Automating Direct Speech Variations in Stories and Games

Stephanie M. Lukin and James O. Ryan and Marilyn A. Walker
Natural Language and Dialogue Systems Lab
University of California, Santa Cruz
1156 High Street, Santa Cruz, CA 95060
{slukin,joryan,mawalker}@ucsc.edu

Abstract
Dialogue authoring in large games requires not only the creation of content, but the subtlety of its delivery, which can vary from character to character. Manually authoring this dialogue can be tedious, time-consuming, or even altogether infeasible. This paper utilizes a rich narrative representation for modeling dialogue and an expressive natural language generation engine for realizing it, and expands upon a translation tool that bridges the two. We add functionality to the translator to allow direct speech to be modeled by the narrative representation, whereas the original translator supports only narratives told by a third person narrator. We show that we can perform character substitution in dialogues. We implement and evaluate a potential application to dialogue implementation: generating dialogue for games with big, dynamic, or procedurally-generated open worlds.

Dialogue authoring in large games requires not only the creation of content, but the subtlety of its delivery which can vary from character to character. Manually authoring this dialogue can be tedious, time-consuming, or even altogether infeasible. The task becomes particularly intractable for games and stories with dynamic open worlds in which character parameters that should produce linguistic variation may change during gameplay or are decided procedurally at runtime. Short of writing all possible variants pertaining to all possible character parameters for all of a game’s dialogue segments, authors working with highly dynamic systems currently have no recourse for producing the extent of content that would be required to account for all linguistically meaningful character states. As such, we find open-world games today filled with stock dialogue segments that are used repetitively by many characters without any linguistic variation, even in game architectures with rich character models that could give an actionable account of how their speech may vary (Klabunde 2013).

Indeed, in general, we are building computational systems that, underlyingly, are far more expressive than can be manifested by current authoring practice. These concerns can also be seen in linear games, in which the number of story paths may be limited to reduce authoring time or which may require a large number of authors to create a variety of story paths. Recent work explores the introduction of automatically authored dialogues using expressive natural language generation (NLG) engines, thus allowing for more content creation and the potential of larger story paths (Monfort, Stayton, and Campana 2014; Lin and Walker 2011; Cavazza and Charles 2005; Rowe, Ha, and Lester 2008).

Figure 1: NLG pipeline method of the ES Translator.

(Walker et al. 2013) explore using a dynamic and customizable NLG engine called PERSONAGE to generate a variety of character styles and realizations, as one way to help authors to reduce the authorial burden of writing dialogue instead of relying on scriptwriters. PERSONAGE is a parameterizable NLG engine grounded in the Big Five personality traits that provides a larger range of pragmatic and stylistic variations of a single utterance than other NLG engines (Mairesse and Walker 2011). In PERSONAGE, narrator’s voice (or style to be conveyed) is controlled by a model that specifies values for different stylistic parameters (such as verbosity, syntactic complexity, and lexical choice). PERSONAGE requires hand crafted text plans, limiting not only the expressibility of the generations, but also the domain.

(Reed et al. 2011) introduce SpyFeet: a mobile game to encourage physical activity which makes use of dynamic storytelling and interaction. A descendant of PERSONAGE, called SpyGen, is its NLG engine. The input to SpyGen is a text plan from Inform7, which acts as the content planner and manager. (Reed et al. 2011) show that this architecture
allows any character personality to be used in any game situation. However their approach was not evaluated and it relied on game specific text plans.

(Rishes et al. 2013) created a translator, called the EST, which bridges a narrative representation produced by the annotation tool Scheherazade, to the representation required by PERSONAGE, thus not requiring the creation of text plans. Fig. 1 provides a high level view of the architecture of EST, described in more detail below. Scheherazade annotation facilitates the creation of a rich symbolic representation for narrative texts, using a schema known as the STORY INTENTION GRAPH or SIG (Elson and McKeown 2010; Elson 2012). A SIG represents the sequence of story events, as well as providing a rich representation of the intentions, beliefs, and motivations of story characters. The EST takes the SIG as input, and then converts the narrative into a format that PERSONAGE can utilize.

