Narrative Prose Generation

Charles B. Callaway
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206 USA
cbcallaw@eos.ncsu.edu

James C. Lester
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206 USA
lester@csc.ncsu.edu

Abstract

Story generation is experiencing a revival, despite disappointing preliminary results from the preceding three decades. One of the principle reasons for previous inadequacies was the low level of writing quality, which resulted from the excessive focus of story grammars on plot design. Although these systems leveraged narrative theory via corpora analyses, they failed to thoroughly extend those analyses to all relevant linguistic levels. The end result was narratives that were recognizable as stories, but whose prose quality was unsatisfactory.

However, the blame for poor writing quality cannot be laid squarely at the feet of story grammars, as natural language generation has to-date not fielded systems capable of faithfully reproducing either the variety or complexity of naturally occurring stories. This paper presents the AUTHOR architecture for accomplishing precisely that task, the STORYBOOK implementation of a narrative prose generator, and a brief description of a formal evaluation of the stories it produces.

1 Introduction

Despite extensive research in the fields of story generation and natural language generation, collaborative research between the two has been virtually nonexistent. A major reason for this is the difficult nature of the problems encountered respectively in these fields. Story generators [Meehan, 1977; Yazdani, 1982; Lebowitz, 1985; Turner, 1994; Lang, 1997], 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 by progressively adding more and more detail. 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 produce multi-paragraph text. What remains is a substantial gap between the narrative plans produced by story generators and the requirements of NLG systems.

This is explained by the historic research programs of these two distinct fields. Story generation originally descends from the application of planning formalisms to the work of sociolinguists such as Vladimir Propp [Propp, 1968], who created story grammars to capture the high-level plot elements found in Russian folktales. Early work (Figure 1) in this area [Meehan, 1977; Yazdani, 1982; Lebowitz, 1985] focuses on the creation of characters and their interactions with plot elements. In the latest of these, Lebowitz states, “Eventually, we expect UNIVERSE to be able to generate connected stories in natural language form over a long period of time. For the moment, we are concentrating on generating plot outlines, and leaving problems of dialogue and other low-level text generation for later.” Moreover, even the most recent story generation systems, such as MINSTREL and JOSEPH [Turner, 1994; Lang, 1997], focus on characters and plot when generating text, without considering the actual linguistic structures found in the texts they are attempting to mimic (Figure 2).

However, the lack of progress in achieving computer-produced stories characterized by high-quality prose is far from one-sided. Rather than *narrative generation*, most full-scale NLG systems [Hovy, 1993; Young, 1996; Horacek, 1997; Lester and Porter, 1997; Mittal *et al.*, 1998; Callaway *et al.*, 1999] instead focus on *explanation generation*, creating scientific or instructional text which 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.*, [Kantrowitz and Bates, 1992; Robin, 1994; Doran, 1998; Cassell *et al.*, 2000]). In addition, the most advanced of these 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. Finally, none of these systems are intended to accept narrative plans from a typical story generator.

To bridge the gap between story generators and NLG systems, we have developed the AUTHOR narrative prose generation architecture [Callaway, 2000] to create high-quality narrative prose comparable to, and in some cases identical to, that routinely produced by human authors. This architecture has been implemented in STORYBOOK, an end-to-end narrative prose generation system that utilizes narrative planning, sentence planning, a discourse history, lexical choice, revision, a full-scale lexicon, and the well-known FUF/SURGE [Elhadad, 1992] surface realizer to produce multi-page stories in the Little Red Riding Hood fairy tale domain.

```
     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.
```

Figure 1: Prose generated by Tale-Spin, 1977

```
  one day it happened that peasant
quarreled with the wife. when this
happened, peasant felt distress. in
response, peasant took a walk in the
woods. peasant found a pit when he
looked under the bush. when this
happened, peasant desired to punish
wife. in response, peasant made it
his goal that wife would be in the
pit.  peasant tricked wife. wife was
in the pit. peasant lived alone.
```

Figure 2: Prose generated by Joseph, 1997

Narrative prose differs linguistically from text found in explanatory and instructional passages in a number of ways:

