ELSEVIER

Artificial Intelligence 139 (2002) 213–252

Artificial Intelligence

www.elsevier.com/locate/artint

Narrative prose generation ☆

Charles B. Callaway [a,*], James C. Lester [b]

[a] ITC-irst, Centro per la Ricerca Scientifica e Tecnologica, Via Sommarive, 18, 38050 Povo, Trento, Italy
[b] Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA

Received 10 January 2002

Abstract

Narrative generation has historically suffered from poor writing quality, stemming from a narrow focus on story grammars and plot design. Moreover, to-date natural language generation systems have not been capable of faithfully reproducing either the variety or complexity of naturally occurring narratives. In this article we first propose a model of narrative derived from work in narratology and grounded in observed linguistic phenomena. Next we describe the AUTHOR architecture for narrative generation and an end-to-end implementation of the AUTHOR model in the STORYBOOK narrative prose generation system. Finally, we present a formal evaluation of the narratives that STORYBOOK produces. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Narrative Generation; Story generation; Natural Language Generation; Revision; Pronominalization; Discourse history; Narrative models; Character dialogue; Discourse markers

1. Introduction

Narrative (story) generators [36,37,44,63] typically address the macro-scale development of characters and plot, slowly refining from high-level narrative goals down to individual descriptions and character actions. This is accomplished with a monolithic planning system that creates text by associating text strings with planning operators (Fig. 1). Meanwhile, work in natural language generation (NLG) focuses on linguistic phenomena at the individual sentence level, and only recently have NLG systems achieved the ability to

☆ This is an extended version of the paper presented at IJCAI-2001, Seattle, WA, USA.

* Corresponding author.
E-mail addresses: callaway@itc.it (C.B. Callaway), lester@adm.csc.ncsu.edu (J.C. Lester).
URLs: http://ecate.itc.it:1024/callaway/ (C.B. Callaway), http://www.multimedia.ncsu.edu/imedia/ (J.C. Lester).

PII: S0004-3702(02)00230-8

```
      ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND.
THERE WAS A NEST IN AN ASH TREE.  WILMA BIRD LIVED IN THE NEST.
THERE WAS SOME WATER IN A RIVER.  WILMA KNEW THAT THE WATER WAS
IN THE RIVER.  GEORGE KNEW THAT THE WATER WAS IN THE RIVER.
ONE DAY WILMA WAS VERY THIRSTY.  WILMA WANTED TO GET NEAR SOME
WATER.  WILMA FLEW FROM HER NEST ACROSS A MEADOW THROUGH A
VALLEY TO THE RIVER.  WILMA DRANK THE WATER.  WILMA WASN'T VERY
THIRSTY ANY MORE.
```

Fig. 1. Prose generated by TALE-SPIN, 1977.

```
      Once upon a time a woodman and his wife lived in a
pretty cottage on the borders of a great forest. They had
one little daughter, a sweet child, who was a favorite with
every one.  She was the joy of her mother's heart.  To
please her, the good woman made her a little scarlet cloak
and hood.  She looked so pretty in it that everybody
called her Little Red Riding Hood.
```

Fig. 2. Prose generated by AUTHOR, 2001.

```
      Once upon a time there lived in a pretty cottage, on
the borders of a great forest, a woodman and his wife who
had one little daughter, a sweet child, and a favorite with
every one.  She was the joy of her mother's heart, and to
please her, the good woman made her a little scarlet cloak
and hood, in which she looked so pretty, that everybody
called her Little Red Riding-Hood.
```

Fig. 3. Prose from *Little Red Riding Hood* [61].

produce multi-paragraph text. What remains is a substantial gap between the story plans produced by narrative generators and the detailed syntactic requirements of current NLG systems.

To bridge this gap between narrative generators and NLG systems, we have developed the AUTHOR narrative prose generation architecture [5,10] to create high-quality narrative prose (Fig. 2) comparable to that routinely produced by human authors (Fig. 3). This architecture has been implemented in STORYBOOK, an end-to-end narrative prose generation system that utilizes a primitive narrative planner along with a complete sentence planner, discourse history, lexical choice module, revision module, and the FUF/SURGE [19] surface realizer to produce multi-page stories in the Little Red Riding Hood fairy tale domain.

Upon receiving a high-level story specification from a narrative planner, STORYBOOK (1) structures it into paragraph and sentence-sized chunks, (2) conducts a discourse history analysis to determine indefinite references and pronominalizations, (3) performs a lexical choice analysis to increase variety among concepts and event relations, (4) maps actors, props and events to semantic/syntactic roles in full linguistic deep structures, (5) revises paragraph-sized groups of deep structures via aggregation, discourse marker insertion, or reordering to eliminate the short, choppy sentences characteristic of text produced by discourse planning systems, and (6) performs surface realization with integrated formatting to produce narrative prose similar to that found in stories written by human authors.

To evaluate the quality of narratives that STORYBOOK produces, we first created a simplified narrative planner capable of generating two different Little Red Riding Hood stories. We then created five distinct versions of STORYBOOK by ablating the discourse history, lexical choice, and revision components to produce a total of 10 story versions which were then formally evaluated by human judges [9]. An ANOVA analysis over a wide range of evaluation criteria showed significant differences between the inclusion or ablation of the individual architectural modules.

2. A model of narrative

Literary theorists have discussed many and varied aspects of the novel and narrative, ranging in levels from the highly abstract [59] to the descriptive [39]. However, current narrative theories are incapable of serving as the foundation for a comprehensive computational model that informs a decision algorithm for narrative generation, and all such theories ultimately resort to non-computational definitions.

However, literary theorists have been very helpful in identifying a substantial number of other concepts useful for analyzing story and narrative. Two of the most broadly used terms in narrative criticism are (from the Russian formalist school) *fabula* and *suzjet* [2,51,56]. Although many flavors of these notions exist, the fundamental meaning of the former is the *deep semantic structure* of the narrative text while the latter refers to the *linear presented order* of events by the author.

In every narrative, there are a number of underlying "facts" that constitute the demonstrable states and activities that occur in the story and the objects and properties involved in those states and activities. Additionally, there are a number of decisions made by the author as to which facts should be mentioned first (or even whether they should be mentioned at all, as in a mystery) and what style considerations should be involved in realizing the actual text. Consider the following Little Red Riding Hood paragraph [61]:

> One day her mother told her she meant to send her to her grandmother—a very old lady who lived in the heart of a neighboring wood—to take her some fresh butter and new-laid eggs and a nice cake. Little Red Riding-Hood was delighted at being sent on this errand, for she liked to do kind things, and it was such a very long time since she had seen her grandmother, that she had almost forgotten what the old lady looked like.

In order to describe the events and states found in this paragraph, we must first enumerate the objects and properties which comprise them [2]. Objects are comprised of three categories, which we call *actors*, *locations* and *props* after the terms from drama. It is relatively easy here to enumerate most objects: a mother, a girl named Little Red Riding Hood, a grandmother, some butter, etc. The events in this passage could also potentially be enumerated as follows: "the mother told the daughter something", "a daughter is going to take something to her grandmother", etc. along with various states: "Little Red Riding

Hood was delighted", "it was a very long time". The *fabula* is the sum total of this factual content that constitutes the story.

In every narrative an author also makes choices about which of the above elements should be mentioned first, which second, and so on. What is meant here is not that one event occurred temporally before or after another event (logical ordering), or that the story occurred in the past relative to the narrator (chronological ordering), but rather the *presentational ordering*: that a specific event or state was mentioned by the author before or after the author decided to mention another particular state or event. We term this strict linear sequence the *narrative stream*, which represents the presentational ordering of the facts from the fabula as found in the text.

The fabula and narrative stream are *separable* in the sense that if the fabula is held constant, a large number of stories can be produced by varying the presentational ordering, and similarly variations in the fabula can also produce different stories. This separation is not unmotivated. A number of aspects of stories lead to their decomposition into fabula and narrative stream. These factors come from a variety of sources and include:

- *Implicit vs. Explicit Information*: Many more events occur in narratives than are actually included in the text. We can imagine that Little Red Riding Hood used her hand to open the door on her way out of her house and into the forest even though that fact is never mentioned. The presentational ordering reflects explicit narrative information, while the fabula is the embodiment of the implicit information that underlies the narrative.
- *Narratorial Distinctions*: Whether the story is told from a particular point of view or not is immaterial as far as the fabula is concerned because it merely records all factual occurrences. However the presentational order must take into account the number of narrators, the choice of first or third person narrator, and whether or not they are diegetic [56].
- *Dialogue vs. Exposition*: Although knowledge between characters in a narrative is communicated one way or another, whether an author renders this communication as quoted dialogue or unquoted exposition is a feature of the presentation, because for the fabula it is only important that that communication occurred and that informational content was communicated.
- *Linearity of Text*: Because text is a linear medium, it forces a sequentialization of information which is reflected in the presentational order. By contrast, the fabula is an essentially parallel knowledge structure because it contains information that can be mentally accessed in any order.

In addition to the fabula and order of presentation, there are a number of stylistic factors which authors use to distinguish their prose, although unfortunately very little work has been done on the practical side of the relationship between style and narrative:

- *Narrator Mode*: An author can choose first, second, or third person narrator(s), and indicate their omniscience, reliability, embodiment, etc.
- *Sentence Complexity*: An author can choose to exclude complex grammatical constructions to yield fairy tales such as *Little Red Riding Hood*, or include them and produce

extremely complex and sophisticated text such as that from James Joyce or Umberto Eco (see [38]).
- *Amount and Timing of Dialogue*: Over the course of a narrative, an author can select from among the various dialogue modes presented in Section 2.3.1.
- *Diction*: An author might wish to use words with particular connotative values, technical terminology, or age appropriateness.

Finally, we must consider the *mechanical realization* of textual prose that convert the fabula, narrative stream, and stylistic directives into the linguistic phenomena found in text. These requirements are different for prose than for other potential media, such as film, which makes use of images and sound.

2.1. Fabula, story ontology, and knowledge representation

When an author composes a narrative, he or she brings to bear a large amount of background cultural knowledge, including experience with previous stories told by others, ideas about how characters should act or talk, and fundamental knowledge of the world and how objects within it interact. This repository of background knowledge from which a narrative planner can draw on for the context of the story is termed the *story ontology*.

The fabula itself consists of highly detailed instances of characters, events and objects that are closely related to those generic elements of the story ontology. In some sense, the plot itself is a subset of the fabula, namely, a particular chronologically ordered set of events which can be arranged in any sensible presentational order. The story ontology contains facts that are true across all stories, while the fabula contains facts that are specific to a particular story.

The fabula and story ontology are the embodiment of the enumerable concepts, concept instances, and concept relations in a narrative. Thus both the fabula and story ontology are organized as a knowledge base. In addition to taxonomic links between those concepts and the ontological hierarchy, the knowledge base encodes relations between concepts in standard semantic network form [3,16,49].

This provides a foundational structure for intensively studying an entire narrative at very low levels of detail. Linguistic structures alone (e.g., story grammars) have not been scalable in terms of the amount of prose needed for generating fairy tales or short stories

```
(NewNarrative MyLRRH-Narrative001 Narrator001)
(AddActor Woodman001 Woodman Person Male)
(AddActor Wife001 Wife Person Female)
(AddLocation Cottage001 Cottage)
(AddLocation Forest001 Forest)
(AddActor Little-Red-Riding-Hood001 Little-Red-Riding-Hood
          Person Female "Little Red Riding Hood")
(AddAlias Daughter001 Daughter Little-Red-Riding-Hood001)
(AddProp Cloak001 Cloak)
(AddProp Hood001 Hood)
```

Fig. 4. Sample Fabula Operators, corresponding to Fig. 2.

Table 1
Fabula operators

Operator	Action
AddScene	Creates the hierarchical scene structure and outlines which characters and props are present during a particular story segment. Such hierarchical structures are typical of published works on story and plot such as [43, Chapter 2].
AddActor	Creates a new named actor character who can appear in scenes and participate in dialogue.
AddProp	Creates a new concept instance that represents a physical or mental object in the narrative world which can be manipulated by characters, pointed to, or referenced.
AddLocation	Declares a concept instance as a type of location. Some concept instances that function as both objects and locations, like *house*, are registered only once using this fabula operator.
AddAlias	Ties two concept instances to each other, allowing multiple names for a single entity while keeping proper track for linguistic purposes. For example, if Little Red Riding Hood is mentioned using her proper name in one sentence and the word “girl” in the next sentence, she should be referenced as *the girl* and not *a girl*.

[22]. While the cost of organizing this much knowledge rigorously is high, it is offset by the flexibility it provides for both the narrative prose generation and narrative planning processes.

