

# Using CNL for Knowledge Elicitation and Exchange across Story Generation Systems

Eugenio Concepción<sup>1</sup>, Pablo Gervás<sup>2</sup>, Gonzalo Méndez<sup>2</sup>, and Carlos León<sup>1</sup>

<sup>1</sup> Facultad de Informática, Universidad Complutense de Madrid

<sup>2</sup> Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid  
{econcepc,pgervas,gmendez,cleon}@ucm.es

**Abstract.** Story generation is a long standing goal of Artificial Intelligence. At first glance, there is a noticeable lack of homogeneity in the way in which existing story generation systems represent their knowledge, but there is a common need: their basic knowledge must be expressed unambiguously to avoid inconsistencies. A suitable solution could be the use of a controlled natural language (CNL), acting both as an intermediate step between human expertise and system knowledge and as a generic format in which to express knowledge for one system in a way that can be easily mined to obtain knowledge for another system – which might use a different formal language. This paper analyses the suitability of using CNLs for representing knowledge for story generation systems.

**Keywords:** CNL, Story Generation, Knowledge Representation

## 1 Introduction

Natural languages allow human communication and knowledge transmission, and they provide an unbeatable expressiveness for concept modelling and structuring. However, for the same reasons, they are substantially complex for automatic processing.

Controlled Natural Languages (CNLs) can be considered as a balance between the expressiveness of the natural languages and the need for a formal representation that can be handled by a computer. A CNL is an engineered subset of natural languages whose grammar and vocabulary have been restricted in a systematic way in order to reduce both ambiguity and complexity of full natural languages [1].

Against this background, CNLs are attractive because of two reasons: first of all, since they are subsets of natural languages, they are naturally easier to write and understand by humans than formal languages; secondly, they can be translated automatically (and often deterministically) into a formal target language and then be used for automated reasoning [1]. CNLs offer an additional advantage: unlike formal languages that require some degree of consensus concerning their syntax, a CNL should be more suitable for different teams to understand each other and therefore to more easily conclude an agreement. Of course, the application of formal languages such as XML, JSON or RDF formats is easy to

achieve when considering interchange formats between systems that have established an agreed data model. This is not normally the case between systems that address storytelling from different perspectives, which are likely to have radically different data models. Interchange formats such as XML, JSON, or RDF will require a different translation procedure to convert knowledge built based on one data model to knowledge built on a different data model. Converting a given data model to and from a common CNL would allow every system to make their knowledge available to all other systems for which such a translation procedure is available.

Story generation systems are a form of expression for computational creativity. Using the words of Gervás [2], a story generator algorithm (SGA) refers to a computational procedure resulting in an artifact that can be considered a story. The term story generation system can be considered as a synonym of storytelling systems, that is, a computational system designed to tell stories.

## 2 Related Work

Storytelling systems require the representation and manipulation of large amounts of knowledge. This involves not only the product itself – stories represented at various levels of detail – but also the knowledge resources that are required to inform the construction processes. This section explores some of the aspects that need to be represented and some examples of how controlled natural language might be applied in specific cases.

The context in which the proposal presented in this paper occurs involves: the complexity of knowledge representation required for story generation systems, the already proven suitability of CNL in storytelling systems, the difficulties in eliciting the required knowledge, and existence of storytelling systems that already contemplate automated transformations across different representation formats as an integral part of their functionality.

### 2.1 Knowledge Representation in Storytelling Systems

There are multiple dimensions when considering knowledge representation for story generation. Gervás and León [3] provided a list of the most relevant classifications, and proposed their own list of suitable dimensions obtained from the different aspects of a narrative: discourse, simulation, causality, character intention, theme, emotion, authorial intention, and narrative structure.

From a historical perspective, formal languages have been the most common way of knowledge representation. The reason for using formal languages is simplicity: they have a well-defined syntax, an unambiguous semantics and are very convenient for automated reasoning. Particularly, in the field of automatic story generation, there is an abundance of examples of this kind.

