A Computer Model that Generates Biography-like Narratives

Samer Hassan[1], **Carlos León**[2], **Pablo Gervás**[3], **Raquel Hervás**[4]
Universidad Complutense de Madrid
Departamento de Sistemas Informáticos y Programación
[1]samer@fdi.ucm.es,[2]cleon@fdi.ucm.es,
[3]pgervas@sip.ucm.es,[4]raquelhb@fdi.ucm.es

Abstract

This paper presents an initial decomposition of the process of creative storytelling into subtasks that are relevant for studying where and how creativity plays a role from a computational point of view. Five basic subtasks are identified: building a world to act as setting for the story (including characters, locations, possible actions), generating a set of events that take place in that world, selecting from that set of events those that are worth telling, identifying a particular sequence in which to tell them, and finding appropriate linguistic realizations for each event in that sequence. To test the model, an initial prototype is presented that operates on logs generated artificially by a social simulation built by a multiagent system. A second module addresses the task of generating a textual narrative for a given log. Examples of system input and output are presented, and their relative merits are discussed. The final section discusses future lines of work that may be worth exploring.

Keywords: Storytelling, emergent creativity, social multiagent systems, natural language generation.

1 Introduction

Storytelling is an intellectual activity that is crucial to understand the way humans perceive the world, understand it, and communicate with one another concerning their own experience of it. As for other areas of cognition closely related with the human language faculty, research in this area has drawn interest from a very early stage, but actual progress has long been delayed by the inherent difficulty and complexity of the phenomena that require modelling. The general circumstances have not changed significantly, in the sense that adequate modelling of the human storytelling capacity is still a far off research goal. However, recent advances in multiagent systems and natural language generation have provided a set of tools that, properly integrated, can be used to build a simple model of the various subtasks involved in elementary storytelling.

An important obstacle on the road to achieving useful models of human storytelling from a computational point of view has been the lack of interaction between researchers addressing this problem from the different fields of artificial intelligence and literary studies. Recent joint efforts have started to establish a set of common assumptions, starting with the identification of a certain consensus on basic terminology and the identification of the set of subtasks involved in the broad general activity that is usually referred to as storytelling. The work of Gervás et al. (2006) distinguishes between several process that are involved in the generic process of building a story from scratch: creating a world, creating a story, and telling a story. In the past - as discussed by Callaway and Lester (2001) -, very little effort has been devoted to model computationally the task of creating a world, most efforts under the label of storytelling were concerned with creating a story, and only in the recent past has the task of actually telling a story as text been addressed.

This paper presents a decomposition of the process of creative storytelling into subtasks that are relevant for studying where and how creativity plays a role from a computational point of view. Five basic subtasks are identified: building a world to act as setting for the story (including characters, locations and possible actions), generating a set of events that take place in that world, selecting from that set of events those that are worth telling, identifying a particular sequence in which to tell them, and finding appropriate linguistic realizations for each event in that sequence. To test the model, an initial prototype is presented that operates on logs generated artificially by a social simulation built by a multiagent system. This simulation carries out the task of specifying the initial world (configuration of the simulation), and provides a log of events for a large set of characters emulating real life behaviour over a certain period of time (generating a set of events). A second module addresses the task of generating a textual narrative for a given log. This module carries out content determination (filtering the non-relevant events out of the total log), discourse planning (organizing a possibly large set of parallel threads of events into a linear narrative discourse), and sentence planning and realization (for the time being performed in a crude manner to allow readable presentation of the generated material). Of course, it is possible to create a different framework for this purpose, considering different stages. In the next sections we explain how the proposed framework is instantiated.

2 Previous Work

In order to develop this system, we have resorted to previous work in the fields of natural language generation and social simulations using multi-agent systems. A brief outline of the relevant studies is given in this section.

2.1 Automatic Story Telling

With a single exception (mentioned below in section 2.2), the label of *storytelling systems* has been used in the past mostly to refer to programs capable of creating a story, in the sense described above. No effort is made to create the world in which the story is to take place, and very rigid methods are employed to render the story in a textual form. In terms of how they model the creative process, Bailey (1999) distinguishes between three different approaches to automated story generation: *author models* - where an attempt is made to model the way a human author goes about the task of creating a story -, *story models* - based on an abstract representation of the story as a structural (or linguistic artefact) -, and *world models* - where generating a story is seen as constructing a world governed by realistic rules and peopled with characters with individual goals, and the story arises from recording how the characters go about achieving their goals.