However, the approach described in (Rishes et al. 2013) is limited to telling stories from the third person narrative perspective. This paper expands upon the EST to enable annotation of direct speech in Scheherazade, that can then be realized directly as character dialogue. We explore and implement a potential application to producing dialogue in game experiences for large, dynamic, or procedurally-generated open worlds, and present a pilot study on user perceptions of the personalities of story characters who use direct speech. The contributions of this work are: 1) we have can modify a single, underlying representation of narrative to adjust for direct speech and substitute character speaking styles; and 2) that we can perform this modeling on any domain.

**ES Translator**

Aesop’s Fable “The Fox and The Crow” (first column in Fig. 2) is used to illustrate the development and the new dialogue expansion of the EST.

**Annotation Schema**

One of the strengths of Scheherazade is that it allows users to annotate a story along several dimensions, starting with the surface form of the story (first column in Fig. 3) and then proceeding to deeper representations. The first dimension (second column in Fig. 3) is called the “timeline layer”, in which the story facts are encoded as predicate-argument structures (propositions) and temporally ordered on a timeline. The timeline layer consists of a network of proposi-

tional structures, where nodes correspond to lexical items that are linked by thematic relations. Scheherazade adapts information about predicate-argument structures from the VerbNet lexical database (Kipper et al. 2006) and uses WordNet (Fellbaum 1998) as its noun and adjectives taxonomy. The arcs of the story graph are labeled with discourse relations. Fig. 4 shows a GUI screenshot of assigning propositional structure to the sentence The crow was sitting on the branch of a tree. This sentence is encoded as two nested propositions sit(crow) and the prepositional phrase on the branch of the tree. Both actions (sit and on) contain references to the story characters and objects (crow and branch of the tree) that fill in slots corresponding to semantic roles. Only the timeline layer is utilized for this work at this time.

![GUI view of propositional modeling](image)

In the current annotation tool, the phrase The fox ... said “You have a voice, madam...” can be annotated in Scheherazade by selecting say from VerbNet and attaching the proposition the crow has a voice to the verb say(fox, able-to(sing(crow))). However, this is realized as The fox said the crow was able to sing (note: in the single narrator realization, everything is realized in the past tense at this time. When we expand to direct speech in this work, we realize verbs in the future or present tense where appropriate). To generate The fox said “the crow is able to sing”, we append the modifier “directly” to the verb “say” (or any other verb of communication or cognition, e.g. “think”), then handle it appropriately in the EST rules described in Section . Furthermore, to generate The fox said “you are able to sing”, instead of selecting crow, an interlocutor character is created and then annotated.
as say(fox, able-to(sing(interlocutor))). We add new rules to the EST to handle this appropriately.

**Translation Rules**

The process of the EST transformation of the SIG into a format that can be used by PERSONAGE is a multi-stage process shown in Fig. 5 (Rishes et al. 2013). First, a syntactic tree is constructed from the propositional event structure. Element A in Fig. 5 contains a sentence from the original “The Fox and the Grapes” fable. The Scheherazade API is used to process the fable text together with its SIG encoding and extract actions associated with each timespan of the timeline layer. Element B in Fig. 5 shows a schematic representation of the propositional structures. Each action instantiates a separate tree construction procedure. For each action, we create a verb instance (highlighted nodes of element D in Fig. 5). Information about the predicate-argument frame that the action invokes (element C in Fig. 5) is then used to map frame constituents into respective lexico-syntactic classes, for example, characters and objects are mapped into nouns, properties into adjectives and so on. The lexico-syntactic class aggregates all of the information that is necessary for generation of a lexico-syntactic unit in the DSyntS representation used by the REALPRO surface realizer of PERSONAGE (element E in Fig. 5) (Lavoie and Rambow 1997). (Rishes et al. 2013) define 5 classes corresponding to main parts of speech: noun, verb, adverb, adjective, functional word. Each class has a list of properties such as morphology or relation type that are required by the DSyntS notation for a correct rendering of a category. For example, all classes include a method that parses frame type in the SIG to derive the base lexeme. The methods to derive grammatical features are class-specific. Each lexico-syntactic unit refers to the elements that it governs syntactically thus forming a hierarchical structure. A separate method collects the frame adjuncts as they have a different internal representation in the SIG.

