Evolving the INES story generation system: from single to multiple plot lines

Eugenio Concepción[1] and **Pablo Gervás**[1,2] and **Gonzalo Méndez**[1,2]
[1]Facultad de Informática
[2]Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
{econcepc,pgervas,gmendez}@ucm.es

Abstract

INES *(Interactive Narrative Emotional Storyteller)* is a story generation system based on the Afanasyev framework. It is focused on generating stories by combining template-based plot generation with an agent-based simulation of characters' interaction. Its design follows the microservice-oriented model established by Afanasyev, in which a Story Director orchestrates the story generation stages, implemented by specialised microservices. While this model is suitable for generating a single plot story, it is insufficient for managing a multiple plots scenario. This paper focuses on describing an evolved version of INES that aims at generating stories that contain different plot lines. In addition to the adoption of changes in the story representation model, the adaptation entails a modification of the operation of INES and includes a new microservice: the Plot Weaver. This component introduces the application of a literary technique referred to as the *"Communicating Vessels"*, in which the different lines evolve in parallel while interacting between themselves.

Introduction

Automated story generation is a research area in Artificial Intelligence focused on developing systems whose result is a story (Gervás 2012). It is closely related to other Computational Creativity areas such as interactive storytelling (Glassner 2009).

Story generation systems present two distinctive characteristics: they strongly depend on knowledge and they can only generate a constrained variety of stories (Gervás and León 2014). A story generator requires knowledge from practically all areas of the collective wisdom, so it needs to be fed with a wide range of information, from the most basic common sense knowledge to the physical world rules. Besides this, due to the technical limitations of the generation process, story generators only generate stories of a certain kind –in terms of theme, rhythm, discourse, etc. Many of these limitations come from the fact that the architecture of these systems is built according to a monolithic design. Hence, a single application concentrates all the required functionality and assets. If this is combined with the lack of architectural mechanisms for collaborating with other systems, the results obtained are quite restrictive. A way of addressing all these limitations is by adopting a distributed architecture, with an emphasis on the collaboration between different systems.

INES (Concepción, Gervás, and Méndez 2018b) is a story generation system based on Afanasyev, a microservice-oriented architectural framework (Concepción, Gervás, and Méndez 2018a). Afanasyev provides a reference architecture that brings a collaborative environment for services sourced from different storytelling systems and orchestrated by a Story Director. INES was originally developed to test the suitability of the framework and also as an evolution of the Charade storytelling system (Méndez, Gervás, and León 2016). The limitation of the current version of INES comes from the fact that it can only generate a single-plot story, being its design inadequate for managing a multiple plots scenario.

Background

TALE-SPIN (Meehan 1977), one of the first story generators, wrote short stories about the inhabitants of a forest. From a technical point of view, it applied planning techniques (Cohen and Feigenbaum 2014) for generating the characters actions –while trying to achieve their goals, and then it wrote up the story by narrating the steps performed by the characters for achieving their goals. **Author** (Dehn 1981) was also a planner but focused on the authorial goals instead of the characters' goals. **Universe** (Lebowitz 1984) generated scripts for a TV soap opera by focusing on characters interaction. Its generation process included a planning stage that kept track of pending goals for developing a partial draft of the story until plot completion. **GESTER**, *GEnerating STories from Epic Rules*, (Pemberton 1989) was one of the first approaches towards generating stories from interacting modules of independent knowledge. The program was a rule-based story generation system that managed information about story structure, in the form of a simplified version of a narrative grammar, and to the possible events and actors of the epic sub-genre. **Brutus** (Bringsjord and Ferrucci 1999) generated short stories about betrayal. It used a very thorough knowledge model for representing the concept and implication of betrayal. It also provided a grammar-based generation model and a literary beautifier, which allowed it to generate high-quality stories, providing texts that could have been written by humans. **MEXICA** (Perez y Perez 1999) was a storytelling system that generated mythological stories about the Mexicas, the early inhabitants of Mexico. It was the first system that brought the character's

