# Creativity in Story Generation From the Ground Up: Non-deterministic Simulation driven by Narrative

**Carlos León**
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
cleon@fdi.ucm.es

**Pablo Gervás**
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
28040 Madrid, Spain
pgervas@sip.ucm.es

### Abstract

Creativity in narrative requires a careful management of knowledge but story generation systems focusing on creativity have typically circumvented this level of detail by using high level descriptions of events and relations. While this has proven effective for plot generation, narrative generation can be drastically enriched with a grounded representation of actions based on low level simulation. This level of detail and robust knowledge representation can form the basis for a conceptual space exploration driven by narrative knowledge, namely by guiding non-deterministic generation of successive simulation states composing a story. This paper presents and updated version of the story generation system *STellA* that implements this hybrid model, along with results and discussion on the relative benefits of the described approach.

## Introduction

Instances of story generation systems usually perform at a relatively abstract level, focusing on the plot and aggregating details that, if processed at a lower granularity level, could enrich a story to the point that these details themselves could potentially be the sources for new narrative constructions and unexpected plot twists (Turner 1992; Pérez y Pérez 1999; Riedl and Young 2010). This lack of fine grain detail is usually due to the technical restrictions that the currently available knowledge representation models impose over the design of complete story generation systems. Classic knowledge representation methods have proven to set the same limits on the implementation of this kind of systems as on many other applications like expert systems (Bell 1985) or ontologies (Rosati 2007), to name a few.

Lower level world-modelling techniques, like simulation, have different features than relation-based knowledge representation. In this context, we consider simulation as a process in which the whole world is modelled in a complete structure evolves step by step according to a certain, fully defined set of rules. This definition is broad enough to contain a number of different approaches to knowledge representation in general and plot generation in particular. Simulation-based modelling, as one of these techniques, can provide a good way to represent the needed information for story generation while relatively different from logic-based approaches. Indeed, simulation has been used to model narrative generation, but it has not been widely used to create explicit models of creativity in narrative. This is probably because the most evident use of simulation is the reproduction of the evolution of a static model in order to examine some results, which seemingly contradicts the need for unpredictability, novelty and freedom usually assumed to play a fundamental role in creativity.

The relatively reduced number of systems that use simulation to model creative processes contrasts with the undeniable success of simulation for gathering results and producing data from grounded models. When seen in the appropriate light, simulation becomes a powerful tool for generating a big amount of artifacts, but only if the generative process is able to complement the robust generation of simulation-produced data with techniques that let the generation produce and explore a conceptual space. In fact, simulation has been applied to story generation in several systems, but these have not put the focus on creative generation (Meehan 1977; Theune et al. 2003; Aylett et al. 2005).

This context suggests that enhancing a process grounded in simulation with already available models used in Computational Creativity is a promising method for producing grounded data and at the same time explore a conceptual space. In particular, creative processes heavily influenced by knowledge representation and management, as story generation, can benefit from the features that both fields offer. Generating a story is a complex process where details can make a huge difference and simulation can provide this level detail when used against a proper model. Together with this granularity, explicit means for traversing a conceptual space trying to generate a story with certain properties can provide a useful pattern for story generation.

This hybrid system mixing simulation and creative exploration for story generation is described along this paper. The current system description is an updated version of the story generation system *STellA* (*Story Telling Algorithm*) (León and Gervás 2011) that mixes a non-constrained simulation-based production of world states and narrative actions as source material for a conceptual space exploration engine. The system controls and chooses simulations in a non-deterministically generated space of partial stories until the generation finds a satisfactory progression of simulations

that are rendered as a story.

The previous design of *STellA* did not include a world simulation as a generative solution. Instead, knowledge was represented by means of logic facts and a elaborated set of domain rules. While this approach was carefully structured to permit incremental knowledge inclusion, the engineering effort for modelling the world became too big. We identified that an even more structured representation (a well defined structure resembling the world model used in simulations) could alleviate the required engineering effort. This paper thus describes the modification of the main generation engine to allow for a simulation-based knowledge representation and world evolution. This includes the design of a new representation system and the creation of a narrative-driven conceptual space exploration based on rules (objectives and constraints) and narrative curves. The previous version of *STellA* included curves and rules, but the way in which they were used was fundamentally different.

## Related Approaches to Automatic Story Generation

In order to avoid ambiguity, we will restrict our analysis here to three levels of conceptual representation of a story, and refer to these as the *fabula* (the complete set of what could be told, organised in chronological order of occurrence), the *discourse* (what has been chosen to tell, organised in the order in which it is to be told) and the *narrative* (the actual way of telling it). Of all existing effort to build plots, the present review will be focusing on those that construct a fabula by means of a process of simulating the actions of a set of characters.

The first story telling system for which there is a record is the Novel Writer system developed by Sheldon Klein (Klein et al. 1973). Novel Writer created murder stories within the context of a weekend party. It relied on a micro-simulation model where the behaviour of individual characters and events were governed by probabilistic rules that progressively changed the state of the simulated world (represented as a semantic network). The flow of the narrative arises from reports on the changing state of the world model. A description of the world in which the story was to take place was provided as input. The particular murderer and victim depended on the character traits specified as input (with an additional random ingredient). The motives arise as a function of the events during the course of the story. The set of rules is highly constraining, and allows for the construction of only one very specific type of story.