Finally, the fabula allows for multiple instances of similar concepts. Taking advantage of inheritance, a system could write a new Little Red Riding Hood story where she and her grandmother were saved by two woodcutters (say, `WOODMAN001` and `WOODMAN002`), or perhaps, a story where Little Red Riding Hood accidentally journeyed to the wrong forest (`FOREST002` vs. `FOREST003`).

To create the fabula from the story ontology, we assume a narrative planner is capable of employing a small set of *fabula operators* (Fig. 4). Before story realization occurs, a narrative planner constructs each concept instance in the fabula by using an appropriate operator (depending on both the ontological type and purpose in the narrative). The fabula operators are briefly described in Table 1 (additional details can be found in [5]).

2.2. Narrative stream and presentational order

While an author is constructing a narrative, he or she takes into consideration the chronological ordering of events and decides whether to present those events in the original chronological order or to create a possibly more “dramatic” sequence. However, an author has more choices than simply deciding on an ordering for those events. Key events may be intentionally omitted, as in mystery novels, and uninteresting events may be completely gapped out for purposes of saving time or for preserving the reader’s focus, such as omitting large parts of the life story of a film character.

In fact, film directing and editing presents an apt analogy for the difference between the fabula and narrative stream. If one imagines Quentin Tarantino, the director of the asynchronous movie *Pulp Fiction*, standing in the film cutting room with a basketful of cut negatives, then those negatives represent the fabula: portions of filmed narrative in many small pieces that have yet to be arranged in any particular order. The actual filming of the movie (the first half of a director's work) corresponds to the creation of the fabula, while the selecting and splicing of the negatives (the second half) corresponds to the imposition of a singular presentational ordering.

In our narrative model, this presentational ordering is imposed by the *narrative stream*, which consists of a sequence of *narrative stream primitives*. These primitives are adapted from work in Speech Act Theory [1,55], which assumes that dialogue can be decomposed into a series of speech acts that describe the utterer's intentions toward affecting the hearer. A typical speech act consists of the *purpose* (or *intention*) the utterer had when speaking as well as the *propositional content* of the utterance.

However, narratives contain a number of features that are not found in the conversations that Austin originally analyzed nor in other later applications of speech act theory, including revisability (rehearsal), dramatic factors (plot structure) and exposition. Thus while speech act theory is helpful in understanding and aiding in the dialogues that all narratives contain, it is less helpful with the expository text that comprises a larger portion of narratives.

In our model, narrative primitives (Fig. 5) are generated in conjunction with the fabula operators by the narrative planner. The primitives are processed in sequence by the *narrative organizer* (Section 3.2) along with the fabula operators and the various stylistic factors, producing a sequence of sentential structures that are eventually converted directly to text. As such, narrative primitives and the fabula can be considered as the interface level for more sophisticated versions of narrative planners than the simple version we employ.

Narrative stream primitives fall into three basic categories:

- *Delimiting Primitives*: Delimiting primitives create the narrative context. They establish scenes, introduce characters and narrators, and describe the author's overall intent with respect to the audience.
- *Base Primitives*: Base primitives provide most of the raw structural content used for creating sentences in exposition and dialogue.
- *Modifying Primitives*: Modifying primitives present information that modifies the content of or adds detail to a base primitive.

Delimiting primitives are similar to a playwright's comments to the director in a dramatic script. Additionally, base primitives are distinguishable from the other primitives because they contain speech acts. Modifying primitives contain the expressive content that differentiates narrative prose from scientific explanation and the output of previous narrative generators (Fig. 1).

2.2.1. Delimiting primitives

As the "directorial" factor in the narrative stream, delimiting primitives serve as the basis for organizing the narrative around *scenes*. A scene is a segment of the narrative

```
((narration-mode historical mixed complex ascii narrated english)
 (narrator-mode narrator001 third-person disembodied)
 (new-scene scene001)
 (new-actor woodman001)
 (new-actor mother001)
 (new-actor little-red-riding-hood001)
 ;;; "Once upon a time there was a woodman and his wife."
 (actor-property exist-being woodman001)
 (refinement and-along-with woodman001 wife001)
 (refinement belonging-to wife001 woodman001)
 (specification exist-being process-step-type once-upon-a-time)
 ;;; "The woodman and his wife lived in a pretty cottage."
 (prop-relationship living-in woodman001 cottage001)
 (refinement and-along-with woodman001 wife001)
 (refinement belonging-to wife001 woodman001)
 (detail cottage001 pretty-appearance)
 ;;; "The cottage was on the borders of a great forest."
 (prop-relationship location-on cottage001 border)
 (refinement container-of border forest001)
 (refinement discourse-reference border multiple-quantity-reference)
 (refinement discourse-reference border principle-subregion-reference)
 (detail forest001 great-sized)
 ;;; "The woodman and his wife had one little daughter."
 (actor-relationship having woodman001 daughter001)
 (refinement and-along-with woodman001 wife001)
 (refinement belonging-to wife001 woodman001)
 (refinement quantifier-value daughter001 unique-one)
 (detail daughter001 little-sized)
 ;;; "The girl was a sweet child."
 (actor-relationship identity girl001 child001)
 (detail child001 sweet-natured)  ...
```

Fig. 5. A sequential narrative stream fragment, corresponding to Fig. 2.

that is contiguous in time, location, and characters. Often this is important in children's narratives: the wolf is a speaking character in Little Red Riding Hood, but not in Jack London's *Call of the Wild*. Without knowing important information about the narrative, it would be possible to mistakenly have Jack London's wolves utter dialogue. In this manner, the equivalent of *type checking* can be applied to the output of a narrative planner.

Delimiting primitives are also responsible for setting global narrative *policies*, such as story setting, narratorial modes, genre, and prose quality. Often these span the entire narrative, although in certain instances, they may change several times (e.g., experimental narratives with multiple types of narrators). Finally, delimiting primitives may be used to indicate local details, such as shifts in topic or time (enabling the realization mechanism to decide when to insert paragraph boundaries), or indications of dialogue direction and content:

- *Narration Mode*: This mode specifies the genre, time period, and other factors that globally affect a narrative as described in the narrative model.

- *Narrator Mode*: Specifies the exact characteristics (person, omniscience, etc.) of any narrators present in subsequent narrative passages.
- *Scene Change*: Scene changes inform the narrative prose generator that the current set of characters may have changed, that a different narration style may be in effect, or that new formatting commands may need to be issued.
- *Scene Actor*: Characters may enter the dramatic stage in the middle of a scene; the discourse history records this so that it can appropriately mark dialogue as having been heard by some characters present but not by others.
- *Define Complex Event*: Although objects and locations are relatively easy to reference, events and actions have a more complex linguistic structure that makes it difficult to refer to them in the text. Event definitions allow a shorthand method to reify events and allow them to be related to each other temporally, causally, etc.
- *Define Group*: Groups of actors and props can be modified either singly or in total. Thus "some butter and cookies" is equivalent to "some butter and some cookies", but "the big dog and woodman" is most likely not equivalent to "the big dog and the big woodman".
- *Dialogue Actors*: Dialogue consists of a sequence of *turns*, where in each turn one participant in the dialogue is the *speaker* and the remaining participants are *hearers*. The dialogue actors primitive marks this distinction.
- *Dialogue Type*: When an utterance occurs, whether direct or indirect, there are many possible realizations (discussed in Section 2.3.1): communicative act ordering, utterer ordering, utterer manner, and communicative intent.
- *Topic Shift*: Occasionally a narrative planner will decide to emphasize some new aspect of a situation during the execution of a scene. Because this is a pragmatic decision, the narrative prose generator cannot be expected to determine on its own whether or not to begin a new paragraph.
- *Format*: The narrative planner may want to indicate a particular font or type style, such as emphasis in "Grandmother, what a **big** nose you have!" or to insert an image in HTML output. This applies to typographical formatting, rather than for specifying sentence length or for linguistic issues.

2.2.2. Base primitives

The primary content in a narrative passage is produced via base primitives, including that found in both exposition and dialogue. Base primitives are similar to the *illocutionary* acts from speech act theory, in that they carry the content component of a higher-level speech act. In our case, these higher-level acts are motivated by narrative concerns rather than dialogic ones.

Although surely incomplete, these primitives are sufficient to produce narratives comparable in size, structure, and content to many existing versions of Little Red Riding Hood. We do not assert that as presented these narrative primitives are sufficient to generate much larger narratives (such as novels) or other genres (such as mysteries or newspaper reporting). However, in principle the narrative stream can be extended with new narrative primitives that reflect the discoveries found in detailed narrative analyses of those genres.

Base primitives consist of a *narrative purpose directive* followed by an event or property relation along with that relation's arguments in propositional form. The directive itself is the narrative primitive that states the *raison d'etre* for the clause and comes from a very small set. Its arguments however, can consist of any of the many thousands of relations and objects contained in the fabula.

- *Actor Relationship*: When an action, event, or description holds between two characters, the `actor-relationship` base primitive is used to explicitly indicate that relationship in the narrative.
- *Actor Action*: This narrative primitive describes an action performed by a character in the current scene, along with the objects that fill the action relation's theta frame (list of thematic arguments).
- *Actor Command*: This primitive is used when one character orders another to perform an action, including whether the person ordered is explicitly mentioned or not, and if the command is immediate or reported by another.
- *Actor Request*: Similarly, characters can request information or props.
- *Actor Property*: The actor-property primitive is used to give descriptions (e.g., appearance, relative height) and properties (e.g., existence, health, absolute height) of characters.
- *Actor Emotion*: This primitive expresses both internal emotional states as well as outward gestures and facial expressions of emotion.
- *Actor Intent*: Similarly, the actor-intent primitive provides for the expression of mental states in which a character is attempting to accomplish something or getting others to assist in accomplishing something.
- *Directed Actor Emotion*: Another primitive that is similar to actor-emotion, the directed-actor-emotion expresses emotion that is explicitly projected at another character or object.
- *Prop Relationship*: Like actors, props have relationships (e.g., spatial distance, ownership, usefulness) to other props and to characters.
- *Prop Property*: Finally, just like characters, props can have inherent and absolute properties and descriptions.

2.2.3. Modifying primitives

Like base primitives, modifying primitives represent clauses or phrases with semantic content. Unlike base primitives, they do not represent "stand-alone" clauses. Instead, they indicate that the main clause generated by a base primitive is related in some specific way to another *attributive*, *subordinating*, or *circumstantial* clause [52, Chapter 8] or phrase. This relationship is indicated with the name of the relation itself, by repeating a concept or concept instance from a previous base primitive, reified event or modifying primitive, and by supplying the new information as additional arguments.

- *Actor-Purpose*: This primitive modifies the main content to express what intent or desired goal a character had when performing some action.
- *Actor-Reason*: This primitive modifies the main clause to express the reason behind the action or property involved in it.

- *Beneficiary*: This primitive specifies the recipient of an object or action.
- *Comparison*: This primitive allows for comparison between two entities (which do not have to be present in the scene).
- *Condition*: This primitive specifies a condition that while pragmatically dependent on the main clause is not syntactically dependent on it.
- *Detail*: This primitive specifies an attributive relationship between an entity and a quality.
- *Discourse Marker*: Specifies that the preceding base primitive is related to the following base primitive in a particular manner.
- *Event Time*: This specifies the relative temporal occurrence of two events.
- *Focus*: When a particular piece of information needs to be stressed by moving it to some position in a sentence, the *focus* primitive handles this in a language-neutral fashion.
- *Manner*: This primitive helps answer the questions *how* or *in what way*.
- *Prop-Purpose*: This primitive is different from the *actor-purpose* primitive in that the intention is directed toward a goal rather than the event of intending itself.
- *Reason*: This primitive answers the questions *why* and *what for*. It also differs from the purpose primitive in that it describes an intellectual exercise rather than an emotional one.
- *Refinement*: Like the *detail* primitive, this primitive allows for direct modification. However, refinement helps to disambiguate by providing extra information instead of providing detail for "artistic" sake.
- *Result*: This primitive describes the outcome of an event.
- *Specification*: This primitive is also similar to the *detail* and *refinement* primitives. It helps answer the questions *when*, *how*, and *to what degree*, and modifies the sentence as a whole rather than a particular element of it.

2.3. Mechanical realization of narrative

Because most NLG systems have concentrated on generating *explanations* rather than *narratives*, the text they generate is by its nature closer to that of TALE-SPIN rather than the prose found in novels. This difference can be explained by noting the most important difference between narrative and explanation: characters and the consequences that their interactions bring.

As explanations do not involve characters, there has been no attempt to reproduce the dialogue that characters engage in, pronouns in non-neuter genders, and a myriad of detailed syntactic constructions associated with the volitions, desires, thoughts, needs and social relationships of characters, such as raising verbs, clefting, complex tenses, and subjunctives. And while systems like TALE-SPIN do have characters, their generation systems are not capable of handling the difficult syntactic requirements that characters bring to a text. The following sections discuss some of these problems.