MINSTREL (Turner, 1994) and MEXICA (Pérez y Pérez and Sharples, 2001) would be classed as examples of author models. Systems based on story grammars (Rumelhart, 1975) fall under the category of story models. Tale-Spin (Meehan, 1977), the classic Story Generator inspired on Aesop's fables, and recent efforts of story telling based on planning (Riedl and Young, 2006) are based on a world model.

A possible explanation of this diversity of approaches can be found if one considers them as partial approximations to the overall complexity of the problem. Under this interpretation, each approach is focusing on a part of the problem - the decisions required from the author, the form of the story, the restrictions imposed by the world -, and simplyfing the whole by omitting the others. Two interesting considerations arise from this hypothesis. ¿From the point of view of modelling the process, it seems that a model that takes several of these factors into account may provide greater representational power, though it may run the risk of becoming too complex to be computationally useful. ¿From the point of view of creativity, it raises the question of whether the perceived creativity of a program based on a partial model - which is modelling and controlling only a subset of the elements in play - arises from the freedom allowed in the elements that are not being modelled. Evaluation of program results should attempt to establish whether the elements that produce the impression of creativity are indeed being modelled by the program.

2.2 Natural Language Generation

Natural language generation is important for a study of storytelling because it involves both a model of the task that need to be carried out to generate a valid text - therefore partially modelling the activity of an author - and a model of the story as linguistic artefact - a story model.

Reiter and Dale (2000) define the general process of text generation as taking place in several stages, during which the conceptual input is progressively refined by adding information that will shape the final text. During the initial stages the concepts and messages that will appear in the final content are decided and organised into a specific order and structure (*content planning*), and particular ways of describing each concept where it appears in the discourse plan are selected (*referring expression generation*). This results in a version of the discourse plan where the contents, the structure of the discourse, and the level of detail of each concept are already fixed. The *lexicalization* stage that follows decides which specific words and phrases should be chosen to express the domain concepts and relations which appear in the messages. A final stage of *surface realization* assembles all the relevant pieces into linguistically and typographically correct text. The tasks of referring expression generation and lexicalization are known as *sentence planning*.

The natural language generation work presented in this paper is mainly centered around content planning.

The work of Callaway and Lester (2002) stands out as the most significant effort in the field of natural language generation to address the specific challenges posed by narrative texts. It relies on having an external planner that defines the outline of the intended story, and it carries out elaborated sentence planning to produce input for a surface realizer.

2.3 Social systems

The role of social systems in the current research is to provide a first approximation of a model of the world, which has been identified as an important ingredient of the storytelling capability. A multi-agent system (MAS) consists of a set of autonomous software entities (the agents) that interact among them and with their environment taking decitions. The agent paradigm assimilates quite well to the individual in a social system. With this perspective, agent-based simulation tools have been developed in recent years to explore the complexity of social dynamics.

The MAS presented in this paper is based on a previous social simulation by Pavon et al. (2006). In that work, the agents were developed with several main attributes: from simple ones such as sex or age, to complex ones, like for example ideology or educational level. The population in the agents' society also experiments demographic changes: individuals are subject to some life-cycle patterns. For example, they get married, reproduce and die, going through several stages where they follow some intentional and behavioural patterns. The agents/individuals can build and be part of relational groups with other agents: they can communicate with other close agents, leading to friendship relationships determined by the rate of similarity. Or, on the other hand, they can build family nuclei as children are born close to their parents.

3 Story Generation

Each of the stages outlined earlier is described in more detail.

3.1 The Social Simulation: Creating the World and the Story

Based on the ideas mentioned, several changes to the original MAS from Pavon et al. (2006) have to be made in the perspective of execution to be able to generate "life logs" of the individuals, which will be the basis for the texts describing the storyline. It is necessary to shift the point of view from trends data acquisition to vital biographies. We do not need numerical data, but semantic content that can be interpreted by the rules as we interpret them, because we want the story generation to be as close as possible to what humans might have done faced with similar sets of events. In this framework, we adapted the MAS for a Fantasy Medieval World, as it is described in León et al. (2007). Thus, for every single individual we have a Name and Last Name. Moreover, this Last Name is inherited: this will be useful for telling the stories of lineages, and for personal events. And every agent has a race, so they can be elves, humans, dwarfs... For each agent there is now a random possibility of dying, allowing the possibility that we can relate this early death to the betrayal of a friend, poisoning by a wife, a mysterious accident... This cause-consequence link is very simply implemented, only based in the last event happenned, but it should be improved with a "memory" of the agent that would lead its future actions (for ex. with a BDI arhitecture).