Overall, Novel Writer operated on a very restricted setting (murder mystery at weekend party, established in the initial specification of the initial state of the network), with no automated character creation (character traits were specified as input). The world representation allows for reasonably wide modeling of relations between characters. Causality is used by the system to drive the creation of the story (motives arise from events and lead to a murder, for instance) but not represented explicitly (it is only implicit in the rules of the system). Personality characteristics are explicitly represented but marked as "not to be described in output". This suggests that there is a process of selection of what to mention and what to omit, but the model of how to do this is hard-wired in the code.

TALESPIN (Meehan 1977), a system which told stories about the lives of simple woodland creatures, was based on planning: to create a story, a character is given a goal, and then the plan is developed to solve the goal. TALESPIN introduces character goals as triggers for action. Actions are no longer set off directly by satisfaction of their conditions, an initial goal is set, which is decomposed into subgoals and events. The systems allows the possibility of having more than one problem-solving character in the story (and it introduced separate goal lists for each of them). The validity of a story is established in terms of: existence of a problem, degree of difficulty in solving the problem, and nature or level of problem solved.

Lebowitz's UNIVERSE (Lebowitz 1985) modelled the generation of scripts for a succession of TV soap opera episodes (a large cast of characters play out multiple, simultaneous, overlapping stories that never end). UNIVERSE is the first storytelling system to devote special attention to the creation of characters. Complex data structures are presented to represent characters, and a simple algorithm is proposed to fill these in partly in an automatic way. But the bulk of characterization is left for the user to do by hand.

UNIVERSE is aimed at exploring extended story generation, a continuing serial rather than a story with a beginning and an end. It is in a first instance intended as a writer's aid, with additional hopes to later develop it into an autonomous storyteller. UNIVERSE first addresses a question of procedure in making up a story over a fictional world: whether the world should be built first and then a plot to take place in it, or whether the plot should drive the construction of the world, with characters, locations and objects being created as needed. Lebowitz declares himself in favour of the first option, which is why UNIVERSE includes facilities for creating characters independently of plot, in contrast to Dehn (Dehn 1981) who favoured the second in her AUTHOR program (which was intended to simulate the author's mind as she makes up a story).

The actual story generation process of UNIVERSE (Lebowitz 1985) uses plan-like units (plot fragments) to generate plot outlines. Treatment of dialogue and low-level text generation are explicitly postponed to some later stage. Plot fragments provide narrative methods that achieve goals, but the goals considered here are not character goals, but author goals. This is intended to allow the system to lead characters into undertaking actions that they would not have chosen to do as independent agents (to make the story interesting, usually by giving rise to melodramatic conflicts). The system keeps a precedence graph that records how the various pending author goals and plot fragments relate to each other and to events that have been told already. To plan the next stage of the plot, a goal with no missing preconditions is selected and expanded. Search is not depth first, so that the system may switch from expanding goals related with one branch of the story to expanding goals for a totally different one. When selecting plot fragments or characters to use in expansion, priority is

given to those that achieve extra goals from among those pending.

The line of work initiated by TALESPIN, based on modelling the behaviour of characters, has led to a specific branch of storytellers. Characters are implemented as autonomous intelligent agents that can choose their own actions informed by their internal states (including goals and emotions) and their perception of the environment. Narrative is understood to emerge from the interaction of these characters with one another. While this guarantees coherent plots, Dehn pointed out that lack of author goals does not necessarily produce very interesting stories. However, it has been found very useful in the context of virtual environments, where the introduction of such agents injects a measure of narrative to an interactive setting.

The Virtual Storyteller (Theune et al. 2003) introduces a multi-agent approach to story creation where a specific director agent is introduced to look after plot. Each agent has its own knowledge base (representing what it knows about the world) and rules to govern its behaviour. In particular, the director agent has basic knowledge about plot structure (that it must have a beginning, a middle, and a happy end) and exercises control over agent's actions in one of three ways: environmental (introduce new characters and object), motivational (giving characters specific goals), and proscriptive (disallowing a character's intended action). The director has no prescriptive control (it cannot force characters to perform specific actions). Theune et al. report the use of rules to measure issues such as surprise and "impressiveness".

In general, approaches to Interactive Storytelling have some degree of simulation as conceived in this work (Aylett et al. 2005; Cavazza, Charles, and Mead 2002; Mateas and Stern 2005). While every approach models the problem of story generation in a specific way, there exist some degree of similarity in the way they perform, namely by chaining sequential states that are driven or selected by an implicit or explicit model of plot quality.

## Knowledge Representation in the Story Generation System: Simulation

Narratives are known to share a relatively high amount of constructions and the complexity of common sense knowledge (Schank and Abelson 1977). Elaborated narratives are as complex as common human knowledge and thus its representation and processing is a long term problem of Artificial Intelligence. As an example, we can borrow a famous scene from *The Hobbit* (Tolkien 1972) in which Bilbo Baggins, when trying to win the game of riddles against Gollum, asks himself "What have I got in my pocket?". While the scene can seem not very complex for human cognition, this seemingly simple event carries a huge amount of information that requires a fine grain representation of characters (property, clothes, value of items), intentions (trying to escape), self-awareness (asking something to himself), emotions (fear), focus and concentration of characters (focusing on something relatively independent from the current context) and many other aspects that confer relative narrative quality and richness.