2.3.1. Character-to-character dialogue

Written character-to-character dialogue [6] takes the form of turn-taking conversational interactions between one or more agents. The written form lacks prosodic and accentual features found in spoken dialogue; instead, the substitution of orthographic punctuation marks serves as a cue allowing the reader to mentally reconstruct the spoken conversation. Additionally, there are modifications to the basic sentential structure that authors may employ in narrative (as opposed to explanatory) prose:

- *Interjections*: To express shock, surprise, and pain among a number of other feelings, characters will frequently utter interjections such as "Oh!" or "Aha!" Interjections are frequently fronted inside a complete sentence:
  "Ouch, you must have just come from the dentist!"
- *Addressee*: In face to face dialogue, characters will frequently address each other by name, nickname, or some other appellation:
  "Carol, don't forget to pick up some milk."
  "All the better to see you with, my dear!"
- *Written stylistic effects*: In order to show emotions and have characters express themselves in nonstandard ways, authors write dialogue that includes onomatopoeiatic effects, simulated yelling, regional dialects, ellipsis, vocal handicaps, or vowel lengthening. Current surface realizers are ill-equipped to deal with these types of textual effects.
  "Ewwww, I can't believe you just did that!"
  "Mom, you CAN'T do that!"
  "B-b-but, it's so s-s-scary!"
- *Combinations*: Furthermore, these modifications to traditional sentential structures can be used in combination:
  "Wow, John, you REALLY hit that ball!"

Although direct quotes can stand alone, more frequently they are accompanied by another unquoted clause which specifies the speaker and the manner in which the speaker conveyed the utterance. There are six major features that influence these possibilities (shown graphically in Fig. 6):

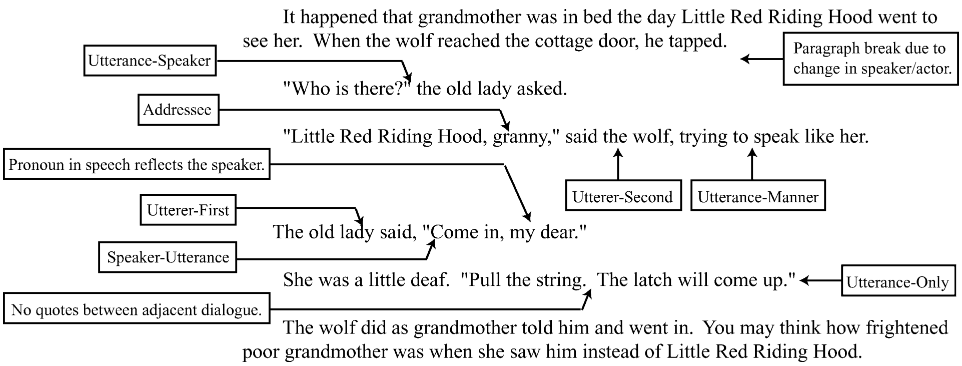

Fig. 6. Some elements of written character-to-character dialogue.

- *Communicative Act Ordering*: There are four positions (in English) that the utterer and the communicative relation can appear in comparison to the quoted dialogue: `utterance-only` as in "⟨sentence⟩", `preposed` as in *John said, "⟨sentence⟩"*, `postposed` as in *"⟨sentence⟩," John said*. and `interposed` as in *"⟨sentence1⟩," John said, "⟨sentence2⟩."*
- *Utterer Ordering*: There are two positions that the utterer and the communicative relation can appear in comparison to each other: `utterer-first` as in *"⟨sentence⟩," John said*. and `utterer-second` as in *"⟨sentence⟩," said John*. Also see [52, Section 14.29].
- *Utterance Manner*: The matrix relation in dialogue is often modified to indicate accompanying action or style in which the communicative act was delivered, such as `adverbial` in *"⟨sentence⟩," John said hurriedly,* with a `prepositional phrase`, a `gerundive clause` or a `co-event clause`.
- *Utterance relation semantics*: The utterance relation is not restricted to the traditional notion of purely communicative acts. For example, emotive verbs are frequently used as in, *"That's my favorite," John smiled*.
- *Use of Pronouns*: English requires an explicit speaker in direct speech with a communicative act, although other languages with morphologically implicit subjects do not, e.g., * *"I love dogs," said*. Also, postposed pronouns are considered archaic: * *"I often go to the mountains on weekends," said he*.
- *Segmentation*: Most notably in replies to questions [41], quoted dialogue does not need to be a complete sentence. Many utterances are fragmentary phrases due to the informal nature of dialogue.

2.3.2. Pronominalization

Pronominalization is the appropriate determination, marking and grammatical agreement of pronouns (*he, she, their, herself, it, those, each other, one, etc*.) as a short-hand reference to a previously mentioned object or event. As opposed to anaphora resolution [23,58,64,65] in understanding, the task in generation is to appropriately insert pronominal anaphora in texts [26,33,42,47,67].

This problem is exacerbated because a pronominalization decision cannot necessarily be made in isolation. E.g., if two grammatically masculine entities occur in a single sentence, converting every reference of the two into pronouns will in many cases make it impossible to resolve references unambiguously. In addition, there are several other problems posed by pronouns in narrative beyond the typical anaphora found in most types of non-narrative texts:

- *Pronouns across scenes*: After a signficant scene boundary (change in time, place, or both; see [42]) in a narrative, authors will typically "reset" pronominalization, at the start of a chapter for example.
- *Restrictions on pronominalization*: Because pronouns cannot have modifiers like nouns, adding an adjective, relative clause, or some other modifier prevents a noun from being replaced by a pronoun. For instance:

  The woodsman had seen the *wicked wolf* earlier that day.

  * The woodsman had seen the *wicked him* earlier that day.

- *Focused nouns*: Especially after a vocally stressed discourse marker or some other marked shift in topic, a word that normally would be pronominalized is often not, as in the second sentence of this example:
  ... A neighbor's child came once or twice a day to tidy the house and bring her [i.e., grandmother] food.
  Now, *grandmother* was very feeble and was often obliged to keep her bed.
- *Pronouns in Floating Circumstantials*: Circumstantial clauses can come either before or after the matrix clause [21] and can contain referents that are also included in the matrix clause. However, a pronominalization decision cannot be made immediately because both aggregation and realization may affect its position with respect to the matrix clause:
  When the wolf reached the cottage door [he tapped].
  [The wolf tapped] when he reached the cottage door.

Pronominalization occurs equally as often in exposition as in dialogue, but dialogue can have slightly different pronominalization rules depending on the relationship between the utterer and the hearer:

- *Speaker self-reference*: First person singular pronouns are used:
  "John thinks John should go eat John's breakfast", John said.
  "I think I should go eat my breakfast", John said.
- *Speaker references hearer(s)*: Second person pronouns are used:
  "John thinks Mary should go eat Mary's breakfast", John said.
  "I think you should go eat your breakfast", John said.
- *Reference to speaker and hearer, or to speaker and a third party*: First person plural pronouns are used:
  "John and Mary should go eat John and Mary's breakfast", John said.
  "We should go eat our breakfast", John said.
- *Reference to a third party*: Third person pronouns are used:
  "Bob and Mary should go eat Bob and Mary's breakfast", John said.
  "They should go eat their breakfast", John said.
- *Position with respect to the quotation*: Change from first to third person:
  "Oh man, I forgot to eat my breakfast!" John muttered to himself while grabbing his shoes.

2.3.3. Discourse markers

Discourse markers constitute a category of mainly semantic relations that have yet to be included in a comprehensive, systematic way in an implemented natural language generation system, and yet they play a vital role in defining the linguistic style of a narrative.

Although models for discourse markers exist, they are either not comprehensive (for example, focusing solely on temporal markers [24]) or consist only of categorical hierarchies [35,52]. Grote and Stede indicate how they believe discourse marker selection could be included into an overall NLG system [25], but focus on task-oriented generation, which uses only a subset of the discourse markers found in narrative prose. [50] provide the first attempt at generating a restricted set of multiple discourse markers in an explanatory domain.

Because of the many discourse marker types and the sentential relations whose semantic content they combine, we mention only a few here. A more complete enumeration can be found in [52, Section 8.137].

- *Temporal*: Including *while*, *after* and *before*, temporal markers indicate the temporal difference between the semantic content of two adjacent sentences.
- *Causal*: Causal markers such as *because* and *despite* introduce a causal dependency between sentences.
- *Enumerative*: One of the most often encountered discourse markers found in narrative prose, this group, including *first*, *next*, and *then*, among many, many others, must be included in a model of narrative prose if it is to faithfully reproduce the types of sentences found in human-produced narratives.
- *Reinforcing*: This second most often encountered group contains words such as *also*, *besides*, *as well*, and *too*, especially in quoted dialogue.

To see how frequently inter-sentential discourse markers are used in narrative prose, consider this Little Red Riding Hood paragraph [61]:

> **Now**, it happened that a wolf, a very cruel, greedy creature, **also** heard her as she passed, and longed to eat her for his breakfast, **but** he knew Hugh, the woodman, was at work very near with his great dog, **and** he was afraid **because** they might hear Little Red Riding Hood cry out **if** he frightened her, **and then** they would kill him. **So, after** following her a little way, he came up to her very gently **whereupon** he said, "Good day, Little Red Riding Hood, where are you going?"

2.3.4. Summary of mechanical realization

We have endeavored to show that there are a very large number of low-level linguistic details inherent in the construction of any narrative text involving actor characters, including but not limited to character dialogue, pronominalization, and discourse markers. Because existing story generators have not attempted to address these issues, they are currently incapable of faithfully reproducing naturally occurring prose, which invariably contains these types of linguistic constructions.

2.4. Limitations

Although the narrative model presented here addresses a broad range of phenomena, it is not comprehensive. Our claim in this respect is that the narrative model is capable of handling the major phenomena that define complex narratives. However, there are at least four limitations that we have identified:

- *Relative Ordering of Narrative Stream Primitives*: Currently, the implementation of the model requires that delimiting primitives precede all other narrative primitives for some sentence, followed by a single base primitive, and then finally by all of the modifying primitives. Furthermore, it requires that when one modifying primitive modifies another, it must follow the second modifying primitive. Although linearity is a necessary property because of the nature of reading or writing text, it is not clear that the ordering requirements must be so strict for the narrative planner.

- *Lack of Syntactic Information*: Because the narrative stream is encoded at the conceptual level, it contains no specific syntactic information. Although in most cases this is considered an advantage (especially for generating multilingual narratives), there may be times when a narrative planner would like to force an utterance to be of a particular syntactic category. This is impossible under the current formulation of our narrative model unless the fabula and lexicon are configured to allow no other syntactic options.
- *Multiple Copies of Concept Instances*: When two or more adjacent narrative stream primitives contain distinct copies of the same concept instance, it is unclear which one is intended to be modified by a subsequent modifying primitive. For instance, if the part of the narrative stream corresponding to the sentence, "The wolf made his way through the forest", (which literally would be "The wolf made the wolf's way through the forest") contains the modifying narrative primitive `(detail wolf001 wicked-attitude)`, which instance of `wolf001` would be modified, the one corresponding to the subject or the one corresponding to the possessive relation? The current narrative model provides no consistent method for discriminating between them. (Our implementation selects the most recently mentioned concept.)
- *Division of Labor with Narrative Planning*: Our narrative model assumes that the narrative planner is responsible for content creation while the narrative prose generator is responsible for organizing that content into grammatical sentences. Implicit in the latter process is the pervasive influence of linguistic knowledge. However, the narrative planner is assumed to need no linguistic knowledge whatsoever. Yet it produces narrative stream primitives and the fabula. So are the narrative stream primitives themselves linguistic in nature? If so, how does the narrative planner produce them; if not, how does the narrative prose generator make linguistic inferences with them? Because the narrative planner we constructed is a research prototype—albeit a non-trivial one—it would be desirable to use a full-scale narrative planner, such as those involved in research directed specifically at narrative planning [66,69] to further explore this question.

