social state. Predicates can have many types, but we focused on `status` predicates. These assign status effects such as *confused* or *feels out of place* to either the initiator or the responder. In our example, Doug's feelings of loneliness are the results of a `status` predicate.

Each instantiation also has a block of `ConditionRule` predicates that describe various prerequisites for that particular instantiation of the social exchange. These can take many forms, but we looked for `trait` predicates that encode ideas such as *the initiator is introverted* or *the responder wears a hat.* Doug's characterization as shy in our example is a result of this type of augmentation.

There are other possibilities for augmentation too. Capturing dialogue is an obvious one, but the SIG formalism does not currently allow for the encoding of direct speech. Further, *Prom Week* dialogues are defined using placeholders, variables, and even randomly selected elements. While this approach serves *Prom Week* very well, it complicates our task considerably, so we decided to leave it for future work. Additionally, each instantiation features a description of the *performance realization*, which is essentially a short string describing the events. Because this text is a mixture of natural language with placeholder values is very difficult to use directly. Finally, it is possible to track changes to the game's representation of social state as a result of each action. One could, for instance, observe how friendship values fluctuate over time.

3.4 The SIG

The penultimate stage transforms the intermediate representation into a valid Story Intention Graph.

The SIG is a semantic graph structure of discourse relations for modeling narrative, not only the moment-by-moment events of the fabula, but also the intentions and goals of its agents. These two domains are represented in different layers of the SIG: the timeline layer and the interpretative layer. Our proof of concept does not model character goals—indeed, doing so is quite difficult—but describes the proceedings in the timeline layer, a theoretically direct representation of the game's observable action.

In order to translate the intermediate representation of the play trace into a SIG, we procedurally assemble hand-encoded SIG subgraphs describing social exchanges, character traits, animations, and statuses.

For practical reasons, we chose to encode the subset of *Prom Week*'s social interactions and features described in our selected set of play traces. We created prototypical SIG subgraphs for each possible component of our retellings. The SIG's moorings to WordNet and VerbNet, although essential to the production of natural language, made this surprisingly difficult. As every SIG node is semantically grounded, the graphs can articulate only very rigid accounts. Our encodings required a liberal measure of interpretation and careful diction.

Social exchanges are the action verbs of any *Prom Week* playthrough; by themselves they represent a baseline retelling, so let us examine them first. We began the encoding process by defining placeholder characters: a male student and female student. In the timeline layer, we encoded each social exchange (with one exception) as three separate story points, or nodes on the timeline layer: an attempted action, a success, and a failure.

The social exchange Share Interest is an instructive example. The first thing to note is that in *Prom Week*, each of the social exchanges has many different instantiations, which include particular lines of dialogue, animations, and the like: these are the player's experience of the action, but we set them aside for the moment in the interest of producing a baseline. So Share Interest can be various specific things, but how do we describe it in a general sense? It is quite difficult to make vague statements in Scheherazade! The system required us to cast the action in concrete terms, so we framed it as one student speaking to the other about something. Unfortunately, Scheherazade is very particular about what that something can be. We could not figure out how to encode the idea of an interest or hobby, let alone in such a way that it could belong to a character and be selected as the thing being spoken about. Our awkward encoding was finally realized as, "Chloe spoke about interest to Doug."

Describing the specific details of these actions is important to the quality of a retelling. Traits, animations, and statuses provide some of this detail. In comparison with social exchanges, encoding traits was straightforward. We created a prototypical student character and added as attributes all of the traits that we needed to represent. The only difficulty here was in translating complex traits such as *wears a hat*. We would have settled for "behatted", but the best we could do was "crowned".

The nature of most animations as simple actions made their encoding relatively easy as well. Effect statuses were difficult. Scheherazade technically allowed us to encode statuses such as *annoyed*, *angry*, *confused*, and so on as statives, but doing so presented two related problems. Scheherazade requires that the duration of statives be explicitly modeled, and it was not clear how long these effects lasted in the game. Scheherazade disallows the same stative to overlap with itself, so it was impossible to encode a student becoming angry on one turn and (more) angry on the next. Additionally, Scheherazade's natural language realization of statives is explicit in an unnatural way. We finally chose to encode the statuses as expressions of some quality, so that rather than becoming lonely, for example, a student feels loneliness.

