A Declarative Model for Simple Narratives

R. Raymond Lang

Computer Science Department, Xavier University
New Orleans, Louisiana, 70125
rlang@xula.edu

Abstract

This paper describes a declarative model for simple narratives. The model describes what it is about a sequence of events such that reporting the sequence constitutes a story. Previous work in story generation has followed one of two tracks: (1) declarative, or isolating the regular structure of stories and then creating text which conforms to that structure, and (2) procedural, that is, modeling and recreating the processes used by human authors. Frequently, researchers in first track were unable to point to a concrete implementation based upon their model; researchers in the second track did not directly address the question of what constitutes a story. By implementing a story grammar, we address both these difficulties.

Background

Anthropology and linguistics intersect when attention focuses on the legends and folklore pertaining to a culture. In the early nineteenth century, Wilhelm and Jakob Grimm published their collections of traditional domestic tales (Grimm 1987) of the German people. Subsequently, Aleksandr Afanasev published his collection of Russian folk tales (Afanasev 1974), (Afanasev 1975) which Vladimir Propp used for his investigations into the morphology of the folktale (Propp 1968). Contemporary investigations into story structure reached a watershed in 1973, when B.N. Colby published a grammar for Eskimo folktales (Colby 1973). Colby was the first to use formal grammars to describe linguistic phenomenon beyond single sentences. We also use a formal grammar to describe narratives; to this end, we have developed a set of structural components along with rules for their composition. Our model is (1) general enough to apply to compilations of the sort described above, and (2) sufficiently detailed to rule out constructions of non-stories. In what follows, we briefly review previous work in story modeling according to whether the work was done from a declarative or procedural perspective.

Previous Work in Declarative Story Modeling

Rumelhart (Rumelhart 1975) develops a model for the organization that takes place in connected discourse but is absent in strings of sentences. In general, it is almost always necessary to infer (unstated) causal relationships in order to understand groups of sentences. These causal relationships relate sentences to each other. Rumelhart presents a grammar describing the inter-sentence bindings that arise in simple stories. The grammar is context-free and consists of syntactic rewrite rules each of which has a corresponding semantic interpretation rule. The primitives are meta-sentence components such as setting, episode, and event. Figure 1 illustrates two rules of his grammar.

1. Attempt → Plan + Application
   ⇒ MOTIVATE (Plan,Application)
2. Application → (Preaction)* + Action + Consequence
   ⇒ ALLOW (AND(Preaction,Preaction,. . .),
   {CAUSE | INITIATE | ALLOW} (Action,Consequence))

Figure 1: Two rules from David Rumelhart's story grammar

In Rumelhart's grammar (as well as those based on this grammar), the relationship a story component has to other components is expressed in "semantic" annotations accompanying the "syntactic" rules. Scare quotes distinguish the story grammar use of the terms syntactic and semantic from conventional use. In story grammars, the terms are intended to mean something like "structural" and "extra-structural," but in fact mean rather "captured by the grammar" and "not captured by the grammar." The "syntactic" structure of a portion of a text makes a particular rule applicable, then the relationship of this component to others is gleaned from the annotation to the rule. Unfortunately, the "syntax" given in story grammars doesn't rule out many constructions; while the "semantic" annotations are not formalized rigorously enough. This deficiency leaves the grammars open to wishful parsing and generation, a serious flaw which proponents of story grammars were unable to overcome. A major part of our work is a rigorous, formal framework used in relating story components to one another(Goldman and Lang 1993), (Lang 1997).

Following Rumelhart's "Notes on a Schema for Stories," a number of researchers expanded on Rumelhart's grammar (Bower 1976), (Frisch and Perlis 1981), (Johnson and Mandler 1980), (Mandler and Johnson 1977), (Stein and Glenn 1979), (Thorndyke 1977) while others attacked the foundations of the possibility of a "grammar for stories"(Black and Bower 1980), (Black and Wilensky 1979), (Garnham 1983), (Wilensky 1982). Eventually, the story grammars project was abandoned as unsuccessful, largely due to the crude state of formal techniques available at the time, but also due to the excessive demands made of story grammars as a cognitive mechanism.