emotions into play in the generation process. **MAKEBELIEVE** (Liu and Singh 2002) generated short fictional stories using common sense knowledge to generate them. It required the user to provide a story about a character as initial seed, for later attempting to continue that story by inferring possible sequences of events that might happen to that character. **TEATRIX** (Machado, Paiva, and Brna 2001; Prada, Machado, and Paiva 2000) was a virtual environment for story creation, designed to help children and teachers to understand the whole process of collaborative story creation. It provides an environment where both drama and story creation are merged into one medium. Architecturally, it is an agent-based system in which each character is performed by an intelligent software agent interacting in the story world. Every character plays a role according to the Propp's folktales model (Propp 1968). **Fabulist** (Riedl and Young 2010) is a whole architecture for automatic story generation and presentation which combines both the author interests and the characters intentionality. **Charade** (Méndez, Gervás, and León 2014; 2016) is a storytelling system focused on the relationships between the characters. By simulating their interactions, it tracks the evolution of their mutual affinities and applies it for generating stories. From an architectural point of view, it is an agent-based architecture implemented in JADE (Java Agent Development Framework) (Bellifemine, Poggi, and Rimassa 1999). Charade aims at implementing an affinity model decoupled from the story domain, that is, the world in which the story takes place or any other context-related attribute of the characters.

Related work

Single-plot stories are not the only model in human-made narrative. There are a good number of stories that contain more than one plot, such as unnatural narratives, a subset of fictional narratives that subvert the physical laws, the logic principles or the standard human limitations (Alber and Heinze 2011). This approach affects not only the facts told in the story, but also its structure: it can contain several interwoven plots, a rupture of the plot natural progression, multiple concurrent plots and many other possibilities. From the beginning of the literature, the unnatural elements have been present in the literary production (Todorov 1975).

In addition to the historical precedents, postmodern literature has adopted much of those unnatural resources, bringing a disruptive narrative design to the stories (Martínez 2011). Contemporary literature is fraught with stories containing several plot lines linked together by means of diverse strategies (Menéndez 2013).

There is not a long record of multiple plot generation and combination in automatic story generation. One of the remarkable works in this respect is the plot weaving algorithm proposed by Fay (Fay 2014). This is a method that takes a set of individual plot threads as input, one for every single character, and generates a new story by tying them. These individual plot threads have been previously extracted from a preexisting story by means of the *Genesis* story understanding system (Winston 2016). This plot weaving algorithm makes sure that characters' plots are compatible and it also takes care of building a consistent timeline for all of the plot elements of the story. This procedure is computationally difficult because it entails selecting the best set of pairings of characters to generic entities to create the best possible combination for the story (Fay 2014).

Porteous et al. (Porteous, Charles, and Cavazza 2016) have developed a remarkable example of multiplot interactive storytelling system. It is focused primarily on facing three challenges: the distribution of the characters across the different subplots, the length of each subplot presentation and the transitions between subplots.

Gervás (Gervás 2018) has recently explored the suitability of combining events from a sequence for generating a plot as a technique for story generation. This approach has partially influenced the solution proposed in this paper.

It is also worth mentioning other efforts in the generation of multiple possibilities in a story, such as the planning approach by Li and Riedl (Li and Riedl 2010) and the *Crystal Island* (Mott, Lee, and Lester 2006) interactive narrative engine. In the first case (Li and Riedl 2010), authors define a plan refinement technique based on partial-order planning that it uses for off-line adaptation of authored narratives with multiple "quests" adapting the plot line to create new plausible sequences of actions. The Crystal Island generation model offers multiple quest subplots that encourage the user goal recognition, combining multiple plot elements to create rich customized stories (Mott, Lee, and Lester 2006). However, whilst these approaches sought to generate multiple quests, they did not provide a procedure to interleave them.

Materials and methods

This section covers the technical basis and the conceptual techniques considered for designing the solution. It firstly provides a review of INES, with emphasis on those aspects that mainly take part in the plot generation step, and some known literary techniques for creating stories based on multiple plots. It also focuses on reviewing the knowledge representation model used by INES, taken from the Afanasyev common knowledge representation structure. Lastly, it describes different literary strategies for writing multiple plot narratives.

The INES story generation system

INES (Interactive Narrative Emotional Storyteller) is based on the Afanasyev framework (Concepción, Gervás, and Méndez 2018a; 2018b). One of the declared objectives of this framework is to facilitate the generation of stories much closer to human-made literature by combining the diverse capabilities of various story generators (Concepción, Gervás, and Méndez 2018a). Afanasyev provides both a microservice-oriented reference architecture and a common knowledge representation model.

The architecture of INES is based on microservices. Figure 1 depicts a high-level view of its original components. The following lines focus on a short review of the relevant aspects of the architecture. A more detailed description can be found in (Concepción, Gervás, and Méndez 2018b).