- The existence of character dialogue with the accompanying difficulties of orthographic markers [Doran, 1998; Callaway, 2001], speaker-hearer relationships, locutional relations and manner clauses, interjections, and changes in pronominalization patterns. For instance, the following would never be found in explanatory text: "Beware the wolves," her mother said in a hushed voice.
- Since explanatory text lacks dramatic characters, there is little need to include personal pronouns, highly idiomatic text about personal needs, or intentional desires such as wanting, needing, or knowing.
- Without character dialogue, explanatory text is usually able to get by using only present verb tenses with an occasional reference to events in the past when discussing sequences of processes. However, dialogue and the complex interactions between characters opens up the need to perform at least simplistic temporal reasoning and realizations in complex present, future and past tenses.
- Because human authors employ widely differing styles in narrative (*e.g.*, Hemingway vs. Joyce) as opposed to explanatory or instructional text which tries to adhere to stricter conventions, a narrative prose generator should be capable of mimicking those different types of styles.
- Finally, a narrative prose generator must conform to common prose formatting conventions, such as knowing when to force paragraph breaks and being able to generate written stylistic effects like onomatopoeia, regional dialects, and emphasis (*e.g.*, "Ewwww!" "B-b-but, it's s-s-scary!" "Mom, you CAN'T do that!")

StoryBook is capable of reproducing these phenomena, and doing so in both grammatically correct English and passable Spanish [Callaway *et al.*, 1999; Callaway, 2001].

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 and 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 the narratives that StoryBook produces, we created a simplified narrative planner capable of generating two Little Red Riding Hood stories expressed in the required high-level story specification. We then created five versions of StoryBook variously ablating the discourse history, lexical choice, and revision components to produce a total of 10 story versions which were then formally evaluated by a panel of judges. The results showed significant differences between the inclusion or ablation of individual architectural components.

2 Narrative Representation

While most researchers in story generation utilize planning mechanisms or story grammars, a growing literature on narratology [Propp, 1968; Segre, 1988; Bal, 1997] posits that narrative consists of the *fabula*, or sum total of knowledge and facts about a narrative world, and the *suzjet*, or the ordering and specifics about what the author presents and at which position(s) it occurs in the linear narrative. The Author architecture adopts this view and computationalizes it to describe the requirements of a narrative planner and a narrative prose generator: the narrative planner is responsible for creating both the fabula and suzjet, while the narrative prose generator is responsible for converting them into textually recognizable narratives.

A narrative world is also populated with a large number of scenes, characters, props, locations, events, and descriptions. The StoryBook implementation explicitly represents this knowledge, which forms the basis of the fabula. Initially, the fabula contains only ontological information, including the existence of broad concepts such as *forest*, *cottage*, and *person*, and concept relations like *next-to*, *mother-of*, and *moves-toward*. StoryBook assumes that a narrative planner is responsible for constructing the specific concept instances that populate a particular story, *e.g.*, Little Red Riding Hood lives in `Cottage001`, which is her house, while her grandmother (`Grandmother001`) lives in a different house, `Cottage002`.

```
;;; Fabula Operators
(NewNarrative Meehan-Narrative000 Narrator001)
(AddActor George-Ant003 George-Ant Ant Male "George Ant")
(AddActor Wilma-Bird004 Wilma-Bird Bird
     Female "Wilma Bird")
(AddLocation Patch005 Patch-Area)
(AddLocation Ground006 Ground-Earth-Area)
(AddLocation Nest007 Nest-For-Birds)
(AddLocation Ash-Tree008 Ash-Tree)
(AddProp Water009 Water)
(AddLocation River010 River)
(AddLocation Meadow011 Meadow)
(AddLocation Valley012 Valley)
(AddAlias Wilma013 Wilma Wilma-Bird004 "Wilma")
(AddAlias George014 George George-Ant003 "George")

;;; Narrative Stream Primitives
(narration-mode historical-fairy-tale mixed-dialogue
     simple-syntax ascii-format narrated english)
(narrator-mode narrator001 third-person disembodied)
(prop-relationship living-near george-ant003 patch005)
(refinement region-of patch005 ground006)
(specification living-near process-step-type
     once-upon-a-time)
(prop-property exist-being nest007)
(specification exist-being location-in ash-tree008)
(prop-relationship living-in wilma-bird004 nest007)
(prop-property exist-being water009)
(specification exist-being location-in river010)
(refinement quantifier-value water009 some)
(define-event being-in015 being-in water009 river010)
(actor-intent knowing wilma013 being-in015)
(specification knowing thought-binder that-binder)
(define-event being-in016 being-in water009 river010)
(actor-intent knowing george014 being-in016)
(specification knowing thought-binder that-binder)
(actor-property personal-condition wilma013 thirsty-state)
(specification personal-condition time one-day)
(refinement intensifier thirsty-state very)
(define-event getting-near017 getting-near none water009)
(actor-intent wanting wilma013 getting-near017)
(refinement quantifier-value water009 some)
(actor-action flying-from wilma013 nest007)
(refinement belonging-to nest007 wilma013)
(specification flying-from across-path meadow011)
(specification flying-from through-path valley012)
(specification flying-from destination river010)
(actor-action drinking-action wilma013 water009)
(actor-property personal-condition wilma013 thirsty-state)
(specification personal-condition duration any-more)
(specification personal-condition polarity negative)
(refinement intensifier thirsty-state very)
```