Story Generation by Author Modeling

Around the same time as Rumelhart's seminal paper on schemas for stories, James Meehan published his dissertation on story generation (Meehan 1976). His system, Tale-Spin, inspired work in story generation from the perspective of author modeling, that is, by modeling the cognitive processes of a human author of stories. Turner's Minstrel (Turner and Dyer 1985), (Turner 1990), (Turner 1991a), (Turner 1991b) and the system described by Okada and Endo (Okada and Endo 1992) are representative samples of author-modeling systems for stories.

Meehan's Tale-Spin is a simulation of a forest world, producing natural language output describing the interactions of characters pursuing goals such as eating and drinking in a context where duplicity and hostility occur along with honesty and friendliness. Although Tale-Spin provides access to the meanings (conceptual dependency forms, in this case) from which the natural language text is constructed, the model by which the meanings themselves are generated is left implicit; and the relationships among the components of a story are deeply entwined in the procedures which drive the simulation.

Michael Lebowitz develops a model of story telling based upon an extensible library of plot fragments (Lebowitz 1985). These plot fragments provide narrative methods to achieve goals of the author. Such goals may be nonsensical from the point of view of the characters but are essential from the perspective of the author. For example, the author of a tale may have a goal to keep lovers apart; and, in pursuit of this goal, he will insert into a story elements that prevent lovers from meeting. It would be absurd for lovers themselves to seek obstacles to their meeting; but as a device for enhancing the dramatic interest of a story, it makes perfect sense for the author to devise such obstacles. Lebowitz's Universe program generates plot outlines using an algorithm very similar to that used in Tale-Spin except that author goals rather than character goals drive the mechanism. The research issue addressed by Lebowitz treats the realization of an author's goals in a story.

Scott Turner and Michael Dyer describe Minstrel (Turner and Dyer 1985), a story-telling program which generates stories that make a point as well as being believable and logically consistent. Turner describes further development of Minstrel in subsequent papers (Turner 1991a), (Turner 1991b), (Turner 1990). The overall design philosophy of Minstrel is to model human story-telling behavior. Turner's primary interest is in modeling human creativity, and he uses King Arthur-style tales as his domain. Although we are working in a superficially similar domain, our objective is a model that is independent of the process human authors undertake when writing a story.

A New Grammar for Stories

In this section we present selected features from our formal model for simple narratives (Lang 1997). Our model takes the form of a definite clause grammar, (hereafter referred to as "the-grammar"). The nonterminals are meta-components such as setting, episode, outcome, etc. The terminals are first-order predicate calculus schemas for the events, states, goals, and beliefs which, when instantiated and rendered into natural language, are the content of a simple narrative. The language described by the-grammar consists of lists of FOPC expressions. Each list in the language of the-grammar is an ordered representation of the facts and events contained in some tale; but the list does not specify the relations among the various terms in it. For example, Figure 2 shows an event list representing a portion of "The Bad Wife" (Afanasev 1975). The list adequately captures the states and events which the story comprises; but it does not represent the relationships among them. For example, nothing in the list indicates that the trick carried out by the peasant at time int(x7, x8) is in service of his goal held during time int(x10, x8) that his wife be in the pit. The information about the relationships among the elements of the event list is specified in the rules of the-grammar.

```
[holds(lives(peasant), int(x1, x2))
holds(married_to(peasant, wife), int(x1, x2))
holds(disobeys(wife, peasant), int(x1, x2))
occurs(quarrel(peasant, wife), int(x2, x12))
occurs(do(peasant, walk(in(woods))), int(x4, x5))
occurs(finds(peasant, pit, under(bush)), int(x5, x6))
goal(peasant, holds(loc(wife, in(pit)), int(x8, x9)), int(x10, x8))
occurs(do(peasant, trick(wife)), int(x7, x8))
holds(loc(wife, in(pit)), int(x8, x9))
holds(alone(peasant), int(x8, x9))
occurs(time_passes, int(x9, x20))
```