Having used Scheherazade to create SIG nodes for each feature, we were able to harvest the underlying representation, essentially serialized Java API calls, to build up a dictionary of generic snippets. With this dictionary, we iterate through the events in our intermediate representation and use snippets to produce SIG nodes: each snippet is instantiated with the appropriate initiator, responder, and time step and becomes a node in the timeline layer of our SIG.

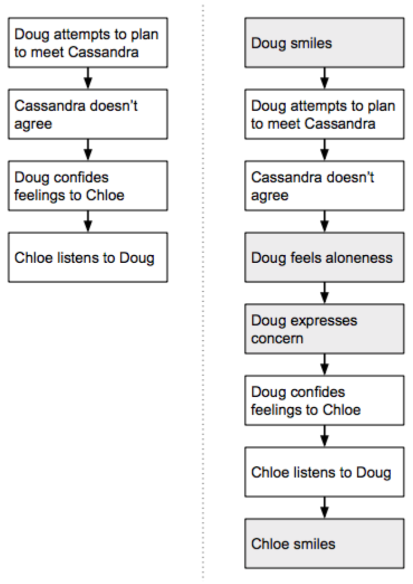

Figure 3: A baseline SIG (left) and an augmented SIG, generated from play trace XML. A baseline SIG encodes social exchanges as attempts and results. The augmentation process incorporates traits, animations, and statuses as additional nodes (shaded).

By way of example, consider Figure 3. If our intermediate representation contains a social exchange where Doug fails to Make Plans with Cassandra, we can grab the Make Plans attempt snippet and the Make Plans rejection snippet and slot in Doug as the initiator and Cassandra as the responder. The augmentation process may have given us several other snippets to instantiate as well. We string these nodes together in the timeline layer to produce a SIG.

At the end of this stage we have everything we need to produce a valid VGL file, which is the format used by Scheherazade to serialize a SIG.

3.5 Natural Language

To obtain natural language output, we relied on Scheherazade's built in verbalizer. We wrote a small command-line utility that uses Scheherazade's API to invoke the verbalizer with the same style parameters used by the GUI. The utility accepts one or more VGL files from the previous stage and prints natural language output.

4. EVALUATION

We present an initial evaluation and discuss its flaws. While we don't yet consider the pipeline complete—there are still several outstanding issues with our surface text output including aggregation, vocabulary, and repetitive results—we were eager to undertake a preliminary evaluation to see if we could learn anything about our approach.

Ideally, a retelling of gameplay should be evaluated in light of the actual experience of playing. At the time of our experiment, this was infeasible. Without on-demand access to *Prom Week* play traces there was no way for us to link output from our pipeline with the experience of the gameplay from which it resulted.

Instead, we turned to Amazon Mechanical Turk to gather feedback about our output. Our goals were two-fold: we hoped to gain a general understanding of the characteristics of our output and to investigate whether the augmentation process meaningfully affected the measurements of those characteristics.

We generated both baseline and augmented output for the ten development play traces, giving a total of twenty retellings. In order to avoid perceptions about characters carrying from text to text, we used a simple script to replace character names by selecting from a list of the most popular names from the previous decade.

Each of these retellings was presented to nine raters who were instructed to read the text and answer following questions:

- How clearly can you imagine the events taking place?
- Is the text coherent?
- How easily were you able to infer the characters' motivations?
- Is the text a story? Use your own definition of what constitutes a story or narrative.

Clarity, coherence, and motivations were judged on a five-point scale ranging from 0 to 4, and we provided textual interpretations for the high, low, and middle ratings. The story criterion was a binary.

The twenty texts yielded 180 ratings. Table 1 shows the mean ratings for the four measures grouped by baseline retellings, augmented retellings, and both combined. The results were normalized to fall between 0 and 1.

We first considered the mean ratings from the group of all retellings and then attempted to find significant differences between the ratings of baseline texts and the ratings of augmented texts.

4.1 Overall Results

Considering the mean ratings for the group of all retellings, we observed acceptance of the text as a story at a rate of just less than 64 percent. The mean scores for clarity, coherence, and ability to infer character motivations hovered just above the midpoint on our scale.

For clarity, the low end of the scale corresponds to raters not being able to picture the events taking place. The mean score of 0.56 (0.27 standard deviation) corresponds roughly to events being ambiguous and seems to indicate that the retellings would benefit from the addition of more detail.