Figure 3: Fabula and Narrative Stream for generating Fig. 1

In addition, STORYBOOK assumes that the narrative planner is responsible for creating a stream of narrative events (the suzjet) that defines the linear ordering of events and descriptions as well as for *content determination*, the NLG term for deciding if particular narrative details or underlying facts are ever mentioned at all (*e.g.*, events can be "too obvious" or perhaps meant to be inferred, as in mystery novels). Also, linearity can vary between different versions of a single story: a strict chronological ordering that states Little Red Riding Hood meets a wolf before travelling to grandmother's house wouldn't necessarily hold in the *in medias res* version.

In order to computationalize the fabula and narrative stream so they can serve as an interface between a narrative planner and a narrative prose generator, the AUTHOR architecture defines a set of *fabula operators* which can be used to construct the fabula from the original story ontology, and a set of *narrative stream primitives* (Figure 3), which define the presentational order and content determination as well as information about what purpose that particular content serves at that point in the narrative.

A typical fabula operator relates a new *concept instance* (indicated by a unique number at the end of the name) to either some element in the story ontology or a previously created concept instance. A typical narrative stream primitive consists of a *narrative directive*, which describes the purpose for its inclusion by the narrative planner as well as the relationship between its arguments. The ordered arguments of a narrative directive are directly tied to either the original concepts in the story ontology or the derived concept instances created by the fabula operators. Furthermore, a partial order is imposed on the narrative stream forcing dependent elements to follow their modifiers (*e.g.* in the phrase "some water" from Figure 3, "water" is introduced in a narrative primitive before the "some" quantifier is introduced).

STORYBOOK currently defines six different fabula operators as well as 35 narrative stream primitives that serve three main functions: *delimitation* (narrator and scene changes, dialogue marking, and formatting directives), *foundation* (important clause-level events and descriptions, rhetorical, intentional, and perlocutionary relations loosely based on Speech Act Theory [Austin, 1962; Searle, 1969]), and *modification* (descriptive elaboration, comparison, manner, reason, time, *etc.*) These have been sufficient to encode three distinctly different multi-page Little Red Riding Hood fairy tales and to allow STORYBOOK's narrative prose generator to create the narrative texts for each.

Finally, STORYBOOK assumes that the fabula and narrative stream operate in an *interlingual* environment, where the knowledge base encodes world knowledge in a language-neutral format [Callaway *et al.*, 1999; Callaway, 2001]. Thus given a single fabula and narrative stream, we should be able to produce fairy tales (or other forms of fictional narratives) in a variety of languages. A significant benefit of this approach is that such a narrative prose generator could also be used to improve the output of a machine translation system in a manner analogous to that of story generation. Regardless of how they are determined, the fabula and narrative stream are sent along with a set of stylistic parameters to the narrative prose generator as described in the following section.

3 The AUTHOR Architecture

To reproduce the complex phenomena that characterize human-generated stories, an effective narrative prose generator must be comprehensive in scope. It must address all of the requirements inherent in sentence planning, lexical choice, formatting, revising, and surface realization. AUTHOR therefore takes a standard "pipelined" approach with components for each of these processes. With the exception of discourse planning, which is here replaced by a narrative planner, STORYBOOK is the first NLG system to incorporate all of these modules into an end-to-end multi-page generation system.

Upon receiving the fabula and narrative stream from the narrative planner, STORYBOOK (Figure 4) first structures it into paragraph and sentence-sized chunks. It then conducts a discourse history analysis to determine pronominalizations and identify seen/unseen concepts. Next, it performs a lexical choice analysis to increase variety. It then maps actors, props and events to semantic/syntactic roles in full linguistic deep structures. Next it revises paragraph-sized groups of sentences via aggregation and reordering to increase propositional density before finally performing surface realization to produce narrative prose similar to that found in stories written by human authors.

3.1 Narrative Organization

Because the narrative stream (Figure 3) is generated by the narrative planner as one long sequence, it must be segmented into groups of narrative stream primitives which reflect natural boundaries such as changes in speaker during dialogue and shifts in scene or topic. Because of the partial order imposed on the narrative stream, this is a relatively straightforward process. In addition, the discourse history and lexical choice modules operate by combing through the narrative stream and recording data in order to make decisions about altering the narrative stream. Because these three procedures involve a similar iterative analysis, they are performed by a single architectural module called the *narrative organizer* whose purpose is to take a flat, linear narrative stream and impose a hierarchical narrative structure onto it.