The operation of this microservices ecosystem is driven by the Story Director. It orchestrates the rest of the ser-

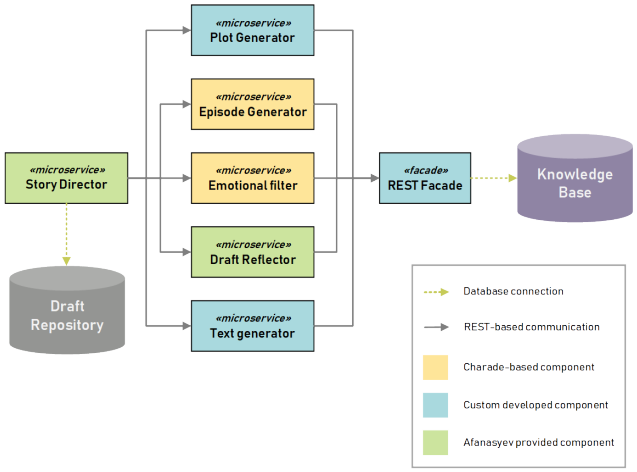

Figure 1: Original architecture of INES.

Episode	Description
Initial state	A peaceful community
Arrival	The arrival of the outsider to the community
Outsider destructive actions	The outsider acts against the members of the community, performing destructive actions, without being uncovered
The outsider revealed	The true evil nature of the outsider is revealed
The rise of the heroes	The The heroes rise from the community and fight against the outsider
Purge	The outsider is purged. The community becomes peaceful again

Table 1: The "Destructive Outsider" story plot template

vices to perform the story generation process. Story Director starts by requesting the Plot Generator to generate a plot. This request includes a list of initial characters, the plot template to apply and the setting in which the story happens. Every plot is a sequence of scenes characterised by a precondition and a postcondition that reflect the state of the world before and after the development of each scene (Concepción, Gervás, and Méndez 2018b; 2018a).

The knowledge representation model provided by Afanasyev tries to cover all the aspects related to the structure and meaning of a story (Concepción, Gervás, and Méndez 2017). This representation has been designed as a hierarchical structure in which the root concept is the **story**. A story represents what both intuitively and narratologically can be considered a story, that is, a narration of events happening in a setting (Concepción, Gervás, and Méndez 2017). It is composed by the two classic narratological components: the plot and the space.

The plot of the story represents its essential structure, providing a sort of scaffolding for the actions and events that happen across the story. In the Afanasyev model, the plot is generated at the beginning by the Plot Generator microservice.

The INES instance for the **Plot Generator** is named **"Audrey"** —after Audrey Hepburn. It is a template-based plot generator which produces outlines from a subset of the cinematographic basic plots compiled by Balló (Balló and Pérez 2007). Audrey aims at building a story plot containing the main scenes that will be completed by the Episode Generator microservice. Its generation model is quite similar to systems like Gester (Pemberton 1989) and Teatrix (Machado, Paiva, and Brna 2001).

The original plot building procedure started by selecting one of the cinematographic templates, namely a conceptual structure with a sketch of the plot, and developed it later by instantiating the roles and the types of actions into real characters and actions. Audrey's REST interface supports the random selection of a template but also the selection of a specific template name. Once a basic template is selected, it instantiates its generic elements to develop a concrete plot. This step entails the use of knowledge about the context in which the story will be set. In this case, the context is inferred from the request parameters.

An example of one of these templates is *"The destructive outsider"*, summarized in table 1 (Concepción, Gervás, and Méndez 2018b).

Audrey queries the knowledge base that contains the main concepts presented in the plot for creating a consistent detail for every episode (Concepción, Gervás, and Méndez 2018b) according to certain setting. In the prior example, the plot mentions a "community", an "outsider", some "destructive actions" performed by the outsider, etc. All these concepts are included in the knowledge base and there are more specific roles and actions which refer to them. For example, the concept of "community" can be instantiated into a "town", a "family" or a "company". In each case, the "outsider" can be a "new sheriff", an "unknown relative" or a "new colleague". By the same token, the knowledge base contains the information required to determine the type of actions that the characters can perform.

Every episode or scene is expressed in terms of a set of attributes that essentially provide information about the story space —including both time and location, the characters that appear in the episode and the state of the world before and after the episode happens (Concepción, Gervás, and Méndez 2018a; 2018b). These last information is represented as the scene precondition and postcondition. They are sets of assertions expressing the restrictions to be considered during the development of the episode's detail. Table 2 shows a sample of all these attributes.