The complexity becomes a problem when trying to represent knowledge by classic means. Logic-based knowledge representations methods have been designed from the early years of Artificial Intelligence and, after the initial optimism (revived with the arrival of expert systems) the complexity of such systems became clear to the point that it is widely accepted that knowledge intensive systems are limited and their use is restricted only to very well known domains (Bell 1985). Many different kinds of formalisms for knowledge representation have appeared along the last years (Trentelman 2009; Sloman 1985), but the basic problems of knowledge representation are still present and relatively unsolved (Sowa 2000; Baral 2003).

Logic-based knowledge representations for story generation has nonetheless been used in several story generation system, but with very restricted domains (Pérez y Pérez 1999; Bringsjord and Ferrucci 1999). This has classically lead to systems that perform well in their respective merits and contributions, but a big amount of rich stories has not been produced so far. In order to partially tackle this issue, the presented version of *STellA* follows the hypothesis that grounding knowledge representation as much as possible is determinant for allowing a story generation system to produce rich content. A rich representation complemented by conceptual space exploration guided by narrative are proposed as a solution for creative story generation. According to this hypothesis, making the simulation more complex could provide more complex worlds and interactions and therefore create a larger conceptual space traversable by the narrative-based driving engine. The system will hypothetically be able to generate many different stories and partially identify which ones are "better" according to a set of given objectives.

### Grounding Knowledge for Storytelling

For the simulation engine to be able to produce states containing content suitable for narrative generation, an appropriate grounded representation and a corresponding set of rules for creating that information are needed. This is a new addition to *STellA*.

Grounding knowledge representation for story generation requires a low level definition of concepts that are usually defined in a more abstract way by most other generation systems (Turner 1992; Pérez y Pérez 1999; Bringsjord and Ferrucci 1999). This results in an additional effort from the beginning since usual constructions inherited from logics as $in(knight, room)$ must be refined so as to represent data better suited for simulation. In the previous example, in order to represent exact position, the data would have to become $position_{knight} = (10, 20)$, assuming that $(10, 20)$ is a valid coordinate inside the $room$. This is the kind of knowledge representation that the proposed system uses.

This approach requires a fixed representation in which every construction or relation is *grounded* in the sense that the system includes mechanisms to process that construction internally. This grounding permits meta-representation of the world, which means that a mental state of the world, for instance, can be represented using the same formalism.

This meta-representation *STellA* is provided with makes

knowledge representation possible at two different levels: first, characters' reasoning uses a set of rules that manage incomplete knowledge (characters can ignore aspects of their surrounding context). Then, the same set of rules is applied to the simulated world, in which there is no uncertain information since the whole state is available. This implies a relative reduced engineering effort compared with the maintenance of two different rule sets.

Domain rules are a determinant part in this model. Narrative generation is a knowledge hungry process and any domain model is by definition incomplete (given the requirements of narrative this would imply modelling all human knowledge). This makes it almost impossible to recreate the needed amount of information in a single prototype, thus imposing the need to design a flexible, improvable system to let it evolve over time and manage a richer set of knowledge constructions.

In order to keep the rule set maintainable, rule coupling has been reduced to a minimum in terms of the structure of the rule set. Rules are organized in a linear way, meaning that no hierarchical topology is imposed over the design. This lets the maintainer include new rules without taking a big structure into account. Additionally, rules can be enabled or disabled at will without affecting the rest of the system since no rule is dependent on any other by design. The semantic coupling between rules still exist, but this is kept to a minimum.

For this independence of rules to be possible, a domain-specific language for rules has been included as part of the generation engine. The rules can query the world state and output *actions* that represent changes in the story, as the next section explains. Querying current state limits the scope in which rules can act, which constraints rule creation and makes them easier to produce. Rules cannot examine the story but only the current simulation. In this way, narrative processes are isolated.

*STellA* offers a set of primitives for querying the current story so that the creation of these rules can be made without knowing the representation details. Figure 1 shows an example of objective rule for creating the story and the use of the story-querying primitives that the current version of the system provides the user with.

$$\begin{aligned} \texttt{finished}(story) \leftarrow\ & hs = \texttt{humans}(story) \\ & hs_d = \texttt{inDungeon}(story, hs) \\ & \texttt{length}(hs_d) == 0 \end{aligned}$$

Figure 1: Example of objective rule for the story generation process. A story must satisfy this rule to be valid.

Rules are able to cope with *incomplete knowledge* in the generation system, which is also a new addition in this updated version of *STellA*. The meta-representation of the world that characters have can be incomplete, and thus some properties of the internal representations can have the $uncertain$ value. When characters reason to decide their next action, use a simple unification mechanism to instance the $uncertain$ value with potentially valid grounded values. For instance, a character ignoring whether an enemy is equipped with a weapon searches over the possibilities and acts according the first plausible solution. More powerful inferencing techniques will be used in future versions.