After the narrative stream primitives have been segmented, the discourse history module opportunistically replaces concept instances with the appropriate definite/indefinite forms and pronominalizations. These features are used to decide when to replace the lexicalizations of concepts and concept instances with the appropriate new linguistic deep structure information.[1] 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 an indefinite determiner or determiner sequence for the mass noun ("some butter"). Knowing whether a concept instance has been seen or not requires computing and tracking several *occurrence* properties for every concept instance:

- *Frequency*: How often a concept instance has been used (lexicalized vs. pronominalized).
- *Last-Usage*: If its most recent use was lexicalized.
- *Recency*: How many distinct concept instances have been used since it was last seen (including gender).
- *Distance*: The number of scenes, paragraphs, or dialogue turns since it was last seen.

Similarly, pronominalization decisions are made to replace repetitive instances of concept instances with appropriate pronouns (*e.g.*, "Grandmother" with the single feminine pronoun "she/her"). Because knowing when an instance is repetitive involves using the same occurrence properties, the lexical chooser similarly checks for excessive repetition of concept instances or relations. If this happens, the lexical chooser may replace elements of each narrative primitive with synonymous concept instances or relations from the fabula. The StoryBook lexical choice module detects:

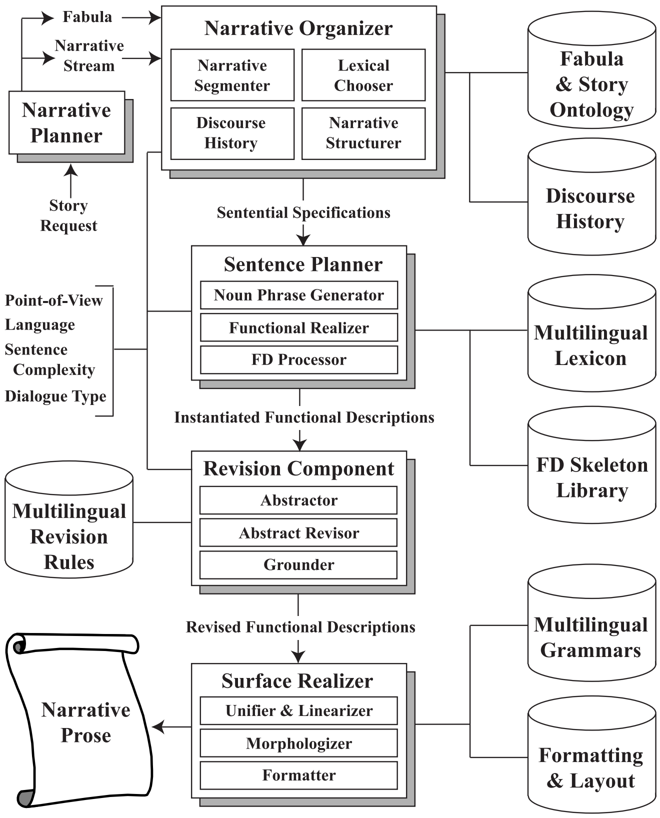

Figure 4: A Narrative Prose Generation Architecture

- *Repetition in Noun Phrases*: Languages typically contain a large number of similar nouns. For example, Little Red Riding Hood might live in a house, cottage, shack, hut, cabin, *etc.*
- *Repetition in Verb Phrases*: Similarly, events have a number of lexicalizations. Little Red Riding Hood can walk through the forest, skip, amble, stroll, *etc.*
- *Repetition in Thematic Role Ordering*: Many event verbs also impose different theta frames even though they describe similar actions. Little Red Riding Hood might give her grandmother the cookies or grandmother might receive the cookies from Little Red Riding Hood.

Although this does not compare to more sophisticated methods [Elhadad, 1992; Stede, 1996] and is by no means suggested as a solution to the problem of lexical choice, it is sufficient to satisfy our goal of preventing repetitive prose. The result of segmentation, discourse history analysis, and lexical choice is thus a modified narrative stream. However, in classic pipelined NLG architectures, discourse planners typically produce a single structure (*e.g.*, a frame) that corresponds to a sentence-sized chunk of the discourse plan. Thus, the job of the *narrative structurer* is to convert the groups of

[1] These "replacements" are more accurately "annotations" on the narrative stream, because future changes by the revision component may alter the circumstances that lead to a particular noun phrase being pronominalized. For instance, the revision component may swap the order of two sentences or change the position of a circumstantial clause from leading a sentence to following it.

narrative primitives into a sequence of specifications suitable for the sentence planner. Additionally, the narrative structurer is responsible for making decisions about tense shifting, especially for character dialogue. In dialogue, conversations usually take place in present tense even though the surrounding narrative is in past tense, and references to prior events are typically in the past tense where in expository text they would be in the past perfect tense (at least for English).

