Case-Based Story Generation through Story Amalgamation

Santiago Ontañón[1], Jichen Zhu[1] and Enric Plaza[3]

[1] Drexel University
Philadelphia, PA, USA 19104
santi@cs.drexel.edu
jichen.zhu@drexel.edu
[2] IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
enric@iiia.csic.es

Abstract. Computational narrative explores techniques through which computers can analyze, understand and, most importantly, generate stories. This paper explores a CBR approach to story generation based on the idea of *story amalgamation*: given a target partial description to the story we want the system to generate, the system will retrieve a set of full stories (represented as cases), and then reuse them by merging them in a way that the result satisfies the target partial description. This story merging is performed via a formal operation we call an *amalgam*. We report on a preliminary study showing the potential of the approach.

1 Introduction

Computational narrative explores the age-old creative form of storytelling by algorithmically analyzing, understanding, and most importantly, generating stories. The various AI techniques developed in story generation can be extended to other forms of interactive entertainment and electronic literature, including computer games and interactive narrative. Many different approaches have been studied for the problem of automatic story generation, such as planning or case-based reasoning, each of which providing different aesthetic affordances in the range of stories that can be generated [21].

This paper explores a CBR approach to story generation based on the idea of *story amalgamation*, i.e. generating new stories by selectively merging parts of previous, existing, stories. Specifically, we explore the idea that if stories are represented as *terms* in a generalization space, then story amalgamation can be carried out by performing formal operations over terms such as *amalgams* [10]. An amalgam is a formal operation between terms in a generalization space that generates a new term that combines as much information as possible from the initial terms. Formally, an amalgam is related to the idea of partial unification, and also to the cognitive process called *conceptual blending* [6].

In this paper we present a CBR system that can generate stories using amalgams in the following way: each case in the CBR system represents a different story; the input to the system is a target partial description of the story we want to generate; given an input target description, the system retrieves a set of similar cases, and then reuses them by amalgamating them in a way that the amalgam satisfies the target partial description.

The remainder of this paper is organized as follows. Sections 2 and 3 provide background on story generation and the amalgam operations respectively. Then Section 4 presents our technical approach. Section 5 shows our preliminary results. The paper closes with related work, conclusions and potential future work.

2 Story Generation

Automatic story generation is an interdisciplinary topic focusing on devising models for algorithmically structuring and producing narrative content and/or discourse. Narrative can be divided into two main parts [3]: *story* and *discourse*, which basically correspond to the story content, and to the way the story is presented respectively. Most story generation systems focus strictly in generating a story, although some are capable of generating a discourse as well [20]. In general, story generation systems can be classified into three main categories [1]: character-centric, author-centric and story-centric:

- Character-centric systems like Tale-spin [9] and the Virtual Storyteller [17] generate stories by simulating characters in a world.
- Author-centric systems, such as the MEXICA system [13], model the author's throught process during the process of story-writing.
- Story-centric systems, such as the Fabulist [15], generate stories by modeling the structural properties of the stories themselves.

The system reported in this paper can be classified as a story-centric system.

Different techniques have been studied in story generation, the most common of which is automated planning. Salient examples of planning-based story generation systems include Tale-spin [9], Universe [8] and Fabulist [15]. However, other techniques, such as CBR and computational analogy have also shown applicability to the problem of story generation. Examples are systems like Minstrel [18], ProtoPropp [7], Riu [12], or the story-translator [16]. In this paper we will explore case-based techniques, and in particular in a technique called *amalgamation*, which can generate solutions to new problems by amalgamating information coming from one or more cases [10].

Story generation is a very challenging task from many points of view. Generating stories means generating coherent plots, believable characters that have common sense, and natural language. Moreover, those stories have to be aesthetically pleasing and creative. In fact, some approaches to story generation, such as Minstrel [18] aim at being general models of computational creativity. Finally, one of the most important open problems in story generation is evaluation [19]. Which are the criteria under which we can compare story generation systems, or even how to evaluate the output of these systems are still open problems.

3 Amalgams

In this paper we will make the assumption that cases are terms in some *generalization space*. We define a generalization space as a partially ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$, where $\mathcal{L}$ is a language, and $\sqsubseteq$ is a subsumption between the terms of the language $\mathcal{L}$. We say that a term ψ_1 subsumes another term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when ψ_1 is more general (or equal) than ψ_2[3]. Additionally, we assume that $\mathcal{L}$ contains the infimum element $\perp$ (or "any"), and the supremum element $\top$ (or "none") with respect to the subsumption order.