Following the cited objective of emulating life behaviours, the MAS presented here introduces context dependant relationships and life events: usual life events were not exciting enough to build a fantasy adventure. And so, an individual can have friends and enemies. Along his path, he can get married and have children, but he also can, randomly, suffer several spells (loss of memory, fireball or even change of sex!), kill horrible monsters (ogres, dragons), get lost in mazes or dark forests, find treasures and magic objects in dangerous dungeons,... In this way we can build a really amazing (and sometimes weird) story, with several characters that evolve and interact among them.

At the end of simulation, this collection of events, together with the agents' characteristics, is exported to a context-independent XML file. This file will be imported by the content planning module that will continue with the process of generating a story from the lives of the most interesting of these agents.

Here we present an example of the output of the social simulation with the important information of each agent. The data for every agent is listed: the initial ones, together with the next generations that appear during the simulation. Here we explain briefly the one corresponding to the individual that will be selected as star of our example story. The data for each character is divided in two main sections. The first one (Table 1) corresponds to the characteristics of the agent. Each attribute of the character has two parameters, expressed as attributes: its ID (identifier of the attribute) and its Value. The value of these keys is, of course, context-dependent: they represent aspects like its race or how religious the character is.

Id	Name	Last name	Race	Sex
i212	Jeanine	Avery	human	female

Table 1: Attributes of a character

The second main section (Table 2) is the collection of life events, associated with the time in which they took place. As in the previous sections, attributes are context-free, but values of these attributes depend on the context. Thus, we can read in the full log that in the year 515, the human Jeanine Avery suffered a spell that transformed her into a frog. Or, analyzing the chain of events, we can see that the impossible love of her youth was, after she grew to be an adult, her formal couple, giving her many children and living happily... at least for some years.

Id	Time	Action	Param
e9	515	spelled	frog
e10	515	impossible love	i229
...			
e14	526	couple	i229
e15	526	child	i258

Table 2: Events of a character

3.2 A New Rule-Based Story Planning System

The work presented in this paper is a new version of a previous content planning story generation system described in León et al. (2007). The input for this program is a description of a set of characters with their facts and attributes, and the output is a filtered and ordered set of those elements, in such a way that the final generated discourse is intended to be much more legible for a human than the original set of facts. The design is oriented to generate stories for a group of agents coexisting and creating relations between them, in this way emulating the story of a real society, with possible emerging sub-stories that can be narrated.

The previous version of the content planning system executed a ruled based algorithm to describe the story of one of the agents, telling about some important facts of some of the most important agents or characters which had any relation with the protagonist. Although the results were somehow interesting, we have carried out a new version able to generate more complex stories, not only focusing on the main character, but also on other agents. With this approach, new narrative structures can be generated, like simple conversations and changes of narrative focus.

3.2.1 Interest

The content planning system uses a rule based algorithm. The rules that directed content determination in the first version of this program were based on some numerical factors that added information to the characters, making it easier to choose which of them should be the protagonist. This new version, however, reduces the role of these values, and only computes and uses one of them, the *interest*. The rest of the information needed for creating an ordered an filtered story is included in the algorithm using more powerful rules and, as we explain later, a *focus*.

The *interest* ($I(X)$) is the importance that we assign to a character because of their facts. We have a table that gives more or less interest to each of the possible facts. Of course, these values are fully domain dependent. Formula 1 shows how to compute this factor:

$$I(X) = \sum_{i=1}^{n} f_i \cdot h(X, i) \qquad (1)$$

where f_i is the interest that we assign by hand for the fact i, X is the character, and $h(X, i)$ is the weight for the appearance of i in the life of X. The value of h is calculated with the type of i (what kind of fact it is) and with the attributes of X (if it is an elf, or an orc). There are, of course, some other ways of assigning this interest. However, doing it in this way we can easily put many information of the domain about the meaning and the importance of each fact.