Because the fabula and story ontology exist by these stages, they can be used as knowledge sources for making appropriate decisions. For example, the discourse history module may examine the gender of a concept instance and thus know it should substitute “she” for “Little Red Riding Hood” without that knowledge having to be explicitly represented in the narrative stream. Similarly, the narrative structurer may examine the lexicon entry of a narrative stream primitive’s primary relation to determine its theta frame and its argument’s semantic type restrictions for error-checking purposes.

3.2 Sentence Planning

The function of the sentence planner is to take a specification for the semantic content of a sentence (or *protosentence*) and to plan the roles (either semantic or syntactic) that each of its elements play in a particular sentence. Because our approach utilizes an off-the-shelf surface realizer that expects particular semantic roles, we require that our sentence planner produce the deep structure linguistic representations known as *functional descriptions* (FDs, Figure 5). Functional descriptions are hybrid semantic/syntactic entities that can be used to produce text via unification with the FUF/SURGE [Elhadad, 1992] surface realizer.

A sentence planner must:

- Guarantee that complex content units are properly and completely packaged within functional descriptions, *e.g.*, complex noun phrases such as “the beautiful cottage where they lived” must be (a) capable of being created as a linguistic deep structure and (b) encapsulated so that it can be manipulated as a whole by succeeding elements of the pipelined NLG architecture.
- Assign thematic roles to concepts. To achieve semantic equivalence between a sentence’s frame specification and the corresponding deep structure, a sentence planner must ensure that relations in the specification are precisely mapped to the appropriate thematic roles in a functional description, *e.g.*, `mother001` could be mapped to `agent` and `nest007` to `located`.
- Robustly construct functional descriptions. A sentence planner must ensure that only FDs that will create grammatical sentences can be constructed. A number of errors that degrade robustness must be curbed, *e.g.*, lack of appropriate lexicalizations, missing semantic roles, and sentential modifiers that conflict with the overall sentential semantics.

Once the sequence of narrative stream primitives has been processed by the sentence planner, the resulting FDs (representing the deep linguistic structures for each protosentence) can be given directly to the surface realizer for text generation. However, because the quality of simple propositional sentences is notoriously poor, StoryBook revises them, iteratively saving each FD while maintaining the paragraph separations imposed by the narrative segmenter and proceeds to send paragraph-sized batches to the revision component (described in the following section) in order to improve overall prose quality.

```
((cat clause)
 (tense past)
 (process ((type existential)))
 (participants ((located ((cat common)
                         (definite no)
                         (lex "nest")))))
 (pred-modif ((location ((cat pp)
                        (prep ((lex "in")))
                        (np ((cat common)
                             (definite no)
                             (lex "ash tree"))))))))
```

Figure 5: Functional Description (FD) for “There was a nest in an ash tree.” from Figure 1, Sentence 2.

3.3 Revision

Revision modules [Dalianis and Hovy, 1993; Robin, 1994; Callaway and Lester, 1997; Shaw, 1998] take a series of protosentences (simple sentences with limited content, *e.g.*, “The wolf saw Little Red Riding Hood”) and rearrange them by *aggregation*, *i.e.* combining protosentences in various ways, or by *migration*, *i.e.* permuting the order of two adjacent protosentences. The REVISOR component [Callaway and Lester, 1997] receives a paragraph-sized group of protosentences from the sentence planner represented as an ordered set of deep-structure functional descriptions.

To illustrate, consider the issue of clause aggregation, a central problem in multi-sentential text generation. Suppose a narrative prose generation system is given the task of constructing a fairy tale and it produces several pages of prose. Although it might accurately communicate the content of the narrative plan, the terseness of each sentence makes the overall effect disjointed; in other words, content without literary form. An entire story comprised solely of protosentences is intolerable for almost any adult reader. (See sample prose in Figures 1 and 2.)

To avoid producing a series of abrupt sentences, a narrative planner could be assigned the task of predicting how particular concepts will be realized in order to optimize clause aggregation and reordering decisions. However, this approach violates modularity considerations and does not scale well: it significantly complicates the design of the narrative planner by forcing it to attend simultaneously to content selection, narrative organization, and complex syntactic issues. Alternatively, the propositions could be grouped by a single-pass realization system. This approach is quite inefficient and also ineffective. Reorganizing, aggregating, and realizing the specifications in a single pass poses innumerable difficulties: the realizer would somehow have to anticipate the cumulative effects of all aggregation decisions with regard to grammaticality, subordination, and lexical choice.

