

Minstrel Reloaded: From the Magic of Lisp to the Formal Semantics of OWL

Federico Peinado and Pablo Gervás

Departamento de Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid
Profesor José García Santasmases s/n, 28040 Madrid, Spain
email@federicopeinado.com, pgervas@sip.ucm.es

Abstract. This paper is a review of a story generation system called Minstrel. It uses complex but hand-crafted Lisp knowledge structures to generate short computer-generated stories within the King Arthur domain. The knowledge representation model of Minstrel is reimplemented using a W3C standard language to analyze the pros and cons of technology updates over this kind of classic AI projects.

Keywords. Story Generation, Story Representation, Computational Narratology, Semantic Technologies

1 Introduction

There are many reasons for revisiting the classics. One is to learn from them, another one is to revive them bringing their old ideas face to face with new trends and technologies. The last one considered in this paper is to open a discussion for fair comparisons between their achievements and the results of our up-to-date –technologically speaking– research projects.

As a first step to these goals, Minstrel, the classic story generator, has been partially reimplemented in order to transfer the original Lisp-based knowledge representation of the application to a more accurate and interoperable framework based on W3C formal semantics. The purpose of this experiment is to give insight into the issues of updating frame-based projects to formal semantics of OWL and the specific problems arising for Minstrel as a specific computational narrative model.

2 Minstrel Overview

Minstrel was developed in the late 1980s by Scott R. Turner. It is a stand-alone application for the automatic generation of short tales (around 200 words in English) set in the universe of King Arthur and the Knights of the Round Table. Focusing on creative heuristics and well known problem-solving methods, Turner presents exciting results claiming that simple but powerful reasoning mechanisms manipulating big amounts of knowledge will lead us to more intelligent systems.

Unfortunately, there is no working version of Minstrel on the web, but the available documentation is meticulous enough to deeply study many details of the implementation.

The knowledge representation is an extension of a Lisp library called Rhapsody. It uses frames, schemas with slots and facets which represent story themes or morals, dramatic effects (suspense, foreshadowing, pacing, dialogue...), world states, character's beliefs and affects, etc.

Minstrel uses a Case-Based Reasoning approach to create new stories reusing parts of old stories taken from an episodic memory. The main goals of the systems are divided in four categories: story theme, plot consistency, dramatic narration and linguistic presentation. Planning Advice Themes (PATs) are more specific goals designed for generalize, specialize, mutate and recombine story themes while the system is running.

The fictional world of the characters is modeled using three concepts: Goal, Act and State. Goals and Acts are divided in two categories: Character-Level and Author-Level, this last one oriented to high-level task concerning the narrative structure of the result. Goals have priority numbers (from 0 to 100) for the planner to put them in order.

Transform-Recall-Adapt Methods (TRAMs) are used to mutate the story in order to obtain creative results.

To sum up¹, the winding network of linked individuals belonging to different knowledge domains is rich, but it is ruled by a constellation of untyped semantic operators and empirically-defined processing algorithms. There are also "ghost individuals" as Planner, Other, Someone or Elsewhere, that Minstrel uses for their internal planning operations.

The assumption is that, from the point of view of Knowledge Engineering, such a knowledge base is hard to maintain, evolve and reuse, even for the same engineer who took all decisions during the development of the system. That is the main reason for recoding a new version using a formal methodology.

3 OWL Overview

The proposal of the W3C to formalize ontologies for interoperability on the web is OWL (Ontology Web Language [1]). The goal of this standard is to formalize the semantics that was created *ad hoc* in old frame systems and semantic networks.

OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. OWL Full is powerful for representing complex statements but not useful for reasoning with them due to their computational properties. OWL DL is the subset of OWL designed for applications that need the maximum expressiveness without losing computational completeness and decidability. It is based on Description Logics, a particular fragment of first order logic, in which concepts, roles, individuals and axioms that related them (using universal and existential restrictions, negation, etc.) are defined.

¹ Readers can find more theoretical and technical details in Turner's PhD thesis [4].

OWL DL has support for polihierarchical reasoning and automatic classification what means knowledge engineers can develop an ontology in a more natural way (e.g. creating a simple taxonomy in which concepts one parent as maximum) and after that let the system put things in order automatically.

Minstrel Reloaded aims for intensive reasoning capabilities and that is the reason why OWL DL is chosen.

4 A List of Knowledge Reengineering Incidents

This section presents a list of incidents detected when transferring Turner’s annotation to the new ontology and knowledge base implementation.

- *Lack of canonical nomenclature.* Identifiers of schemas in Minstrel vary between adjectives and names. Singular and plural names both are used to represent concepts. Lowercase and uppercase letters are used with no clear criteria in the nomenclature, because there are inconsistencies along the identifiers used in the examples of Turner’s thesis.
- *Lack of modularization.* Schemas of totally different domains are put all together at the same level, due to unexisting prefixes, suffixes, namespaces or separated files to identify each domain.
- *Reification of types/Proprietary “is-a” relationship.* Instead of assigning a primitive concept to an individual to represent its type, Minstrel does an extensive use of reification, creating an individual for each type of goal, act or state. A different interpretation of this phenomenon is to consider that Minstrel is using a proprietary “is-a” relationship to assign concepts to individuals, which is usually an embedded relationship in any formal knowledge representation system (e.g. John –individual– “is-a” Male –concept–). Indeed OWL DL does not allow any relation between concepts and individuals apart from the built-in “is-a” relationship. Minstrel has to manage some “back-links”. These are called “inverse roles” in the terminology of OWL and they are included as a standard feature in the W3C language. “Connections” in Minstrel are special roles which establish a link between *another roles*. This kind of “metaroles” is not allowed in DL.
- *Consistency checking.* DL reasoners can check consistency of any DL model with restrictions automatically. On the other hand Minstrel requires a specific library and a set of plans exclusively developed for that task. Both systems can check consistency automatically but in Minstrel this is application-specific and handcrafted whereas using DL ensures that this does not have to be implemented.
- *Untyped roles.* Because of the Lisp features, relationships in Minstrel have no types, additional semantics, domain or range. By default DL roles can link any individuals of role domain with any individual of the role range, but designers can define roles as functional, symmetric, transitive, etc. There is also no taxonomy of roles (e.g. goalType, actType and stateType should be classified under a more generic “type” role). This presents serious difficulties