Next, for any two terms ψ_1 and ψ_2 we can define their *unification*, ($\psi_1 \sqcup \psi_2$), which is the *most general specialization* of two given terms, and their *anti-unification*, defined as the *least general generalization* of two terms, representing the most specific term that subsumes both. Intuitively, a unifier (if it exists) is a term that has all the information in both the original terms, and an anti-unifier is a term that contains all the information that is shared by the two original terms. Depending on $\mathcal{L}$, anti-unifier and unifier might be unique or not.

The notion of *amalgam* can be conceived of as a generalization of the notion of unification over terms. The unification of two terms (or descriptions) ψ_a and ψ_b is a new term $\phi = \psi_a \sqcup \psi_b$, called unifier. All that is true for ψ_a or ψ_b is also true for ϕ.; e.g. if ψ_a describes "a red vehicle" and ψ_b describes "a German minivan" then their unification yields the description "a red German minivan." Two terms are not unifiable when they possess contradictory information; for instance "a red French vehicle" is not unifiable with "a red German minivan". The strict definition of unification means that any two descriptions with only one item with contradictory information cannot be unified.

An *amalgam* of two terms (or descriptions) is a new term that contains *parts from these two terms*. For instance, an amalgam of "a red French vehicle" and "a German minivan" is "a red German minivan"; clearly there are always multiple possibilities for amalgams, since "a red French minivan" is another example of amalgam. The notion of amalgam, as a form of partial unification, was formally defined in [10], where its relationship with the notion of merging operators [4], is also discussed.

Definition 1. (Amalgam) *The set of* amalgams *of two terms ψ_a and ψ_b is the set of terms such that:*

$$\psi_a \curlyvee \psi_b = \{\phi \in \mathcal{L}^+ | \exists \alpha_a, \alpha_b \in \mathcal{L} : \alpha_a \sqsubseteq \psi_a \ \wedge \ \alpha_b \sqsubseteq \psi_b \ \wedge \ \phi = \alpha_a \sqcup \alpha_b\}$$

where $\mathcal{L}^+ = \mathcal{L} - \{\top\}$

Thus, an amalgam of two terms ψ_a and ψ_b is a term that has been formed by unifying two generalizations α_a and α_b such that $\alpha_a \sqsubseteq \psi_a$ and $\alpha_b \sqsubseteq \psi_b$ —i.e. an amalgam is a term resulting from combining some of the information in ψ_a with some of the information from ψ_b. Formally, $\psi_a \curlyvee \psi_b$ denotes the set of all

[3] In machine learning terms, $A \sqsubseteq B$ means that A is more general than B, while in description logics it has the opposite meaning, since it is seen as "set inclusion" of their interpretations.

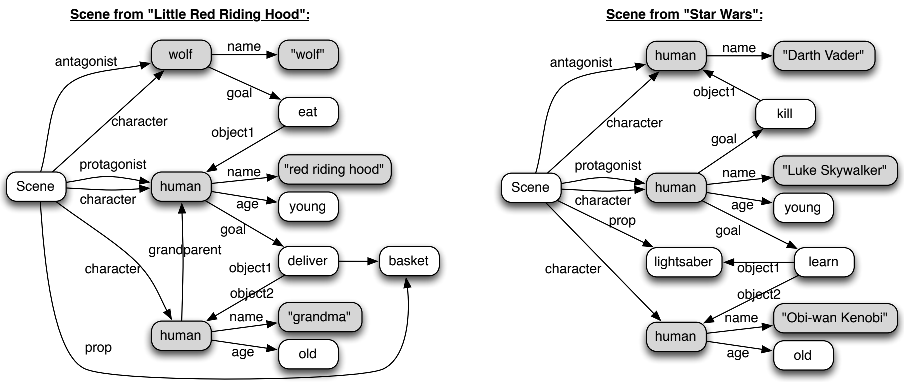

Fig. 1. Two of the scenes used in our experiments.

possible amalgams; however, whenever it does not lead to confusion, we will use $\psi_a \curlyvee \psi_b$ to denote one specific amalgam of ψ_a and ψ_b.