At the second stage, the algorithm traverses the syntactic tree in-order and creates an XML node for each lexico-syntactic unit. Class properties are then written to disk, and the resulting file (see element E in Fig. 5) is processed by the surface realizer to generate text.

**Dialogue Realization**

The main advantage of PERSONAGE is its ability to generate a single utterance in many different voices. Models of narrative style are currently based on the Big Five personality traits (Mairesse and Walker 2011), or are learned from film scripts (Walker et al. 2011). Each type of model (personality trait or film) specifies a set of language cues, one of 67 different parameters, whose value varies with the personality or style to be conveyed. In (Reed et al. 2011), the SpyGen engine was not evaluated. However previous work (Mairesse and Walker 2011) has shown that humans perceive the personality stylistic models in the way that PERSONAGE intended, and (Walker et al. 2011) shows that character utterances in a new domain can be recognized by humans as models based on a particular film character.

Here we first show that our new architecture as illustrated by Fig. 1 and Fig. 5 lets us develop SIGs for any content domain. We first illustrate how we can change domains to a potential game dialogue where the player could have a choice of party members, and show that the EST is capable of such substitutions. Table 1 shows different characters saying the same thing in their own style. We use an openness to experience model from the Big 5 (Mairesse and Walker 2011), Marion from Indiana Jones and Vincent from Pulp Fiction from (Lin and Walker 2011), and the Otter character model from (Reed et al. 2011)’s Heart of Shadows.

<table>
<thead>
<tr>
<th>Character Model</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marion (Indiana Jones)</td>
<td>“Because you save me from the dungeon pal, I will fight with you!”</td>
</tr>
<tr>
<td>Vincent (Pulp Fiction)</td>
<td>“Oh God I will fight with you!”</td>
</tr>
<tr>
<td>Otter (Heart of Shadows)</td>
<td>“Oh gosh I will fight with you because you save me from the dungeon mate!”</td>
</tr>
</tbody>
</table>

With the EST, an author could use Scheherazade to encode stock utterances that any character may say, and then have PERSONAGE automatically generate stylistic variants of that utterance pertaining to all possible character personalities.
This technique would be particularly ripe for games in which character personality is unknown at the time of authoring. In games like this, which include *The Sims 3* (Electronic-Arts 2009) and *Dwarf Fortress* (Adams and Adams 2006), personality may be dynamic or procedurally decided at runtime, in which case a character could be assigned any personality from the space of all possible personalities that the game can model. Short of writing variants of each dialogue segment for each of these possible personalities, authors for games like these simply have no recourse for producing enough dialogic content sufficient to cover all linguistically meaningful character states. For this reason, in these games a character’s personality does not affect her dialogue. Indeed, *The Sims 3* avoids natural-language dialogue altogether, and *Dwarf Fortress*, with likely the richest personality modeling ever seen in a game, features stock dialogue segments that are used across all characters, regardless of speaker personality.

When producing the EST (Rishes et al. 2013) focused on a tool that could generate variations of Aesop’s Fables such as *The Fox and the Crow* from Drama Bank (Elson and McKeown 2010), (Rishes et al. 2013) evaluated the EST with a primary focus on whether the EST produces correct retellings of the fable. They measure the generation produced by the EST in terms of the string similarity metrics BLEU score and Levenshtein Distance to show that the new realizations are comparable to the original fable.

After we add new rules to the EST for handling direct speech and interlocutors, we modified the original SIG representation of *The Fox and the Crow* to contain more dialogue in order to evaluate a broader range of character styles, along with the use of direct speech (second column of Fig. 2). This version is annotated using the new direct speech rules, then run through the EST and PERSONAGE. Table 2 shows a subset of parameters, which were used in the three personality models we tested here: the laid-back model for the fox’s direct speech, the shy model for the crow’s direct speech, and the neutral model for the narrator voice. The laid-back model uses emphasizers, hedges, exclamations, and expletives, whereas the shy model uses softener hedges, stuttering, and filled pauses. The neutral model is the simplest model that does not utilize any of the extremes of the PERSONAGE parameters.
User Perceptions

Here we present a pilot study aimed at illustrating how the flexibility of the EST when producing dialogic variations allows us to manipulate the perception of the story characters. We collect user perceptions of the generated dialogues via an experiment on Mechanical Turk in which the personality models used to generate the dialogic version of “The Fox and The Crow” shown in Fig. 6 are modified, so that the fox uses the shy model and the crow uses the laid-back model. We have three conditions; participants are presented with the dialogic story told 1) only with the neutral model; 2) with the crow with shy and the fox with laid-back; and 3) with the crow with laid-back and the fox with shy.