Figure 2: An event list representing a portion of "The Bad Wife"

The Story Rule

We model a story as having two sub-components, a setting and an episode list, which both have temporal intervals associated with them. The relationship between these two components is that the interval associated with the setting must meet that associated with the episode list. The rule show in Figure 3 expresses this.

```
story(story(Setting, Ep_list)) --->
    setting(Setting, S_time),
    episodes(Ep_list, E_time),
    {meets(S_time, E_time)}.
```

Figure 3: Starting rule for stories

The left hand side of the rule states that a story is a labeled pair `story(Setting, Ep_list)`. The right hand side states (1) that `Setting` and the temporal interval `S_time` must satisfy the rule for a setting; (2) `Ep_list` and the temporal interval `E_time` must satisfy the rule for episodes; and (3) the temporal intervals `S_time` and `E_time` must meet the constraint meets, in other words, the end point of `S_time` must be the starting point of `E_time`.

Rules for Episodes

The `episodes` rule in Figure 4 defines this component as a non-empty list of components of the form `ep(Ev,ER,A,O)`. Each element of an episodes list, `ep(Ev, ER, A, O)`, has four parts:

1. `Ev`, an initiating event
2. `ER`, an emotional response on the part of the protagonist
3. `A`, an action response on the part of the protagonist
4. `O`, an outcome or state description which holds at the conclusion of the episode.

The second argument of the episodes rule is the temporal interval during which all the episodes in the list take place. The temporal constraints in the episodes rule capture the relations among the intervals associated with the first episode, the remaining episodes, and the overall episodes list. The episode rule describes the relations among an episode's four components and the associated temporal intervals.

The action response rule describes (1) the form of the action response and the outcome and (2) the constraints on and the relationships among these two components and their associated temporal intervals. The predicate `term_time/2` relates a terminal expression to a temporal interval. `wm_call/1` is used to invoke the world model.

The left hand sides of most rules specify other components which the right hand side relates to the component described by the rule. By this means, the grammar is able to capture constraints within the rule itself rather than as a “semantic annotation.” Also, by adding arguments to the non-terminals in the rules, the grammar achieves the expressive power of a Chomsky type 0 grammar, as Black and Wilensky conclude a story grammar must be (Black and Wilensky 1979).

```
episodes(episodes([ep(Ev, ER, A, O)|Es]), int(Start, End)) --->
   episode(ep(Ev, ER, A, O), P, int(Start, Mid)),
   episode_rec(Es, P, int(Mid, End)).

episode(ep(ev(Ev, Ev_time), emot(Em, Resp_time)), A, O), Ep_time) --->
   story_event(Ev, Ev_time),
   wm_call([emot_reaction, Ev, Em]),
   emot_response(Ev, Em, Ev_time, Resp_time),
   wm_call([act_motiv, Em, A]),
   action_response(A, O, Act_time, Outcm_time),
   {starts(Ev_time, Ep_time),
    finishes(Outcm_time, Ep_time),
    meets(Ev_time, Resp_time),
    starts_before(Resp_time, Act_time)}.

action_response(act_resp(Act, ef(Efct)), outcm(State), Act_time, Efct_time) --->
   wm_call([effect, Act, Efct]),
   wm_call([consequence, Efct, State]),
   {term_time(Act, Act_time),
    term_time(Efct, Efct_time),
    term_time(State, Efct_time)},
    t_meets(Act_time, Efct_time)},
   [Act, Efct, State].
```

Figure 4: Three rules from the grammar for stories

```
story(setting(... ),
 episodes([ep(ev(quarrel(peasant, wife)),
              emot(distress),
              act_resp(...),
              outcm(alone(peasant))),
           ep(ev(...),
              emot(...),
              act_resp(...),
              outcm(...)),
           ep(...)]))
```

Figure 5: Partial parse tree of "The Bad Wife"

Figure 5 shows a parse tree for "The Bad Wife" [2, pp. 56-57] with the first episode shown in partial detail.