3.2.2 *Focus on the Characters*

Interest is not enough for creating a good story. We also add some other rules to perform discourse planning, based on templates for creating an ordered story and telling stories about several characters. The rules we have chosen model in a very simple way the mental rules humans use to apply on these kind of creative tasks. However, as explained before, the previous version was only able to generate a story for a single character. We have now a different set of rules for the system: the *focus*.

The *focus* establishes which character is the main one at some stage of the story. We can consider it as a light that illuminates a particular actor in a scene. When some character has the focus, the rules generate parts of its story, as if the attention of the public centred only on a particular actor in a play. The focus can be also be understood as a "token" that is passed between the characters, and the character that possesses this "token" can add some of their facts, in the order that it decides based on some rules.

One of the most important aspects of the focus is when to change it. If the focus is only established on a single character, the system will never be able to generate a story of more than one of them. So we have to add some rules that decide when to change the focus, and which character should be the next "main actor". These rules are introduced in the next section.

3.2.3 *Rules*

The rules that govern the behaviour of the system are based on the explicit information stored in the *interest*. These rules first decide which character is the main one, choosing the character with higher interest. The story of this character will direct the main thread of the narration. So once we have chosen this character, we give it the *focus*. Then the character applies some rules to decide which facts are going to be added to the structure of the story. These rules are based on the *interest* of the facts (we only tell the most important aspects of an agent life) and the time (usually, ordering facts in a time sequence). In this way we create an ordering between the facts, creating the discourse.

When the rules for narrating a part of the life of a character decide to add to the final story a fact related to another character, we change the focus to that new character. For example, if a relation with high interest (like an important enemy) appears in the life of some character, we change the focus to that enemy, trying to show the most relevant facts about him. However, in the application of the rules along the generation process, we always keep the information of which character is the protagonist, to be able to change the focus back to him/her.

3.3 Sentence Planning

The final generation of the story is not only a nice way of showing the results. It can make the discourse interesting or boring, even if the order of the facts resulting from *discourse planning* is good or bad, respectively. Thus, we cannot ignore this step if we want to evaluate the generated content. It is not the same to say "Elrond was an Elf. He had a daughter called Arwen. Elrond was friend of Aragorn.", as to say "Elrond the Elf, father of Arwen, was friend of Aragorn the King". The final form of sentences not only gives beauty to the text, but may also convey information not actually present in the data structure. We can infer, in the second sentence, that Elrond is somebody important, as Arwen, and Aragorn is going to play a main role in the story. This knowledge is not contained in the first sentence. To achieve computational modelling of these characteristics is currently beyond the scope of this paper, but we intend to address it in future work.

Two different stages can be clearly differentiated during sentence planning: referring expression generation and lexicalization. In the work presented in this paper both of them have been treated in a simple way, with the intention of providing a first approximation to solve the problem while identifying the kind of decisions that must be taken in the future.

For the generation of references we have implemented a solution where all occurrences of the concepts mentioned are treated as definite references, and without using pronouns. Even when these references are quite simple, it is necessary to handle information about their number - singular or plural - and whether they are proper nouns or not. This information is stored in an elementary knowledge base containing the required information for each of the concepts appearing in the discourse generated by the content planning stage.

During the lexicalization stage two distinct tasks are addressed. On the one hand, for each reference appearing in the text a lexical tag must be chosen. For the moment the bare name of the concept is used, but in the future the system would work with a dictionary where every concept has assigned one or more lexical tags that are appropriate to express the meaning of the reference. On the other hand, for each action that is present in the discourse not only the lexical tag corresponding to the verb is required, but also the structure of the resulting sentence. For example, a suitable sentence for the conceptual representation of the action 'to be born' would be in passive voice and usually accompanied by a locative or temporal adjunct, as in "She was born in Rivendel" or "She was born in 1980". This information is stored in a syntactic knowledge base, where each kind of sentence is associated with the type of appropriate adjunct that can accompany it. In the current implementation the set of possible verbs is quite reduced, as well as the adjuncts corresponding to them.

3.4 An Example

Now, we show a real example of our application. The multi-agent system is capable of running parametrized simulations, changing the number of characters, probabilities of the facts, years of simulation, and all other attributes of the system. Once executed, the system generates logs in XML.

