Representing Story Plans in SUMO

Jeffrey Cua
Center for Human Language Technologies
De La Salle University, Manila, Philippines
cuajeffreyleonardcompro1@yahoo.com

Ethel Ong
College of Computer Studies
De La Salle University, Manila, Philippines
ethel.ong@delasalle.ph

Ruli Manurung
Faculty of Computer Science
University of Indonesia, Jakarta, Indonesia
maruli@cs.ui.ac.id

Adam Pease
Articulate Software
Angwin, California, USA
apease@articulatesoftware.com

Abstract

Automatic story generation systems require a body of commonsense knowledge about the basic relationships between concepts we find everyday in our world in order to produce interesting narratives that describe human actions and world events. This paper presents an ongoing work that investigates the use of Suggested Upper Merged Ontology (SUMO) to represent storytelling knowledge and its inference engine Sigma to query actions and events that may take place in the story to be generated. The resulting story plan (*fabula*) is also represented in SUMO, allowing for a single story representation to be realized in various human languages.

1 Introduction

People combine words and events from their knowledge source of words, their meanings and their relationships in order to tell stories about their lives, their communities, and their daily experiences. In order for computers to achieve the same level of expressiveness to provide a more fluent man-machine interaction, they must be provided with the same collection of knowledge about the basic relationships between things and events.

Picture Books (Solis et al, 2009), an automatic story generator that generates story text for children from a given input set of picture elements (backgrounds, characters and objects), utilized a semantic ontology whose design has been adapted from ConceptNet (Liu and Singh, 2004). The background serves as the setting of the story and is also used to determine the theme. Semantic concepts needed by the story planner, specifically objects, story events, and character actions are classified according to the semantic categories of ConceptNet, namely things, spatial, events, actions, and functions. This mapping approach constrained the flexibility of the system, as new themes would entail repopulating the sequences of possible events manually into the knowledge base. Events and actions are selected according to their associated themes, and not marked with preconditions that specify constraints under which certain actions can be performed and the corresponding consequential events that may arise.

Swartjes (2006) developed a story world ontology containing two layers, the upper story world ontology and the domain-specific world ontology. The upper story world ontology is independent of any story structures or story domains and models a vast amount of possible actions and events. It is also limited to high-level concepts that are meta, generic or abstract to address a broad range of domain areas. A domain-specific story world ontology, on the other hand, applies the upper story world ontology to a certain story domain.

Kooijman (2004) suggests the use of the Suggested Upper Merged Ontology (SUMO) as an upper ontology to capture the semantics of world knowledge. SUMO (Niles and Pease, 2001) is an

Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity, pages 40–48,
Los Angeles, California, June 2010.

open source formal and public ontology. It is a collection of well-defined and well-documented concepts, interconnected into a logical theory. It numbers some 20,000 terms and 70,000 axioms. Axioms are in first-order logic form (with some higher order extensions) and reflect commonsense notions that are generally recognized among the concepts. They place a constraint on the interpretation of concepts and provide guidelines for automated reasoning systems such as Sigma (Pease, 2003). Formal terms in SUMO are mapped to synsets in WordNet (Pease, 2006).

There are other noteworthy ontologies that can be considered. Like SUMO, Cyc (Lenat, 1995) is a large-scale, language-independent and extensible knowledge base and commonsense reasoning engine, but it is proprietary and its open-source version, OpenCyc[1], has no inference rules. DOLCE (Gangemi, 2003) is a small-scale descriptive ontology with a cognitive orientation. BFO (Smith, 1998) is another small-scale upper ontology supporting domain ontologies developed for scientific research domain, such as biomedicine. Thus, no ontology other than SUMO had the characteristics of being comprehensive enough to include formalizations that represent detailed elements of everyday life (e.g., *furniture*, *breaking an object*, *emotion*), being open-source, having expressiveness of at least first order predicate calculus so that arbitrary rules about actions and consequences can be represented, having an associated open-source first-order inference engine, and a language generation capability so that stories can be automatically presented in multiple human languages

This paper presents SUMOs (SUMO Stories), an automatic story generator that uses first-order logic to declaratively describe models of the world, specifically those aspects of the world that represent storytelling knowledge for children's stories of the fable form. The story planner then utilizes an open source browsing and inference engine Sigma to infer this knowledge to generate a story plan (*fabula*) also in first-order logic form.