## Non-deterministic generation of Narrative Actions

If the described simulation process generates only one single sequence of actions and corresponding states, the room for creativity would be marginal. According to most frameworks of computational and non-computational creativity, the creation or exploration of a conceptual space, trying to produce unexpected and valuable artifacts is a determinant part of the creative process (Boden 1999; 2003).

This update of *STellA* performs the exploration of the corresponding conceptual space generatively, that is, iteratively creating new states for subsequent simulation. This has been modeled and implemented as a non-deterministic process in which a certain simulation step can yield not one but many steps. From a classical Artificial Intelligence perspective, the conceptual space generated by *STellA* is a tree rooted in the original state (the base state from which the generation happens). Each intermediate node of the conceptual tree contains a partial simulation state that, when processed, generates possibly many candidate states that can be subsequently expanded, in this way modelling non-determinism.

While state exploration works for expanding the conceptual space, connecting the simulation with the creation of a narrative structure requires a more detailed process. The grounded data coming from each generation step must be processed carefully because the state changes that a simulation step yields are heterogeneous from a narrative perspective.

The changes happening from a simulation state to the next one that are produced in the non-deterministic expansions are referred to as *narrative actions*, which are a new addition to *STellA*. During the development of the described system the number of these actions has grown as more different kinds were detected. It is important to note that the way in which simulation is implemented in *STellA* affects the kind of actions that are produced and thus its identification, but the next list is likely to be applicable to other approaches as well:

- *Character perception* actions define the parts of the simulation that are perceived by the characters. This includes perceiving the surrounding objects, being aware of health and position, updating or forgetting the position of an object that has moved and so on. The generation of these actions are currently model as a non-deterministic process in which perceptions have a probability to happen. The algorithm then orders perceptions by probability, creating sets of perceived elements non-deterministically. Perception actions are the link between the complete world happening in the simulation and the inner representation of it that every agent in the story (every character) has.

- *Deus ex* actions are generated without any causal requirement. They must be consistent with the current state, but do not need to respond to any character need of model. Deus ex actions model events that are too serendipitous to need a detailed model, like a character stumbling upon a rock when running or raining. These actions are generated non-deterministically and have a probability of happening in their definition that is used by the generator to order these actions by their chance of occurring and not by pure randomness. This has been designed so to keep a complete model not depending on random numbers.
- *Character desires* actions are the output of a reasoning process that emulates character decisions. These decisions include eating if the character is hungry, trying to escape an enemy or maybe attacking him or her. These actions confer a relative degree of believability (Riedl 2004). Character desires actions, which are generated in a non-deterministic way, have both an associated probability and a *priority*. This priority is used by the characters in the next step of simulation to order desires and try to satisfy the most prioritized ones first.
- *Character intentions* complete *desires* and *perception* so as to reproduce a classic agent-like narrative model (Bratman 1987). Intentions are generated according to perceptions (beliefs in the classic model) and desires, which means that the representation of the external world in not taken into account when creating intentions (only the character's internal representation). This allows for a simpler creation of rules since less information must be taken into account. Character intentions actions are non-deterministic too and have an associate probability just like the other kinds of actions. Trying to go to some location that the character desires to be in or trying to attack the enemy that the character desires to be dead are examples of intentions. The difference between *doing* and *trying to do* is subtle but very influential in narrative generation since it permits richer character interaction.
- *Physical world* actions are non-deterministic and model causality of physical events that, under certain conditions, will necessarily happen with a certain probability. Things that fall to the ground if nothing holds them or moving an object if it is pushed with enough force are examples of physical world actions. This kind of actions have the additional role of representing *success* of *failure* of character intentions. In this way, a character can try an action and the physical state will decide whether the intention succeeded or not.

This division makes sense from the point of view of story generation. The focus and detail on character behavior is clear and considered to be very important in narrative. This is complemented with serendipitous events and world physics in a broad sense. Probabilities are used to order actions in such a way that the main algorithm produces candidate updated versions of the current state of the simulation and gives priority to the most likely ones. Creativity can be explored by choosing less likely states, which is planned as part of the future enhancements of *STellA*.

These five kinds of narrative actions are extracted from the simulation. Formally speaking, the output of each step of the simulation non-deterministically yields a set of new states along with their corresponding actions. This can be formally described as:

$$\langle state, e, p, d, i, w \rangle$$

where $state$ is the current state of the simulation, $e$ is the set of *deus ex* actions generated from that step, $p$ is the set of *character perception* actions, $d$ is the set of *character desires* actions, $i$ is the set of *character intentions* actions and $w$ is the set of *physical world* actions.

A *fabula* generated by *STellA* is then a list of tuples:

$$[\langle state, e, p, d, i, w \rangle]$$

The generation can be represented formally in terms of a generative function $\gamma$ that accepts a $state$ and returns a non-deterministic set of tuples:

$$\begin{aligned}\gamma(state) = \{&\langle state_0, e_0, p_0, d_0, i_0, w_0 \rangle, \\ &\langle state_1, e_1, p_1, d_1, i_1, w_1 \rangle, \\ &\dots \\ &\langle state_n, e_n, p_n, d_n, i_n, w_n \rangle\}\end{aligned}$$

Having explained and formalize how to generate a conceptual space of stories from a grounded simulation, it is still necessary to complete the system by including a way to traverse this space an find valuable artifacts, namely valid stories.