TALE-SPIN [4] is one of the earlier story generators that produced stories about the inhabitants of a forest. It was a planning solver system that took as inputs a collection of characters with their corresponding objectives, found a

solution for characters goals, and finally wrote up a story narrating the steps performed for achieving those goals. TALE-SPIN knowledge representation relied on Conceptual Dependency Theory [5]. TALE-SPIN output can be defined as a trace through a problem-solving process where the problems were limited to a specific area of knowledge, named the problem domain, which was defined by a set of primitives, a set of goal states or problems, and procedures for achieving these goals. All this knowledge was expressed as a formal language.

Minstrel [6] was a story generation system that told stories about King Arthur and his Knights of the Round Table. Its building units were a collection of goals and the plans to satisfy them. Story construction in Minstrel operated as a two-stage process involving a planning stage and a problem-solving stage which reused knowledge from previous stories. The knowledge representation in Minstrel used an extension of a Lisp library called Rhapsody, a tools package that provided the user with ways to declare and manipulate simple frame-style representations, and a number of tools for building programs that used them. Minstrel used Rhapsody for defining frames, schemas with slots and facets which represent story themes or morals, dramatic effects, world states, characters beliefs and affects.

Mexica [7] was a computer model designed to generate short stories about the early inhabitants of Mexico. It used several knowledge structures for supporting its storytelling model: an actions library, a collection of stories for inspiring the new ones, and a group of characters and locations. The story generation process took as input a file of primitive actions for creating an in-memory data structure after processing. It also created additional structures by transforming the file of **Previous Stories** into the **Concrete, Abstract and Tensional Representations**. The data structure built by the initial step was called **Primitive Actions Structure**, and it served as a repository for the primitive actions, which consists of an action name and several sets representing characters and their circumstances. Relations in Mexica representing emotional links and tensions between characters were modelled by means of formal languages in terms of three attributes: type (love or friendship), valence (positive or negative) and intensity. Mexica knowledge base also contained stories created by humans representing well-formed narratives, expressed as action sequences.

MAKEBELIEVE [8] was a short fictional story generation system that used a subset of common sense from the ontology of the **Open Mind Common Sense Knowledge Base** [9] for describing causality. Binary causal relations were extracted from these sentences and stored as crude trans-frames. MAKEBELIEVEs original knowledge base has been continued subsequently by the **Open Mind Common Sense ConceptNet** [10]. A trans-frame [11] is a type of diagram used for representing the common information related to an action. Minsky used the Trans primitives from **Conceptual Dependency Theory** [5] as inspiration for trans-frame concept. Hence, these data structures can be used for representing a stereotyped situation.

## 2.2 Use of CNL in Storytelling Systems

There is not a long record of uses of CNL in the context of storytelling.

Inform [12] was a toolset for creating interactive fiction. From version 7 on, Inform provided a domain-specific language for defining the primary aspects of an interactive fiction like the world setting, the character features, and the story flow. The provided domain-specific language used a CNL, similar to Attempto Controlled English [13].

The StoryBricks [14] framework was an interactive story design system. It provided a graphical editing language based on Scratch [15] that allowed users to edit both the characters features and the logic that drove their behaviour in the game. By means of special components named story bricks, users could define the world in which characters live, define their emotions, and supply them with items. Story bricks were blocks containing words to create sentences in natural language when placed together. They served to define rules that apply under certain conditions during the development of the story in the game.

In the extended ATTAC-L version [16], authors introduced a model which combined the use of a graphical Domain Specific Modeling Language (DSML) for modelling serious games narrative, ATTAC-L [17], with a CNL to open the use of the DSML to a broader range of users, for which they selected Attempto Controlled English [13]. It allows describing things in logical terms, predicates, formulas, and quantification statements. All its sentences are built by means of two word classes: function words (determiners, quantifiers, negation words, etc.) and content words (nouns, verbs, adverbs and prepositions). The main advantage is that Attempto Controlled English defines a strict and finite set of unambiguous constructions and interpretation rules.

## 3 Knowledge Elicitation for Storytelling Systems

Storytelling systems are extremely knowledge hungry. Generated stories are only as good as the knowledge they have been derived from. Given the thirst for knowledge of story generation systems, knowledge elicitation has always been a significant concern for researchers in this area.