The terms α_a and α_b are called the *transfers* of an amalgam $\psi_a \curlyvee \psi_b$. α_a represents all the information from ψ_a which is *transferred* to the amalgam, and α_b is all the information from ψ_b which is transferred into the amalgam.

Intuitively, an amalgam is *complete* when all which can be transferred from both terms into the amalgam has been transferred, i.e. if we wanted to transfer more information, α_a and α_b would not have a unifier.

For the purposes of story generation, we introduce the notion of asymmetric amalgam, where one term is fixed while only the other term is generalized in order to compute an amalgam.

Definition 2. (Asymmetric Amalgam) *The* asymmetric amalgams $\psi_s \overrightarrow{\curlyvee} \psi_t$ *of two terms* ψ_s *(*source*) and* ψ_t *(*target*) is the set of terms such that:*

$$\psi_s \overrightarrow{\curlyvee} \psi_t = \{\phi \in \mathcal{L}^+ | \exists \alpha_s \in \mathcal{L} : \alpha_s \sqsubseteq \psi_s \quad \wedge \quad \phi = \alpha_s \sqcup \psi_t\}$$

In an asymmetric amalgam, the target term is transferred completely into the amalgam, while the source term is generalized. The result is a form of partial unification that conserves all the information in ψ_t while relaxing ψ_s by generalization and then unifying one of those more general terms with ψ_t itself. Finally, an asymmetric amalgam is *maximal* when all knowledge in ψ_s that is consistent with ψ_t is transferred to the solution ψ'_t —i.e. $\psi'_t \in \psi_s \overrightarrow{\curlyvee} \psi_t$ is maximal iff $\nexists \psi''_t \in \psi_s \overrightarrow{\curlyvee} \psi_t$ such that $\psi'_t \sqsubset \psi''_t$.

4 Story Generation through Story Amalgamation

In this paper we want to explore the idea of generating stories by amalgamating previously existing stories that correspond to cases in a CBR system. For that purpose, we have designed a system that solves the following task:

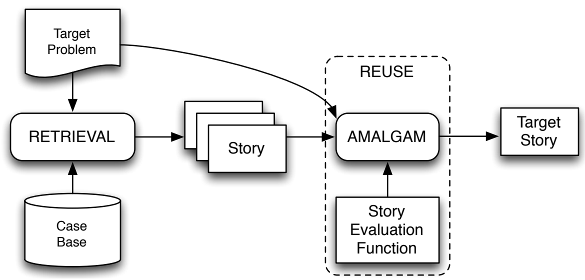

Fig. 2. Overview of the story generation approach studied in this paper.

Given A case base $\Delta = \{\psi_1, \dots \psi_m\}$, where each case ψ_i is a story, and a partially specified target story ψ_t

Generate A story ψ'_t such that $\psi_t \sqsubseteq \psi'_t$ (i.e. the generated story satisfies the partial specification provided as input) by amalgamating a collection of cases from the case base.

From the previous description it can be seen that we are proposoing to represent stories as terms in a generalization space. Specifically, we have used the *feature term* formalism [2, 14] to represent stories. Figure 1 shows two sample stories used in our experimentation. The first one represents the opening scene of the famous "Little Red Riding Hood" tale, with three characters: red riding hood, her grandmother and the wolf. As can be seen, the story specifies that red riding hood wants to deliver a basked of food to her grandmother, and the wolf wants to eat red riding hood. For this experimentation, we have avoided representing the notion of time in our stories, and thus each case in the case base represents just a "scene". Full stories could be represented by a collection of scenes composed using time relations. However, for our purposes, scenes like the ones shown in Figure 1 are complex enough to test the potential of our approach. As can be seen, for each scene, we represent the set of characters and props, their relations, and the goals of each character in the scene, some scenes also contain actions.

The target given to the system is a partially specified story, which means that it is a term (similar to those in Figure 1), but where only some parts of a scene are represented. For example, we could just specify that we want to have 2 characters, or that we want a story with a "character wanting to kill another character". Basically, the target specifies the constraints that the generated story has to satisfy.

Figure 2 shows an overview of our proposal. When the user provides a new target ψ_t (a partially specified story), the system retrieves a set of k cases from the case-base by using a similarity measure. In our experiments, we used the S_λ similarity metric defined in [11] (which basically measures the amount of information shared between two terms), to find the cases that are most similar to the provided problem. Let us call R to the set of retrieved cases.