## Narrative Drives the Simulation: Curves, Objectives and Constraints

Simulation is a flexible and powerful tool for representing the state of a story and the transitions between states. However, producing a sequence of states that, when appropriately rendered, are acceptable as a narrative, requires control over the generation. *STellA* uses three types of mechanisms to drive the simulation: *objectives*, *constraints* and *narrative curves*.

The presented generation process is fed with a set of *objectives* that the story must satisfy in order to be suitable to be accepted as finished and valuable by the system. This version of the story generation system models objectives as a group of boolean functions receiving a story. The user can thus use these to create declarative definitions of the kind of wanted story. *Objectives* are used post-hoc. When a partial story is reached by the system, it is checked against the set of these objectives and all of them must accept the story as valid. Figure 1 shows an example.

Along with *objectives*, providing the system with means to restrict the generation is needed. Non-determinism in story generation is a powerful modelling tool, but unrestricted production of stories degenerates in a very big conceptual space whose whole traversal is intractable (León and Gervás 2010). This is not only consistent from a computational perspective but also from the point of view of creativity in story generation: the set of stories that can be generated from any starting state is very large.

This characteristic is inherent to the domain of story production and cannot be eluded. The computational generation can, however, filter out those intermediate states that are not promising and should not be explored, as humans seemingly do (Sharples 1999). The current model uses *constraints* for avoiding exploring branches of the traversal process that are unpromising. The implementation of constraints is analogous to the implementation of objectives as constraints are defined in terms of declarative rules using the same kind of formalism and query primitives. Constraints, however, are used in the generation during the expansion of new states to be simulated and forbid the exploration of those candidates states that do not satisfy them.

The use of constraints compared to objectives therefore leads to a less strict definition. In practical terms, constraints are usually less restrictive with regard to their scope: experience suggests that constraints are defined in term of specific features that a story should not have, while objectives tend to describe general aspects of a narration. Figure 3, showing an example of a constraint, exemplifies this.

$$
\begin{aligned}
\texttt{promising}(story) \leftarrow\ & hs = \texttt{humans}(story) \\
& \forall h_i \in hs : \\
& \quad \forall h_d \in hs - \{h_i\} : \\
& \quad\quad d_i = \texttt{distance}(story, h_i, h_d) \\
& av = \texttt{average}(d_0, d_1, \ldots, d_n) \\
& \texttt{av} <= threshold
\end{aligned}
$$

Figure 2: Example of constraint rule. A partial story not satisfying a constraint rule will not be accepted as promising and its corresponding state will not be explored.

*STellA* uses a generalized version of *tension curves* to drive story generation. The design of these curves as a way to drive plot generation has been studied in previous versions of *STellA* (León and Gervás 2011; León and Gervás 2012). The main objective underlying this method is to represent the evolution of a set of narrative properties of a story as curves. As the conceptual space is traversed to find a suitable story, this evolution is iteratively compared with a set of objective curves. This comparison informs the traversal on every step and this information can be used as an additional source for deciding when a partial story is promising and whether a story is finished.

Previous versions of *STellA* also considered these methods for plot generation, but they were applied differently. Objectives and constraints did were not as powerful as they are in this version regarding their both their expressive power and their scope. While the current version allows for evaluation of a complete story, previously only states were considered, additionally, full access to the world representation is allowed now. Curves have a more general definition now since they define generic metrics (distances, average values and others) and previous versions needed more elaborated definitions. This has been made easier by the use of a simulation-based representation.

Algorithm 1 describes the overall generation algorithm. The non-determinism occurs, as previously described, when generating candidate sets of *deus ex*, *character desires* and *character intentions* actions. The generation algorithm iterates until a satisfying story is found and filters those exploratory branches that are unpromising according to the constraints imposed in the execution.

**Data**: the current partial story $[\langle state, e, p, d, i, w\rangle]$
objective curves
objective function
constraint function
**Result**: a set of candidate new tuples
**while** *current story is not finished according to curves and objectives* **do**
  $\sigma \longleftarrow$ last state tuple from current story
  $p \overleftarrow{\text{non-det}}$ *perception* for $\sigma$ ordered by probability
  $e \overleftarrow{\text{non-det}}$ *deus ex* for $\sigma$ ordered by probability
  $d \overleftarrow{\text{non-det}}$ *desire* for $\sigma$ ordered by probability
  $i \overleftarrow{\text{non-det}}$ *intention* for $\sigma$ ordered by probability
  $w \overleftarrow{\text{non-det}}$ *physical world* for $\sigma$ ordered by probability
  $\sigma' \leftarrow$ apply $(e, p, d, i, w)$ to $\sigma$
  $curves_{\sigma'} \leftarrow$ compute current curves for $\sigma'$
  new story $\longleftarrow$ current story $+ \sigma'$
  **if** $curves_{\sigma'} \approx curves_{objective} \wedge$ *new story satisfies constraints* **then**
    **foreach** $\sigma'$ **do**
      explore generation from $\sigma'$
    **end**
  **else**
    reject $\sigma'$
  **end**
**end**
**return** current story

**Algorithm 1:** Story generation algorithm in *STellA*

## Example Output

The described model has been implemented in three main modules:

1. The core engine for generating stories, containing the non-deterministic algorithms and basic narrative data structures.
2. The simulation engine defining the basic data structures and rules for the simulation to happen.
3. The set of rules both for generating actions and for defining story objectives.