An important aspect of revision in NLG is the concept of *discourse constraints*, which specify a partial order on the sequence of functional descriptions. For example, the narrative planner might hand down a narrative constraint stating that, in a particular narrative passage, a sequence of events are causal in nature and that to reorder them in some fashion could destroy that causality in the mind of the reader. Additionally, because narrative prose includes character dialogue, it is important to prevent the reordering of character utterances. Thus, discourse constraints are employed to restrict aggregation and migration revisions that would affect particular types of clause elements across critical semantic boundaries. STORYBOOK utilizes the multilingual version of the REVISOR component described in [Callaway and Lester, 1997] to perform all of these tasks.

3.4 Surface Realization

The revision component passes the series of revised functional descriptions one by one to the surface realizer, which is responsible for producing the actual readable text that readers see. STORYBOOK employs the FUF surface realizer, which is accompanied by the SURGE (Systemic Reusable Grammar of English) grammar. SURGE, written as a *systemic grammar* [Halliday, 1976] in the FUF formalism, is the largest generation grammar in existence in terms of coverage, containing large portions of Quirk's Comprehensive Grammar of English [Quirk *et al.*, 1985] in an HPSG [Pollard and Sag, 1994] interpretation.

Modifications were made to SURGE to allow for dialogue orthography [Callaway, 2001], integrated formatting to produce LATEX, HTML, and XML, as well as a number of grammatical additions to account for syntactic constructions encountered during our corpus analyses (*i.e.*, linguistic phenomena we encountered in narratives that were not present in our analyses of explanatory and instructional text). This allows STORYBOOK to produce webpages as output that include pre-generated graphics specified in the narrative stream as well as boldface, italics, and font size embedded into individual sentences. Furthermore, we implemented a Spanish version of SURGE as described in [Callaway *et al.*, 1999] and also augmented it to produce character dialogue, *etc.*

4 Implementation and Evaluation

STORYBOOK is an end-to-end generation system capable of producing multi-page narrative prose in the Little Red Riding Hood domain like that found in Figure 6, which required 74 fabula operators and 253 narrative stream primitives to generate. STORYBOOK is implemented in HARLEQUIN LISP on a Dell Precision Workstation 410 using a 600 MHz Pentium III processor with 512 MB of memory. The initial story ontology consists of approximately 500 concepts and 300 relations (including their lexicon entries) covering three different Little Red Riding Hood narratives.

STORYBOOK consists of approximately 10,000 lines of Lisp (for narrative organization, sentence planning, and revision, but not surface realization). In addition, there are approximately 30 revision rules which are presently being modified to work for Spanish. The Spanish version of SURGE is approximately the same size as the English version. During story writing, surface realization is by far the largest consumer of time, usually requiring 90% of the 45–90 seconds needed to generate a two to three page narrative.

```
     Once upon a time, a woodcutter and his wife
lived in a small cottage.  The woodcutter and his
wife had a young daughter, whom everyone called
Little Red Riding Hood.  She was a merry little
maid, and all day long she went singing about the
house.  Her mother loved her very much.

     One day her mother said, "My child, go to
grandmother's house. We have not heard from her
for some time.  Take these cakes, but do not stay
too long.  And, beware the dangers in the forest."

     Little Red Riding Hood was delighted because
she was very fond of her grandmother.  Her mother
gave her a well-filled basket and kissed her
goodbye.

     The road to grandmother's house led through
the dark forest, but Little Red Riding Hood was
not afraid and she went on as happy as a lark.
The birds sang her their sweetest songs while the
squirrels ran up and down the tall trees.  Now
and then, a rabbit would cross her path.

     Little Red Riding Hood had not gone far when
she met a wolf.

     "Hello," greeted the wolf, who was a
cunning-looking creature.  "Where are you going?"

     "I am going to my grandmother's house,"
Little Red Riding Hood replied.

     "Ah, well then, take care in the forest, for
there are many dangers."  And then the wolf left.

     Little Red Riding Hood was not in a hurry.
Indeed, she gathered wild flowers and chased the
pretty butterflies.

     Meanwhile the wolf ran ahead very quickly
and soon arrived at grandmother's house.  He
knocked on the door gently.  The old lady asked,
"Who is there?"

     The wolf replied, "It is Little Red Riding
Hood, your granddaughter."

     And so the old lady opened the cottage door.
The wolf rushed in immediately and devoured the
lady in one bite.  Then he shut the door and
climbed into the old lady's bed.

     Much later Little Red Riding Hood arrived at
grandmother's house. She knocked on the door and
shouted, "Grandmother, it is Little Red Riding
Hood."

     "Pull the string.  The door will open."

     And so Little Red Riding Hood opened the
door and walked in.  "Grandmother, what big eyes
you have."

     "All the better to see with, dear."

     "Grandmother, what big ears you have."

     "All the better to hear with, dear."

     "And, grandmother, what big teeth you have!"

     "All the better to eat up with!" yelled the
wolf.

     And then the wolf jumped up and devoured
Little Red Riding Hood in one bite.
```