3. An architecture for narrative prose generation

We have designed a start-to-finish architecture for creating organized paragraphs of coherent exposition and dialogue from a fabula, narrative stream, and various stylistic factors. The architectural components gradually convert the conceptual description of the specification in a "pipelined" fashion [53] into an increasingly linguistic form before the final linguistic deep structure is realized as the textual narrative. A standard pipelined architecture for narrative generation would consist of the following components:

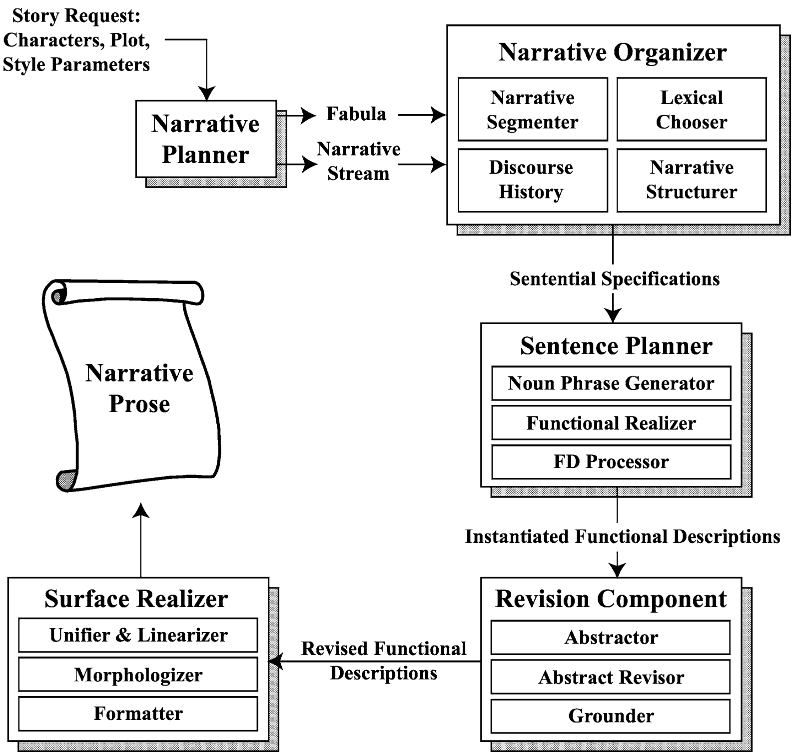

Fig. 7. A narrative prose generation architecture.

- *Narrative planner*: Creates a sequence of characters, props, locations, descriptions and events driven by the narrative plot.
- *Sentence planner*: Creates the deep linguistic representation for individual sentences from the narrative plan.
- *Surface realizer*: Converts deep linguistic structures into a sequence of surface strings using a grammar.

To improve prose quality, we introduce several additional modules into the generic deep generation architecture pipeline. Although the focus here is on *narrative* generation instead of generation for explanations or other types of communication, in principle all of the architectural modules we describe are necessary for scaling up to high-quality multi-page text generation. These modules[1] consist of a *discourse history*, a *lexical choice* component, a *revision* module, and a *narrative formatter* (Fig. 7). Each of these modules is responsible for a key attribute relating to the quality of the final narrative prose related to some aspect of the narrative design criteria:

[1] Although we aren't the first to propose these individual modules, STORYBOOK is the first system to include all of them in a working multi-page text generator.

- *Discourse History*: A narrative prose generator must keep track of which entities have been mentioned in exposition or dialogue and provide for pronominalization, definiteness, and contextual deixis.
- *Lexical Choice*: To ensure that the narrative prose is not repetitive, the generator should allow alternate lexicalizations and appropriately restructure the theta roles [19,60].
- *Revision*: To create complex sentences found in human-produced narrative, a revision component [8,54,57] can enhance the original prose with subordinating clauses, reordering of sentences and aggregation, and introducing appropriate discourse markers as described in Section 2.3.3.
- *Narrative Formatting*: Because most deep generation systems have been geared toward explanation generation, they lack facilities necessary for narrative prose such as dialogue formatting, as discussed in Section 2.3.1.

As an example, consider how this architecture would go about writing a Little Red Riding Hood fairy tale. A narrative planner first generates a sequence of fabula operators and narrative primitives representing the characters and events for that story. These are sent to the narrative segmenter (part of the narrative organizer, Fig. 7) which examines the narrative primitive stream to determine boundaries such as paragraphs, sentences, and quoted dialogue.

Additionally, the sentences are examined for appropriate alterations by the lexical choice component and the discourse history module makes changes to individual linguistic elements depending on the content of preceding sentences. The final component of the narrative organizer, the narrative structurer, orders the segmented narrative stream into a sequence of sentential specifications.

The sentence planner extracts semantic type information from each sentential specification and selects a linguistic deep structure template. The template is then instantiated with deep structure noun phrases extracted from the narrative stream. The sentence planner sends the instantiated sentential deep structures along with ordering constraints from the narrative organizer to the revision module. The revision system uses revision operators to compare drafts for each paragraph-sized chunk of sentence structures, and invokes the appropriate revision rules to create more syntactically complex sentence structures.

After all of the sentence structures have been revised, they are passed to the FUF/SURGE surface realization system [19]. FUF creates surface sentence strings from the sentential structures by unifying them with a grammar, ordering the syntactic elements with a linearization module, and marking the results with the correct morphology. In addition, the resulting text is modified to satisfy appropriate narrative formatting constraints, and can also be marked up with formatting commands, such as HTML and other XML languages and LaTeX commands [30]. Finally, the resulting text is presented as a complete narrative to the reader.

3.1. Narrative planner

Because the focus of our research is on the production of high quality narrative prose rather than on methods for the production of narrative plots, the narrative planner is not greatly elaborated in the architecture we present here. Other researchers have investigated this area, frequently termed *story generation* [37,44,63]. Although our narrative planner allows variations in the initial conditions as mentioned in our narrative model, it makes certain assumptions that restrict the type of narratives it can produce. These assumptions include restricting it to fairy tale genres, historical time periods, limited types of plots, and a set cast of characters.

The narrative planner has four main tasks:

(1) *Character Design*: The planner should decide how many and which types of characters are necessary, and what roles they play in the narrative (e.g., hero, villain, etc.)
(2) *Plot Construction*: The planner should select a plot that includes a set of goals, the characters, the props and locations that allow the characters to achieve those goals, and the sequence of events that lead from the initial story setting through the final plot resolution.
(3) *Scene Construction*: The planner should order the events in the plot in some coherent structure and determine the presence or absence of characters and props as the plot progresses.
(4) *Generation of Fabula and Narrative Stream*: The planner should embed the characters and props into the fabula and create a narrative stream that reflects the orderly progression of events, descriptions and states.

The narrative planner used in STORYBOOK is necessarily simple, as our work concentrates on devising the narrative interface level (modeling the fabula and narrative stream) and constructing the narrative prose generation system. Our narrative planner consists of a finite state automata, where a path to a terminal state defines a story. Each node along the path contributes fabula operators and narrative stream primitives to the growing narrative plan. Note that this procedure explicitly carries out step (4) above, with steps (1)–(3) being implicitly conducted during the process of constructing rather than executing the FSA.

Upon completion of the fabula and narrative stream, those structures are sent along with particular stylistic directives to the first component of the narrative prose generator, the Narrative Organizer.

3.2. Narrative organizer

The narrative organizer functions as the initial organizational stage of a standard explanation generator. Its task in narrative generation is convert the raw semantic content from the narrative stream and fabula into a series of manageable sentence-sized chunks. This is accomplished in four stages:

(1) *Narrative Segmentation*: Because the narrative stream is generated by the narrative planner as one long sequence, it must be segmented into groups of one to several narrative stream primitives whose content can fit into a simple propositional sentence.
(2) *Discourse History*: After the narrative stream primitives have been segmented, the discourse history module searches linearly from beginning to end and opportunistically annotates concept instances with the appropriate definite/indefinite forms, contextual deixes, and pronominalizations.

(3) *Lexical Choice*: To provide variation in the narrative prose, the lexical chooser monitors the discourse history to check for excessive repetition of concepts or relations. If this occurs, the lexical chooser may replace narrative primitive relations with similar relations from the fabula and reorder the concepts accordingly.
(4) *Narrative Structuring*: Once the narrative stream primitives have been segmented and analyzed, the actual sentential structures that represent the propositional content of the sentence must be created for each group of primitives, where each group represents a single sentence.

Because the fabula and story ontology exist by these stages, they can be used to make appropriate linguistic decisions. For example, the discourse history module may examine the gender of a concept instance and thus know to substitute "she" for "Little Red Riding Hood". Similarly, the narrative structurer may examine the lexicon entry of a narrative stream primitive's main relation to determine its theta frame and its arguments' semantic type restrictions.

3.2.1. Narrative stream segmenter

The narrative stream segmenter recognizes two major types of discourse segments: sentence breaks and paragraph breaks. Although the narrative stream usually contains some explicit paragraph breaks, it contains no explicit indication of when a sentence break should occur. Nor does the narrative stream segmenter have the final say over when a sentence break will occur, as the revision component may at a later time aggregate two sentences together, thus destroying a sentence boundary imposed by the segmenter.

In narrative prose, there are four reasons for inserting paragraph boundaries:

- *Scene Boundaries*: When actors move to a new location or time, or when the exposition or dialogue switches to different characters at a different location and/or time, a paragraph break (or chapter/section break, for longer narratives) is inserted to help the reader realize that a shift has occurred.
- *Dialogue Boundaries*: In conversations of standalone dialogue (as opposed to exposition) where the "floor" alternates between two or more speakers, paragraph breaks are inserted to indicate to the reader that a different character is now speaking [23,58,62].
- *Topic Boundaries*: As in well-established forms of writing such as essays, major shifts in topic are indicated by the use of paragraph breaks. Because this decision cannot be made locally (due to focalization effects among others), the segmenter relies on the narrative planner to make this decision.
- *Size Boundaries*: Paragraph breaks due to size constraints are the last resort of the narrative segmenter when the previous three constraints fail to create paragraphs of reasonable size. Although some writers use very small or very large paragraphs to create certain dramatic effects, our narrative segmenter assumes that paragraphs should be from three to twelve sentences in length if at all possible, and tries to balance the size of adjacent paragraphs.

Three of the four paragraph break types are indicated by the narrative planner, through the `new-scene`, `dialogue-actors`, and `topic-shift` narrative stream primitives. Forcing paragraph breaks due to size constraints is currently computed by the narrative segmenter using strictly numerical criteria.

The second type of discourse segment is the sentence break. The segmenting point for sentences can be determined based on the type of the narrative stream primitive. Because delimiting primitives do not carry semantic content that is realized as prose in the actual narrative, they can be disregarded here. Because modifying primitives are subordinate clauses or phrases that depend on a base primitive for realization, they too can be disregarded.

Thus, to segment the narrative stream into sentence-sized packets, the narrative segmenter simply marks the beginning of a new sentence when it encounters a base primitive in the narrative stream and marks the end of the sentence as it encounters a delimiting primitive or the next base primitive in the sequence. Note that this requires the narrative planner to produce all modifying primitives after the base primitive they modify (see Section 2.4).

3.2.2. Discourse history

The discourse history is one of the most important modules for improving the quality of narrative prose. The discourse history module can be viewed as a filter that removes or alters elements of sentences that, while grammatically correct or function well in an isolated sentence, detract from the coherence of the discourse environment as a whole. The discourse history module examines the segmented narrative stream for three factors:

- *Definite vs. Indefinite*: Although in general noun phrases are indefinite when mentioned for the first time in discourse and definite subsequently, this is not always the case. For instance, mass nouns behave differently as do noun phrases that refer to objects that can be considered to be in "plain view" of the discourse participants, among other reasons. Also, concept instances which are related to a previously mentioned concept instance via the `AddAlias` fabula operator (Section 2.1) should not be marked as indefinite even when they are mentioned the first time.
- *Contextual Deixis*: Occasionally, dialogue participants will make references to contextual elements that are neither people nor objects. For example, if Little Red Riding Hood enters Grandmother's house and says "This can't be right", she is referring to the contextual *situation* and not to any object or person inside Grandmother's house.
- *Pronominalization*: Narratives contain a variety of pronominalized forms, including anaphora and dialogue pronouns.

As mentioned in Section 2.3.2, one of the jobs of a pronominalization algorithm in generation is to create pronominal anaphora [26,33,42,47,67]. One of the more popular approaches is to reverse existing models of centering theory [23,58,64] for parsing to generate rather than to resolve anaphora.

Our corpora analyses have not uncovered any instances of either exposition or dialogue requiring the application of centering theory to drive pronominalization. Given that most works on centering theory concentrated on task-oriented dialogues, it may be that different domains or more complicated subject matter would require a more comprehensive approach. Alternatively, it may be the case that the narratives we studied are not of sufficient complexity, or that different genres require different pronominalization algorithms.

Instead of a more complicated theory, we use a straightforward counting approach [7] by keeping records of which concepts and concepts instances have been seen and in what contexts. In addition, this counting approach allows us to easily reset the discourse history when we reach a scene break. Furthermore, it allows us to interleave the marking of definiteness with the marking of pronominalizations. The algorithm we developed considers locally the current discourse leaf node and the rhetorical structure above it, and globally makes use of the following data:

- *Nominal element distance*: How many nominal elements have been seen since this particular element was last used.
- *Recency*: How many different, distinct nominal elements have been seen since its last use.
- *Sentential distance*: How many sentences (or paragraphs) have appeared since the last usage of this nominal element.