Probably, the most influential attributes in terms of consistency keeping across the scenes are the precondition and the postcondition. They are included as a part of the Scene - Frame - State structure. These attributes are a collection of assertions about the state of the story world before and after the scene occurs. They contain assertions related to any existant referenced in the scene —that is, characters, beings

Attribute	Value
Precondition	John is a friend of William
Postcondition	John is a friend of William John performs friendly actions to William
Characters	John, William
Time	Story relative time in which the episode happens
Location	Spatial reference in the story world

Table 2: The basic attributes of an episode

and objects.

Techniques for plots interweaving

This section reviews some feasible techniques for plot interweaving taken from the Literary and Narratological studies.

The *"Chinese Box"* technique (Menéndez 2013) consists in the inclusion of nested stories inside a larger one. Every nested story can be related to one or several characters of the main plot line, or even consist of a separate one with the purpose of explaining some happenings of the main story. This approach has been applied in great works of literature such as *The Arabian Nights* (Vernet 1990) and *Don Quixote* (Cervantes 2011).

The technique of the *"Communicating Vessels"* (Menéndez 2013) is based on constructing a story by alternating at least two differentiated parallel plot lines. An example of this technique can be seen in Madame Bovary (Flaubert 1857), which contains a chapter that alternates two apparently disconnected plot lines. The only commonality between them is the temporal context in which the actions and happenings they contain are occurring. The resulting effect is a contamination between the two plot lines that, taken in isolation, would produce a different understanding in the reader. In other words, when two alternating story passages are interwoven together, proximity and alternation generate mutual influence. This influence can be applied to the tone, the tension, or the atmosphere that every story transmits to the other.

However, this technique is not limited to these types of influence. A stronger influence is also possible at the plot level. That means that both lines can develop a closer connection and converge at a certain point in the story. This can be achieved by using shared characters in both plots, acting as a hook between them.

Regardless of how strongly connected the plot lines are, it is important that there is a balance between the two plots, in order to avoid that one predominate over the other.

Despite the prior explanation has focused on the application of the technique on only two plots, it is also applicable to more than two plot lines. An example of several overlapping plots in which the characters intersect is Pulp Fiction (Tarantino 1994).

Proposed solution

This section focuses on detailing the approach adopted for implementing the multiple plot interweaving techniques described above and the consequent evolution of the existing architecture of INES, putting special stress on the new central component of the solution: the Plot Weaver service.

Multiple-plot generation process

The structure of a story plot in INES is based on a sequence of scenes, each of which is defined by a set of preconditions, a set of postconditions, the time in which it happens and a spatial reference in the story space (Concepción, Gervás, and Méndez 2018a; 2018b). According to this model, the weaving of scenes from different plots should take into consideration the logical consistency when combining their respective pre and postconditions. Following this reasoning, there are two ways of weaving scenes from different plots: combination and juxtaposition. The first strategy combines two scenes from different plots into a new one. This operation is feasible if and only if both the preconditions and the postconditions from the two scenes are respectively consistent among themselves from a logical point of view. The juxtaposition approach creates a combined sequence of scenes by putting one after another. In this case, the postcondition of every scene must be consistent with the precondition of the scene that goes after it. The latter is the approach adopted for the design of the Plot Weaver. This microservice, that will be concisely described later, is responsible for implementing the interweaving of the plot lines.

So, in order to weave the plots, the preconditions and the postconditions of the involved episodes must be consistent. If not, the Plot Weaver skips one episode in the plot line and tries to match with the next one. There is always a chance that the plot lines are simply incompatible so the response in this case would be an error, and the Story Director would have to select a different pair of plots to merge.

The new generation process starts with the Story Director requesting the Plot Generator to create a first plot. The request includes as parameters a template, the characters' list and a theme. The *template* parameter is the logical name which references the plot template that must be instantiated to create the plot. This instantiation is strongly linked with the *theme* parameter, that represents the setting in which the story will take place. It is also a logical name referencing a particular context in the knowledge base. For example, when applying a plot template such as "The Destructive Outsider" summarised in Table 1, the "Community" can be a "Nineteenth Century Western North American Town" or a "Middle-Class Family", depending on whether the theme is "Far West" or "Family Drama".

The generated plot is the skeleton of a story draft that is persisted in the Draft Repository. This draft contains information about the setting —time and space in the story world–, a list of the characters mapped with the roles required by the template and the theme, and of course the plot line.