Figure 6: Example text produced by STORYBOOK

Previous story generation projects such as TALE-SPIN [Meehan, 1977], UNIVERSE [Lebowitz, 1985], and JOSEPH [Lang, 1997], which actually generated narrative prose were never subjected to an empirical evaluation to determine qualitatively or quantitatively how well their systems produced narratives. Also, narrative prose generation is currently at such an early stage of development that its evaluation should be conducted in a manner that is qualitatively different from work in more mature areas such as machine learning.

In order to assess the utility and overall contributions of our deep generation architecture to the task of narrative prose generation, we conducted a formal evaluation of the STORYBOOK system. Its purpose was to establish a baseline for future NLG systems by judging the performance of three key architectural components that differentiate shallow NLG systems from deep NLG systems: the discourse history module, lexical choice module, and revision module.

Formally comparing human-authored narratives with those produced by computer presents a difficult problem: there is no known objective metric for quantitatively evaluating narrative prose in terms of how it performs *as a story*. Simple metrics exist for evaluation at the sentence level (*e.g.*, number of words, depth of embedding, *etc.*), but a narrative *per se* cannot be considered to be merely a collection of sentences that are not related to each other. We instead opted for a computer *vs.* computer style of evaluation involving the ablation of the three architectural components mentioned above.

To stand in for the narrative planner (which is beyond the scope of this work), we created a modestly sized finite state automaton (containing approximately 200 states) capable of producing two stories, comprising two and three pages respectively. Furthermore, we fixed the content of those stories (*i.e.*, the fabula and narrative stream were identical) and ran five different versions of STORYBOOK on each story: (1) all three architectural components working, (2) revision turned off, (3) lexical choice turned off, (4) the discourse history turned off, and finally (5) a version with all three components turned off. This resulted in ten total narratives which we presented to our test subjects. While the two versions differed in the sense that particular modules were either ablated or not, the two stories differed because they were created from two separate paths through the planning automaton. Thus, Story #1 had some different events, descriptions, and props than Story #2 did.

Twenty test subjects graded each narrative over nine grading factors (representing various stylistic and linguistic criteria) according to an A–F scale. We then converted the results to a quantified scale where A = 4.0, B = 3.0, C = 2.0, D = 1.0, and F = 0.0 and tallied and averaged the final scores. 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) over the following grading factors:

- *Overall*: How is the story as an archetypal fairy tale?
- *Style*: Did the author use an appropriate writing style?
- *Grammaticality*: How would you rate the syntactic quality?
- *Flow*: Did the sentences flow from one to the next?
- *Diction*: How appropriate were the author's word choices?
- *Readability*: How hard was it to read the prose?
- *Logicality*: Did the story seem out of order?
- *Detail*: Did the story have the right amount of detail, or too much or too little?
- *Believability*: Did the story's characters behave as you would expect?

The results of the ANOVA analysis point to three significant classes of narratives due to the architectural design of the narrative prose generator. The most preferred narrative class, consisting of Versions (1) and (3), were not significantly different from each other while they were rated significantly higher than all other versions. In addition, Version (2) scored significantly better than the third class formed by Versions (4) and (5), each of which lacked a discourse history.

The results indicate that discourse history and revision components are extremely important, while lexical choice improved text significantly in Story #1 but not in Story #2. Upon analysis of the comments in their evaluations, it became clear that a 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 (1) compared less favorably to Version (3) 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 a more adult audience was expected or if a larger-scale component were utilized.

5 Conclusions

Full-scale linguistic approaches to narrative prose generation can bring about significant improvements in the quality of text produced by story generators. By integrating off-the-shelf NLG components and adding a well-defined computational model of narrative, we can create a new generation of story systems whose written prose quality far surpasses that of previous attempts. This approach has been implemented in STORYBOOK, a narrative prose generator that produces multi-page fairy tales in near realtime. This deep structure approach has also been formally evaluated, suggesting that the architectural modules responsible for a significant improvement in prose quality are components not found in shallow (template-based) generation systems currently employed by most story generators. It is hoped that this work begins to bridge the traditional gap between story generators and NLG systems.