The mean coherence rating of 0.50 (0.25 standard deviation) suggests that the texts were found to be somewhat coherent. However, this is more difficult to interpret. In retrospect, our phrasing of the question was too coarse: it did not allow us to distinguish between coherence of the event sequences and their textual renderings. We suspect both factors are at play. *Prom Week* games are a series of social interactions that are not necessarily causally related to each other, even if they are justified by the game's underlying model of social state. And Scheherazade's sometimes archaic spellings ("annoied" for "annoyed") and stilted sentence constructions can seem strange.

Table 1: Normalized mean ratings of the generated retellings for the four evaluation criteria by group.

	Baseline	Augmented	All
Clarity	.56	.55	.56
Coherence	.51	.50	.50
Motivations	.56	.58	.57
Story	.64	.63	.64

The mean of 0.57 (0.26 standard deviation) for motivations indicates that, on average, raters found that motivations were present but uncertain. Considering the simplicity of our system, we interpret this as a positive result. It suggests that the retellings leave adequate room for the reader to ascribe a degree of depth to the characters.

4.2 Baseline versus Augmented

We hypothesized that augmenting the retellings would lead to increased ratings across the four criteria. We thought that the extra detail provided by augmentations—animations, status results, and traits—would provide hooks for the reader's imagination. For instance, in our running example, Doug's loneliness and concern, absent from the baseline text, are made explicit in the augmented rendering. And what is Chloe smiling about? Is she happy it didn't work out for Doug? We can't be sure, but the smile is a clue that there is some mental process at work.

Our results did not bear out our hypothesis. For each of the four measurements—clarity, coherence, motivation, and story—we ran a paired difference test between the 90 baseline and 90 augmented ratings. We used a two-tailed dependent t-test but did not find a significant difference between the groups for any of the measurements.

We partially account for this by pointing to quirks in Scheherazade's surface realizer. In particular, it tends to do funny things with sentence aggregation. We noted one instance in the example earlier: Doug's loneliness is told before the event causes it. In the development phase, we decided to overlook these obvious temporal inversions in favor of more flowing text. In retrospect, it is clear that this was a major hurdle for readers who were not primed to expect such inversions. So while the augmented texts contained more detail than the baseline retellings, they were also arguably less natural.

4.3 Limitations and Future Prospects

We are aware of several issues with the evaluation presented here. We noted problems with the formulation of the questions we asked, and we acknowledge that nine raters per retelling is a smaller sample size than we would have liked.

But perhaps the most significant limitation of the experiment is that the retellings are divorced from any experience of gameplay. Indeed, participants in our experiment had no way of knowing that the texts they read were retellings of game events or at all related to *Prom Week*: we simply presented the retellings free from any context or background.

In some sense, our experiment had the undesirable and unintended effect of evaluating *Prom Week* as a story generator. Here, it is crucial to understand that stories in this game are intended to emerge dynamically from player behavior; this aspect has been described by its developers using the metaphor of the player *sculpting* a story [22]. So while *Prom Week* playthroughs may be chaotic or incoherent, this

Table 2: Coverage describes the system in terms of the percentage of selected game state features that it can express.

Category	Encoded	Total	Percentage
Characters	14	14	1.00
Social Exchanges	17	37	.45
Animations	25	74	.33
Status Results	10	41	.24
Traits	7	78	.08
Total	73	244	.30

will more likely happen as the result of player actions than any system behavior. And, in any event, the experience of play imposes order on that chaos. We are generating semantic encodings of events that unfolded in gameplay. Whether or not they stand by themselves as good stories is not the point; we are interested in the degree to which they reflect what the player experienced. We would like to know whether we can produce some artifact that can complement the experience of play.

To address these concerns we are working on designing a new evaluation. As a first step we have begun to lay the groundwork for tighter integration with *Prom Week* itself. This will allow us to gather and process play traces immediately after play sessions with the goal of presenting participants with a natural language retelling of the very events they brought about in play.

Because we're creating a semantic representation of game state, we think it's important to examine any resulting artifacts in the context from which they are derived. We will ask participants to play a *Prom Week* level and then, using the trace from that session, we will generate a retelling. We will then ask participants to rate how well the retelling reflects their experience playing the game.

4.4 Coverage

Another measure by which we evaluate our work is coverage, the quantity of events and characteristics that our system recognizes divided by the quantity that it will actually describe.