The core engine (1) corresponds to the implementation of Algorithm 1 and the simulation engine (2) has been implemented according to the model previously described. A rule set (3) for an example prototype has been created for demonstration purposes. This rule set and the sample world place the action in a dungeon from which humans must escape.

The simulated world is a two-dimensional grid in which every entity is placed in one single cell. Basic actions of

characters are $move$ in eight directions, $attack$ adjacent enemies, $eat$ food, $escape$, $protect$ themselves and others, $take$ and $drop$ objects and $apply$ objects on other entities (for healing an ally, for instance). Characters and creatures can sense their surroundings and use an A* based pathfinder to go from one place to another. Characters loose energy for being injured and doing things. The initial state includes 3 humans (located at one edge of the dungeon) and 5 creatures (located at the opposite edge, nearby the exit). Humans *desire* to escape and creatures are hungry and will try to eat the humans. Food, shields and weapons are spread out over the dungeon (10 items in total). The layout of the dungeon and the location of objects have been randomized.

Three objective curves have been used to drive the generation in this example. These curves have simple definitions and try to capture the evolution of measurable aspects of the story that, in the current domain, match to some extent specific features of the narrative arc:

- *danger*, the perceived danger in the story, computed as the mean distance between humans and creatures.
- *success*, the level of success of characters, computed as the difference between humans that have escaped the dungeon and the number of humans that have died.
- *richness*, an additional measurement to ensure that the generation is rich enough, computed as the number of different actions that happen in the story. Richness avoids monotonous stories in which characters just find their way to the exit without any conflict.

The input objective curves for the generation are a monotonously increasing line for *danger*, *richness* and *success*, forcing the generation to produce a story with an ending in which many things have happened (richness), the creatures surround the characters at the end (danger) and all characters escape (success).

In order to keep the demonstration prototype simple, one single objective function has been used: no humans must remain in the dungeon (Figure 1). Analogously, the only constraint used for the example forbids states in which the group of humans splits up, the average distance between humans must be lower that a certain threshold (Figure 3).

An example execution would start as follows: the generation starts as shown in Algorithm 1. First, the initial state is tested against the objective function which is not satisfied because there are 3 humans in the dungeon. Perception actions are computed and every cognitive entity (humans and creatures) update their internal representation of the world with their surrounding area. Deus ex rules are processed and no action is triggered, then desire rules are examined. A human with low energy desires to get food with a high priority (escaping is postponed) and the other two still decide to escape. All creatures decide to look for food. When intention actions are generated, all characters decide to move to find what the desire and this move is realized as a successful physic action because no obstacle limits their movement.

After this step, the current values for the objective curves are computed and compared against the objective curves. The difference between the current and the objective curves is acceptable by the system (being the first step yields the resulting comparison negligible according to the thresholds). This state is thus valid and new other candidates from the initial state are similarly generated and filtered. Then one of this states in chosen (the current prototype choses the one with a higher number of actions) and the generation continues until the system has found a satisfying story.

Then, the sequence of states and their corresponding actions are converted into a textual story. The rendering of the generated *fabula* as a *discourse* has been carried out with simple, ad-hoc rules to improve the apparent result. Figure 5 shows an example. Some redundant, easy to infer events and states were filtered (Figure 4) and sequential order was used (that is, events are told in the same order as they occur). The focus on the current prototype has not been put on the quality of the discourse and only a simple method has been used. Better narrative discourse planning, however, will be tackled in future versions of *STellA*. Figure 6 shows a fragment of the rendered output. The fragment has been selected by hand, but the whole story has been taken as-is without any form of curation or human intervention. Figure 7 shows part of the underlying representation corresponding to the text in Figure 6. The example shown corresponds to the sentence "the knight was hungry".

$$
\begin{aligned}
\texttt{promising}(\mathit{story}) \leftarrow\ & hs = \texttt{humans}(\mathit{story}) \\
& \forall h_i \in hs : \\
& \quad \forall h_d \in hs - \{h_i\} : \\
& \quad\quad d_i = \texttt{distance}(\mathit{story}, h_i, h_d) \\
& av = \texttt{average}(d_0, d_1, \ldots, d_n) \\
& \texttt{av} <= \mathit{threshold}
\end{aligned}
$$

Figure 3: Example of constraint rule. A partial story not satisfying a constraint rule will not be accepted as promising and its corresponding state will not be explored.

