A Top-Down Design Methodology based on Causality and Chronology for Developing Assisted Story Generation Systems

Carlos León

Dpto. de Ingeniería del Software e I. Artificial
Universidad Complutense de Madrid
cleon@fdi.ucm.es

Pablo Gervás

Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
pgervas@sip.ucm.es

ABSTRACT

Assisted story generation systems do not include automatic generation of content due to the fact that creating generation algorithms is a challenging task, usually carried out in research projects on Artificial Intelligence. This paper proposes a top-down design methodology in which the knowledge representation is based on causality and chronology. The proposed methodology partially eases the process by dividing the narrative generation system in two parts: a generic engine and a domain definition based only on a specific set of predicates. The theoretical model and an implemented case study with resulting stories are presented.

Keywords

Story generation, top-down design, assisted creativity, artificial intelligence, knowledge representation

INTRODUCTION

There exist many methods for writing stories by hand, basically oriented to train storytellers. These methods basically give more or less generic rules to engage an audience or facilitate the understanding of the message being told. However, while these influence or improve the creativity of the generated narrative content, they fully rely on the effort of the storyteller and do not use computers.

In this paper we propose a design methodology for story generation systems. This methodology tries to offer a more general approach to story generation in such a way that the domain dependent knowledge and the generation engine are clearly isolated. The set of high-level relations that must be defined as the domain definition for the engine to work is kept to a minimum. In this way, the maintainability of the knowledge base in particular and the generation system in general is improved. The top-down approach makes it simpler to solve the growth of the knowledge base in the top levels which, by definition, are more intensely defined and thus less susceptible to changes at a large scale.

OTHER STORY GENERATION SYSTEMS

TALE-SPIN tackles story generation as a problem solving process [1]. This design proposes a basic core engine reasonably independent from the set of actions that can be performed. Grammar-based story generation systems also propose a top-down approach for the design of the generation systems [2,3]. They do not scale well because the dependency of the added knowledge with the rest of the knowledge base is not well managed. MINSTREL [4] domain-specific knowledge with TRAMS, which are base cases of narrative schemas that can be adapted to fit a new story. MEXICA [5] introduced tensional guidelines, which are typical progressions of some emotional variable in time. This has also inspired part of this work, as explained later. BRUTUS' engine consists on a large knowledge base that encodes a significant amount of common sense knowledge and narrative [6]. While this is powerful in principle, it suffers from the knowledge acquisition bottleneck and the inclusion of additional knowledge heavily restricts the improvement of the system. Another knowledge-intensive system is presented by León and Gervás in a story generation system based on evaluation and selection of stories [7].

DOMAIN INDEPENDENT PART OF THE STORY GENERATION ENGINE

One of the objectives of this research is to improve the ratio between the effort needed to improve the generation capabilities and the number of different stories that can be generated. In this research we have created a top-down design pattern for defining a domain definition without affecting the basic generation engine. This means that a new domain definition can be used to generate new types of stories without having to rewrite the engine. Additionally, the top-down definition makes it a bit easier to define the domain, as it will be explained. The basic engine relies on 5 relations that must be instanced for each implementation.

- *requirement* – this relation links a set of facts that must be true in a partial story with another one that requires them to be true.
- *cause* – this relation pairs a set of event descriptions with another one that *could* be caused by this set.

- *provoke* – this relation represents a stronger version of causality. When something is provoked, it means that it must happen in the story.
- *simultaneous* – this relation sets whether a fact can take place during the same interval as others or not.
- *duration* – this basic relation informs the basic engine about the basic duration of an event description.

These relations are defined in a state-independent way. This means that the domain-independent part of the engine expects a definition of these relations as generic rules that do not get instanced themselves, that is, the definition of the relations do not have access to the partial story. The core engine is responsible for finding a relation that helps to add a new event description to the partial story and for instantiating the variables.

Core Generation Engine

The proposed core engine performs story generation as a generic state space search problem. The story itself represents the state, and it is updated as the story is completed. According to this, the final objective is to generate a story that fulfils the user requirements, that is, to reach a state in which all user's constraints are satisfied. How these constraints are set is explained later. For the sake of understandability, it can be assumed that the user's constraints are a set of rules.

TOP-DOWN DEFINITION OF THE DOMAIN-DEPENDENT PART OF THE STORY GENERATION ENGINE

Since the relations are so high-level, definition of complex instantiations of the relations can be too complex. This is where the top-down approach shows its benefits. Instead of trying to describe a full computational representation of the content needed for creating a knowledge base for the generators, this research proposes a way to define the domain specific knowledge in a simpler, more straightforward way. This is possible due to the fact that a general scope knowledge base is not being created, but a knowledge base for narrative plot generation oriented towards assisted creativity. By restricting the range of application strong assumptions can be made, and this helps to reduce the amount of hand-input knowledge. The 5 relations defined in the previous section can be made more specific in many different ways. Instead of implementing the relations directly with rules, they can be implemented with more abstract definitions. These definitions help to manage the reduction of maintainability of a field-specific domain base. The relations can be further defined in terms of more specific relations.

DRIVING THE GENERATION THROUGH CURVES OF DESIRED VALUES AND RULES

Since the presented design is meant to ease the development of assisted creativity story generation systems, some means to let the user set what she wants the system to generate are proposed. Three different, complementary methods are considered in the model:

- A set of objectives for generation defined as logic rules. This set contains the characteristics that the final story must have.
- A set of constraints defined as logic rules. These constraints describe what the user does not want to be in the story.
- A set of curves that represent the sequential evolution of some set of user defined and domain dependent variables in time.

CONCLUSION

Automated story generation systems can reduce the amount of needed work in storytelling when used as an assisting tool, and this can help to spend the effort on the creative part. Since developing this kind of tools requires a complex handling of Artificial Intelligence techniques, this paper proposes a methodology to ease the creation of story generators. Several assumptions that can be made because of the constrained domain (narrative) help to restrict the constraints of the problem so that a simpler definition is possible. In this paper a detailed description of the model has been presented.

ACKNOWLEDGMENTS

Research funded by Spanish Ministry of Science and Innovation's TIN2009-14659-C03-01 and GR35/10-A (group 921332).

REFERENCES

1. Meehan, J. (1976). The Metanovel: Writing Stories by Computer. PhD thesis, Yale University.
2. Lang, R. R. (1999). A declarative model for simple narratives. In Proceedings of the AAAI Fall Symposium on Narrative Intelligence, pages 134–141. AAAI Press.
3. Rumelhart, D. E. (1975). Notes on a schema for stories. Representation and Understanding: Studies in Cognitive Science, pages 211–236.
4. Turner, S. (1992). MINSTREL: A Computer Model of Creativity and Storytelling. PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA.
5. Pérez y Pérez, R. (1999). MEXICA: A Computer Model of Creativity in Writing. PhD thesis, U. of Sussex.
6. Bringsjord, S. and Ferrucci, D. (1999). Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a StoryTelling Machine. Lawrence Erlbaum Associates, Hillsdale, NJ.
7. León, C., Gervás, P. Towards a Black Box Approximation to Human Processing of Narratives based on Heuristics over Surface Form. AAAI 2010 Fall Symposium on Computational Models of Narrative. Arlington, Virginia, pp. 31-38.