We note that the pipeline is resilient to missing encodings: if it can't find a mapping for a particular social exchange, for example, it will simply omit it from the retelling.

Table 2 shows encodings by category. Only character and social exchange encodings are required for baseline retellings; to cover all play traces at baseline level would only require an additional 20 encodings.

Of all the building blocks we identified, we encoded 73—about 30 percent. *Prom Week* games vary wildly in terms of content, so in absolute terms, this figure only gets us complete coverages of about 280 of the 11,323 traces. Encoding the remaining 171 items would allow us to render any *Prom Week* play trace at the current level of fidelity. This is a relatively small amount of work for the generation potential.

5. CONCLUSION

Because they have massive state spaces, authorial burden is especially pronounced in games that are underpinned by rich simulations; as such, this class of games represents perhaps the strongest use case for procedural content generation. In this paper, we have outlined ongoing work in which we procedurally generate natural language retellings of playthroughs from play trace data in the simulationist game *Prom Week*. Specifically, we have shown that it is possible to translate game state into a semantic encoding that can be used for NLG.

We identified features of game state we could encode and combine into retellings. This approach proved to be fruitful, subjectively speaking, but given the limited number of social exchanges, the results sometimes seemed repetitive. Alternative encodings for each game might help to alleviate this issue.

In general, our results suggest that we're able to take a record of game state and transform it into a natural language story. While we weren't able to show any statistically significant effects from the augmentation process, there were many options left unexplored and we are hopeful that future work will validate the approach.

Scheherazade's comparatively archaic and limited vocabulary made it difficult to express some of the social interactions common to high-schoolers. While it is possible to modify Scheherazade with custom verbs, we left that for future work and adopted more uncomfortable formulations. For instance, the social exchange BACK ME UP! was encoded as an imploration for support.

One of the reasons we chose the SIG as our target semantic encoding was for its modeling of character belief states and goals. While we did not take advantage of that functionality in this work, the potential is there. However, our initial exploratory efforts to that end brought to light numerous difficulties. With *Prom Week*'s data in particular, we found that although the underlying simulation is indeed quite rich, it doesn't actually encode character goals or motivations. The best we could do is to infer trivial goals such that when Doug performs ASK OUT on Chloe, he has the goal of dating her. It is tempting to cast this as a general goal and to interpret Doug's further actions in terms of it, but this is problematic: the player may perform a series of completely contradictory social exchanges at will. Attempting to unify the caprices of any character's social interactions into an appropriate goal is left as future work. This particular issue points to the more general problem of *story understanding*, and we might glean insight from prior work in that area (e.g., [5]).

Relatedly, our evaluation exposed a problematic treatment of any and all *Prom Week* playthroughs as stand-alone stories, something which speaks to the simulationist challenge of *story recognition* that we introduced above. Again, this is the issue that these games currently have no way of recognizing the very narratives that may emerge from their underlying simulations. Here, we have shown that it is probably not reasonable to assume that intact playthroughs will constitute coherent stories. Indeed, it would appear that stronger narrative retellings of gameplay should recount only carefully selected subsets of in-game events; how these might be selected represents an open challenge.

In the short term, we intend to explore adding the EST to our pipeline, leveraged for its ability to perform narratological transformations. In particular, it would let us interface with PERSONAGE to use the Big Five personality model to change the tone of the output, and use RealPro [12] for surface realization. We also hope to take advantage of its ability to transform narratives into first-person.

Looking to the future, we envision a more complete system applied to more complex games, with mechanisms to select interesting subsets of the fabula to recount. In this work, we have demonstrated the feasibility of a single piece of the machinery needed for games to generate natural language artifacts as or after they are played.

6. REFERENCES

[1] T. Adams and Z. Adams. *Slaves to Armok: God of Blood Chapter II: Dwarf Fortress*. Bay 12 Games, 2006.

[2] N. D. Allen, J. R. Templon, P. S. McNally, L. Birnbaum, and K. J. Hammond. Statsmonkey: A data-driven sports narrative writer. In *AAAI Fall Symposium: Computational Models of Narrative*, 2010.

[3] C. B. Callaway and J. C. Lester. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, 2002.

[4] M. Cavazza and F. Charles. Dialogue generation in character-based interactive storytelling. In *AIIDE*, pages 21–26, 2005.

[5] M. T. Cox. An empirical study of computational introspection: Evaluating introspective multistrategy learning in the meta-aqua system. In *Proceedings of the third international workshop on multistrategy learning*, pages 135–146, 1996.