6 Acknowledgements

The authors wish to thank Joy Smith of NC State University for her help with the statistical analysis; Bruce Porter and Eric Eilerts for their development of the KM knowledge base language; Michael Elhadad and Jacques Robin for their development of FUF/SURGE; Michael Young for helpful discussion on narratives; the anonymous reviewers for their insightful comments. Support for this work was provided by the IntelliMedia Initiative of North Carolina State University.

References

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

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

[Callaway and Lester, 1997] Charles B. Callaway and James C. Lester. Dynamically improving explanations: A revision-based approach to explanation generation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 952–58, Nagoya, Japan, 1997.

[Callaway *et al.*, 1999] C. Callaway, B. Daniel, and J. Lester. Multilingual natural language generation for 3D learning environments. In *Proceedings of the 1999 Argentine Symposium on Artificial Intelligence*, pages 177–190, Buenos Aires, Argentina, 1999.

[Callaway, 2000] Charles Callaway. *Narrative Prose Generation*. PhD thesis, North Carolina State University, Raleigh, NC, 2000.

[Callaway, 2001] Charles Callaway. A computational feature analysis for multilingual character-to-character dialogue. In *Proceedings of the Second International Conference on Intelligent Text Processing and Computational Linguistics*, pages 251–264, Mexico City, Mexico, 2001.

[Cassell *et al.*, 2000] J. Cassell, M. Stone, and H. Yan. Coordination and context-dependence in the generation of embodied conversation. In *International Natural Language Generation Conference*, Mitzpe Ramon, Israel, 2000.

[Dalianis and Hovy, 1993] Hercules Dalianis and Eduard Hovy. Aggregation in natural language generation. In *Proceedings of the Fourth European Workshop on Natural Language Generation*, Pisa, Italy, 1993.

[Doran, 1998] Christine Doran. *Incorporating Punctuation into the Sentence Grammar: A Lexicalized Tree Adjoining Grammar Perspective*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998.

[Elhadad, 1992] Michael Elhadad. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. PhD thesis, Columbia University, 1992.

[Halliday, 1976] Michael Halliday. *System and Function in Language*. Oxford University Press, Oxford, 1976.

[Horacek, 1997] Helmut Horacek. A model for adapting explanations to the user's likely inferences. *User Modeling and User-Adapted Interaction*, 7(1):1–55, 1997.

[Hovy, 1993] Eduard H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385, 1993.

[Kantrowitz and Bates, 1992] M. Kantrowitz and J. Bates. Integrated natural language generation systems. In R. Dale, E. Hovy, D. Rosner, and O. Stock, editors, *Aspects of Automated Natural Language Generation*, pages 247–262. Springer-Verlag, Berlin, 1992.

[Lang, 1997] R. Raymond Lang. *A Formal Model for Simple Narratives*. PhD thesis, Tulane University, New Orleans, LA, 1997.

[Lebowitz, 1985] M. Lebowitz. Story-telling as planning and learning. *Poetics*, 14(3):483–502, 1985.

[Lester and Porter, 1997] James C. Lester and Bruce W. Porter. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101, 1997.

[Meehan, 1977] J. Meehan. Tale-Spin, an interactive program that writes stories. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.

[Mittal *et al.*, 1998] V. Mittal, J. Moore, G. Carenini, and S. Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–469, 1998.

[Pollard and Sag, 1994] C. Pollard and I. Sag. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, 1994.

[Propp, 1968] V. Propp. *Morphology of the Folktale*. University of Texas Press, Austin, TX, 1968.

[Quirk *et al.*, 1985] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Comprehensive Grammar of the English Language*. Longman Publishers, 1985.

[Robin, 1994] Jacques Robin. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. PhD thesis, Columbia University, December 1994.

[Searle, 1969] J. Searle. *Speech Acts*. Cambridge University Press, Cambridge, England, 1969.

[Segre, 1988] Cesare Segre. *Introduction to the Analysis of the Literary Text*. Indiana University Press, Bloomington, IN, 1988.

[Shaw, 1998] James Shaw. Segregatory coordination and ellipsis in text generation. In *COLING-ACL-98: Proceedings of the Joint 36th Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 1220–1226, Montréal, Canada, 1998.

[Stede, 1996] Manfred Stede. *Lexical Semantics and Knowledge Representation in Multilingual Sentence Generation*. PhD thesis, University of Toronto, Toronto, Ontario, 1996.

[Turner, 1994] Scott R. Turner. *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.

[Yazdani, 1982] Masoud Yazdani. How to write a story. In *Proceedings of the European Conference on Artificial Intelligence*, Orsay, France, July 1982.

[Young, 1996] R. Michael Young. Using plan reasoning in the generation of plan descriptions. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1075–1080, 1996.