- for visualization and organization due to the number of current roles: 56. Untyped roles make the system more error prone and more difficult to debug.
- *Overridden concepts*. “The values and meanings of the slot fillers for state schemas vary depending upon the sub-type of states” [4] (p. 30). In Minstrel a specialization of a concept can have a different structure from its parent, and it can also change the meaning of inherited slots according to the values found in other slots, as it happens in the “Decision” slot of a Plan Advise Theme, for instance. This behaviour is not tolerated in OWL.
 - *Unclear polihierarchy*. It is not clear if there is a polihierarchy of concepts in Minstrel or not. According to the different taxonomies presented in the documentation, there are some special cases of individuals belonging to more than one concept. Goals and Acts are divided in Author-Level Goals/Acts and Character-Level Goals/Acts. Due to the polihierarchical reasoning, there is no need for artificial duplication of concepts in the OWL taxonomy.
 - *Anonymous individuals*. Planner, Other, Someone or Elsewhere are individuals that Minstrel uses for their planning operations. Thanks to the Open World Assumption in OWL DL, it is not necessary to put a name to these individuals for their existence to be inferred by the system.
 - *Defined concepts*. Minstrel’s concepts are not defined with axioms or restrictions. Talking about violent Goals, Turner gives a definition like this: “a violent Goal is a Goal that has a violent Act as part of its plan”. This informal definition matches quite well with the idea of “defined concepts” in OWL DL, the definitions of which are used by DL reasoners to classify automatically concepts or individuals according to their relationships.

5 Discussion

Minstrel Reloaded is just a reimplementaion of the main knowledge schemas of the original system in a modern ontology and knowledge base. Every concept of the original system and some basic individuals of the King Arthur domain have been implemented. The rest of scenes, stories, plans and other individuals of the knowledge base are not implemented because they are not listed exhaustively in the book of Turner. To evaluate the whole system is not possible yet because the processes that deals with this knowledge have not been developed, so in this section it follows a discussion on knowledge representation and the practical results.

The OWL DL files for the basic ontology² includes 65 concepts and 56 roles, the knowledge base³ has 99 individuals.

Minimum indispensable changes have been required for implementing the new version of Minstrel’s knowledge base. Concepts and individuals have been separated. Concepts are put together in the ontology (what we have called *MinstrelOnto*) and individuals have been placed in the knowledge base (*MinstrelKB*). The “minstrel” namespace has been created. The priority of goals is represented

² <http://federicopeinado.com/projects/minstrel/onto/MinstrelOnto.owl>

³ <http://federicopeinado.com/projects/minstrel/onto/MinstrelKB.owl>

as a datatype property, indeed a simple integer because its value cannot be trivially restricted to an interval in OWL DL, as it was in the original system.

The advantages of standard semantics are revealed when creating and editing the ontology and the knowledge base with powerful tools as Protégé-OWL[2], sharing them on the Internet or validating and classifying the model using a DL reasoner as Pellet [3].

6 Conclusions

A partial reimplementaion of the knowledge structures of Minstrel, the classic story generator, has been created, focusing on knowledge representation. Several problems have been found during the process, most of them concerning lack of reusability related to the lack of formal semantics of complex applications written in Lisp. There are also problems related to the artificial restrictions of OWL DL, which are justified by theoretical requirements for efficient and sound DL reasoning.

One of the design problems that Turner found as a knowledge engineer was the construction of extensible taxonomies of concepts. Discipline and methodology are still requirements for this task, but with the help of automatic validators, classifiers and ontology engineering tools it is reasonable to envisage a future in which systems like Minstrel can be developed by a heterogeneous team of domain experts without the need of such an enormous effort and sacrifice.

Next steps go through the improvement of this resource (currently shared on the web) for being explored and used for a wider audience, developers of narrative applications and even research communities working in the field of interactive storytelling.

Acknowledgements

This research is funded by the Spanish Ministry of Education and Science (TIN2005-09382-C02-01 and TIN2006-14433-C02-01 projects), Complutense University of Madrid and the G.D. of Universities and Research of the Community of Madrid (UCM-CAM-910494 research group grant).

References

1. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and A. Stein. OWL web ontology language reference. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
2. M. Crubézy, O. Dameron, R. W. Fergerson, H. Knublauch, M. A. Musen, N. F. Noy, D. Rubin, S. W. Tu, and J. Vendetti. Protégé project. <http://protege.stanford.edu/>.
3. Mindswap. Pellet OWL reasoner. <http://www.mindswap.org/2003/pellet/index.shtml>.
4. S. R. Turner. Minstrel: A computer model of creativity and storytelling. Technical Report UCLA-AI-92-04, Computer Science Department, University of California, 1992.