Using first-order logic enables a less restricted semantics compared to description logic, which is commonly used for knowledge representation of large ontologies. Though having lesser constraints will have an impact on the speed of inference, it is overcome by the advantage of having greater representational capability. In particular, the axiomatic nature of actions and their consequences, so essential for reasoning about narrative structures, is not supported by description logics, which focus on category and instance membership reasoning.

Section 2 provides a background on the knowledge required by story generation and how these were represented in Picture Books, which is used as the basis for the storytelling knowledge. Section 3 discusses the representation of the storytelling knowledge to SUMO. The SUMOs architecture depicting the interaction between the story planner and Sigma to derive the story plan is then presented in Section 4. The paper concludes with a summary of what we have accomplished so far, and presents further work that can be done.

2 Storytelling Knowledge

Theune and her colleagues (2006) presented five levels of the different aspects of a story that must be represented in the semantic network. These are the story world knowledge, character representations, a causal and temporal network to represent plot structures, representational model of narratological concepts, and the representation of the story's potential effects on the user. Only the first four levels are included in this study.

According to Swartjes (2006), a story is composed of a story world where the story takes place, the characters that interact in the story world, and the associated objects. Consider the story generated by Picture Books in Table 1 about *Rizzy the rabbit* who learns to be honest (Hong et al, 2008).

The afternoon was windy. Rizzy the rabbit was in the dining room. She played near a lamp. Rizzy broke the lamp. She was scared. Mommy Francine saw that the lamp was broken. Rizzy told Mommy Francine that Daniel broke the lamp. Daniel the dog told her that he did not break the lamp. Daniel was upset. He got punished. Mommy Francine told Daniel that he was grounded. He cried. Rizzy felt guilty. She told Mommy Francine that she broke the lamp. Mommy Francine told Rizzy that she should have been honest. Rizzy apologized to Mommy Francine. Mommy Francine forgave Rizzy. Rizzy apologized to Daniel. He forgave her. Mommy Francine told Rizzy to be honest. She told her that being honest is good. From that day onwards, Rizzy always was honest.

Table 1. Sample story generated by Picture Books (Hong et al, 2008)

[1] OpenCyc web site, http://www.opencyc.org/

The story elements in Table 1 were determined from the background (i.e., *dining room*), the characters (i.e., *Rizzy and her mommy Francine*) and object (i.e., *lamp*) that the child user places into his/her picture using the Picture Editor of the system in Figure 1.

The background serves as the main setting of the story, and combined with the selected objects, is used to determine the theme. Consider the *bedroom* setting. If the associated object is a *lamp*, then the theme is about bravery (i.e., *do not be afraid of the dark*). If the object is a set of *toy blocks*, the theme can be about being neat. In Picture Books, such associations are manually determined and entered into the database. In SUMOs, these associations should be inferred automatically through axioms that should be commonsense, and not be explicit encoding of narrative knowledge.

Figure 2. Picture Editor (Hong et al, 2008)

Stories generated by Picture Books follow a basic plot dictated by Machado (2003) that flows from negative to positive and comprises four subplots, namely the problem, rising action, solution and climax. The theme is subdivided into these four subplots, each representing a major event in the story.

Each subplot contains at least two author goals representing the goal of the scene and the corresponding consequence of the goal. An author goal is translated into one or more character goals, each representing an action performed by the character (main, secondary, or adult character) in order to achieve the author goal. A character goal translates directly to one declarative sentence in the generated story. Table 2 shows the author goals and the character goals for some of the sentences in the story in Table 1.

The design of the character goal is based from the action operators of Uijlings (2006) which is easily transformed to a declarative sentence in active voice using the surface realizer *simpleNLG* (Venour and Reiter, 2008). In the case of Picture Books, however, the approach resulted in a story where every sentence describes an action or a feeling (i.e., *scared*, *guilty*, *upset*) that is performed by the character, as seen in Table 1.