Following this first plot generation, the Story Director requests the Plot Generator to create a second plot. This time, the request includes as parameters a new characters' list containing a subset of the characters' list of the first plot, their roles, the setting —time and space–, and the theme of the existing draft. In addition, the Plot Generator will need to

locate a proper template which can fit the character roles required, according to the theme. In order to do do this, it must query the template repository to get all the available templates that consider the involved roles. If no match is found, then the characters' list will not contain any common character with the prior plot line. This case will produce a story with two parallel plot lines that take place in the same setting, but apparently unrelated.

The procedure described above shows how the Story Director keeps the global consistency between the two generated stories. After having generated the two plots, the weaving process can start. The Story Director requests the Plot Weaver to merge the plots of the two generated stories.

The weaving of the plots is performed according to a sequence of steps. Taking the plot of the first story as the master line, the Plot Weaver proceeds by finding a compatible scene in the second plot line and merging it with the first one to insert a new scene in the master plot line. This means that the weaving process always takes the sequence of episodes of this first plot and proceeds by looking for compatible episodes in the second plot. The ideal outcome of this procedure is the generation of an interwoven plot line consisting in an alternation of scenes from the first and the second plot. The limitation of this strategy is that non-merged scenes from the source plots could remain, which would be included at the end of the resulting plot.

Evolved INES architecture

Adding the objective of supporting multiple-plots generation to the current operational requirements, the original generation model is insufficient. This need adds a new functional driver to the existing architecture. The adaptation of INES for interweaving plot lines is based on the definition of a set of plot merging heuristics and the modification of the INES microservices ecosystem to include this stage in the generation process. Figure 2 depicts the current architecture after evolving INES to support multiple plot generation.

The original INES model considered only one plot per story. This entails that the Story Director only requests the Plot Generator for a plot line once for every story. In the evolved version, the Story Director requests twice the Plot Generator to get the plot lines to merge and must include new controls along its inner logic to prevent inconsistencies.

As described previously, the original REST interface of the Plot Generator supported requests with no parameters, so it instantiated a random template, and requests with the template to be applied to generate the plot. In this case, generating a consistent story requires that the different plots share a common story space. This entails not only location and time, but also the set of participating characters. So that, the REST interface of Audrey —the Plot Generator, has been modified to accept all these new parameters. In order to adapt the architecture for mixing two plot lines into a single story, it is also necessary to include a new component. However, the adaptation of this microservice has entailed more changes than merely adapting its interface. Audrey uses a inner template repository for managing the templates it instantiates during the plot generation. The original design of this component only allowed for querying templates by

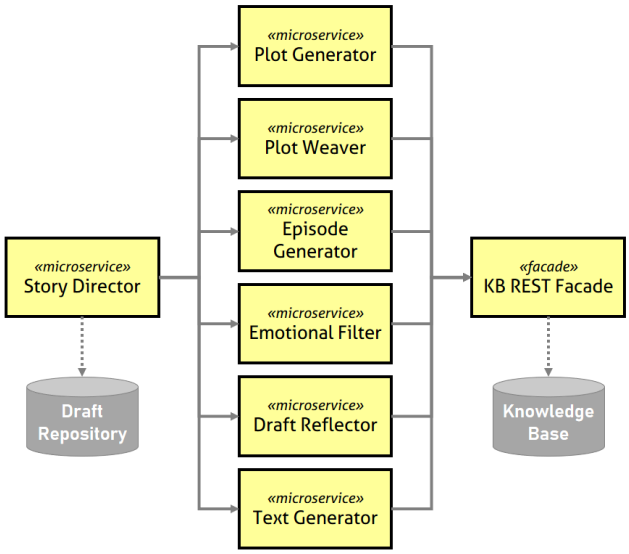

Figure 2: The evolved architecture of INES.

name. Due to the need for having a way of selecting templates by the roles they involve, the signature of the template repository has been adapted to provide queries based on roles. Thus, the updated Plot Generator can choose a template to apply according to certain restrictions. This functionality is essential to implement the multiple plot generation procedure.

The Plot Weaver is the microservice devoted to perform the plot weaving stage. It is implemented according to the Strategy pattern in order to select and apply the selected weaving heuristic. Initially, there have been considered two heuristics derived from the "Comunicating Vessels" technique. The simplest way of mixing the plots is by a mere alternation of episodes, a kind of "unrelated juxtaposition" of episodes from the two plot lines. In this basic case, the sets of characters of the two original plots can be disjointed. A more elaborated way is the linking of the episodes according to their compatibility in terms of state of the story world. In this case, there are common characters among the two stories.