These features are used to opportunistically decide when to replace the lexicalizations of concepts and concept instances with the appropriate linguistic deep structure information. For example, a decision to make "wolf" or "butter" be indefinite when they are first mentioned in the discourse context may result in an indefinite article for the count noun ("a wolf") or either no article or a determiner sequence for the mass noun ("butter", "some butter").

Similarly, a pronominalization decision may replace a repetitive instance of "Grandmother" with the single feminine pronoun "she/her/herself". Because knowing when an instance is repetitive involves using the discourse history data, it is necessary to perform the segmentation procedure before pronominalization during the construction of the discourse history table. Otherwise, it is difficult to determine how many sentences ago a concept instance was last seen, or whether there was a topic shift or scene shift since.

After the discourse history module has examined the original narrative stream, the end result is an improved, modified narrative stream.

3.2.3. Lexical chooser

Lexical choice is, according to [19, p. 7], "the process of selecting open-class lexical items in a linguistic structure". The purpose of the lexical chooser is to select these open-class lexical items[2] in furtherance of the goal of improving narrative prose quality. Although Elhadad's model worked well for improving explanatory text in a limited domain, there is no known, general lexical choice algorithm for unrestricted narrative

[2] *Closed-class* items are those like English prepositions where the lexical set is considered complete, i.e., no new prepositions are being added to the English language. Open-class items are all other types of words, such as nouns and verbs.

prose. Thus, for our purposes, we employ a much less sophisticated lexical chooser whose sole responsibility is to prevent lexical repetition and encourage lexical variation.

Verbs and nouns are extremely important open-class lexical items because of their frequency in sentences. Additionally, repetition is possible in other features besides syntactic categories. We thus focus on three important areas where we have observed frequent repetition in narrative prose:

- *Repetition in Noun Phrases*: Languages typically contain many similar nouns that are connotatively different in their use. For example, Little Red Riding Hood might live in a house, cottage, shack, hut, cabin, etc.
- *Repetition in Verb Phrases*: Similarly, events have varying connotations. Little Red Riding Hood can walk through the forest, amble, stroll, etc.
- *Repetition in Theta Role Ordering*: Many event verbs also impose different theta and case frames even though they describe similar actions. For example, Little Red Riding Hood might give her grandmother the cookies or grandmother might receive the cookies from Little Red Riding Hood.

To provide for variation in narrative prose, the lexical chooser monitors the accumulating discourse history to check for excessive repetition of concepts, relations and narrative stream primitives. If this occurs, the lexical chooser may replace the offending concepts with related concepts found in the story ontology. If a particular relation is repetitive, an alternate relation from the story ontology can be located and its argument concepts reordered accordingly. If it discovers repetition of the narrative stream primitives themselves, it could swap that narrative stream primitive for another closely related one.

Swapping is accomplished in STORYBOOK by augmenting the story ontology with a large variety of relations that are nearly synonymous but with different lexicalizations and/or theta and case role orderings, and a selection method for distinguishing which concepts and relations are suitable for substitution. Although this does not compare to more sophisticated methods [19,60] and is not recommended as a final solution to the problem of lexical choice, it is sufficient to satisfy our goal of preventing repetitive prose.

3.2.4. Narrative structurer

Once the narrative stream has been segmented, it can be characterized as a sequence of *narrative stream primitive groups*, where each group is headed by a single narrative stream base primitive and zero or more modifying primitives. However, in the classic pipelined NLG architecture, discourse planners typically produce a set of singular structures that correspond to sentence-sized chunks of the discourse plan. Thus, the job of the narrative structurer is to convert the narrative primitive groups into individual sentential specifications.

Each sentential specification contains high-level semantic information about the roles that the arguments should play in a sentence. However, there is no direct mapping between the content of the narrative stream primitives and the names of the roles in a sentential specification. In order to construct a correct sentence, we must first map from the intentional relation found in the base narrative primitive to the fabula and lexicon, which contain the semantic theta frame for the relation of interest, and then map the narrative primitives' arguments to the new sentential specification.

When a narrative stream primitive group is received, the narrative structurer creates an empty sentential specification, and then iterates through the base and modifying primitives. The base primitive directs the structurer to retrieve the appropriate theta frame for its relation from the fabula via the lexicon, and then adds the appropriate content from the base primitive to the named roles specified by the theta frame. Similarly, for each modifying primitive, the structurer retrieves the new relation's theta frame and either adds the values of the modifying primitive to the named roles of the new theta frame or modifies a pre-existing role created by the base primitive.

In addition to thematic roles, the narrative structurer is responsible for making decisions about dialogue realization and tense shifting. It must also account for the dialogue elements presented in Section 2.3.1, such as whether the utterer is to be included, and if there are any manner or co-event clauses. The result of iterating through all narrative stream primitive groups is a completed set of sentential structures which can be sent to the sentence planner. After this step, the narrative organizer has transformed the original narrative stream into a sequence of sentential structures that represent the propositional content of simple sentences. We now discuss the sentence planner, which converts those sentential specifications into linguistic deep structures.

3.3. Sentence planning and revision

The function of the sentence planner is to take a specification for the semantic content of a sentence and to plan the roles (either semantic, syntactic or both) that the specification's elements play in a particular sentence. Because our approach utilizes an off-the-shelf surface realizer that expects particular semantic roles, we require that a sentence planner produce semantic sentential specifications. The sentence planner takes these knowledge-based sentence structures and converts them into linguistic deep structure representations known as *functional descriptions* (FDs), which are hybrid semantic/syntactic entities that can be used to directly produce text [19].

Once the sentence planner has created the FDs (representing the deep linguistic structure), they can be sent directly to the surface realizer for text generation. However, because the quality of a sequence of simple propositional sentences is notoriously poor, the AUTHOR architecture instead maintains the paragraph separations imposed by the narrative segmenter and proceeds to send each paragraph-sized batch of FDs to the revision component in order to improve overall prose quality.

To dynamically improve narratives while at the same time combating the problems of complexity and efficiency, STORYBOOK uses a revision component for narrative prose generation that operates on *abstract narrative plans*. Encoded in a minimalist representation consisting of only those features that are most critical for making revision decisions, abstract narrative plans promote efficiency by reducing the complexity of drafts. This model of revision-based narrative generation employs a non-monotonic unification framework to compose and edit multiple drafts of narratives. Specifically, it focuses on clause aggregation and clause reordering, problems that have been the subject of increasing attention from a variety of perspectives in the natural language generation community [15, 54,57].

Given the initial *ground level* results from a narrative plan, the REVISOR revision component [8] transforms it into an abstract narrative plan, which contains only the most critical lexical, syntactic, and semantic features needed for revision. This abstract narrative plan represents the first draft. REVISOR then iteratively refines draft after draft by applying revision operators and evaluating the resulting narrative plans with respect to quantitative discourse, style and length constraints. When a final draft has been constructed, it is re-transformed into a (more complex) ground-level narrative plan which is then sent to the surface realizer for text generation.

To address these problems, we have developed a revision system (a predecessor of this system that operates on explanatory prose is described in [8]) that dynamically improves multi-page narratives by searching through an abstraction space of narrative drafts. Rather than enacting revisions by reasoning about all of the syntactic and semantic details of discourse plans, it abstracts away all but the most essential aspects of a narrative plan and performs all manipulations on drafts encoded in the abstracted representation. By conducting its search through this abstraction space, it efficiently evaluates candidate revision operators, applies selected operators to create new drafts, and retracts operators to return to previous drafts (via standard backtracking inherent in the unification process).

A typical abstract revision operator that performs clause aggregation considers two adjacent clauses in the abstract planning space (Fig. 8, parts (a) and (b)). The two abstract clauses as well as the abstract revision operator itself (c) are encoded in the FUF functional unification formalism [19]. In the example, two independent sentences are combined into a single sentence where the second sentence is transformed into a subject relative clause as indicated in (d) when the revision rule finds no violated constraints.

Once upon a time there was a tiny house.

```
((CAT  ABSTRACT-FD)
 (REVISION-SYNTAX    INDEPENDENT-SENTENCE)
 (REVISION-SEMANTICS    EXISTENTIAL-PROPERTY)
 (MATRIX-RELATION    EXIST-BEING)
 (SUBJECT-INDEX    NONE)
 (OBJECT-INDEX    HOUSE001))
```

(a)

The house was in a dark forest.

```
((CAT  ABSTRACT-FD)
 (REVISION-SYNTAX    INDEPENDENT-SENTENCE)
 (REVISION-SEMANTICS    SIMPLE-LOCATION)
 (MATRIX-RELATION    BEING-LOCATED-IN)
 (SUBJECT-INDEX    HOUSE001)
 (OBJECT-INDEX    FOREST001))
```

(b)

```
((CAT   ABSTRACT-FD)
 (REVISION-SYNTAX    INDEPENDENT-SENTENCE)
 (REVISION-SEMANTICS
          SIMPLE-LOCATION)
 ({^ REVISION-SEMANTICS}
          EXISTENTIAL-PROPERTY)
 ({^ OBJECT-INDEX}    OBJECT-INDEX)
 (REVISION-SYNTAX    REMOVE)
 (REVISION-SYNTAX
          SUBJECT-RESTRICTIVE-RELATIVE)
 (LINKED?   TRUE))
```

(c)

Once upon a time there was a tiny house, which was in a dark forest.

```
((CAT   ABSTRACT-FD)
 (REVISION-SYNTAX    INDEPENDENT-SENTENCE)
 (REVISION-SEMANTICS    EXISTENTIAL-PROPERTY)
 (MATRIX-RELATION    EXIST-BEING)
 (SUBJECT-INDEX    NONE)
 (OBJECT-INDEX    HOUSE001))
((CAT  ABSTRACT-FD)
 (REVISION-SYNTAX    SUBJECT-RESTRICTIVE-RELATIVE)
 (REVISION-SEMANTICS    SIMPLE-LOCATION)
 (MATRIX-RELATION    BEING-LOCATED-IN)
 (SUBJECT-INDEX    HOUSE001)
 (OBJECT-INDEX    FOREST001))
```

(d)

Fig. 8. An example of a revision operator in STORYBOOK.

The revision component improves the narrative in the following three phases:

(1) *Narrative Plan Abstraction*: It maps the initial *ground level* narrative plan to an *abstract narrative plan* by excising all but the most essential syntactic and semantic features of each sentential specification.
(2) *Abstract Narrative Revision*: It applies revision operators to the abstract narrative plan to create a *draft tree*, where each node is a draft derived from its parent by applying an operator. At each iteration, it selects an operator by evaluating (a) a candidate operators' preconditions and (b) a set of *revision predicates* that specify desirable narrative properties.
(3) *Narrative Plan Grounding*: Because abstract narrative plans cannot be realized as text—the vast majority of linguistic detail has been removed—we must map the final draft to a ground level narrative plan. We reconstitute a ground level plan by first locating the syntactic and semantic information that was stored away during the abstraction phase and then re-integrating it into the form specified by the abstract narrative plan.

The revised ground level narrative plan (the revised sequence of sentential specifications) is then passed to the surface realizer, which produces the actual text for the final story.

3.4. Surface realizer

Once the Sentence Planner has created a group of functional descriptions from the original narrative plan, the functional descriptions (FDs, or deep linguistic structures) are passed one by one to the final stage of narrative prose generation, surface realization. The role of the surface realizer is to take an FD and create text for presentation to the reader. Creating text requires detailed and extensive mechanical knowledge about how language is organized, and to this end surface realizers fulfill five major functions:

(1) *Add closed-class and default lexical items*: Syntactic categories such as articles, prepositions, and pronouns which are semantically specified but not lexically specified in the initial functional description are lexicalized by the surface realizer's grammar and then added to the FD.
(2) *Ensure grammaticality of utterances*: Functional descriptions that represent ungrammatical sentences should be rejected.
(3) *Properly order all lexical items*: Because (English) text proceeds in a (left-to-right) sequential order, the surface realizer should ensure that the text produced conforms to the language's linear precedence rules.
(4) *Adjust lexical items for morphology*: Most languages change lexical stems to account for gender, number, and other grammatical features.
(5) *Add punctuation and other formatting directives*: The surface realizer should be capable of reproducing all orthographical features, and augmenting the surface text with markup commands for HTML, XML, LaTeX, or other display-oriented presentation languages [30].