The fragment chosen and shown in Figure 6 exemplifies the level of detail that *STellA* is able to achieve. Specific focus on some generated events can shed more light on what *STellA* is able to do. For instance, when the knight is injured by the attack of the red creature, a new set of possible next steps in the simulation are generated. In some of them, the barbarian is not aware of the event and thus there is no reaction. According to the rules, these have a low probability of happening because the barbarian and the knight are nearby. In some others, chosen before because of their higher probability, the barbarian detects the attack. Since there is a rule stating that humans defend themselves against the creatures, the barbarian could non-deterministically choose what to do, either defend or ignore the knight. The system performs a space search to choose the best option among these two, that is, non-deterministically explores partial simulations from the current one and chooses the chain that fits the curves better. Since defending the knight maximizes the number of alive heroes, that one is chosen. In this way, the simulation and the narrative-based conceptual space search pro-

```
[...]
if event action is "pass" then
    filter event
end

if event action is "move" and
       character does not face enemy then
    filter event

if event action is "get tired" and
       character's energy > 100 then
    filter event
end
[...]
```

Figure 4: Simple event filtering for demonstration purposes. The current prototype includes ad-hoc rules for redundant or excessively detailed events.

```
[...]
if kindOf(entity) = "knight" then
    print "the knight "
end

if energy(entity) < 1500 then
    print "was hungry"
end

ifenergy(entity) < 1500 then
    print "blocked "
    print attackerOf(entity)
    print " with "
    print objectDefense(entity)
end
[...]
```

Figure 5: Example rule for discourse and textual generation in *STellA*. The current version addresses simple text for demonstration purposes.

duce rich, meaningful stories.

The grounded representation allows a fine level of granularity in the action and the narrative information leads to relatively interesting scenes according to the formal metrics described in term of narrative curves and specific requirements encoded as objectives and constraints. Generating detailed interactions can provide rich content that an accurate discourse planner can aggregate where needed. However, this does not mean that any form of verbose or redundant generation can be easily fixed by a discourse planner. The content generator should be able to provide reasonably meaningful and useful content letting the discourse planner decide what is relevent for each kind of discourse.

## Discussion

The empirical evidence during the development suggests that the initial effort needed for grounding knowledge pays off soon. While more research and comparable measurements are needed to make any strong claim, the development process and the relative effort to include rules in the system is relatively reduced as the system evolves.

As previously detailed, many simulation-based story generation systems have already been created. *STellA* contributes to the field by focusing on *creativity* and exploration of a conceptual space. More specifically, several studied story generation systems perform a guided simulation in which some sort of general objectives (be it author or character goals) are pursued and fulfilled in a valid story (Lebowitz 1985; Dehn 1981; Theune et al. 2003). While the conjunction of goals and simulation links these systems with the presented version of *STellA*, the taken approach here is conceptually different: the simulation happens with *no narrative information* and the simulation is let to progress nondeterministically thus producing a growing tree of plausible states. Narrative is only included as an external process in which these successive simulations are selected as partial artifacts in the conceptual space. This puts a clear division between content generation with robust grounded generation and detailed filtering based on narrative rules. This somehow resembles the engagement and reflection model described by Sharples (Sharples 1999) and implemented in MEXICA (Pérez y Pérez 1999) in the sense that a model of creativity receives the focus.

Other story generation systems rely on the underlying narrative-like features of logging the simulation of character actions and put little or no effort on making an explicit narrative model (Klein et al. 1973; Meehan 1977). This clearly contrasts with the approaches taken by *STellA*, which specifically focus on using narrative to control which simulations are plausible according to the current objectives.

*STellA* explicitly addresses creativity both as a model and as objective. From a theoretical point of view and according to the theoretical framework described by Boden (Boden 2003) and formalized by Wiggins (Wiggins 2006), the nondeterministic simulation process would generate the conceptual space, and the mechanisms described to select and filter states would match the definition of the traversal function. The evaluation function would be composed by a mix of the curves and the objective function. The current prototype, however, is not reaching any high form of narrative creativity. The kind of story generation that *STellA* tries to achieve necessarily implies a complex management of knowledge and narrative structures. Before trying to create highly valu-

[...]
the knight was hungry.
the barbarian was injured.
the knight desired to protect the barbarian.
the green creature wanted to eat the barbarian.
the green creature tried to attack the barbarian.
the knight blocked the green creature with the shield.
the red creature tried to attack the knight.
the red creature succeeded when trying to attack the knight.
the knight was injured.
the barbarian desired to protect the knight.
the barbarian used the healing potion on the knight.
the barbarian desired to attack the green creature.
the knight desired to protect the barbarian.
the green creature tried to attack the barbarian.
the knight failed to block the green creature with the shield.
the green creature succeeded when trying to attack the barbarian.
the barbarian died.
the knight took the sword.
the knight desired to attack the green creature.
the knight tried to attack the green creature.
the knight succeeded when trying to attack the green creature.
the green creature died.
[...]