The Implementation

A major failing of previous models for narratives is that they were so ambiguous and poorly specified that it was difficult or impossible to implement them. This weakened the claim that these models of narratives were, in fact, *computational* in nature. We present a concrete implementation in support of our claim that our model, based on our theory of rational intention in autonomous agents (Goldman and Lang 1993), (Lang 1997), is indeed a computational model describing a non-trivial class of narratives. The implementation, named Joseph, produces randomly generated natural language narratives conforming to `the-grammar`.

Components of Joseph

The tasks of the Joseph story generation system are divided among the following components:

story grammar: At the core of Joseph is the implementation of `the-grammar`. The story grammar defines structured series of story components.

grammar interpreter: The grammar interpreter defines the search strategy of the generation process. We use depth-first, iteratively deepening search (Korf 1987) plus random choice to find a sequence of grammar rule rewrites which defines a valid story.

temporal predicates: A sequence of events constituting a story must satisfy temporal relations specified in `the-grammar`. We implement Allen's seven temporal relations (Allen 1984) in order to enforce temporal constraints on story components.

world model: A story must have content as well as form. Our story grammar produces abstract representations of stories; the grammar specifies terminals as schemas but does not specify the bindings of variables contained in those schemas. The potential instantiations of terminals are drawn from a set of actions which the characters may perform and fluents which vary during the course of the tale; this set of actions, fluents, and characters is enumerated in a world model.

natural language output unit: The story grammar and the world model define event lists, sequences of events which when rendered into natural language constitute the content of a story. These event lists are accompanied by the parse tree describing the structure of the story. These two data structures, the event list and the parse tree, are rendered into natural language text to produce the final output.

Figure 6 illustrates the interaction among these five components. The grammar interpreter initiates the generation process by invoking the top level rule for a story. When the generation process reaches a terminal, the grammar rule specifies a *leaf schema* and requests that the world model instantiate it. A leaf schema determines the form of the terminal and specifies how the world model elements fit into the parse tree. When the grammar interpreter has produced a completely instantiated parse tree and event list, these two structures are sent to the natural language output unit, which maps them to surface text.

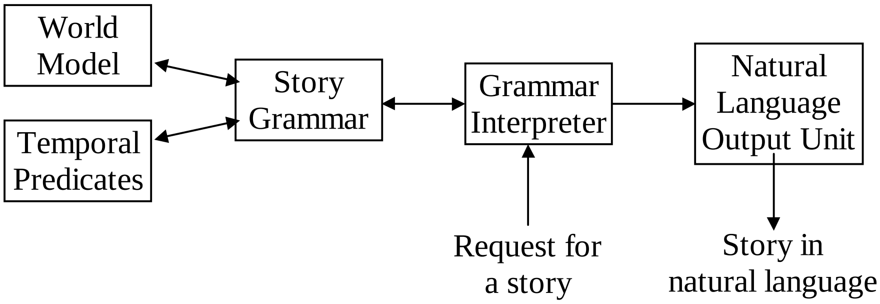

Figure 6: Schematic diagram of the relationship among the five Joseph components

The World Model

Coherence relations such as causality and goal-directedness serve as the ‘‘glue” which hold together the states and events which the story comprises. By the form of a story, we mean these coherence relations that hold among the constituent components. By a story’s content we mean the component states and events themselves. Our model separates the rules which govern the form of a simple narrative from the elements which make up its content. We implement this distinction by packaging the grammar terminals into a separate world model. This has the added advantage of easier extensibility.

As the story grammar generates a tree representing a story, it calls the world model to provide components to instantiate the terminals. For example, two terminal components of an episode are an event and the protagonist’s reaction to that event. However, the grammar specifies neither the event nor the reaction. To instantiate these components, the grammar invokes the world model, which specifies the events that may occur and what a character’s reaction to an event will be. There are 11 predicates which the-grammar requires in a world model. All except for hero/1 accept an argument World which is the list of terminals (i.e. states and events) that have been generated up to the point the world model predicate is invoked. The predicates use the World argument to check for preconditions of relations.