[6] D. Elson and K. McKeown. A tool for deep semantic encoding of narrative texts. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 9–12, 2009.

[7] D. K. Elson. Detecting story analogies from annotations of time, action and agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative, Istanbul, Turkey*, 2012.

[8] P. Gervás. Stories from games: Content and focalization selection in narrative composition. In *Actas del Primer Simposio Español de Entretenimiento Digital*, page 25, 2013.

[9] M. Helgeson. L.A. Noire: Rockstar resets the bar with its upcoming crime thriller. *Game Informer*, 2010.

[10] M. Hoek, M. Theune, and J. Linssen. Generating Game Narratives with Focalization and Flashbacks. 2014.

[11] I. D. Horswill. Architectural issues for compositional dialog in games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[12] B. Lavoie and O. Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the fifth conference on Applied natural language processing*, pages 265–268. Association for Computational Linguistics, 1997.

[13] W. G. Lehnert. Plot units and narrative summarization. 5(4):293–331, 1981.

[14] S. M. Lukin, J. O. Ryan, and M. A. Walker. Automating direct speech variations in stories and games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[15] F. Mairesse and M. A. Walker. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488, 2011.

[16] J. McCoy, M. Treanor, B. Samuel, M. Mateas, and N. Wardrip-Fruin. Prom week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 319–321. ACM, 2011.

[17] J. McCoy, M. Treanor, B. Samuel, A. Reed, M. Mateas, and N. Wardrip-Fruin. Social story worlds with comme il faut. *IEEE Transactions on Computational Intelligence and AI in Games PP (99)*, pages 1–1, 2014.

[18] J. McCoy, M. Treanor, B. Samuel, A. A. Reed, M. Mateas, and N. Wardrip-Fruin. Prom Week: Designing past the game/story dilemma. In *FDG*, pages 94–101, 2013.

[19] J. McCoy, M. Treanor, B. Samuel, A. A. Reed, M. Mateas, and N. Wardrip-Fruin. Prom week: Designing past the game/story dilemma. In *FDG*, pages 94–101, 2013.

[20] N. Montfort. Natural language generation and narrative variation in interactive fiction. In *Proceedings of the AAAI Workshop on Computational Aesthetics*, 2006.

[21] J. C. Osborn, B. Samuel, J. A. McCoy, and M. Mateas. Evaluating play trace (dis) similarity metrics. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[22] A. A. Reed. A sequence of possibilities. Master's thesis, University of California, Santa Cruz, 2011.

[23] A. A. Reed, B. Samuel, A. Sullivan, R. Grant, A. Grow, J. Lazaro, J. Mahal, S. Kurniawan, M. A. Walker, and N. Wardrip-Fruin. A step towards the future of role-playing games: The spyfeet mobile rpg project. In *AIIDE*, 2011.

[24] E. Rishes, S. M. Lukin, D. K. Elson, and M. A. Walker. Generating different story tellings from semantic representations of narrative. In *Interactive Storytelling*, pages 192–204. Springer, 2013.

[25] J. P. Rowe, E. Y. Ha, and J. C. Lester. Archetype-driven character dialogue generation for interactive narrative. In *Intelligent Virtual Agents*, pages 45–58. Springer, 2008.

[26] J. O. Ryan, C. Barackman, N. Kontje, T. Owen-Milner, M. A. Walker, M. Mateas, and N. Wardrip-Fruin. Combinatorial dialogue authoring. In *Interactive Storytelling*, pages 13–24. Springer, 2014.

[27] B. Samuel, J. McCoy, M. Treanor, A. A. Reed, M. Mateas, and N. Wardrip-Fruin. Introducing story sampling: Preliminary results of a new interactive narrative evaluation technique. In *Ninth International Conference on the Foundations of Digital Games*, 2014.

[28] Team Bondi. *L.A. Noire*. Rockstar Games, 2011.

[29] M. A. Walker, J. Sawyer, C. Jimenez, E. Rishes, G. I. Lin, Z. Hu, J. Pinckard, and N. Wardrip-Fruin. Using expressive language generation to increase authorial leverage. *Intelligent Narrative Technologies*, 6, 2013.

[30] N. Wardrip-Fruin. *Expressive Processing: Digital fictions, computer games, and software studies*. MIT Press, 2009.