The FUF surface realization system [19] employs an extension of the Functional Unification Grammar (FUG) approach. The main component of the surface realizer is the *grammar*, a large data structure that consists of a series of named disjunctions that represent the paths that can be taken to achieve a grammatical sentence. The grammar itself is represented as a very large functional description. The process for checking the grammaticality of a particular functional description is termed *unification*. Unification takes two FDs, one representing the grammar and one representing a FD of some syntactic category (usually a sentence), and attempts to determine whether they are compatible (whether they unify successfully). Failure to do so indicates that the FD does not represent a valid, grammatical sentence.

Thus the sentence planner iteratively passes the sentential functional descriptions to the surface realizer, which unifies each one with the grammar. The grammar imposes default lexical values, linear precedence constraints, morphology and punctuation. The resulting augmented sentential functional description is then *linearized* to enact the linear precedence constraints, and the resulting linearized lexical items are concatenated together to produce a final text string. Once the text strings for all sentential functional descriptions are collected, they are concatenated together and presented to the reader.

3.5. Summary

We have designed an extended pipelined architecture for narrative prose generation that converts a narrative plan into high quality narrative prose. Narrative prose generation begins when a narrative planner constructs a fabula and narrative stream representing a particular story. The narrative plan populates the fabula and narrative stream with characters, props, events, descriptions, and scenes. The narrative stream is then segmented and passes through the discourse history filter and lexical chooser, which increase the linguistic variety and force the narrative stream to conform to human story expectations.

Next, the narrative stream is converted into sentential structures representing the semantic content of the segmented narrative streams. The noun phrase generator is then used to create linguistic deep structures representing concepts and the sentence planner then creates a sequence of functional descriptions. Afterwards, the revision component further improves the prose quality by reordering and aggregating single proposition sentences into complex sentences. Finally, the list of revised sentential functional descriptions is passed to the surface realizer to produce formatted text[3] and presented to the user.

Although we focus mainly on narrative prose generation, the architectural components presented here are not narrative-specific. For example, the sentence planner and revision module were originally designed to assist the KNIGHT [40] explanation generation system in producing instructional text in the domain of college-level botanical anatomy and physiology, and STORYBOOK has also generated text from the New York Times. The "pipeline" architecture has been applied extensively [53] in NLG, and none of the

[3] STORYBOOK is capable of producing text in ASCII or marked up according to HTML and LaTeX specifications.

architectural models presented here are completely novel. However, STORYBOOK is the first system to combine all of them into a single working system that supports the production of multi-page prose generation.

An example narrative produced by STORYBOOK is described in Appendix A, along with a description of the most relevant and interesting details.

4. The STORYBOOK evaluation

The existence of three optional architectural modules (discourse history, lexical choice, and revision) allows us to conduct an *architectural ablation* experiment. By selectively removing a component, the resulting text of a story will be changed in some way. The narrative planner, sentence planner and surface realizer are vital components; without them text cannot be produced at all. However, removing the other modules will result in text that we expect to be degraded in some fashion. Thus without the discourse history, the system will be unable to produce pronouns or appropriately mark nouns for definiteness. Without the revision component, the system will produce a minimal number of propositions per sentence due to the lack of clause aggregation. Finally, removing the lexical choice module will result in a decrease in the variability of the lexical forms of verbs or nouns.

Given these three architectural modules, there are 2^3 or 8 possible pairwise comparisons between the presence or absence of each component when used to produce a narrative. Due to constraints on the logistics of the evaluation process, we decided to utilize only five of those pairwise comparisons: the two all-or-none approaches and the three approaches where one specific architectural module is ablated.

The remaining three approaches would evaluate the enhancement that each module adds to the whole rather than what is missing when each is removed. This approach leads to a somewhat more effective comparison, because as more modules are removed from the generation process, the resulting prose becomes progressively less desirable and thus unwanted effects from the absence of multiple architectural modules might overlap and affect a test subject's experience in ways that could not be easily teased apart when analyzing the data.

The ablation of these architectural modules can have a significant impact on text quality, even over very small text segments, as is shown in Table 2.

4.1. Evaluation methodology

To investigate the behavior of the STORYBOOK system [9], we created a modestly sized narrative planner (implemented as an FSA containing approximately 200 states), enough to produce two stories comprising two and three pages respectively. Furthermore, we fixed the content of those stories and ran the five different versions of STORYBOOK described above on each one. This resulted in ten total narratives which we presented to our test subjects using the grading factors shown in Fig. 9. While versions were different in the sense that certain modules were either ablated or not, the two stories differ because they were created from two different sets of fabula operators and narrative stream primitives,

Table 2
Selected texts for versions A through E

Version	Example text
A: Complete Revision: Yes Lexical Choice: Yes Discourse History: Yes	She had not gone far when she met a wolf. “Hello”, greeted the wolf, who was a cunning looking creature. He asked, “Where are you going?” “I am going to my grandmother’s house”, she replied.
B: No Revision Revision: No Lexical Choice: Yes Discourse History: Yes	She had not gone far. She met a wolf. “Hello”, greeted the wolf. The wolf was a cunning looking creature. He asked, “Where are you going?” “I am going to my grandmother’s house”, she replied.
C: No Lexical Choice Revision: Yes Lexical Choice: No Discourse History: Yes	She had not gone far when she met a wolf. “Hello”, said the wolf, who was a cunning looking creature. He said, “Where are you going?” “I am going to my grandmother’s house”, she said.
D: No Disc. Hist. Revision: Yes Lexical Choice: Yes Discourse History: No	Little Red Riding Hood had not gone far when Little Red Riding Hood met the wolf. “Hello”, greeted the wolf, who was the cunning looking creature. The wolf asked, “Where is Little Red Riding Hood going?” “Little Red Riding Hood is going to Little Red Riding Hood’s grandmother’s house”, replied Little Red Riding Hood.
E: Nothing Revision: No Lexical Choice: No Discourse History: No	Little Red Riding Hood had not gone far. Little Red Riding Hood met the wolf. “Hello”, said the wolf. The wolf was the cunning looking creature. The wolf said, “Where is Little Red Riding Hood going?” “Little Red Riding Hood is going to Little Red Riding Hood’s grandmother’s house”, said Little Red Riding Hood.

(1) *Overall*: On an absolute scale of how good fairy tales should be in general, evaluate the story on an A–F scale (A, B, C, D, F).
(2) *Style*: Did the author use a writing style appropriate for fairy tales?
(3) *Grammaticality*: How would you grade the syntactic quality of the story?
(4) *Flow*: How well did the sentences flow from one to the next?
(5) *Diction*: How interesting or appropriate were the author’s word choices?
(6) *Readability*: How hard was it to read the prose?
(7) *Logicality*: Did the story omit crucial information or seem out of order?
(8) *Detail*: Did it have the right amount of detail, or too much or too little?
(9) *Believability*: Did the story’s characters behave as you would expect?

Fig. 9. Grading factors presented to readers.

each implemented as an FSA (cf. Section 3.1). Thus Story #1 potentially has different characters, different events and properties, and different props than Story #2 has.

A total of twenty students were selected from North Carolina State University's Departments of English and Communication via first-come first-serve email notices. All of the students were registered in upper division or graduate courses in those departments. Each subject was asked to read the directions and ask for clarifications before the evaluation proceeded and was assigned a random version of each story for evaluation. Subjects were not informed prior to their completion of the questionnaire that the narratives were generated by software. Subjects were paid $25.00 for their participation.

Because each subject compared two versions of Story #1 to each other and two versions of Story #2 to each other, every subject saw a total of four narratives. To prevent subjects from evaluating the same types of stories in succession, we devised the following policy:

(1) Each subject would read four distinct story versions out of the total of five, two from each story (e.g., subject #1 read versions A and B from Story #1, and versions D and E from Story #2). No subject would read the same version twice. All subjects would read the 2 versions of Story #1 first.
(2) Each version would be read by the same total number of subjects (i.e., each version of every story would be read by 8 separate subjects).
(3) Each pairwise comparison of different versions would be read by two separate subjects (e.g., Subjects #1 and #11 would both read versions A and B of Story #1 and versions D and E of Story #2).
(4) For each pair of subjects reading the same two versions, the narratives would be presented in opposite order (e.g., Subject #1 read version A first and then version B, while Subject #11 read version B first followed by version A).
(5) Subjects would be randomly assigned narrative versions on a first-come first-serve basis; all subjects would perform their evaluations within a few hours of each other at a single location.

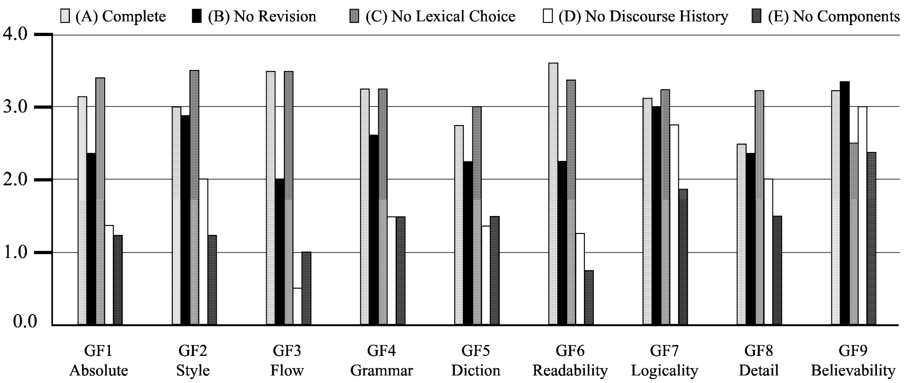

Fig. 10. Means for Story #2: 8 evaluations per Version × Grading Factor × Story.

Table 3
Significance values (with Bonferroni adjustment): $* = p < 0.01$, $** = p < 0.001$

Grading factors	GF1	GF2	GF3	GF4	GF5	GF6	GF7	GF8	GF9	ALL
COMPLETE VS. NO REV.	n.s.	n.s.	**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
COMPLETE VS. NO L. C.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
COMPLETE VS. NO D. H.	**	*	**	**	**	**	n.s.	*	n.s.	**
COMPLETE VS. NOTHING	**	*	**	**	**	**	n.s.	n.s.	*	**
NO REV. VS. NO L. C.	*	n.s.	**	*	*	*	n.s.	n.s.	n.s.	**
NO REV. VS. NO D. H.	**	*	**	**	*	**	n.s.	n.s.	n.s.	**
NO REV. VS. NOTHING	**	n.s.	*	**	n.s.	**	n.s.	n.s.	*	**
NO L. C. VS. NO D. H.	**	**	**	**	**	**	*	**	*	**
NO L. C. VS. NOTHING	**	**	**	**	**	**	*	**	**	**
NO D. H. VS. NOTHING	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.

Subjects graded each narrative following the instructions according to an A–F scale, which we then converted to a quantified scale where $A = 4.0$, $B = 3.0$, $C = 2.0$, $D = 1.0$, and $F = 0.0$. The resulting scores were then tallied and averaged. The means for Story #2 is shown in Fig. 10.

To determine the quantitative significance of the results, we performed an ANOVA test over both stories. The analysis was conducted for three independent variables (test subject, story, and version) and nine grading factors (labeled GF1–GF9, as described in Fig. 9). Because not all possible grading combinations were performed (only 80 observations, or $20 \times 2 \times 2$, out of a possible 200, or $20 \times 2 \times 5$, due to crossover and time constraints), we performed the mixed procedure analysis. Interactions between variables were only significant for grading factor #9 at 0.0300 for story $*$ version.

The combined results of the ANOVA analysis (Table 3) point to three significantly different equivalence classes of narratives due to the architectural design of the narrative prose generator. The most preferred narrative class, versions A & C (COMPLETE VS. NO LEXICAL CHOICE), were not significantly different from each other overall while they did differ significantly from all other versions (although there were similarities in particular grading factors such as GF2, *style*, between versions A & B). Interestingly, the affinity for versions A & C is strongly correlated for Story #2 (Fig. 10) but only weakly for Story #1. A two-tailed paired t-test evaluating this difference illustrated that versions A & B were not significantly different when only story #1 was considered, but were significantly different in Story #2. The opposite was true for versions A & C when the scores for each story were compared individually.

4.2. Discussion

Indisputably, versions D & E form the least preferred narrative class, differing significantly from all other versions while not differing significantly from each other. Because the architectural commonality between these two versions was the lack of a discourse history (corresponding to a lack of pronominalization and definiteness marking) while versions A, B, and C all utilized a discourse history, we conclude that this architectural component is exceptionally important in the design of a narrative prose generator. Effects of pronominalization and topicalization were previously studied by

[28] although that work focused on recall rates while our evaluation concentrated on the expressed preferences of subjects.

As hypothesized in advance, the full version (Version A) scored quite well while versions lacking a discourse history (Versions D & E) scored poorly. A surprise resulting from the analysis was the mild preference subjects had for the version missing the lexical choice component (Version C) over the full-fledged version. While related work on word choice in spontaneous dialogues has concluded that dialogue participants tend to converge onto a limited set of words (the phenomenon of *lexical entrainment*) [4], fictional narrative by and large does not exhibit the spontaneity reflected in such dialogues.