1. `hero(Agent)` Agent may be the protagonist of a story.

2. `fact(Fact, World)` Fact can be added to the setting of a story. This predicate checks World for preconditions of facts and also to avoid supplying duplicate facts.

3. `ep_prim(Event, World)` Given a world state World, Event can initiate an episode. Our world model is constructed by analyzing several narratives in Afanasev’s collection of Russian folk tales. The clauses of this predicate contain those events our analysis identified as initiating episodes. For example, the following clause says that an episode can begin with a quarrel between the protagonist (Agent) and his (or her) spouse.

```
ep_prim(quarrel(Agent, Spouse), World) :-
   holds_in(married_to(Agent, Spouse),
            World).
```

4. `effect(Act, Result, World)` When done in World, Act causes Result to happen if Result is an event, or causes Result to hold when Result is a state. These Results are possible effects of an action. If an Act appears in multiple clauses of effect/3, then the world model will choose a clause at random to instantiate a Result.

5. `consequence(State1, State2, World)` Given World, the consequence of State1 is State2. This predicate specifies entailment for states; that is, if State1 holds in World, then State2 holds simultaneously. For example, in a World in which Person is married to Spouse, the consequence of Spouse being in a pit is that Person lives alone.

```
consequence(loc(Spouse, in(pit)),
            alone(Person), World) :-
   holds_in(married_to(Person, Spouse),
            World).
```

6. `plan(Title, Steps, World)` The plan Title consists of this list of Steps. A plan in Joseph is a compound act. The Title of the plan is a term denoting the compound act as a whole, and the Steps are the individual components of the compound act. The grammar allows a plan to appear wherever an action in service of a goal can appear. In the following example, the plan entitled con_with(Demon, set_of(Mark) consists of four steps: (1) wait for Demon to possess the daughter of one of the Marks, (2) pretend to exorcise the Demon, (3) collect money from the Mark, and (4) repeat.

```
plan(con_with(Demon, set_of(Mark)),
     [act(wait(Demon,
      possess(daughter(Mark)))),
      act(pretend(exorcise, Demon,
          daughter(Mark))),
      act(collect(money, Mark)),
      act(repeat(scam))], _).
```

7. `plan_efct(Plan_title, Effect, World)` Executing in World the plan entitled Plan_title causes Effect, which may be either a state or an event. As with effect/3, these are possible effects, and the world model will provide an effect at random if a Plan_title appears in multiple clauses of plan_efct/3. The example below states that, when the above plan is carried out on merchants, the result is that the protagonist possesses wealth. Observe that the world model assumes that the protagonist is the agent of the plan. This assumption works in Joseph since only the protagonist of a story may adopt goals and carry out actions or plans in pursuit of them.

```
plan_efct(con_with(imp,set_of(merchant)),
            state(possess(Prot, wealth)),
            World) :-
   protagonist(Prot, World).
```

8. `emot_reaction(Event, Emotion, World)` When Event happens in World, the protagonist will feel Emotion. For example, in a World in which the protagonist is married to Spouse, if the Spouse goes to

Palestine, the protagonist will miss the Spouse. This predicate makes an assumption similar to that of plan_efct/3: it is implicit in emot_reaction/3 that every event happens to the protagonist and that the protagonist is the character that will feel the emotion. This assumption is based on the restriction that Joseph stories revolve around a single protagonist.

```
emot_reaction(go(Spouse,Place),
              miss(Spouse), World) :-
   protagonist(Prot, World),
   holds_in(married_to(Prot, Spouse),
            World),
   foreign_land(Place).
```