Then, the system generates a new story by combining information from all the retrieved cases in R, using the amalgam operation, to generate a story that satisfies the target. This process is performed in two steps:

1. Retrieved cases amalgamation: in a first step, all the retrieved cases are amalgamated in order to obtain a combination of the parts from the retrieved cases that are consistent (and could later be used to generate a new story). Specifically, given the set of retrieved cases $R = \{\psi_1, ..., \psi_k\}$, this process constructs an amalgam ψ_R in the following way:

$$\psi_R = \psi_1 \curlyvee \psi_2 \curlyvee ... \curlyvee \psi_k$$

   where, $\psi_1 \curlyvee \psi_2 \curlyvee \psi_3 = (\psi_1 \curlyvee \psi_2) \curlyvee \psi_3$, i.e. to perform the amalgam between a set of n terms, we amalgamate the first two, the result is amalgamated with the third, and so on. Notice that ψ_R is not unique, since there are many different possible amalgams that can result from amalgamating a set of given terms. As detailed below, we use an evaluation function that gives a score to each one of them, and select the amalgam that maximizes such evaluation function.
2. Then, given ψ_R, the final story is obtained by performing the asymmetric amalgam of ψ_R with the target ψ_t, obtaining $\psi'_t = \psi_R \overrightarrow{\curlyvee} \psi_t$. As before, ψ'_t might not be unique, and the final story is selected by evaluating all the possible amalgams with an evaluation function and selecting the one that maximizes this function. ψ'_t represents an amalgam that is ensured to satisfy the partial description ψ_t, and contains as much information as possible from ψ_R.

As described above, the amalgam operation between two terms $\psi_1 \curlyvee \psi_2$ doesn't define a single term, but a space of possible amalgams. Each of them obtained by combining different sets of information from the two input terms. That means that the process defined above just defines the space of possible stories that can be generated. In order to determine which of all the possible amalgams is the one the system will produce, we need to introduce some additional criteria that determines which amalgams are better than others. In [10] we introduced the notion of *preservation degree*, which measures how much of the information of the input terms is present in the amalgam. Using the idea of the preservation degree, the system can be programmed to find amalgams that maximize the preservation degree, and thus that generates stories that contain as much information as possible form the retrieved cases. However, we can define other criteria for amalgam selection, which would bias the system towards generating certain types of amalgams that correspond to the stories we are interested in.

Specifically, we have experimented with the following criteria (ll of them are defined to provide a numerical score to the resulting amalgam; the system was programmed to output the amalgam that maximizes such score):

– Preservation Degree: as defined in [10], just tries to maximize the information transferred from the input terms into the amalgam. It assigns larger scores to amalgams that transfer more information from the input terms.

– Compactness: when amalgamating stories using the preservation degree, the amalgam operation tends to add all the characters that exist in the input stories to the amalgam. However, this does not always result in interesting stories. This measure assigns a score to each story computed as the preservation degree minus the number of variables in the resulting story (each of the nodes in the graphs shown in Figure 1 corresponds to a variable in the term representing the story).

In order to generate amalgams that maximize the previous criteria, we have used a straightforward greedy search method over the amalgam space. This method doesn't ensure obtaining the amalgam that maximizes the criteria, but is computationally efficient and provides good results (as shown in the next section). The next section shows some example stories generated by our system using the different evaluation functions.

Other evaluation functions that could be explored. For example: *novelty* (as studied in [5]), which would favor stories that differ from those already in the case base. Another interesting measure could be measure the coherency of the character relations in the story, if they are not related in a complex but coherent way to other characters, the story is likely to be uninteresting. Exploring further evaluation functions is part of our future work.

5 Illustration

In this section we show example stories generated by our system using different evaluation functions. To generate the stories described in this section, we used the following partially specified story as the input problem to our system: $\psi_t =$ "the story must have at least three characters, one named King Arthur, the other named Merlin, and the other is a dragon; King Arthur is the protagonist and the dragon is the antagonist; the story also must involve a sword called Excalibur".