Upon analysis of the comments in the evaluations specifically comparing versions A & C, it became clear that one principal reason was the test subjects' belief that the increased lexical variation might prove too difficult for children to read (even though we provided no indication that the target audience was children) and thus Version A compared less favorably to Version C due to the more complex and varied words it contained. It is not clear whether a lexical choice component would play a much more significant role in subject matter where the target audience was more mature.

Another possible explanation for the correlation of versions A & C is that the model of lexical choice employed in STORYBOOK is not very sophisticated. If a better model had been used [19,60], this difference might have been more significant. We do not think this result shows that, in general, lexical choice does not add significant value to natural language generation.

The fact that Version B scored less favorably compared to Versions A and C indicates that revision is an important aspect of narrative prose generation. Subjects frequently commented that Version B was "very choppy" or "didn't seem to have good grammar" (even though it was as grammatically correct as the other versions). These comments can be accounted for by the two main functions of the revision component: joining small sentences together and combining sentences with repetitive phrases together while deleting the repetitions. This is related to previous work in reading comprehension on propositional content. This line of work, e.g., [34], has shown that reading rate increases as the number of propositions per sentence increases. Here, however, we have shown that a larger number of propositions per sentence is preferred more than a small number of propositions per sentence (stylistically), although there is undoubtedly an upper limit.

Another important note is that there is a difference among the grading factors themselves. Grading factors (2)–(7) (style, flow, grammar, diction, readability and logicality) directly relate to elements governed by the parameters and rules of the various architectural components of the narrative prose generator. However, grading factors #8 and #9 (detail and believability) are more closely related to the content of the plot line, and as such could be expected to remain relatively constant since the "factual" content of the narratives was held constant across all versions of each story. Despite this, these two grading factors also showed significant differences between versions. Given that the perceptions of the test subjects might have carried over from their responses to previous questions, a future evaluation might randomize the order in which these questions are posed to investigate whether this effect persists.

Finally, there appears to be a link between the appeal of the story content itself and the increase in the absolute (GF #1) and total means for versions A, B, and C. Story #1 is a “classic” Brothers’ Grimm fairy tale in the sense that it has a gruesome ending that serves as a lesson to young children. Thus our Story #1 ends with the wolf devouring Little Red Riding Hood and her grandmother. More modern stories have happier endings, however, and this is reflected in our Story #2 which ends with a woodcutter killing the wolf and extracting the unharmed Little Red Riding Hood and her grandmother. A large number of our test subjects, worried about the potential impact on children (even though we provided no indication that the target audience was children), complained about the “horrible” ending of Story #1 in their written comments and this reader bias appears to have affected the overall scores.

5. Related work

Despite extensive work in both narrative generation and natural language generation, little research has explored the fertile intersection between the two. The lack of progress can be attributed to the distinct problems encountered by the two fields. Narrative generators [37,44,63] typically address the macro-scale development of characters and plot, slowly refining from the topmost narrative goal level down to individual descriptions and character actions. This approach originally descends from the application of planning formalisms to the work of sociolinguists such as Vladimir Propp [51], who created story grammars to capture the high-level plot elements found in Russian folktales.

Because of their focus at the level of plot and characters, narrative generators are incapable of addressing the fundamental linguistic challenges posed by multi-paragraph or multi-page prose as described in Section 2.3. However, these challenges are starting to be addressed by both linguists and computational linguists apart from narrative generation. For example, there has been work in punctuation for extended discourse and dialogue [6,14,18,31,48], as well as work on direct/indirect dialogue [41] and quotation inversion [13].

In addition, much research has been devoted to anaphora resolution [17,23,58,64,65] based on Centering Theory. NLG researchers have attempted to implement the ideas of Centering Theory in their generation systems [26,33,42,47,67], but it is extremely difficult to write a single NLG system that simultaneously generates the large quantity of texts needed to exhibit discourse-level phenomena while consistently employing the deep linguistic representations needed to determine appropriate pronominal forms.

Recent work has also been increasingly successful in the analysis of both the distribution of and the role played by discourse markers, and has greatly extended our knowledge over even the most expansive previous accounts of discourse connectives [52]. For example, corpus studies have been carried out for general discourse markers [35] and temporal discourse markers [24] to aid in the process of discourse marker selection. Additionally, several NLG systems have been implemented for automatically inserting discourse markers appropriately [20,25,50].

Finally, there has been much progress in natural language generation driven by both knowledge bases and tutorial systems. Most full-scale NLG systems [11,27,29,40,45,46, 68] focus on *explanation generation*, creating scientific or instructional text. However, such text significantly differs in the distribution and frequency of syntactic, semantic, and orthographic features from that found in narrative prose, although a few projects do address some of these issues, e.g., [12,18,32,54]. However, the most advanced NLG systems are still not capable of producing more than two paragraphs of text, while the vast majority of naturally occurring narratives are at least several pages long.

6. Conclusions

Narrative prose generation is an exceedingly complex task that involves a diversity of interacting computational mechanisms. Given a narrative specification, a narrative prose generator must consider a broad range of linguistic phenomena to dynamically create extended (multi-page) narratives. Although the limitations of traditional approaches to narrative generation were not immediately apparent, it is clear that the quality of the prose they could produce is unsatisfactory. Without considering narrative generation as a linguistic problem, dynamically creating narrative prose that rivals that of human authors is practically impossible. In contrast, aggressively applying techniques in natural language generation can go a long way toward achieving this goal.

We have proposed a computational model of narrative and an architecture for narrative prose generation. To empirically investigate these, they have been implemented in a full-scale narrative prose generator that can produce multi-page narratives at the deep structure level, avoiding many linguistic problems encountered by previous narrative generators. This system was empirically evaluated, demonstrating that the addition of key architectural modules leads to a significant improvement in the quality of the prose it generates.

By necessity, the work described in this article integrates research from a variety of disciplines. By performing original analyses of low-level linguistic phenomena found in naturally occurring narratives, treating narratological models of *fabula* and *suzjet* in a more computational fashion than previously attempted, and extending existing techniques and architectures in NLG, this work provides a foundation for a new generation of narrative generators whose prose quality significantly surpasses that of previous efforts.

In summary, it is encouraging that the stories produced by a narrative prose generator could begin to approach the quality of that produced by human authors. However, because this implementation was constructed for the purpose of producing fairy tales, it remains to be seen how well it would perform with different characters and plots, different genres (mystery/detective, science fiction, historical fiction), and styles (for instance, Ernest Hemingway vs. James Joyce). Furthermore, many of the grand promises of natural language generation, such as automatically generating texts in multiple languages and culturally specific texts from a single representation remain unfulfilled.

This work represents a small step toward the larger goal of creating the foundation for expressive narrative generation. As they mature, interactive narrative generators will play an important role in applications ranging from education and journalism to business and entertainment. To make significant progress in this direction, it will be important to leverage increasingly sophisticated models of narrative as well as advances in multimodal dialog systems and multilingual natural language generation. We will be exploring these lines of investigation in our future work.

Acknowledgements

The authors wish to thank Patrick FitzGerald, Robert St. Amant, and Michael Young for their many very helpful discussions and comments on the direction of the work; Michael Elhadad and Jacques Robin for their development of FUF/SURGE; Bruce Porter and Eric Eilerts for their development of the KM knowledge language; the IJCAI, CogSci and AIJ reviewers whose advice significantly improved the organization and presentation of the results; Rob Holte for the idea of abstract search spaces for revision; and Joy Smith for her assistance with the statistical analysis. Support for this work was provided by the IntelliMedia Initiative of North Carolina State University.

Appendix A. A complete narrative produced by STORYBOOK

This three-page example narrative was produced by STORYBOOK, an implementation of the AUTHOR architecture. Every sentence was produced from a fully specified deep structure by STORYBOOK and the FUF surface realizer [19].

STORYBOOK is implemented in HARLEQUIN LISP and runs on a Dell Precision Workstation 410 using a 600 MHz Pentium III processor with 512 MB of memory. The creation of a 3 page narrative using STORYBOOK requires about 1 minute and a half, with the bulk of that time consumed by the surface realizer (during unification of the FDs with the SURGE grammar).

Some notable features include:

[P02]: Indirect dialogue, revised from 2 separate sentences

[P03]: Pronominalizations with multiple competing antecedents

[P06]: Initiation of an extended dialogue sequence

[P10]: Addressee in an utterance (specific to narrative)

[P13]: Dialogue in *interposed* mode (see Section 2.3.1)

[P18]: Manner clause specifying how the utterance was delivered

[P20]: Indirect question addressed to the reader

[P21]: Four sequential discourse markers (see Section 2.3.3)

[P34]: Multiple pronominalizations (see Section 2.3.2)

[**P01**] Once upon a time, there was a woodman and his wife who lived in a pretty cottage on the borders of a great forest. They had one little daughter, a sweet child, who was a favorite with everyone. She was the joy of her mother's heart, and to please her, the good woman made her a little scarlet cloak and hood. She looked so pretty in it that everyone called her Little Red Riding Hood.

[**P02**] One day her mother told Little Red Riding Hood she meant to send her to her grandmother to take her some fresh butter and new-laid eggs and a nice cake.

[**P03**] Her grandmother was a very old lady, who lived in the heart of a neighboring wood. She was delighted at being sent on this errand, for she liked to do kind things. It was a very long time since she had seen her grandmother, and so she had almost forgotten what the old lady looked like.

[**P04**] The sun was shining brightly, but it was not too warm under the shade of the old trees. Little Red Riding Hood went on her way singing and gathering great bunches of wild flowers. She wanted to give them to her grandmother. She sang so sweetly that a cushat dove flew down from a tree and followed her.

[**P05**] Now, it happened that a wolf, a very cruel, greedy creature also heard Little Red Riding Hood as she passed, and he longed to eat her for his breakfast. But he knew Hugh, the woodman, was at work very near with his great dog. He was afraid Hugh and the dog might hear Little Red Riding Hood and she might cry out if he frightened her. He was afraid they might kill him.

[**P06**] So the wolf followed Little Red Riding Hood a little way and came up to her very gently. He said, "Good day, Little Red Riding Hood, where are you going?"

[**P07**] "To see my grandmother," said the child. "To take her a present from mother of eggs and butter and cake."

[**P08**] "Where does your grandmother live?" asked the wolf.

[**P09**] "Quite in the middle of the wood," Little Red Riding Hood replied.

[**P10**] "Oh! I think I know the house. Good-bye, Little Red Riding Hood." And he ran off as fast as he could go.

[**P11**] Little Red Riding Hood was not in a hurry, because there were many things in the wood to amuse her. She ran after the white and yellow butterflies that danced before her and sometimes, caught one. But she always let it go again, for she never liked to hurt any creature.

[**P12**] And then there were the merry, cunning little squirrels to watch. They were cracking nuts on the branches of the old trees. Every now and then, a rabbit would hurry away through the tall ferns. A great bee came buzzing near Little Red Riding Hood, and she would stop to watch it gathering honey from the flowers and the wild thyme. So she went on very slowly. By-and-by she saw Hugh, the woodman.

[**P13**] "Where are you going, Little Red Riding Hood," said the woodman, "all alone?"

[**P14**] "I am going to my grandmama's," said the child. "Good-bye. I must make haste now for it is becoming late."

[**P15**] While Little Red Riding Hood was playing in the wood, the great wolf galloped on as fast as he could to the old lady's house. Grandmother lived all by herself, and once or twice a day a neighbor's child came to tidy it and to get her food.

[**P16**] Now, grandmother was very feeble and she was often obliged to keep her bed. It happened that she was in bed the day she went to see grandmother. When he reached the cottage door, the wolf tapped.

[**P17**] "Who is there?" asked the old lady.

[**P18**] "Little Red Riding Hood, granny," said the wolf, trying to speak like her.

[**P19**] "Come in, my dear," said the old lady, who was a little deaf. "Pull the string and the latch will come up."

[**P20**] The wolf did as grandmother told him and went in. And you may think how frightened poor grandmother was when she saw him instead of Little Red Riding Hood.

[**P21**] Now, the wolf was quite hungry after his run and he soon ate the poor old lady up. Indeed she was not enough for his breakfast, and so he thought he would like to eat sweet Little Red Riding Hood also. Therefore the wolf dressed himself in her night-cap, got into bed and he waited for the child to knock at the door.

[**P22**] By-and-by Little Red Riding Hood reached her grandmother's house and tapped at the door.

[**P23**] "Come in," said the wolf, in a squeaking voice. "Pull the string and the latch will come up."