Subplot #1	
Author goal 1.1:	
Goal of the scene	Child is doing an activity
Character goal	*<character> plays <object>*
Resulting text	*Rizzy the rabbit played near a lamp.*
Author goal 1.2:	
Goal consequence	Child caused a problem
Character goal	*<character> destroys <object>*
Resulting text	*Rizzy broke the lamp.*
Subplot #2	
Author goal 2.1:	
Goal of the scene	Child lied
Character goal	*<main character> told <adult character> that <secondary character> <did the action>*
Resulting text	*Rizzy told Mommy Francine that Daniel the dog broke the lamp.*
Author goal 2.2:	
Goal consequence	Another child gets punished
Character goal #1	*<secondary character> receives <punishment>*
Resulting text #1	*Daniel the dog got punished.*
Character goal #2	*<adult character> issues <punishment> to <secondary character>*
Resulting text #2	*Mommy Francine told Daniel that he was grounded.*

Table 2. Sample author goals and character goals associated with the theme *Being Honest* (Hong et al, 2008)

The story planner of Picture Books utilizes two types of knowledge, the operational knowledge and the domain knowledge. The operational knowledge contains a static description of the different backgrounds and their associated themes and objects, the child characters and their corresponding parent characters, as well as the occupation of the parents. For each theme, the set of character goals needed to instantiate the major events in the theme are also specified.

The domain knowledge, on the other hand, contains a semantic description of objects and events that can occur, as well as actions that can be performed. For example, *breaking an object* results to *getting punished*, and *grounded* is a form of *punishment*.

Character goals are instantiated by accessing the semantic ontology to search for concepts that are directly related to the input concept. There are two search methods. The first method searches for another concept that has a relationship with the given concept while satisfying the semantic category. For example, ontoSpatial("play") triggers a search for all concepts connected to *play* within the *spatial* semantic category, such as the semantic relationship locationOf("play", "park"). The second method searches for a path that semantically relates the two given concepts. For example, ontoAction("vase", "method of destruction") triggers a search for a path to relate how a *vase* can be destroyed, and yields the following relationships:

```
CapableOf("break", "vase")
Isa("method of destruction", "break")
```

3 Representing Storytelling Knowledge in SUMO

A crucial part of the work involved in the development of SUMOs is the representation of the storytelling knowledge and the evolving story plan in SUMO and the use of the Sigma reasoning engine to infer story facts and events.

The storytelling knowledge represented in SUMO includes the semantic description about concepts, objects and their relationships. From a given input set of story elements comprising the selected background, characters, and objects, a query is sent to Sigma to determine a possible starting action that can be performed by the main character in the story. The story then progresses based on the relationships of character actions and reactions, which are the stored facts in SUMO.

Similar to Picture Books, the resulting story plan is created based on a pre-authored plot of problem, rising action, resolution and climax. But instead of attaching the next set of actions and emotions of characters to author goals, in SUMOs, the set of actions that a character can do – reaction to events and objects, experience emotions such as joy and sadness, and subsequent actions based on their emotions – are represented in SUMO logic.

The storytelling knowledge was formulated using a set of predicates that can be classified into four main types. Factual predicates specify properties of characters, objects, and locations. Semantic predicates define the semantic relationships between concepts. Actions and events predicates define the causal relationships between actions and events. Thematic predicates represent a new set of predicates to relate story themes to actions.

3.1 Conceptualizing Story Characters, Objects, and Backgrounds

Factual predicates represent the characters, their roles, the locations, and the objects that may comprise a story. The *class* and *subclass* axioms of SUMO[2] are used to define the set of characters, objects and locations.