The Plot Weaver provide as the default strategy the alternation of episodes from the two plots to combine –by juxtaposition. The resulting plot line will be a sequence of episodes picked from the two initial plots. Beside this, it also provides the option of merging the plots by sharing a subset of every plot's characters. This operation is the most complex and requires the Story Director to share certain parameters during the plot generation stage. So, it has to request the Plot Generator twice, for generating the two plots to combine, and provide shared information as parameters:

- Setting reference, as mentioned before, in order to instantiate the plot template according to a particular setting in the knowledge base, the Plot Generator requires a reference for this setting —e.g. "Far West", "Present day", "Epic Fantasy".

Episode	Description
Initial state	A frustrated character regrets his / her fruitless life
Temptation	The character is tempted by another character representing the evil forces
Pact with evil	The main character agree to serve the evil cause in exchange for a new satisfactory way of life
Evil actions	The main character performs evil actions induced by evil
Enlightenment	The main character becomes aware of being enslaved by evil
Redemption	The main character performs a saving action and dies. Evil is defeated

Table 3: The "Faust" story plot template

- Characters list, containing their name and their role in the plot template. The Story Director has to select which characters will be shared between the two plots in order to request the Plot Generator to include them.

In addition, the Story Director must take care of not requesting the Plot Generator using the same theme —e.g. requesting to generate two plots for interweaving based on "The destructive outsider", and choosing a compatible plot template for the second plot.

The interface of the Plot Weaver is also a stateless REST-based API, as the rest of the microservices of INES. In a first version, the Plot Weaver is only capable of combining two plot lines. An obvious precondition for these two plots to be merged is that they share a subset of the characters involved in their respective plot lines. For assuring this precondition, the Story Director must analyse the characters' roles of the first plot before requesting the Plot Generator for the second time. It needs to identify a set of matching roles between the first plot line and any of the available plot templates for the second plot line. For this reason, the REST interface of the Plot Generator includes a new operation for requesting information about the available plot templates and their metadata —such as characters' roles.

An example of interwoven plot story

The following lines introduce an example of story generation by plot interweaving according to the "Communicating Vessels" technique. It is structured around a first plot based on the "Destructive Outsider" template and a second one based on the "Faust" template. Tables 1 and 3 show the detail of both plot templates.

Table 4 shows a sample story which combines the two plots, generated by applying a juxtaposition approach with shared characters. The story combines two plots based on two different templates. The white rows contains the episodes from the first plot —based on "The destructive outsider" template, while the gray rows contains the episodes from the plot based on "Faust". Both plots share the same setting and

Episode	Actions
A peaceful community	Mary and John work together on their farm William helps John with the farm tasks Mary invites William to dinner Jeff visits John Jeff gives a present to John
Frustrated character regrets his life	Jeff feels miserable Jeff thinks that he is weak Jeff hates Carlson Jeff wants to arrest Carlson
The arrival of the outsider	Adam arrives at the city Adam buys a ranch Jeff welcomes Adam John welcomes Adam Mary invites Adam to dinner
Temptation	Adam offers help to Jeff Adam offers money to Jeff Adam tells Jeff to arrest all the gunmen
Outsider destructive actions	Adam wants John's farm Adam sneakily burns down John's barn
Pact with evil	Adam blames Carlson for burning down John's barn Jeff accepts Adams' money Jeff arrests Carlson
Conflict	Adam offers Mary to buy her farm Mary accepts Adam's offer John refuses to sell his farm John gets angry with Mary
Evil actions	Adam shoots John John is injured
The outsider revealed	Mary witnesses Adam shooting John Mary tells Jeff that Adam is a killer John tells Jeff that Adam is a killer Jeff gets angry with Adam
Enlightenment	Jeff realises that Carlson did not burn out John's barn Jeff releases Carlson Jeff says sorry to Carlson
The rise of the heroes	John faces Adam John demands Adam to leave Adam refuses to leave the town
Redemption	Jeff arrests Adam Adam shoots Jeff Carlson shoots Adam Adam dies Jeff dies
Conclusion	Mary says sorry to John Carlson is freed

Table 4: A sample story based on mixing two plots

a subset of the characters. The story takes place in the Far West and the characters of the whole story are the following:

- John, a farm owner married with Mary
- Mary, John's wife
- William, a farmer friend of Mary and John
- Jeff, the Sheriff
- Carlson, a gunman
- Adam, a cattle baron, an outsider

William is a character that only appears in the first plot, while Carlson only appears in the second one. Despite this, the combined plot line remains consistent.