At this stage, the story generation application reads the resulting XML file, and outputs a text. This example is the result of a simulation of the life of 200 initial characters and their descendants over a time span of 80 years. The system has inferred who is the most important character, and it produces the following rendition of her mortal life:

> *Jeanine Avery was born in 520. Jeanine Avery was saved by the priest. Jeanine Avery killed the ogre. Jeanine Avery was involved in the battle. Jeanine Avery was enchanted with the marvellous spell of the frog. Jeanine Avery killed the dragon. Jeanine Avery was lost in the forest. Jeanine Avery met Luisa Brandagamba. Luisa Brandagamba was born in 529. Luisa Brandagamba met Jeanine Avery. Jeanine Avery killed the ogre. Jeanine Avery fell desperately in love with Bobbie Beasttongue. Jeanine Avery inherited the castle. Jeanine Avery met Pogor Brandagamba. Pogor Brandagamba was born in 529. Pogor Brandagamba killed the ogre. Pogor Brandagamba met Jeanine Avery. Jeanine Avery grew up. Jeanine Avery fell desperately in love with Bobbie Beasttongue. Jeanine Avery met Haurk Avery. Haurk Avery was born in 542. Haurk Avery found the magic sword. Haurk Avery met Jeanine Avery. Jeanine Avery was lost in the labyrinth. Jeanine Avery found the treasure. Jeanine Avery was enchanted with the marvellous spell of the memory. Jeanine Avery was enchanted with the marvellous spell of the frog. Jeanine Avery was involved in the battle. Jeanine Avery killed the ogre. Jeanine Avery killed the dragon. Jeanine Avery was involved in the battle. Jeanine Avery was lost in the forest. Jeanine Avery found the treasure. Jeanine Avery killed the dragon. Jeanine Avery was betrayed and killed by Morrain Avery.*

4 Discussion

An important advantage of the proposed model is that it allows separate discussion of the degree of creativity achieved at each stage, and considerations of whether the corresponding creativity can be attributed to the program, the programmer, or randomness.

4.1 World Construction: How Creative is it?

Even though we have defined a full context where the action takes place, there still remains lot of work to do for building the world. This MAS can be configured in many ways that lead to completely different results. But this fact does not mean that there is a way of predicting the result: as a social system, its nature is non-deterministic. Although we can partly control the agents' behaviour, for the most part it remains uncontrolled. Only through evaluation and statistics can wededuce which configuration is the most convenient to our aim.

To achieve this, the system was tested changing the values of the parameters and analyzing the results obtained for each configuration. Initial testing was directed to find the base parameters for other testing processes. It revealed that hundreds of individuals were needed to provide enough interactions to extract emergent behaviour and to express enough complexity, following logical reasoning. We found too that if we wanted to tell the life story of a person, we needed to know at least a number of decades of its life events. On the other hand, we can not use thousands of agents due to efficiency issues (the analysis of a log of hundred of MB is costly in computational terms). Thus, we left constant the size of the initial population (200) and the time of simulation (50 years) for all the configurations tested (together with other minor parameters like the space size or those dealing with graphical presentation), and only the demographical parameters are modified. We chose 200 agents because it is a number that the system can handle in reasonable periods of time, but big enough to produce good results.

We say a system is "stable" if it returns a similar output in different executions with the same configuration. New evaluations with those fixed parameters revealed that the stability of the social simulation system was critically affected by the parameters chosen. Furthermore, we found that the same parameters were dramatically influencing the diversity of the population. And diversity is the main path to find creativity.

So, we identified these critical parameters as:

- Mean of children per couple: a simple number that specifies the mean of the normal distribution that define how many children a couple have. It has two typical parameters, corresponding to the means in a developed country (2) and in a typical African one (5).
- With/without initial kids: a complex parameter that reflects the amount of kids in the initial population. In the original sociological MAS there were no kids initially, because all the data of the agents were imported from surveys... and kids do not complete surveys. We can force the appearance of kids reducing

the ages of all the agents of the initial population.

The importance of both parameters can be easily explained. About the first one, we can say that if every couple has more children, there will be more interactions, more friends, more complexity. If we add the fact that sons and daughters are born close to their parents forming family nuclei, the system dynamics tend to self-organize in clusters of friends and families concentrated in the space. Newly born agents grow in a rich environment full of people, so they can generate lots of events and easily find a spouse, better than more isolated ones. Evolution does the rest. After decades of simulation, only the clusters survive and grow.