Children's stories of the fable form are portrayed by animals that can capture the imagination and attention of the readers. Animal characters are given names, such as *Ellen the elephant*, *Rizzy the rabbit*, and *Leo the lion*, to give the impression that the characters are friends that the children are getting to know better through reading the story (Solis et al, 2009). Representing this in SUMO entails the use of the *subclass* axiom to represent class inheritance as shown below:

```
(subclass RabbitCharacter StoryCharacter)
```

Class definitions include slots that describe the attributes of instances of the class and their relations to other instances (Noy, 2001). A character in SUMOs has the attributes *type* (whether adult or child), *gender*, and *name*. An example axiom to represent a female child *RabbitCharacter* whose name will be "*Rizzy*" is shown below. Similar axioms are defined for all the other characters.

```
(=>
   (and
    (instance ?RABBIT RabbitCharacter)
    (attribute ?RABBIT Female)
    (attribute ?RABBIT Child))
   (name ?RABBIT "Rizzy"))
```

Backgrounds and objects are also defined using the *subclass* axiom and inherit from existing classes in SUMO, for example,

[2] SUMO Ontology Portal, http://www.ontologyportal.org/

```
(subclass LivingRoom Room)
(subclass Lamp LightFixture)
(subclass Lamp ElectricDevice)
(attribute Lamp Fragile)
```

Further definitions can be provided for *living room* to differentiate it from other rooms, such as being disjoint from bathroom, and has a primary purpose of supporting social interaction, as shown below. Similarly, the definition for *lamp* can also be extended to distinguish it from other electric light fixtures, e.g., a lamp is moveable unlike a chandelier, but is plugged in when operating unlike a flashlight.

```
(=>
   (instance ?R LivingRoom)
   (hasPurpose ?R
       (exists (?S)
          (and
             (instance ?S SocialInteraction)
             (located ?S ?R)))))
(disjoint  LivingRoom Bathroom)
```

3.2 Representing Semantic Concepts

Aside from the properties of objects that are modeled using the *attribute* axiom, semantic relationships that may hold between two concepts involving types of activities or actions, character emotions, locations of objects, and abilities of characters or objects must also be modeled. Table 3 shows sample semantic relationships for these concepts as represented in Picture Books, following the semantic categories of ConceptNet (Liu and Singh, 2004).

Objects	IsA (doll, toys)
Activities	IsA (play games, activity)
Concepts	IsA (grounded, punishment) IsA (disorder, problem) IsA (no appetite, problem) IsA (dizzy, discomfort) IsA (itchy, discomfort)
Emotions	IsA (happy, emotion) IsA (scared, emotion)
Reaction to Events	EffectOf (break object, scared) EffectOf (meet new friends, smile)
Location	LocationOf (toys, toy store)
Capability	CapableOf (lamp, break) CapableOf (glass of water, break) CanBe (toys, scattered)

Table 3. Semantic relationships in Picture Books based on ConceptNet (Hong et al, 2008)

In SUMOs, all *isA(entity1, entity2)* relations were replaced with the axiom *(subclass entity1 entity2)*. To specify that an entity is in a location, i.e., *locationOf(toys, toy store)*, first, we create an instance of a *toystore* and then specify that a certain *toy* instance is in that *toystore*, as follows:

```
(=>
   (instance ?TOYSTORE ToyStore)
   (exists (?TOY)
     (and
        (instance ?TOY Toy)
        (located ?TOY ?TOYSTORE))))
```

The *capability* axiom is used to conceptualize the capability relation *(capability ?process ?role ?obj)*. It specifies that *?obj* has the specified *?role* in the *?process*. For example, a *lamp* or a *glass* is the patient (receiver) of the process *breaking*, while a *toy* is the patient for the process *scattering*.

```
(capability Breaking experiencer Lamp)
(capability Breaking experiencer Glass)
(capability Scattering experiencer Toy)
```

Reaction to events is expressed using the *if-else* axiom of SUMO, for example, if a child character causes an accident (a damage), then he/she will feel anxiety. Emotions are represented using the *attribute* relation.

```
(=>
  (and
    (instance ?ACCIDENT Damaging)
    (instance ?CHARACTER StoryCharacter)
    (attribute ?CHARACTER Child)
    (agent ?ACCIDENT ?CHARACTER))
  ((attribute ?CHARACTER Anxiety)))
```

3.3 Conceptualizing Actions and Events

Swartjes (2006) noted that organizing actions and events, and causally relating them, is an essential step in story generation. Independent of the story plot, the causes and effects of character actions can be used to describe the events that form the story.