Let us start with a simple example, where we set $k = 1$ (i.e. the system retrieves only 1 case). In this scenario, the case being retrieved is the one corresponding to the "star wars" story in Figure 1, which has similarity 0.46. Figure 3 shows the two stories that were generated by using the preservation degree and the compactness evaluation functions. As can be seen in the figure, the story generated using the preservation degree evaluation function is very complex and merely puts together all the characters, props and relations from the retrieved story and from the target without any interesting results. Using the compactness evaluation function however, results in a more interesting story, where the system has amalgamated the two villains of the target and of the retrieved case (the dragon from the query is now called "Darth Vader" and happens to be the father of the main character "King Arthur"). The main character, wants to learn how to use the sword "excalibur" from "Merlin", and plans to kill the dragon with it. This shows that, by determining an adequate evaluation function can have a huge impact in the resulting story, since the space of possible amalgams is quite large and varied. Specifically, the space of amalgams that was explored by the

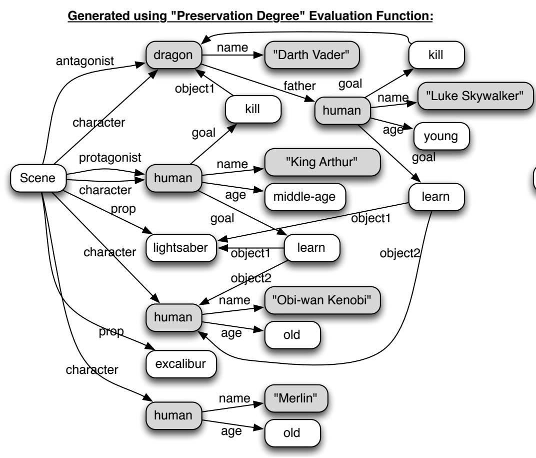

Fig. 3. Two stories generated by retrieving only one case.

greedy search algorithm using the preservation degree evaluation function contained 1897 amalgams, and was explored in 4.89 seconds; using the compactness evaluation function only 109 amalgams were explored, in 0.7 seconds.

We ran an experiment where we set $k = 2$ (the system retrieved 2 cases). In this scenario, the two cases that were retrieved are the ones shown in Figure 1. The space of amalgams being explored using preservation degree for first amalgamating the two cases to find ψ_R contained 1835 amalgams, and the space of amalgams explored when amalgamating ψ_R with the target contained 6681 amalgams and was explored in 30.02 seconds. When using the compactness evaluation function the spaces of amalgams contained 508 and 571 amalgams respectively, and were explored in 5.76 seconds. We don't show the resulting stories due to lack of space, but the story resulting with the compactness evaluation function contained three characters: King Arthur, Merlin and a Dragon called "wolf". Merlin is the grandfather of King Arthur, and King Arthur wants to deliver a basket of food to him. King Arthur also wants to learn how to use Excalibur from Merlin, so he can defeat "wolf" the dragon (who wants to eat King Arthur). As can be seen, this story combines parts from both "Red Riding Hood" and "Star Wars", but using the characters specified in the target story.

In summary, we have seen that by defining a small collection of base stories, a very large number of new stories can be generated by amalgamating them in different ways, and this can be exploited for story generation purposes with interesting results.

6 Related Work

Story generation using CBR approaches has been explored in the past. However, not through amalgams or merging operators, which is the main contribution of this paper. One of the early systems to the CBR for story generation was MINSTREL [18]. MINSTREL is a generic model that generates stories by executing TRAMS (Transform Recall Adapt Methods), which are generic operators that encode different problem solving procedures. MINSTREL was designed to be a model of human creativity, and as such TRAMS explore ways in which problems can be solved in creative ways. Other CBR approaches to story generation have focused on other problems, such as incorporating semantic knowledge into the retrieval and adaptation process [7], or on generating stories that are different from those in the case base (in order to show originality) [5].

Similar to CBR, some systems, such as SAM [12] use computational analogy to generate stories. SAM takes as input a partially specified story and a predefined complete story, and completes the partial story by analogy with the complete one. Notice that this is similar to the way the system presented in this paper works when only one case is retrieved. The main different with SAM is that SAM uses structure mapping theory in order to find the best analogy from the source to the target story, while in our work we use an evaluation function that can capture different aspects of the amalgam in order to decide which is the best amalgam to select.