On the subject of whether to have children in the initial population or not we can say that, without them, the size of the population tends to get lower and lower with time, because there is not a new generation that substitutes the oldest ones that are dying. And moreover: the characters with an "interesting life" are the ones that have enough time to do lot of "interesting things"... If lot of them begin the simulation with 45 years or more, and they tend to die around 65, they do not have much time to "be heroic". The same could be said if an agent is born 5 years before the end of simulation: its probabilities to be "interesting" are very rare.

We can define four different configurations based on these two parameters: Without/2, Without/5, With/2, With/5. We will analyze some evaluation results of them as shown in the following figures and explanations. These figures reflect the analysis of dozens of executions and tests.

In the four configurations, we will observe the total number of individuals simulated (the 200 initial ones plus the born ones) and the size of the family (average and maximum, minimum gives no information). The initial people have no parents and no initial relationships between them.

In Without/2 the number of individuals has a very high stability, with variations around the 5% between executions. Because to both critical parameters tend to reduce the population there are only a few births. Thus, average size of family is very close to the minimum. In Without/5, with each parameter pushing in a different direction, we can see that the "Without" one is stronger: the population still decreases. The family size doubles and the maximums are incredibly higher: three times the Without/2 (around 15).

In With/2, again we can see how "With" prevails: the population increases, but only around a 45%. Here we appreciate a logical increase of the average family size (around 3.0), but less than it could be expected: although here we have a rise of population, it is because we avoid deaths, not because children. We can realize the difference better looking at maximum family size, that here reaches only 10. In With/5, we have what we could expect: an incredible grow of the population size. With crazy executions always completely different (with differences that reach the 200 births), it results an average growth of 275%, an average family of 6.2 and a maximum family size of 18. Now, none of these facts could surprise us. The unstable executions of With/5 are shown in Figure 1 compared with the stability of others.

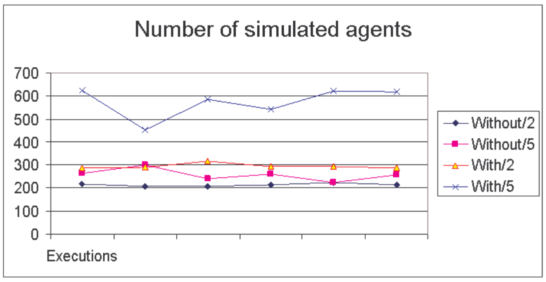

Figure 1: Comparison between configurations, attending to the number of simulated agents

After reviewing all the possibilities, the chosen one would be the one that has a good amount of interactions together with lot of diversity in the population and a reasonable family size.

Dealing with races, we can see how the micro-decisions (taken by the agent) determine the total evolution in a kind of butterfly-effect (chaotic). We force the initial population to have approximately 20% of each one of the five races. But it does not matter which configuration we choose: the percentage will never remain stable. The reason is simple: each agent becomes friend or enemy of someone depending on the similarity between them. And when an agent has to choose its spouse, it chooses the most similar of its friends (some other minor restrictions are included). But an elf and an orc have nothing in common: they dislike each other, and will be extremely difficult for them to marry. Because of that, the number of orcs tends to decrease (if one is in a environment without orcs, he will not find a couple nor have children). Humans can easily cross with other races (according with fantasy middle-age stories), so their adaptability allows them to increase their number more than other races. This way we are modelling the fantastic world, and it makes sense: we will not find strange when the human Aragorn hates orcs and gets married with the elf Arwen. Figure 2 reflects this explanation graphically.

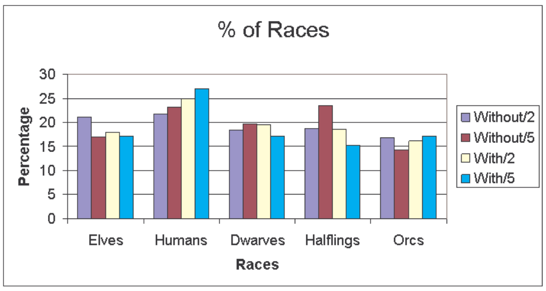

Figure 2: Percentage of races depending on the configuration

Even though we have reviewed the main possibilities of the demographical model types, we cannot decide which one would be the best one for our tale. This is a task of the next step: story building.