Actions define activities that can be performed by a character in the story, such as *play*, *tell a lie*, or *cry*. Events, on the other hand, occur in the story as a result of performing some actions, such as a *lamp breaking* as a result of a character or an object hitting it. Swartjes (2006) further notes that events are not executed by a character.

Action predicates are used to define the actions that may take place given a set of world state. Consider the axiom below which provides a set of four possible actions – *RecreationOrExercise*, *Looking*, *Maintaining*, and *Poking* – that can be performed (as an agent) or experienced by a child character who is situated *near* a *lamp* object in the story world. These four actions are subclasses of the *IntentionalProcess* of SUMO.

```
(=>
 (and
  (orientation ?CHARACTER ?OBJECT Near)
  (instance ?CHARACTER StoryCharacter)
  (attribute ?CHARACTER Child)
  (instance ?OBJECT Lamp))
 (and
  (capability RecreationOrExercise
               experiencer ?CHARACTER)
  (capability Looking experiencer ?CHARACTER)
  (capability Maintaining experiencer ?CHARACTER)
  (capability Poking experiencer ?CHARACTER)))
```

Again, the *capability* relation is used but in this instance, to specify that the character has the role of experiencing the specified process. While both the agent and the experiencer roles represent the doer of a process, an experiencer does not entail a causal relation between its arguments.

Event predicates are used to model explicit events that may take place as a result of some character actions. Consider again the *exists* axiom below which states that an instance of an event (in this case *damaging*) can occur when there is a child character (the *agent*) playing near a fragile object. The *subprocess* axiom is used to represent a temporally distinguished part of a process and also expresses a chain of cause and effect subprocesses for *playing* and *damaging*. The recipient (*patient*) of the event is the object.

```
(=>
 (and
  (agent ?X ?CHARACTER)
  (instance ?CHARACTER StoryCharacter)
  (attribute ?CHARACTER Child)
  (instance ?OBJECT Object)
  (attribute ?OBJECT Fragile)
  (instance ?X RecreationOrExercise)
  (orientation ?CHARACTER ?OBJECT Near)
 (exists (?DAMAGE)
  (and
   (instance ?DAMAGE Damaging)
   (subProcess ?DAMAGE ?X)
   (agent ?DAMAGE ?CHARACTER)
   (patient ?DAMAGE ?OBJECT))))
```

Although suitable for inference, the given axiom does not fully capture the desired truth as the notion of time is not represented. The axiom says "*if a child plays at any point in time, and is near an object at any point in time (not necessarily while playing), then the object gets damaged during playing*". The more accurate axiom below uses *holdsDuring* to show that the time frames of the actual playing and being near the object are the same, thus increasing the likelihood of the character who is playing to cause the damage.

```
(=>
 (and
  (instance ?X RecreationOrExercise)
  (agent ?X ?CHARACTER)
  (instance ?CHARACTER StoryCharacter)
  (attribute ?CHARACTER Child)
  (instance ?OBJECT Object)
  (attribute ?OBJECT Fragile)
  (holdsDuring (WhenFn ?X)
     (orientation ?CHARACTER ?OBJECT Near))
 (exists (?DAMAGE)
  (and
   (instance ?DAMAGE Damaging)
   (subProcess ?DAMAGE ?X)
   (agent ?DAMAGE ?CHARACTER)
   (patient ?DAMAGE ?OBJECT))))
```

As the representation shows, SUMO is quite capable of encoding temporal properties of events with its temporal qualification. However, inferencing with rules involving time relations between events is currently not supported by Sigma (Corda et al, 2008). Nevertheless, efforts are underway to perform true higher-order logical inference (Sutcliffe et al, 2009).

The next step involves deriving axioms to represent the different ways in which an object can be damaged depending on its attribute, for example, fragile objects can break while paper-based objects such as books and paintings can be torn. Consideration must also be made to determine if a damage is an accident or intentional.

3.4 Conceptualizing Story Themes

Themes can also be mapped to SUMO as thematic predicates, and the story planner can identify a theme either based on the first action that was performed, or based on user selection. In the latter case, when Sigma returns all possible actions, the planner can choose one based on the theme.