Recent attempts have been made to address this problem via crowdsourcing [18]. In this work, a number of human authored narratives are mined to construct a *plot graph*, which models the author-intended logical flow of events in the virtual world as a set of precedence constraints between plot events. Knowledge is refined from a set of typical narratives in natural language on a given topic collected via Amazon Mechanical Turk (AMT). The crowd workers are required to follow a simplified grammar and a number of restrictions that resemble closely a CNL. These narratives are parsed and merged into a combined representation in terms of plot graph for the domain being explored, which is later used to inform the process of constructing narratives.

In this regard, the Genesis system [19] can also be considered as an interesting example of knowledge mining from stories written in simplified English. Genesis

was developed for studying the story understanding process, including the human ideological bias. It takes as inputs the text in Genesis English and a set of constraints representing the cultural and ideological context of the human reader that it will emulate. As a result, the system builds a graph-based representation using common sense rules. This knowledge structure allows the system not only to analyse problems, but also to answer questions and generate conclusions.

To inform the development of the Dramatis system for modelling suspense [20], O'Neill carried out an effort of knowledge engineering driven by methods adapted from qualitative research. The goal was to collect typical reader genre knowledge while simultaneously limiting engineer bias. The process was to acquire a corpus of natural language text and the conversion of that corpus into the knowledge structures required by Dramatis.

Although no CNL were strictly used for these tasks, the potential for their use in this context is clear.

### **3.1 Transformation across Representations in Storytelling Systems**

Gervás [21] attempts to model the procedures for composing narrative discourse from non-linear conceptual sources. It establishes algorithmic procedures for constructing a discourse – characterised by being a linear sequence of statements – to describe a set of facts known to have happened – which may involve events affecting a number of characters at different locations on overlapping periods of time. The composition procedure starts from a description of the set of events to be considered, produces an intermediate representation that captures the typical human view of events – restricted to what might have been perceived by a given character at a given moment in time – and proposes an algorithmic procedure to build a sequence of spans of narrative discourse, each capturing perception by an individual character. These spans of narrative discourse are rendered first as a simple conceptual description and then as pseudo-text. Table 1 shows examples of the original input as algebraic notation for a chess game, the conceptual representation of the composed discourse, and the pseudo-text rendering.

This conceptual description is built in such a way that the system can interpret it to reconstruct a version of the exhaustive description of the world that it started from. This is used by the system to validate the decisions taken during the composition of the discourse. The ability to convert automatically from one to another across different formats of knowledge representation can play a crucial role in the proposal described in this paper.

## **4 Using CNL for Knowledge Elicitation and Exchange across Story Generation Systems**

There are two important conclusions that can be extracted from the material presented so far.

1. e4 e5	9. Nc3 Qe8	focalizes_on bp5	The fifth black pawn was
2. Nf3 Nc6	10. Bxf6 gxf6	is_at lbb behind	three squares north of the
3. Bc4 Bc5	11. Nh4 Kh8	is_at bp4 same	centre of the board. The
4. O-O Nf6	12. Qh5 Be6	is_at bp4 same	fifth black pawn found the
5. h3 d6	13. Qxh6+ Kg8	is_at bq behind	black left bishop was behind.
6. d3 O-O	14. Nd5 Bxf2+	is_at bk behind	The fifth black pawn found
7. Bg5 h6	15. Rxf2 Qd7	move_to bp4 1/south	the fourth black pawn was
8. Bh4 b6	16. Nxf6# 1-0	4 later	same. The fifth black pawn
		arrives_at rbk left	found the black queen was
		...	behind. The fifth black
			pawn found the black king
			was behind. The fifth black
			pawn saw the fourth black
			pawn moving south a square.
			Four days later, the fifth
			black pawn saw the black
			right knight arriving left.
			...

**Table 1.** Original input as algebraic notation for a chess game, conceptual representation of the composed discourse, and pseudo-text rendering.