9. `action_motiv(Emotion, Action, World)` Action is a potential action for someone in World who has Emotion. The Actions specified in this predicate are unplanned, spontaneous reactions to external events. In the following example, in a World in which Self is the protagonist, Self is at home, and Baba Yaga has come to the home, fear of Baba Yaga will motivate the protagonist to hide himself behind the stove.

```
action_motiv(fear_of(baba_yaga),
             hide(Self, behind(stove)),
             World) :-
   holds_in(loc(Self, home), World),
   happened_in(comes(baba_yaga, home),
               World).
```

10. `goal_motiv(Emotion, Goal, World)` In World, Emotion motivates one to adopt Goal. In the example below, in a World in which the protagonist's Spouse has been captured by a king and the protagonist misses the Spouse, the protagonist will adopt as a goal that the Spouse be rescued.

```
goal_motiv(miss(Spouse),
           holds(rescued(Spouse), _),
           World) :-
   protagonist(Prot, World),
   holds_in(married_to(Prot, Spouse),
            World),
   happened_in(capture(king, Spouse),
            World).
```

11. `intention(Prot, Goal, Action, World)` In World, Prot believes Action is a means to achieve Goal. Action may be either a primitive action or the title of a plan. For example, in a World in which the protagonist is married to Spouse but protagonist lives alone, the protagonist believes that retrieving Spouse will achieve the goal of relieving melancholy.

```
intention(Prot, relieved(melancholy),
          retrieve(Spouse), World) :-
   holds_in(married_to(Prot, Spouse),
            World),
   holds_in(alone(Prot), World).
```

The model Joseph uses to generate stories supports creation of stories resembling Russian folk tales. The predicates in the model are representations of the characters, events, actions, effects, emotions, and goals appearing in eight arbitrarily selected Russian folk tales.

Output Samples

This section presents selected stories randomly generated by this implementation. The simplest possible story in our model consists of a setting followed by a single episode composed of an event to which the protagonist reacts without forming any goals.

once upon a time there lived a dog. one day it happened that farmer evicted cat. when this happened, dog felt pity for the cat. in response, dog sneaked food to the cat. farmer punished dog.

A slightly more complicated single-episode story has the protagonist adopt a goal and carry out action(s) in pursuit of that goal. Stories with goals are more complex because (1) the system must constrain the protagonist's actions to those intended to achieve the goal, and (2) the system must track the effects of these actions to determine if the goal is met.

once upon a time there lived a cossack. one day it happened that imp possessed daughter of a boyar. when this happened, cossack felt love for the daughter of a boyar. in response, cossack made it his goal that he would be married to the daughter of a boyar. cossack exorcised the imp from the daughter of a boyar. cossack was married to daughter of a boyar.

Our implementation also produces multiple-episode stories. Episodes may be arranged in two ways: sequentially or nested one inside another. Episode nesting takes place when the world model instantiates an action's effect to an event rather than a state. The tale below illustrates nested episodes. The nested episode is emphasized in the surface text. The action of the outer episode (taking a walk in the woods) does not have a state as its effect. Instead, this action triggers an event (finding a pit) which initiates a new episode contained within the other.

once upon a time there lived a peasant. peasant was married to wife. 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.

The final example illustrates goal failure. Characters do not always reach their goals. When a character adopts a goal, the grammar specifies that it make some series of attempts to achieve it. The world model enumerates (1) the actions that an agent may takes toward a given goal and (2) the effects these actions have. The grammar tries to match the effects of the actions with the goal. When it succeeds, the goal has been met. If the world model does not provide an action having an effect that entails the goal or unifies with it, then the goal fails. Our theory of rational intention (Goldman and Lang 1993), (Lang 1997) specifies conditions for an agent to give up a goal. The Joseph system incorporates these conditions implicitly.

once upon a time there lived a peasant. peasant was hungry. one day it happened that the peasant met christ. when this happened, peasant felt awe. in response, peasant begged christ to provide food. christ told peasant to eat ram. when this happened, peasant felt obedient. in response, peasant made it his goal that ram would be eaten. peasant trapped ram. ram whacked peasant. peasant believed it impossible that ram would be eaten. peasant was hungry.

Shortcomings and Limitations

This model is limited in a number of ways. We discuss some of the most important limitations here.