4 System Architecture

The architecture of SUMOs, shown in Figure 2, has two main modules, the Story Editor and the Story Planner, both of which interact with Sigma[3] to retrieve story facts from the SUMO ontology as well as to assert new axioms representing the developing story plan back to SUMO.

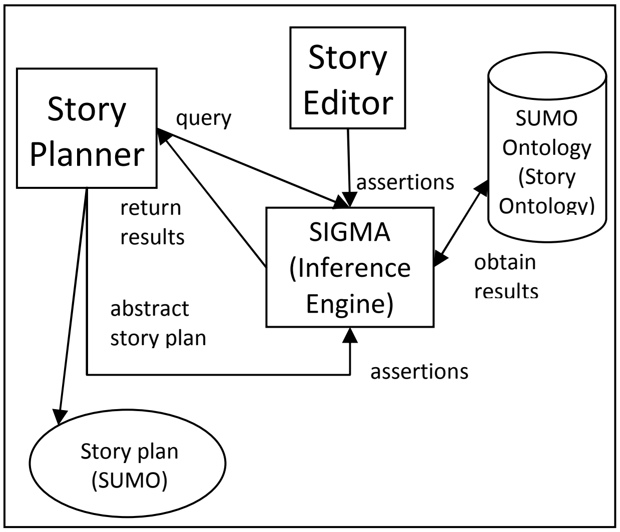

Figure 2. Architecture of SUMOs

The Story Editor handles the generation of assertions corresponding to the input picture elements specified by the user.

The Story Planner is responsible for planning the flow of events in the story. It uses a meta-knowledge about children's story comprising of five phases – introduction, problem, rising action, solution, and climax. The planner determines and phrases the queries that are sent to Sigma and generates additional axioms based on the query results in order to expand the story plan. The generated axioms are asserted back to Sigma for inclusion in the SUMO ontology to be used again for further inferencing.

Queries sent to Sigma can be classified into three categories. Concept-based queries concern classes and instances, and are used to determine direct and indirect subclass and class-instance relationships while relation-based queries infer knowledge by considering transitivity, symmetry and inversion of relations (Corda et al, 2008). Action-based queries identify a set of actions based on the current world state to drive the story. A fourth category, time-event queries, currently not supported by Sigma, should reason about temporal and event-based specifications.

The interaction between the Story Planner and Sigma in Figure 2 raises an issue of search control. In Picture Books and SUMOs, information that guides the story planning can be *bottom-up*, i.e. the actions and events are determined based on what is possible within the story ontology, e.g. through the various capability axioms, or *top-down*, i.e. actions are selected based on Machado's narrative subplot knowledge. Currently, the Story Planner is responsible for managing the process. However, if both these sources of knowledge and constraints can be represented in first-order logic, the search control of the story planning process can be recast as a theorem proving task, i.e. one that searches for a proof that satisfies all constraints. This is a future research direction.

The following section presents a more detailed trace of system operation and the contents of a story plan in first-order logic.

4.1 Generating Story Plans

The first part of the story plan contains assertions to represent the initial elements of the story. Using the story in Table 1 as an example, lines 1 to 6 below assert the main child character and her parent, while lines 7 to 8 assert the background and the object, respectively.

```
1>  (instance Rabbit1 RabbitCharacter)
2>  (attribute Rabbit1 Child)
3>  (attribute Rabbit1 Female)
4>  (instance Rabbit2 RabbitCharacter)
5>  (attribute Rabbit2 Adult)
6>  (attribute Rabbit2 Female)
7>  (instance LivingRoom1 LivingRoom)
8>  (instance Lamp1 Lamp)
```

The next step involves initializing the locations of these story elements. Currently, it is setup that all objects would be situated in the background and the first child character would always be near the first object, as shown in the assertions below.

```
9>  (located Rabbit1 LivingRoom1)
10> (located Lamp1 LivingRoom1)
11> (orientation Rabbit1 Lamp1 Near)
```

This, however, creates the assumption that the child character is already in the location near objects which he will interact with, which may not

[3] Sigma Knowledge Engineering Environment, http://sigmakee.sourceforge.net

necessarily be true and reduces the flexibility of the system. In order to create more varied stories, the initial location can be identified based on the theme and the first event that the user would want to likely happen in the story.