First, that story generations are faced with a significant challenge of acquiring knowledge resources in the particular representation formats that they use. The difficulty of expressing human knowledge in formal languages is a considerable obstacle. In this context, the use of a CNL would provide the means for quicker development of required resources in a format easier to write for human experts. There are ongoing efforts to build such information via crowdsourcing and/or to learn this information via information extraction techniques, but all efforts along these lines either:

- have met with limited success,
- need to rely on huge amounts of hand annotation,
- require procedures of controlled edition very similar to CNL.

Second, that every story generation system defines its own format for knowledge representation, optimised to support its storytelling process. Although the development of a common formalism for representing knowledge would provide a major breakthrough, this is unlikely to happen in the near future (see Gervás and León [3] for a detailed discussion of the problems involved). Under the circumstances, the use of a CNL for codifying resources for storytelling systems might provide some relief. If authors of storytelling systems were to develop the initial version of their resources in a commonly agreed CNL, and then develop the appropriate automated transformations to generate knowledge in their own preferred format, the same resources written in CNL might be of use to researchers developing different storytelling systems. By simply writing the appropriate transformations into their own preferred format, much of the already available knowledge could be reused.

The main advantage of using a CNL is that it can be expressed by domain experts and then it can be translated to the variety of formal languages used in different systems. This feature allows the creation of a common language not only for expressing the different aspects involved in narrative generation, but also for exchanging knowledge resources across different storytelling systems. This might also pave the way for the development of common benchmarks for testing storytelling systems. A relevant conclusion mentioned by Gervás and León [3] is that the same information may be represented through different data structures without affecting its essence, or a data structure can be extended for representing additional types of information.

## 5 A Case Study: the STella System

STella (Story Telling Algorithm) [22, 23] is a story generation system that controls and chooses states in a non-deterministically generated space of partial stories until it finds a satisfactory simulation of events that is rendered as a story. This simulation has been modeled as a knowledge intensive approach in which the whole world domain is explicitly represented as a simplistic view of a realistic environment. At each step, candidate updated versions of the current state are computed and the most likely ones are identified by computing their likelihood in terms of their plausibility and their narrative properties. Candidate partial stories are evaluated based on how well they satisfy a given set of constraints and how their tension curves compare with a set of target curves. The results of this process are used to decide when a partial story is promising and whether a story is finished.

### 5.1 Knowledge Representation in STella

STella is a resource hungry system. The underlying knowledge driving the generation has a big impact on the final quality of the output. One of the main characteristics of STella is its heavy dependency on a core set of rules defining the whole universe the system is capable of producing stories about. The application of a CNL can highly improve the situation by letting external sources be added as knowledge.

The rules are considered part of the domain, and these basically operate by producing sequences of *snapshots* and *actions*. Snapshots are states of the world. A snapshot describes exact character positions, affinities, items, and every other detail of the world. Actions contain information about what led from the previous state to the current one. Then, a story is formally defined as show in Equation 1.

$$story = \{(s_1, [a_{1,1}, a_{1,2}, \dots, a_{1,n}]), \dots, (s_z, [a_{z,1}, a_{z,2}, \dots, a_{z,n}])\} \quad (1)$$

where  $s_x$  is a snapshot and  $a_{i,j}$  is an action. Each pair is called a *state*, and a sequence of states form a story. Actions have their own vocabulary and correspond

to specific structures like  $take(character, item)$  or  $approach(character, place)$ . Snapshots are defined according to a fixed ontology that structures the world. In STella, the world is a matrix, and every entity fills exactly one cell. Big entities, as houses, are composed of small entities (bricks). Each entity has its own set of attributes. Characters, in particular, are the most developed and detailed entities and are described in terms of properties commonly influencing narratives:

- physical properties for moving and interacting with the environment,
- affinities between characters,
- an internal representation of the world, which does not have to be the same as the real world. Characters use this for planning.
- Roles (moral tendency) and traits (special capabilities)

Rules, then, receive a current state (a pair of a snapshot and the actions that led to it from the previous one) and non-deterministically output a new set of actions. The non-deterministic aspect is not relevant for this discussion, and more information can be found on the literature [23]. The creation of rules is, then, driven by the data model.