First, the implementation assumes that stories will have only a single protagonist, enforcing a fixed point of view which at best requires us to do some "force-fitting" to represent some tales and at worst leaves us unable to represent some tales at all. For example, the opening episodes of "Ivan and the Devil" (Afanasev 1975) relate the events leading up to the drowning of an old man. The man's adult son, Ivan, continues as the central character of the story. Ivan meets and eventually outwits a character who turns out to be the devil, thereby winning his father's release from hell. This story has two central characters: the old man in the first part of the story, then Ivan for the remainder of the tale. Our model does not capture stories in which the protagonist shifts from one character to another; however, work is progressing on refining the model so that it can represent such tales. A related issue is that of representing the beliefs and goals of multiple agents. Representing such items would require a mechanism to detect when there is a conflict between two agents' goals. Goal conflicts could be used as a factor in determining the structure as well as the content of a story (Wilensky 1982).

Second, the representation of plans is insufficiently flexible to represent unforeseen circumstances or unexpected events that happen during the plan's execution. This is an instance of a more general problem with the world model inasmuch as it is necessary to identify in advance of generation the actions, goals, effects, etc. that may appear in a story. Ideally, the world model predicates should be able to provide novel terms on the basis of a limited number of axioms describing how the world operates; such a set of predicates would have as few clauses as possible which relate a specific action to a specific effect. The present implementation is ontologically promiscuous in this regard.

Third, the mechanism by which Joseph matches the effects of actions to desired goal states does not always detect indirect goal achievement (that is, an action taken in service of a goal has an effect which is not the goal, but has the goal state as a consequence).

Despite these shortcomings, the Joseph story generation system represents a significant achievement since it is the first such system constructed from an explicit, formal model for stories. Our model uses a well-established temporal logic to represent states and events, and is informed by a new, first-order theory of rational intention in autonomous agents which allows us to describe goal-directed behavior.

References

(Afanasev 1974) Afanasev, A. N., Russian Folk Tales. New York: E.P. Dutton and Co., 1974.

(Afanasev 1975) Afanasev, A. N., Russian Fairy Tales. New York: Pantheon Books, 1975.

(Allen 1984) Allen, J. F., ''Towards a general theory of action and time," Artificial Intelligence, vol. 23, pp. 123--154, 1984.

(Black and Bower 1980) Black, J. B. and Bower, G. H., ''Story understanding as problem-solving," Poetics, vol. 9, pp. 223--250, 1980.

(Black and Wilensky 1979) Black, J. B. and Wilensky, R., ''An evaluation of story grammars," Cognitive Science, vol. 3, pp. 213--230, 1979.

(Bower 1976) Bower, G. H., ''Experiments on story understanding and recall," Quarterly Journal of Experimental Psychology, vol. 28, pp. 511--534, 1976.

(Colby 1973) Colby, B. N., ''A partial grammar of eskimo folktales," American Anthropologist, vol. 75, pp. 645--662, 1973.

(Frisch and Perlis 1981) Frisch, A. M. and Perlis, D., ''A re-evaluation of story grammars," Cognitive Science, vol. 5, pp. 79--86, 1981.

(Garnham 1983) Garnham, A., ''What's wrong with story grammars?," Cognition, vol. 15, pp. 145--154, 1983.

(Grimm 1987) Brothers Grimm, T., The Complete Fairy Tales of the Brothers Grimm. New York: Bantam Books, 1987.

(Goldman and Lang 1993) Goldman, R., Lang, R. R., "Intentions in Time", Second Symposium on Logical Formalizations of Commonsense Reasoning, Austin, Texas, 1993.

(Johnson and Mandler 1980) Johnson, N. S. and Mandler, J. M., ‘‘A tale of two structures: Underlying and surface forms in stories,” Poetics, vol. 9, pp. 51--86, 1980.

(Korf 1987) Korf, R., ‘‘Search,” in Encyclopedia of Artificial Intelligence (Shapiro, S., ed.), Wiley-Interscience, 1987.