4.2 Story Construction: How Creative is it?

We have a complete world, with a demographical evolution, that serves as context. Now we focus on the facts to be told in the story, that have two main characteristics: they have to be "interesting" (our characters are adventurers, not trees that only know how to grow, reproduce and die) and they have to be restricted to this world (Aragorn will not have a helicopter). Because of the first characteristic, we chose to include the random context events (as it has been described) and the relationship of "enemies". Because of the 2nd one, we chose the events carefully, applying our knowledge of the fantasy context. In this way we can analyze, as was done in the previous subsection, the friends, enemies and number of events of each configuration, from the "interest" point of view. Maximums are particularly interesting because they will reflect the "heroic" characters, that will do more things than usual, and so will be more attractive for an exciting story.

For Without/2, with not much population, the interactions between agents are not significant. This simulation is dramatically poor: an average of not even 4 friends per individual and maximums of 10 reveals it. With many dying in the beginning, the average number of events does not grow more than 20... Although this amount is doubled by the "heroes", it is too poor to be considered. We should have much more "interesting" ones to analyze, so we can decide about a really amazing one, or about crossing two interesting and connected stories. For Without/5 we increased the amount of friends/enemies by a 50%, but still not enough: 5 friends and 2 enemies for a whole life is not what we are expecting. The events have nearly no increase, and the maximum is just a 15% higher.

In With/2 we can see, at last, good averages. The amounts of friends/enemies have doubled respect to Without/5, and in the special characters we can see dozens of friends/enemies. Besides, a big increase in the number of events is observed . This configuration could be selected for our purposes. With/5, as in the other subsection, gives crazy results. We can see a huge increase in the number of friends/enemies (the maximum, with 62, doubles With/2!) and, even though the average of events did not change a lot, the maximum is incredibly high: an average maximum of 125 means that the amount of heroic characters is very significant. Figure 3 shows the main event results.

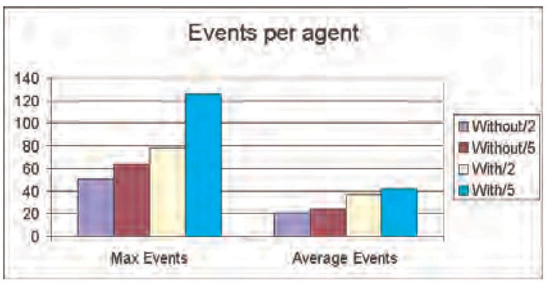

Figure 3: Comparison between configurations, attending to the number of events per agent

A note about efficiency: even though all the configurations ran with no problems of time or memory, the With/5, due to its huge amount of agents and interactions, is significantly slower than the others. Moreover, it outputs a XML log much bigger than the other configurations. Attending to the whole discussion and reviewing the possibilities commented, the chosen configuration will have a good amount of interactions, together with the diversity necessary for finding many "heroes". The With/2 and With/5 are good candidates. Choosing between them depends only on the length and complexity of the story that wants to be told and efficiency issues of the other modules.

A good story must explain why a fact occurs in the story. Only with random events this is, of course, not possible (there is no explaining possible for a random event). We can see, in the example (3.4), that in the story there are some gaps. These gaps are relative to the randomness of the system. This is an important issue that has to be improved.

4.3 Content Planning: How Creative is it?

To obtain a creative content planning system is definitely not an easy task. A truly creative program able to emulate human creativity should have mechanisms for understanding the source logs, creating an intention for the discourse plan, and performing some operations for threading the final story. Of course, nowadays this a very ambitious project, and we have to give little steps towards this ideal system.

In this paper we have presented some progress. As explained before, textual content generation from the facts of the story are generated with rules. This set of rules can be more or less large and complex, but in this kind of systems they will always direct the generation, and thus the quality and the creativity of the resulting text.

In this sense, the creativity relies on the quality of the rules, which is directly linked to the creativity of the human responsible for writing them. In this way, we can say that the human puts the creativity into the system, and the system only reproduces the information written by the human.

This information is not only explicitly deposited on the system with the rules, but also with the *interest* and the *focus*. As explained before, *interest* is stored in a table created by a human, and the focus traffic is guided by rules. Again, the creativity of the content planner is dependent on the human creativity. It is important to note that in this version the rules that govern focus changing are not very good, and the focus swings too much from one character to another. This has to be improved.

4.4 Sentence Planning: How Creative is it?

For the moment the implementation of the sentence planning stages of the process are systematic, and therefore they can not be considered creative. Certain improvements are possible which may result in a higher quality of the output texts. However, this improvement in quality is in truth more related with issues of style than with creativity.