## 5.2 Fundamental Features of a CNL for STella

In order to design a CNL for the knowledge base of STella, the data model must be covered by the language. While it is not a trivial task, the fact that the data model is well established and structured means the vocabulary can be easily described and expressed in natural language. In particular, the sentences to use must be able to describe preconditions and actions. Let us examine the following case:

When a character is alive and it moves north, its  $y$  coordinate has to be increased.

State information and actions are used in the sentence:

- *a character is alive* refers to a state. The rule is valid for *any* character that is alive.
- *it moves north* is an action, it corresponds to  $move(?x, north)$ .
- *its  $y$  coordinate must be increased* applies a change in the state:  $y = y + 1$ .

In order to transform the sentence into a rule, a template must be filled in like this:

**precondition**  $\forall ?c \in Characters, alive(?c)$

**action**  $move(?c, north)$

**postcondition**  $?c.y \leftarrow ?c.y + 1$



Quantified state and action information must be addressed, which is, given the data model, doable. The problem is the generality needed in the kind of changes that must be applied in the postconditions. In this case, the simple *move* rule has updated the value of the *y* location component of the character ?*c*. For more complex changes (like, for instance, traversing all available items in ?*c*'s inventory), the CNL should cover a non-trivial set of constructions.

It is hypothesized that these complex constructions are, in general, mostly covered by a number of basic operations (traversing lists, accessing elements with given properties or finding elements in the world). Making this constructions atomic and suitable for composition can led to a simpler, reasonable expressive CNL. More in-depth study must be conducted in order to gain better insight on what this set of properties is.

## 6 Discussion

The construction of a CNL with the desired properties is a significant challenge. Namely because, to cover all the various aspects of representation relevant for storytelling systems as a whole, it would have to address at least all aspects described by Gervás and León [3]. From a simplified point of view, two major layers of representation can be considered. Firstly, an orchestration layer, whose main concern is related to the dynamic flow of the story. Secondly, a characterization layer, which is focused on representing the static features of the elements that define the story. The orchestration layer is related to the discourse sequence aspect, the causal aspect, and the intentional aspect. On the other side, the characterization layer includes the remaining aspects: the simulation aspect, the theme aspect, the emotional aspect, the authorial aspect, and the narrative structure aspect. It is both necessary and important to emphasise that these layers, and the aspects associated to them, are mutually interwoven. This means that changes in the data related to one aspect typically will cause changes in other aspects. For example, a change in the feelings of a character (emotional aspect) could determine his/her course of action, or modify his/her objectives (intentional aspect).

In addition, the use of a domain-specific glossary would serve not only for establishing a proper definition of the knowledge domain, but also for reducing the risk of polysemy. One of the potential issues with CNL is that they are not specifically designed to address word sense disambiguation. The definition of a CNL usually focuses on analysing just some key words that are relevant for building the discourse representation structure.

In a first-approach, it is possible to define a basic modelling for just some aspects of the narration. One suitable candidate could be the sequential aspect through the use of a planning modelling language, since a good part of the story generation systems architecture is based on a planner. A common language for modelling planners is PDDL (Planning Domain Description Language) [24], which is designed to formalize dynamic models, where actions guide the model through a series of state. This first step would consist of developing a match

between the knowledge modelled in PDDL and a collection of primitives for describing the same information.

## 7 Conclusions and future work

This paper discussed the suitability of using a CNL for eliciting and exchanging knowledge in the context of a range of story generation systems. As shown above, there have been precedents of the use of CNL in the interactive storytelling domain with satisfactory results. The advantages of using a CNL for elicitation of knowledge resources had been demonstrated in the past. The potential for providing a compatible format for the exchange of knowledge across systems would be a major positive contribution to the field.

Future work involves not only the complete development of a CNL for covering the knowledge needed in this domain, but also the development of evaluation techniques that validate the suitability and portability of this representation over a wide range of story generation systems. In particular, a short term goal of the authors is to establish a CNL that can serve to develop knowledge resources that might as common ground data for a shared evaluation task for storytelling systems.