From the initial set of assertions, the story planner issues its first concept-based query to Sigma with "(name Rabbit1 ?X)" to determine a name for the main character, *Rabbit1*, and receives "*Rizzy*" as a result. This is asserted to the story plan as:

12> (name Rabbit1 "Rizzy")

The next query is the first action-based query used to determine the first action to start the story flow. Given "(capability ?X experiencer Rabbit1)", which is intended for identifying the set of possible starting actions that the main character, *Rabbit1*, can perform with the object in the background, Sigma returns the following list (assuming the story facts given in the previous section):

X = [RecreationOrExercise, Looking,
Maintaining, Poking]

Assuming the planner selects *RecreationOrExercise*, the following assertions are then added to the story plan:

13> (instance RecOrEx1 RecreationOrExercise)
14> (agent RecOrEx1 Rabbit1)

At this point, the introduction phase of the story plan has been completed. The problem phase begins with a query to identify any instances of problems that can occur, i.e. "(instance ?X Damaging)". Damaging the object *lamp* causes its attribute to be changed, and again we query Sigma for this change of state with "(attribute Lamp1 ?X)" yielding the result *broken*, and the corresponding emotional state of the character "(attribute Rabbit1 ?X)". The following assertions were added to the plan:

15> (instance (sk0 Rabbit1 Lamp1
RecOrEx1) Damaging)
16> (attribute Lamp1 Broken)
17> (attribute Rabbit1 Anxiety)

While a full explanation of skolemization is not possible here for space reasons, we note that the second argument of assertion #15 (derived from Sigma's answer to the query) stands for the existence of an unnamed term, in this case, that there is an instance of a *Damaging* process. The agent (*Rabbit1*), patient (*Lamp1*), and the action (*RecOrEx1*) that caused the problem were all provided in the query result.

4.2 Generating Surface Text

SUMO-based story plans provide a form of *interlingua* where story details are represented in logical form. The logical representation allows generation of the same story in different languages (that are connected to WordNet). Sigma already has a language generator, with templates for English, and an initial set for Tagalog (Borra et al, 2010). Work is currently underway to enhance the existing language generator in Sigma and make the generated text more natural. Sigma can then be used to generate stories automatically from the knowledge asserted in the story generation process.

5 Conclusions and Further Work

The paper presented a preliminary work aimed at representing storytelling knowledge in SUMO and using Sigma as inference engine to assist the planner in generating story plans. Further work focuses on modeling the emotional state of the character as a result of some event (e.g., feeling worried, guilty or scared due to causing some problems in the world state), changes in character traits as the story progresses (e.g., from negative trait to positive trait as the story flows from rule violation to value acquisition), and enhancing the representation for story themes. Once a set of knowledge has been developed, these should be evaluated systematically through validation of the rules for logical consistency with the theorem prover. A future goal is to apply the metrics proposed by Callaway & Lester (2002) in StoryBook to evaluate with actual users if the generated stories are better and more varied as compared to that of Picture Books.

Although SUMO is quite capable of representing time and sequences, reasoning with temporally qualified expression is challenging for any theorem prover. The works of (Sutcliffe et al, 2009) to extend the inference engine to handle reasoning over temporal relations should be explored further to allow SUMOs to generate story plans that consider temporal relations between actions and events.

Finally, story generators will benefit its readers if the generated stories are narrated orally. SUMOs can be explored further to model various emotions to provide annotations in the surface story text which will then be fed to a text to speech tool for speech generation.

References

Borra, A., Pease, A., Roxas, R. and Dita, S. 2010. Introducing Filipino WordNet. In: *Principles, Construction and Application of Multilingual Wordnets: Proceedings of the 5th Global WordNet Conference*, Mumbai, India.

Callaway, C. B., and Lester, J. C. 2002. Narrative Prose Generation. *Artificial Intelligence*, 139(2):213-252, Elsevier Science Publishers Ltd., Essex, UK.