Figure 6: Fragment of a resulting story generated by *STellA* after the narrative-driven simulation process.

```
"knight0" : {position : (5, 51),
             energy : 1288,
             desire : {
                 desire : "escape",
                 agent : "knight0"
             },
             items : {"shield0"},
             kindOf : "knight",
             strength : 100,
             speed : 3,
             sight : 7,
             weigth : 90,
             known : {
                 "knight0" : {...},
                 "creature0" : {...},
                 "wall26" : {...},
                 "wall27" : {...},
                 [...]
             }
            }
```

Figure 7: Fragment of the underlying representation corresponding to the text in Figure 6.

able stories, the detailed development line tries to build a robust framework that can be further improved with more knowledge. The preliminary results show that world representation can be made richer by simulation and that a creative process can be model by non-deterministic generation and explicit filtering and identification of valuable artifacts.

## Conclusions and Future Work

Simulation is a powerful tool for modelling interactions and can produce grounded information. This information, when properly identified, can be used for driving story generation when enriched with narrative knowledge and generate a conceptual space of stories.

This paper has described the development of an updated version of *STellA*, a story generation system that implements this model that mixes simulation and conceptual space exploration driven by narrative constructions. An example output generated by the current implementation is described and the relative benefits and drawbacks of the proposed solution are discussed.

The system will continue to be developed according the discussed assumptions, namely that generating successive story states by simulating relations between characters and constructing a conceptual space by using narrative information is a plausible method for generating rich stories that can be deemed as creative by unbiased observers (Colton and Wiggins 2012). Thorough work, however, is still to be done for the system fully support these assumptions: the simulation must support richer constructions and the generation process based on narrative must be improved with more general information about narrative, probably with general models borrowed from narratology.

Studying how driven non-determinism and probabilities can lead to better results in terms of novelty is a key aspect of the future improvements of *STellA*. The future work contemplates producing and evaluating stories that include unlikely events in such a way that novelty and quality are ensured to some measurable extent.

## Acknowledgments

This paper has been partially supported by the projects WHIM 611560 and PROSECCO 600653 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

## References

Aylett, R. S.; Louchart, S.; Dias, J.; Paiva, A.; and Vala, M. 2005. Lecture notes in computer science. London, UK, UK: Springer-Verlag. chapter Fearnot!: An Experiment in Emergent Narrative, 305–316.

Baral, C. 2003. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. New York, NY, USA: Cambridge University Press.

Bell, M. 1985. Why expert systems fail. *The Journal of the Operational Research Society*.

Boden, M. 1999. Computational models of creativity. *Handbook of Creativity* 351–373.

Boden, M. 2003. *Creative Mind: Myths and Mechanisms*. New York, NY, 10001: Routledge.

Bratman, M. E. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Bringsjord, S., and Ferrucci, D. 1999. *Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a Story-Telling Machine*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cavazza, M.; Charles, F.; and Mead, S. J. 2002. Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation* 13:121–131.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI*, 21–26.

Dehn, N. 1981. Story generation after tale-spin. In *In Proceedings of the International Joint Conference on Artificial Intelligence*, 16–18.

Klein, S.; Aeschliman, J. F.; Balsiger, D.; Converse, S. L.; Court, C.; Foster, M.; Lao, R.; Oakley, J. D.; and Smith, J. 1973. Automatic novel writing: A status report. Technical Report 186, Computer Science Department, The University of Wisconsin, Madison, Wisconsin.

Lebowitz, M. 1985. Storytelling as Planning and Learning. *Poetics* 14:483–502.

León, C., and Gervás, P. 2010. The Role of Evaluation-Driven rejection in the Successful Exploration of a Conceptual Space of Stories. *Minds and Machines* 20(4):615–634.

León, C., and Gervás, P. 2011. A top-down design methodology based on causality and chronology for developing assisted story generation systems. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, 363–364. New York, NY, USA: ACM.

León, C., and Gervás, P. 2012. Prototyping the use of plot curves to guide story generation. In *Third Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012)*.

Mateas, M., and Stern, A. 2005. Structuring content in the Faade interactive drama architecture. In *Proceedings of AIIDE*, 93–98.

Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *In Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 91–98.

Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. Ph.D. Dissertation, The University of Sussex.

Riedl, M., and Young, M. 2010. Narrative planning: Balancing plot and character. *J. Artif. Intell. Res. (JAIR)* 39:217–268.

Riedl, M. 2004. *Narrative Planning: Balancing Plot and Character*. Ph.D. Dissertation, Department of Computer Science, North Carolina State University.

Rosati, R. 2007. The limits of querying ontologies. In *In Proceedings of the Eleventh International Conference on Database Theory (ICDT 2007)*, 164–178. Springer-Verlag.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale, NJ: L. Erlbaum.

Sharples, M. 1999. *How We Write*. Routledge.

Sloman, A. 1985. *Why We Need Many Knowledge Representation Formalisms*. Cognitive studies research papers. University of Sussex, Cognitive Studies Programme.

Sowa, J. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks/Cole.

Theune, M.; Faas, E.; Nijholt, A.; and Heylen, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 204–215.

Tolkien, J. R. R. 1972. *The Hobbit ; There and back again*. George Allen and Unwin London, [3d ed.]. edition.

Trentelman, K. 2009. *Survey of knowledge representation and reasoning systems*. Defence Science and Technology Organisation Edinburgh, S. Aust.

Turner, S. 1992. *MINSTREL: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA.

Wiggins, G. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7).