7 Conclusions and Future Work

In this paper we have presented a preliminary study concerning the possibilities of using story amalgamation for story generation. We have presented a CBR system that can generate stories by retrieving and amalgamating stories and showed examples of its execution. One of the most interesting properties of the amalgam operation is that the amalgam of two input stories does not define a single story but a space of possible amalgams of the two stories. Therefore, it is possible to provide an evaluation function to the system that biases the story generation procedure towards specific kind of stories, and allows the inclusion of additional domain knowledge.

As part of our future work, we would like to explore the scalability of the approach to larger stories, and study further evaluation functions that capture the vast amount of existing narratology knowledge. Additionally, we would like to perform formal comparisons of the story generation capabilities of the proposed system with other CBR story generation systems.

Acknowledgements. This research was partially supported by project Next-CBR (TIN2009-13692-C03-01).

References

[1] Bailey, P.: Searching for storiness: Story-generation from a reader's perspective. In: Proceedings of the AAAI Fall Symposium on Narrative Intelligence (1999)
[2] Carpenter, B.: Typed feature structures: an extension of first-order terms. In: Saraswat, V., Ueda, K. (eds.) Proceedings of the International Symposium on Logic Programming. pp. 187–201. San Diego (1991)

[3] Chatman, S.: Story and Discourse: Narrative Structure in Fiction and Film. Cornell University Press (June 1978)
[4] Cojan, J., Lieber, J.: Belief merging-based case combination. In: Case-Based Reasoning Research and Development (ICCBR'09). Lecture Notes in Artificial Intelligence, vol. 5650, pp. 105–119 (2009)
[5] Díaz-Agudo, B., Plaza, E., Recio-García, J.A., Arcos, J.L.: Noticeably new: Case reuse in originality-driven tasks. In: ECCBR 2008. pp. 165–179 (2008)
[6] Fauconnier, G.: Conceptual blending and analogy. In: Gentner, D., Holyoak, K.J., Kokino, B.N. (eds.) The Analogical Mind: Perspectives from Cognitive Science. MIT Press, Cambridge, MA (2001)
[7] Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on cbr. Journal of Knowledge-Based Systems 18(4-5), 235–242 (2005)
[8] Lebowitz, M.: Creating characters in a story-telling universe. Poetics 13, 171–194 (1984)
[9] Meehan, J.: The Metanovel: Writing Stories by Computer. Ph.d., Yale University (1976)
[10] Ontañón, S., Plaza, E.: Amalgams: A formal approach for combining multiple case solutions. In: Case-Based Reasoning. Research and Development, 18th International Conference on Case-Based Reasoning, ICCBR 2010. pp. 257–271 (2010)
[11] Ontañón, S., Plaza, E.: Similarity measuress over refinement graphs. Machine Learning Journal 87(1), 57–92 (2012)
[12] Ontañón, S., Zhu, J.: The sam algorithm for analogy-based story generation. In: AIIDE 2011 (2011)
[13] Pérez y Pérez, R., Sharples, M.: Mexica: A computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence 13(2), 119–139 (2001)
[14] Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases. In: Veloso, M., Aamodt, A. (eds.) Case-Based Reasoning, ICCBR-95, pp. 265–276. No. 1010 in Lecture Notes in Artificial Intelligence, Springer-Verlag (1995)
[15] Riedl, M.: Narrative Generation: Balancing Plot and Character. Ph.D. thesis, North Carolina State University (2004)
[16] Riedl, M., León, C.: Generating story analogues. In: AIIDE 2009. The AAAI Press (2009)
[17] Theune, M., Faas, E., Nijholt, A., Heylen, D.: The virtual storyteller: Story creation by. In: Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference. pp. 204–215 (2003)
[18] Turner, S.R.: Minstrel: a computer model of creativity and storytelling. Ph.D. thesis, University of California at Los Angeles, Los Angeles, CA, USA (1993)
[19] Zhu, J.: Towards a new evaluation approach in computational narrative systems. In: Proceedings of the Third International Conference on Computational Creativity (ICCC). pp. 150 – 154. Dublin, Ireland (2012)
[20] Zhu, J., Harrell, D.F.: Daydreaming with intention: Scalable blending-based imagining and agency in generative interactive narrative. In: AAAI 2008 Spring Symposium on Creative Intelligent Systems. pp. 156–162. AAAI Press, Stanford, CA (2008)
[21] Zhu, J., Ontañón, S.: Towards analogy-based story generation. In: First International Conference on Computational Creativity (ICCC-X). Lisbon, Portugal (2010)