This example shows the most complete form of combination that the current design can support. The two plot lines involved in the generation of the story contained a good number of shared compatible roles, so the outcome looks quite united. In addition, the scenes from the first plot perfectly alternate with the scenes from the second one. In this case, the postcondition of every scene from the first plot has been compatible with the precondition of the equivalent from the second plot, but this is not necessarily the norm. In many cases, the Plot Weaver will need to pass to the next scene in the first plot until it can insert the scene in the second. Moreover, the number of scenes in the plots to interweave can perfectly be different, and this will entail that the mixed plot line will contain several consecutive scenes from the same plot line.

It is also worth to mention that, in many cases, the preconditions and postconditions basically refer to facts that rarely affect to scenes from different plot lines. This means that there could be many combinations in terms of scene ordering that also would keep the global consistency of the story. The resulting ordering is mainly due to the previously described interweaving procedure, which always takes first a scene from the first plot and tries to find the next compatible scene in the second plot. Table 4 shows several examples of this. For example, the scenes "A peaceful community" and "Frustrated character" are interchangeable without affecting the consistency of the story, as well as the "Outsider destructive actions" and "Pact with evil". On the other side, the "Temptation" scene will be pointless if it happened before the "Arrival of the outsider". In this case, the Plot Weaver would have applied the procedure to establish a consistent ordering of the scenes.

Conclusions and future work

The presented adaptation entails a good number of validations in terms of knowledge representation and consistency. The assurance of a consistent merging of two different plot lines, putting together episode by episode from the two plots, is not an easy task. Despite the Plot Weaver checks the proper fitting of the respective precondition and postcondition of the episodes, there can be inconsistencies at a global level.

A significant case that can easily occur in the current model is the reappearance of a character killed in an episode of one plot in later episodes of the other plot, creating a kind of blocking inconsistency —the affected plot could not continue in a consistent way with the merged plot. This is caused by the fact that, despite the match of the corresponding preconditions and postconditions, the current version of the Plot Weaver does not consider the whole plot line, so there can emerge inconsistencies from a global point of view. For example, in one of the two plot lines, a common character can die. It is perfectly possible that this character does not appear in the episode that follows the one in which he / she dies. This circumstance makes the postcondition of the first episode to be consistent with the precondition of the following one. But, the character suddenly appears later, in an episode from the plot line which did not include the death of this character. This situation can be amended by including long-term conditions, that are propagated across the whole plot lines. In further iterations, these checks will be faced to guarantee a fully consistent story. On the other hand, from a positive point of view, this kind of situations could be interesting for developing stories according to an unnatural narrative plan, what could be specifically explored in a future line of research. The next natural step in this adaptation process will be the development of the mechanism that holds this need. The evolution of this model will provide generated lessons that will be helpful for making better decisions, and in marking paths for future investigation. One of this paths can be deepening in the development of a more pervasive plot interweaving, in which a Plot Weaver works with incomplete drafts. In this scenario, the plot generation and the episode generation stages in the different generation stages would directly interchange events between them, during their own activity. This approach, more decentralized in the sense of the events will not be mediated by the Story Director, would bring more creative wealth, but also more complexity to the process.

The Plot Weaver supports the application of different weaving strategies. One of the promising candidates to be considered in future versions is the "Chinese Box" technique (Menéndez 2013), which considers the development of nested plot lines inside a larger one. This adaptation would entail not only a modification of the Plot Weaver strategy, but also the model of generation applied by the Plot Generator.

Another aspect that can be analyzed in the future is the ability of merging more than two plot lines in a single story. Many of the already designed heuristics will still be useful, but probably we would need to design new ones to address the complexities associated to this new requirement and introduce more thorough draft evaluation mechanisms (Gervás and León 2016; Tapscott et al. 2016).

Acknowledgments

This paper has been partially funded by the projects IDiLyCo: Digital Inclusion, Language and Communication, Grant. No. TIN2015-66655-R (MINECO/FEDER) and InVITAR-IA: Infraestructuras para la Visibilización, Integración y Transferencia de Aplicaciones y Resultados de Inteligencia Artificial, UCM Grant. No. FEI-EU-17-23.

References

Alber, J., and Heinze, R. 2011. *Unnatural narratives-unnatural narratology*, volume 9. Walter de Gruyter.

Balló, J., and Pérez, X. 2007. *La semilla inmortal: los argumentos universales en el cine*. Ed. Anagrama.

Bellifemine, F.; Poggi, A.; and Rimassa, G. 1999. Jade–a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, 33. London.

Bringsjord, S., and Ferrucci, D. 1999. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press.