After reading one of these tellings, we ask participants to provide adjectives in free-text describing the characters in the story. Figs. 7 and 8 show word clouds for the adjectives for the crow and the fox respectively. The shy fox was not seen as very “clever” or “sneaky” whereas the laid-back and neutral fox were. However, the shy fox was described as “wise” and the laid-back and neutral were not. There are also more positive words, although of low frequency, describing the shy fox. We observe that the laid-back and neutral crow are perceived more as “naïve” than “gullible”, whereas shy crow was seen more as “gullible” than “naïve”. Neutral crow was seen more as “stupid” and “foolish” than the other two models.

Table 3 shows the percentage of positive and negative descriptive words defined by the LIWC (Pennebaker, Francis, and Booth 2001). We observe a difference between the use of positive words for shy crow and laid-back or neutral, with the shy crow being described with more positive words. We hypothesize that the stuttering and hesitations make the character seem more meek, helpless, and tricked rather than the laid-back model which is more boisterous and vain. However, there seems to be less variation between the fox polarity. Both the stuttering shy fox and the boisterous laid-back fox were seen equally as “cunning” and “smart”.

This preliminary evaluation shows that there is a perceived difference in character voices. Furthermore, it is easy to change the character models for the EST to portray different characters.

Table 3: Polarity of Adjectives describing the Crow and Fox (% of total words)

<table>
<thead>
<tr>
<th></th>
<th>Crow</th>
<th>Pos</th>
<th>Neg</th>
<th>Fox</th>
<th>Pos</th>
<th>Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral</td>
<td>13</td>
<td>29</td>
<td></td>
<td>38</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Shy</td>
<td>28</td>
<td>24</td>
<td></td>
<td>39</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Laid-back</td>
<td>10</td>
<td>22</td>
<td></td>
<td>34</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Conclusion

In this paper, we build on our previous work on the EST (Rishes et al. 2013), and explain how it can be used to allow linguistically naïve authors to automatically generate dialogue variants of stock utterances. We describe our extensions to the EST to handle direct speech and interlocutors in dialogue. We experiment with how these dialogue variants can be realized utilizing parameters for characters in dynamic open worlds. (Walker et al. 2015) generate utterances using PERSONAGE and require authors to select and edit automatically generated utterances for some scenes. A similar revision method could be applied to the output of the EST.

As a potential future direction, we aim to explore the potential of applying this approach to games with expansive open worlds with non-player characters (NPCs) who come from different parts of the world and have varied backgrounds, but currently all speak the same dialogue in the same way. While above we discuss how our method could be used to generate dialogue that varies according to character personality, the EST could also be used to produce dialogue variants corresponding to in-game regional dialects. PERSONAGE models are not restricted to the Big Five personality traits, but rather comprise values for 67 parameters, from which models for unique regional dialects could easily be sculpted. Toward this, (Walker et al. 2013) created a story world called Heart of Shadows and populated it with characters with unique character models. They began to create their own dialect for the realm with custom hedges, but to date the full flexibility of PERSONAGE and its 67 parameters has not been fully exploited. Other
recent work has made great strides toward richer modeling of social-group membership for virtual characters (Harrell et al. 2014). Our approach to automatically producing linguistic variation according to such models would greatly enhance the impact of this type of modeling.

Acknowledgments This research was supported by NSF Creative IT program grant #IIS-1002921, and a grant from the Nuance Foundation.

References
Lin, G. I., and Walker, M. A. 2011. All the world’s a stage: Learning character models from film. In AIIDE.
Monfort, N.; Stayton, E.; and Campana, A. 2014. Expressing the narrator’s expectations.