During the generation of referring expressions some of the decision processes involved can be improved. For instance, imagine we are referring to a girl about which the system knows that is pretty and is daughter of a king. If the references to this concept are directly translated from the available information we would obtain sentences such as “The girl was pretty. She was the daughter of a king”. However, the final text would be more natural if it is capable of inferring that the girl is a princess and referring to her as a princess in the rest of the text. This involves providing the system with a certain amount of knowledge that it can use to simulate more inteligent behaviour.

Also in the lexicalization tasks there is space for improvement. When choosing the lexical tag to use for a concept, it seems common that the dictionary had more than one word for each of these concepts. Depending on the previous appearances of the concept, the system can choose between synonyms or hypernyms using heuristics about the style of the discourse in a specific moment, or the emotion that the text is trying to transmit.

5 Conclusions

We have presented a system where interactions between agents over a long period of time can be told in natural language automatically.

We have shown a particular way of generating the stories, based on rules. We have explained a three-step process for performing this task, and we have verified that for *discourse planning*, the rule-system is very dependent on the domain, and the desired type of story.

The results of the system are less impressive - when rendered in a readable text format - than they might have been if the system included an elaborate sentence planning module. The current version is just a skeleton implementation that lets down an otherwise acceptably selected and planned discourse.

The division of story telling into five tasks is envisaged as a generic analysis of the process, in the sense that it should be applicable to all storytelling systems. This does not apply to the sequential manner in which they are carried out in the prototype described. For different systems, the results of some of these tasks may be provided as input (for instance, descriptions of the world are given to story telling systems based on planning, or discourse plans provided to Callaway’s StoryBook system). A different alternative is to avoid altogether explicit modeling of some of these intermediate results. For instance, the creation of the story world may not take place explicitly within the system, and simply be left to emerge in the reader’s mind from the sequence of events that is built by the system. This does not make the proposed model less valid. Whether a particular task is modeled explicitly in any given system, outsourced to the user (or a different system), or left to emerge implicitly from the results of other tasks, it remains true that a story (and hence a story telling system) may/should be evaluated at these five different levels.

Acknowledgements

This work is partially supported by the Spanish Ministry of Education and Science project TIN2006-14433-C02-01, and research group grant UCM-CAM-910494, jointly funded by Universidad Complutense de Madrid and the Comunidad Autónoma de Madrid (Dirección General de Universidades e Investigación).

References

Bailey, P. (1999). Searching for storiness: Story-generation from a reader’s perspective. In *Working Notes of the Narrative Intelligence Symposium, AAAI Fall Symposium Series.*, Menlo Park, CA. AAAI Press.

Callaway, C. and Lester, J. (2001). Narrative prose generation. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1241–1248, Seattle, WA.

Callaway, C. B. and Lester, J. C. (2002). Narrative prose generation. *Artif. Intell.*, 139(2):213–252.

Gervás, P., Lönneker-Rodman, B., Meister, J. C., and Peinado, F. (2006). Narrative models: Narratology meets artificial intelligence. In Basili, R. and Lenci, A., editors, *International Conference on Language Resources and Evaluation. Satellite Workshop: Toward Computational Models of Literary Analysis*, pages 44–51, Genova, Italy.

León, C., Hassan, S., and Gervas, P. (2007). From the event log of a social simulation to narrative discourse: Content planning in story generation. In *AISB’07, Artificial and Ambient Intelligence*.

Meehan, J. R. (1977). Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass. Morgan Kaufmann.

Pavon, J., Arroyo, M., Hassan, S., and Sansores, C. (2006). Simulacion de sistemas sociales con agentes software. In *Actas del Campus Multidisciplinar en Percepcion e Inteligencia, CMPI-2006*, volume I, pages 389–400.

Pérez y Pérez, R. and Sharples, M. (2001). Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(2).

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

Riedl, M. and Young, R. M. (2006). Story planning as exploratory creativity. *New Generation Computing - Special Issue on Computational Creativity*, 24(3-4).

Rumelhart, D. (1975). Notes on a schema for stories. In Bobrow, D. G., editor, *Representation and Understanding. Studies in Cognitive Science*, New York. Academic Press.

Turner, S. R. (1994). *The Creative Process: A Computer Model of Storytelling*. Lawrence Erlbaum, Hillsdale, NJ.