Cervantes, M. 2011. *Don Quixote*. Penguin Random House.

Cohen, P. R., and Feigenbaum, E. A. 2014. *The handbook of artificial intelligence*, volume 3. Butterworth-Heinemann.

Concepción, E.; Gervás, P.; and Méndez, G. 2017. A common model for representing stories in automatic storytelling. In *6th International Workshop on Computational Creativity, Concept Invention, and General Intelligence. C3GI 2017*.

Concepción, E.; Gervás, P.; and Méndez, G. 2018a. Afanasyev: A collaborative architectural model for automatic story generation. In *5th AISB Symposium on Computational Creativity. AISB 2018*.

Concepción, E.; Gervás, P.; and Méndez, G. 2018b. Ines: A reconstruction of the charade storytelling system using the afanasyev framework. In *Ninth International Conference on Computational Creativity, ICCC 2018*.

Dehn, N. 1981. Story generation after tale-spin. In *IJCAI*, volume 81, 16–18.

Fay, M. P. 2014. *Driving story generation with learnable character models*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Flaubert, G. 1857. *Madame Bovary*. Michel Lévy.

Gervás, P., and León, C. 2014. The need for multi-aspectual representation of narratives in modelling their creative process. In *5th Workshop on Computational Models of Narrative*, OASIcs-OpenAccess Series in Informatics.

Gervás, P., and León, C. 2016. Integrating purpose and revision into a computational model of literary generation. In *Creativity and Universality in Language*. Springer. 105–121.

Gervás, P. 2012. Story generator algorithms. In *The Living Handbook of Narratology*. Hamburg University Press.

Gervás, P. 2018. Storifying observed events: Could i dress this up as a story? In *5th AISB Symposium on Computational Creativity*. University of Liverpool, UK: AISB.

Glassner, A. 2009. *Interactive storytelling: Techniques for 21st century fiction*. AK Peters/CRC Press.

Lebowitz, M. 1984. Creating characters in a story-telling universe. *Poetics* 13(3):171–194.

Li, B., and Riedl, M. O. 2010. An offline planning approach to game plotline adaptation. In *AIIDE*.

Liu, H., and Singh, P. 2002. Makebelieve: Using commonsense knowledge to generate stories. In Dechter, R., and Sutton, R. S., eds., *AAAI/IAAI*, 957–958. AAAI Press / The MIT Press.

Machado, I.; Paiva, A.; and Brna, P. 2001. Real characters in virtual stories. In *International Conference on Virtual Storytelling*, 127–134. Springer.

Martínez, A. 2011. *Escribir teatro*. Barcelona, Spain: Alba Editorial.

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.

Méndez, G.; Gervás, P.; and León, C. 2014. A model of character affinity for agent-based story generation. In *9th International Conference on Knowledge, Information and Creativity Support Systems, Limassol, Cyprus*, volume 11, 2014.

Méndez, G.; Gervás, P.; and León, C. 2016. On the use of character affinities for story plot generation. In *Knowledge, Information and Creativity Support Systems*. Springer. 211–225.

Menéndez, R. 2013. *Cinco golpes de genio*. Barcelona, Spain: Alba Editorial.

Mott, B.; Lee, S.; and Lester, J. 2006. Probabilistic goal recognition in interactive narrative environments. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, 187.

Pemberton, L. 1989. A modular approach to story generation. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, 217–224. Association for Computational Linguistics.

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

Porteous, J.; Charles, F.; and Cavazza, M. 2016. Plan-based narrative generation with coordinated subplots. In *European Conference on Artificial Intelligence (ECAI 2016)*, volume 285, 846–854. IOS Press.

Prada, R.; Machado, I.; and Paiva, A. 2000. Teatrix: Virtual environment for story creation. In *International Conference on Intelligent Tutoring Systems*, 464–473. Springer.

Propp, V. 1968. Morphology of the folk tale. 1928.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Tapscott, A.; Gomez, J.; León, C.; Smailovic, J.; Znidarsic, M.; and Gervás, P. 2016. Empirical evidence of the limits of automatic assessment of fictional ideation. In *C3GI@ ESSLLI*.

Tarantino, Q. 1994. Pulp fiction.

Todorov, T. 1975. *The fantastic: A structural approach to a literary genre*. Cornell University Press.

Vernet, J., ed. 1990. *The Arabian Nights*. Editorial Planeta.

Winston, P. H. 2016. The genesis story understanding and story telling system a 21st century step toward artificial intelligence. Technical report, Center for Brains, Minds and Machines (CBMM).