(Lang 1997) Lang, R. Raymond, A Formal Model for Simple Narratives. Doctoral Dissertation, Tulane University, 1997.

(Lebowitz 1985) Lebowitz, M., ‘‘Story telling and generalization,” in Proceedings of the Seventh Annual Conference of the Cognitive Science Society, (Berkeley, California), pp. 100--109, 1985.

(Mandler and Johnson 1977) Mandler, J. M. and Johnson, N. S., ‘‘Remembrance of things parsed: Story structure and recall,” Cognitive Psychology, vol. 9, pp. 111--151, 1977.

(Meehan 1976) Meehan, J. R., The Metanovel: Writing Stories by Computer. PhD dissertation, Yale University, 1976.

(Okada and Endo 1992) Okada, N. and Endo, T., ‘‘Story generation based on dynamics of the mind,” Computational Intelligence, vol. 8, pp. 123--160, 1992.

(Propp 1968) Propp, V., Morphology of the Folktale. University of Texas Press, 1968.

(Rumelhart 1975) Rumelhart, D. E., ‘‘Notes on a schema for stories,” in Representation and Understanding: Studies in Cognitive Science (Bobrow, D. G. and Collins, A., eds.), pp. 211--236, New York: Academic Press, Inc., 1975.

(Stein and Glenn 1979) Stein, N. L. and Glenn, C. G., ‘‘An analysis of story comprehension in elementary school children,” in New Directions in Discourse Processing (Freedle, R. O., ed.), vol. 2, pp. 53--120, Norwood, New Jersey: Ablex Publishng Corporation, 1979.

(Thorndyke 1977) Thorndyke, P. W., ‘‘Cognitive structures in comprehension and memory of narrative discourse,” Cognitive Psychology, vol. 9, pp. 77--110, 1977.

(Turner and Dyer 1985) Turner, S. R. and Dyer, M. G., ‘‘Thematic knowledge, episodic memory and analogy in minstrel, a story invention system,” in Proceedings of the Seventh Annual Conference of the Cognitive Science Society, (Berkeley, California), pp. 371--375, 1985.

(Turner 1991a) Turner, S. R., ‘‘A case-based model of creativity,” in Proceedings of the Eleventh Annual Conference of the Cognitive Science Society, (Chicago, Illinois), 1991.

(Turner 1991b) Turner, S. R., ‘‘A case-based model of creativity,” in Workshop on Creativity, AAAI-91, (Anaheim, California), 1991.

(Turner 1990) Turner, S. R., ‘‘MINSTREL: A model of computer storytelling,” in Workshop on Interactive Fiction and Synthetic Realities, AAAI-90, (Boston, Massachusetts), 1990.

(Wilensky 1982) Wilensky, R., ‘‘Points: A theory of the structure of stories in memory,” in Strategies for Natural Language Processing (Lehnert, W. G. and Ringle, M. H., eds.), pp. 345--374, Hillsdale, N.J.: Lawrence Erlbaum, 1982.

References

[1] (Afanasev 1975)
[2] (Afanasev 1974)
[3] (Allen 1984)
[4] (Black and Bower 1980)
[5] (Black and Wilensky 1979)
[6] (Bower 1976)
[7] (Colby 1973)
[8] (Frisch and Perlis 1981)
[9] (Garnham 1983)
[10] (Grimm 1987)
[11] (Goldman and Lang 1993)
[12] (Johnson and Mandler 1980)
[13] (Korf 1987)
[14] (Lang 1997)
[15] (Lebowitz 1985)
[16] (Mandler and Johnson 1977)
[17] (Meehan 1976)
[18] (Okada and Endo 1992)
[19] (Propp 1968)
[20] (Rumelhart 1975)
[21] (Stein and Glenn 1979)
[22] (Thorndyke 1977)
[23] (Turner and Dyer 1985)
[24] (Turner 1991a)
[25] (Turner 1991b)
[26] (Turner 1990)
[27] (Wilensky 1982)