Corda, I., Bennett, B., and Dimitrova, V. 2008. Interacting with an Ontology to Explore Historical Domains. *Proceedings of the 2008 First International Workshop on Ontologies in Interactive Systems*, 65-74, IEEE Computer Society.

Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. 2003. *AI Magazine*, 24(3):13-24, Association for the Advancement of Artificial Intelligence.

Kooijman, R. 2004. De virtuele verhalenverteller: voorstel voor het gebruik van een upper-ontology en een nieuwe architectuur. *Technical Report*. University of Twente, Department of Electrical Engineering, Mathematics and Computer Science.

Hong, A., Solis, C., Siy, J.T., and Tabirao, E. 2008. *Picture Books: Automated Story Generator*. Undergraduate Thesis, De La Salle University, Manila, Philippines.

Lenat, D.B. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure, *Communications of the ACM*, 38(11).

Liu, H. and Singh, P. 2004. Commonsense Reasoning in and over Natural Language. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 293-306, Wellington, New Zealand, Springer Berlin.

Machado, J. 2003. Storytelling. In *Early Childhood Experiences in Language Arts: Emerging Literacy*, 304-319. Clifton Park, N.Y., Thomson/Delmar Learning.

Niles, I. and Pease, A. 2001. Towards A Standard Upper Ontology. *Proceedings of Formal Ontology in Information Systems* (FOIS 2001), 2-9, October 17-19, Ogunquit, Maine, USA.

Noy, N. and McGuinness, D. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* and *Stanford Medical Informatics Technical Report SMI-2001-0880*, March 2001.

Ong, E. 2009. Prospects in Creative Natural Language Processing. *Proceedings of the 6th National Natural Language Processing Research Symposium*, De La Salle University, Manila, Philippines.

Pease, A. 2006. Formal Representation of Concepts: The Suggested Upper Merged Ontology and Its Use in Linguistics. *Ontolinguistics. How Ontological Status Shapes the Linguistic Coding of Concepts*. Schalley, A.C. and Zaefferer, D. (ed.), Vorbereitung Berlin, New York.

Pease, A. 2003. The Sigma Ontology Development Environment. *Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems*, vol. 71 of CEUR Workshop Proceeding series.

Riedl, M. and Young, R.M. 2004. An Intent-Driven Planner for Multi-Agent Story Generation. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 186-193, Washington DC, USA, IEEE Computer Society.

Smith, B. 1998. The Basic Tools of Formal Ontology. *Formal Ontology in Information Systems*, Nicola Guarino (ed), IOS Press, Washington. *Frontiers in Artificial Intelligence and Applications*, 19-28.

Solis, C., Siy, J.T., Tabirao, E., and Ong, E. 2009. Planning Author and Character Goals for Story Generation. *Proceedings of the NAACL Human Language Technology 2009 Workshop on Computational Approaches to Linguistic Creativity*, 63-70, Boulder, Colorado, USA.

Sutcliffe, G., Benzmüller, C., Brown, C.E., and Theiss, F. 2009. Progress in the Development of Automated Theorem Proving for Higher-order Logic. *Automated Deduction*, *22nd International Conference on Automated Deduction*, Montreal, Canada, August 2-7, 2009. *Proceedings of the Lecture Notes in AI*, vol. 5663, 116-130, 2009, Springer.

Swartjes, I. 2006. *The Plot Thickens: Bringing Structure and Meaning into Automated Story Generation*. Master's Thesis, University of Twente, The Netherlands.

Theune, M., Nijholt, A., Oinonen, K., and Uijlings J. 2006. Designing a Story Database for Use in Automatic Story Generation. *Proceedings 5th International Conference Entertainment Computing*, Cambridge, UK. Lecturer Notes in Computer Science, 4161:298-301, Heidelberg, Springer Berlin.

Uijlings, J.R.R. 2006. *Designing a Virtual Environment for Story Generation*. MS Thesis, University of Amsterdam, The Netherlands.

Venour, C. and Reiter, E. 2008. *A Tutorial for Simplenlg*. http://www.csd.abdn.ac.uk/~ereiter/simplenlg

WordNet. 2006. *WordNet: A Lexical Database for the English Language*. Princeton University, New Jersey.