## Acknowledgements

This paper has been partially supported by the project WHIM (611560) funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program

## References

1. Schwitter, R.: Controlled natural languages for knowledge representation. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. COLING '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 1113–1121
2. Gervás, P.: Story generator algorithms. In: The Living Handbook of Narratology. Hamburg University Press (2012)
3. Gervás, P., León, C.: The need for multi-aspectual representation of narratives in modelling their creative process. In: 5th Workshop on Computational Models of Narrative. OASICs-OpenAccess Series in Informatics (2014)
4. Meehan, J.R.: Tale-spin, an interactive program that writes stories. In: In Proceedings of the Fifth International Joint Conference on Artificial Intelligence. (1977) 91–98
5. Schank, R.C., Abelson, R.P.: Scripts, plans, and knowledge. In: Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1. IJCAI'75, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1975) 151–157
6. Turner, S.R.: Minstrel: A Computer Model of Creativity and Storytelling. PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA (1993) UMI Order no. GAX93-19933.

7. Perez y Perez, R.: MEXICA: A Computer Model of Creativity in Writing. PhD thesis, The University of Sussex (1999)
8. Liu, H., Singh, P.: Makebelieve: Using commonsense knowledge to generate stories. In Dechter, R., Sutton, R.S., eds.: AAAI/IAAI, AAAI Press / The MIT Press (2002) 957–958
9. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open mind common sense: Knowledge acquisition from the general public. In: On the move to meaningful internet systems 2002: Coopis, DOA and Odbase, Springer (2002) 1223–1237
10. Liu, H., Singh, P.: ConceptNet – a practical commonsense reasoning tool-kit. *BT technology journal* **22**(4) (2004) 211–226
11. Minsky, M.: *Society of mind*. Simon and Schuster (1988)
12. Reed, A.: *Creating Interactive Fiction with Inform 7*. Cengage Learning (2010)
13. Fuchs, N.E., Schwertel, U., Schwitter, R.: Attempto controlled english not just another logic specification language. In: *Logic-based program synthesis and transformation*. Springer (1998) 1–20
14. Campbell, M.: A new way to play: Make your own games. *New Scientist* **211**(2829) (2011)
15. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: Programming for all. *Commun. ACM* **52**(11) (November 2009) 60–67
16. Broeckhoven, F.V., Vlieghe, J., Troyer, O.D.: Using a controlled natural language for specifying the narratives of serious games. In: *8th International Conference on Interactive Digital Storytelling, ICIDS 2015*. (2015) 142–153
17. Broeckhoven, F.V., Troyer, O.D.: Attac-l: A modeling language for educational virtual scenarios in the context of preventing cyber bullying. In: *2nd International Conference on Serious Games and Applications for Health, IEEE* (May 2013) 1–8
18. Li, B., Lee-Urban, S., Johnston, G., Riedl, M.O.: Story generation with crowd-sourced plot graphs. In: *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI '13* (2013)
19. Winston, P.H.: The genesis story understanding and story telling system a 21st century step toward artificial intelligence. Technical report, Center for Brains, Minds and Machines (CBMM) (2016)
20. O’Neill, B.: *A Computational Model of Suspense for the Augmentation of Intelligent Story Generation*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia (2013)
21. Gervás, P.: Composing narrative discourse for stories of many characters: A case study over a chess game. *Literary and Linguistic Computing* **29**(4) (2014) 511–531
22. León, C., Gervás, P.: A top-down design methodology based on causality and chronology for developing assisted story generation systems. In: *8th ACM Conference on Creativity and Cognition, Atlanta* (11/2012 2011)
23. León, C., Gervás, P.: Creativity in Story Generation From the Ground Up: Non-deterministic Simulation driven by Narrative. In: *5th International Conference on Computational Creativity, ICC3 2014, Ljubljana, Slovenia* (2014)
24. McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A. and Veloso, M., Wilkins, D.: PDDL - the planning domain definition language. (1998)