[**P24**] Little Red Riding Hood thought her grandmother must have a cold, as she spoke so hoarsely. But she went in at once, and there lay her granny in bed, as she thought.

[**P25**] "If you please, granny, mother has sent me with some butter and eggs."

[**P26**] When she saw the wolf, Little Red Riding Hood felt frightened. She had nearly forgotten her grandmother, but she did not think she had been so ugly.

[**P27**] "Grandmother, what a great nose you have," Little Red Riding Hood said.

[**P28**] "All the better to smell with, my dear," said the wolf.

[**P29**] "And, grandmother, what large ears you have."

[**P30**] "All the better to hear with, my dear."

```
[P31]    "Ah! Grandmother, and what large eyes you have."

[P32]    "All the better to see with, my dear," the wolf said, showing his
   teeth, for he longed to eat the child up.

[P33]    "Oh, grandmother, and what great teeth you have!" said Little Red
   Riding Hood.

[P34]    "All the better to eat you up with," growled the wolf.  He jumped
   out of bed, rushed at her and would have eaten her up.  But just at
   that minute the door flew open and a great dog tore him down.  The
   wolf and the dog were still fighting when Hugh, the woodman, came in
   and they were still fighting when he killed the wicked wolf with his
   axe.

[P35]    Little Red Riding Hood threw her arms round the woodman Hugh's
   neck, kissed him and thanked him again and again.  "Oh, you good, kind
   Hugh," she said.  "How did you know the wolf was here, in time to save
   me?"

[P36]    "Well, after you had passed, I remembered that he had been seen
   about the wood lately," said Hugh.  "I thought I would just come after
   you and I thought I would see if you were safe.  When we came near
   your grandmother's house, Trim sniffed and he ran to the door and he
   whined.  He pushed it open, as you had not shut it closed, and he
   rushed in.  I followed him and between us we have killed the wolf."

[P37]    Then Hugh took the child home and her mother and father could not
   thank him enough for saving their Little Red Riding Hood.  She was
   immediately clasped in her delighted mother's arms.
```

References

[1] J.L. Austin, How to Do Things with Words, Oxford University Press, New York, 1962.

[2] M. Bal, Narratology: Introduction to the Theory of Narrative, 2nd Edition, University of Toronto Press, Toronto, ON, 1997.

[3] D.G. Bobrow, T. Winograd, An overview of KRL, a knowledge representation language, Cognitive Sci. 1 (1977) 3–46.

[4] S. Brennan, Lexical entrainment in spontaneous dialog, in: Proc. International Symposium on Spoken Dialogue, Philadelphia, PA, 1996.

[5] C.B. Callaway, Narrative prose generation, Ph.D. Thesis, North Carolina State University, Raleigh, NC, 2000.

[6] C.B. Callaway, A computational feature analysis for multilingual character-to-character dialogue, in: Proc. Second International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, 2001, pp. 251–264.

[7] C.B. Callaway, J.C. Lester, Pronominalization in discourse and dialogue, in: Proc. 40th Meeting of the Association for Computational Linguistics, Philadelphia, PA, 2002.

[8] C.B. Callaway, J.C. Lester, Dynamically improving explanations: A revision-based approach to explanation generation, in: Proc. IJCAI-97, Nagoya, Japan, 1997, pp. 952–958.

[9] C.B. Callaway, J.C. Lester, Evaluating the effects of natural language generation techniques on reader satisfaction, in: Proc. Twenty-Third Annual Conference of the Cognitive Science Society, Edinburgh, Scotland, 2001, pp. 164–169.

[10] C.B. Callaway, J.C. Lester, Narrative prose generation, in: Proc. IJCAI-2001, Seattle, WA, 2001, pp. 1241–1248.
[11] C.B. Callaway, B.H. Daniel, J.C. Lester, Multilingual natural language generation for 3D learning environments, in: Proc. 1999 Argentine Symposium on Artificial Intelligence, Buenos Aires, Argentina, 1999, pp. 177–190.
[12] J. Cassell, M. Stone, H. Yan, Coordination and context-dependence in the generation of embodied conversation, in: Proc. International Natural Language Generation Conference, Mitzpe Ramon, Israel, 2000.
[13] C. Collins, P. Branigan, Quotative inversion, Natural Language Linguistic Theory 15 (1) (1997) 1–41.
[14] R. Dale, Exploring the role of punctuation in the signalling of discourse structure, in: Proc. Workshop on Text Representation and Domain Modelling, T.U. Berlin, Berlin, 1991, pp. 110–120.
[15] H. Dalianis, E. Hovy, Aggregation in natural language generation, in: Proc. Fourth European Workshop on Natural Language Generation, Pisa, Italy, 1993.
[16] E. Davis, Representations of Commonsense Knowledge, Morgan Kaufmann, San Mateo, CA, 1990.
[17] B. Di Eugenio, Centering in Italian, in: M.A. Walker, A.K. Joshi, E.F. Prince (Eds.), Centering in Discourse, Oxford University Press, Oxford, 1998.
[18] C. Doran, Incorporating punctuation into the sentence grammar: A lexicalized tree adjoining grammar perspective, Ph.D. Thesis, University of Pennsylvania, Philadelphia, PA, 1998.
[19] M. Elhadad, Using argumentation to control lexical choice: A functional unification implementation, Ph.D. Thesis, Columbia University, New York, 1992.
[20] M. Elhadad, K. McKeown, Generating connectives, in: Proc. Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, 1990, pp. 97–101.
[21] M. Elhadad, K. McKeown, J. Robin, Floating constraints in lexical choice, Comput. Linguistics 23 (2) (1997) 195–240.
[22] A. Garnham, What's wrong with story grammars?, Cognition 15 (1983) 145–154.
[23] B.J. Grosz, The representation and use of focus in a system for understanding dialogs, in: Proc. IJCAI-77, Cambridge, MA, 1977, pp. 67–76.
[24] B. Grote, Representing temporal discourse markers for generation purposes, in: Proc. Discourse Relations and Discourse Markers Workshop, Montréal, QB, 1998, pp. 22–28.
[25] B. Grote, M. Stede, Discourse marker choice in sentence planning, in: Proc. 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, ON, 1998, pp. 128–137.
[26] R. Henschel, H. Cheng, M. Poesio, Pronominalization revisited, in: Proc. 18th International Conference on Computational Linguistics, Saarbrücken, Germany, 2000.
[27] H. Horacek, A model for adapting explanations to the user's likely inferences, User Modeling and User-Adapted Interaction 7 (1) (1997) 1–55.
[28] M.L. Hoover, Effects of textual and cohesive structure on discourse processing, Discourse Processes 23 (1997) 193–220.
[29] E.H. Hovy, Automated discourse generation using discourse structure relations, Artificial Intelligence 63 (1993) 341–385.
[30] E. Hovy, Y. Arens, Automatic generation of formatted text, in: Proc. AAAI-91, Anaheim, CA, 1991.
[31] B. Jones, What's the point? A (computational) theory of punctuation, Ph.D. Thesis, The University of Edinburgh, Scotland, 1996.
[32] M. Kantrowitz, J. Bates, Integrated natural language generation systems, in: R. Dale, E. Hovy, D. Rosner, O. Stock (Eds.), Aspects of Automated Natural Language Generation, Springer, Berlin, 1992, pp. 247–262.
[33] R. Kibble, R. Power, An integrated framework for text planning and pronominalisation, in: Proc. First International Conference on Natural Language Generation, Mitzpe Ramon, Israel, 2000, pp. 194–200.
[34] W. Kintsch, J.M. Keenan, Reading rate and retention as a function of the number of propositions in the base structure of sentences, Cognitive Psychology 5 (1973) 257–274.
[35] A. Knott, C. Mellish, A data-driven method for classifying connective phrases, J. Language Speech 39 (1996).
[36] R.R. Lang, A formal model for simple narratives, Ph.D. Thesis, Tulane University, New Orleans, LA, 1997.
[37] M. Lebowitz, Story-telling as planning and learning, Poetics 14 (3) (1985) 483–502.
[38] G.N. Leech, M.H. Short, Style in Fiction: A Linguistic Introduction to English Fictional Prose, Longman, London, 1981.
[39] T.M. Leitch, What Stories Are, The Pennsylvania State University Press, University Park, PA, 1986.

[40] J.C. Lester, B.W. Porter, Developing and empirically evaluating robust explanation generators: The KNIGHT experiments, Comput. Linguistics 23 (1) (1997) 65–101.
[41] R.E. Longacre, The Grammar of Discourse, Plenum Press, New York, 1983.
[42] K.F. McCoy, M. Strube, Taking time to structure discourse: Pronoun generation beyond accessibility, in: Proc. Twenty-First Conference of the Cognitive Science Society, Vancouver, Canada, 1999, pp. 378–383.
[43] R. McKee, Story, ReganBooks, New York, 1997.
[44] J. Meehan, TALE-SPIN—an interactive program that writes stories, in: Proc. IJCAI-77, Cambridge, MA, 1977.
[45] V. Mittal, J. Moore, G. Carenini, S. Roth, Describing complex charts in natural language: A caption generation system, Comput. Linguistics 24 (3) (1998) 431–469.
[46] J.D. Moore, Participating in Explanatory Dialogues, MIT Press, Cambridge, MA, 1995.
[47] E. Not, M. Zancanaro, Exploiting the discourse structure for anaphora generation, in: Proc. DAARC'96—Discourse Anaphora and Anaphor Resolution Colloquium, Lancaster University, Lancaster, UK, 1996.
[48] G. Nunberg, The Linguistics of Punctuation, Center for the Study of Language Information, Stanford, CA, 1990.
[49] B. Porter, J. Lester, K. Murray, K. Pittman, A. Souther, L. Acker, T. Jones, AI research in the context of a multifunctional knowledge base: The botany knowledge base project, Technical Report #AI88-88, University of Texas at Austin, Austin, TX, 1988.
[50] R. Power, C. Doran, D. Scott, Generating embedded discourse markers from rhetorical structure, in: Proc. Seventh European Workshop on Natural Language Generation, Toulouse, France, 1999.
[51] V. Propp, Morphology of the Folktale, University of Texas Press, Austin, TX, 1968.
[52] R. Quirk, S. Greenbaum, G. Leech, J. Svartvik, A Comprehensive Grammar of the English Language, Longman, London, 1985.
[53] E. Reiter, Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible?, in: Proc. Seventh International Workshop on Natural Language Generation, Kennebunkport, ME, 1994, pp. 163–170.
[54] J. Robin, Revision-based generation of natural language summaries providing historical background, Ph.D. Thesis, Columbia University, New York, 1994.
[55] J. Searle, Speech Acts, Cambridge University Press, Cambridge, 1969.
[56] C. Segre, Introduction to the Analysis of the Literary Text, Indiana University Press, Bloomington, IN, 1988.
[57] J. Shaw, Segregatory coordination and ellipsis in text generation, in: COLING-ACL-98: Proc. Joint 36th Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, Montréal, QB, 1998, pp. 1220–1226.
[58] C.L. Sidner, Focusing in the comprehension of definite anaphora, in: M. Brady, R. Berwick (Eds.), Computational Models of Discourse, MIT Press, Cambridge, MA, 1983, pp. 267–330.
[59] F.K. Stanzel, A Theory of Narrative, Cambridge University Press, Cambridge, 1979.
[60] M. Stede, Lexical semantics and knowledge representation in multilingual sentence generation, Ph.D. Thesis, University of Toronto, Toronto, ON, 1996.
[61] P.G. Thomson, Little Red Riding Hood, Little Cavalier Series, Cincinnati, OH, 1884.
[62] D. Traum, A computational theory of grounding in natural language generation, Ph.D. Thesis, University of Rochester, Rochester, NY, 1994.
[63] S.R. Turner, The Creative Process: A Computer Model of Storytelling and Creativity, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
[64] M.A. Walker, Centering, anaphora resolution, and discourse structure, in: M.A. Walker, A.K. Joshi, E.F. Prince (Eds.), Centering in Discourse, Oxford University Press, Oxford, 1998.
[65] B. Webber, A Formal Approach to Discourse Anaphora, Garland, NY, 1979.
[66] P. Weyhrauch, Guiding interactive drama, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.
[67] C.-L. Yeh, C. Mellish, An empirical study on the generation of anaphora in chinese, Comput. Linguistics 23 (1) (1997) 169–190.
[68] R.M. Young, Using plan reasoning in the generation of plan descriptions, in: Proc. AAAI-96, Portland, OR, 1996, pp. 1075–1080.
[69] R.M. Young, Notes on the use of plan structures in the creation of interactive plot, in: Workshop Notes of the AAAI-99 Fall Symposium on Narrative Intelligence, Cape Cod, MA, 1999, pp. 164–167.