On the Use of Character Affinities for Story Plot Generation

Gonzalo Méndez, Pablo Gervás, and Carlos León

Facultad de Informática, Universidad Complutense de Madrid, Madrid, Spain,
{gmendez,pgervas,cleon}@ucm.es,
WWW home page: http://nil.fdi.ucm.es

Abstract. One of the aspects that is used to keep the reader's interest in a story is the network of relationships among the characters that take part in that story. We can model the relationship between two characters using their mutual affinities, which allow us to define which interactions are possible between two characters. In this paper we present a model to represent characters' affinities and we describe how we have implemented this model using a multi-agent system that is used to generate stories. We also present the result of one experiment to measure the evolution of the affinities between two characters throughout a story.

Keywords: computational creativity, narrative, story generation, multi-agent simulation, character affinities

1 Introduction

In the book *The Thirty-Six Dramatic Situations* [17] Polti explores the assertion made by Gozzi (author of *Turandot*) saying that there can only be thirty-six tragic situations. Polti analyses what these thirty-six situations are, their variations, and what characters are involved. At the end of the book, he begins his conclusions by saying that, to obtain the nuances of the situations, the first thing he did was to *"enumerate the ties of friendship or kinship between the characters"*. A century before that, Goethe had already proposed his theory (maybe just metaphorical) of *elective affinities* [5] to depict human relations, specially marriages, and he showed how affinities between characters can be represented by a topological chart.

Even in modern TV shows which expand for several seasons, one of the aspects that create more engagement with spectators are the relationships that exist between characters and the way in which they evolve from the beginning of the first season to end of the last one.

Through all of theses examples, we can see that the affinity between characters is an important factor to take into account when generating stories, and one that can help us to maintain the necessary narrative tension to keep the reader interested in the story.

In the following sections, we present a model of character affinities and the way in which we have implemented it using a multi-agent system that is used to generate stories based on the relationships between characters.

2 Related Work

The first story telling system for which there is a record is the Novel Writer system developed by Sheldon Klein [8]. Novel Writer created murder stories within the context of a weekend party. It relied on a micro-simulation model where the behaviour of individual characters and events were governed by probabilistic rules that progressively changed the state of the simulated world (represented as a semantic network). The flow of the narrative arises from reports on the changing state of the world model. A description of the world in which the story was to take place was provided as input. The particular murderer and victim depended on the character traits specified as input (with an additional random ingredient). The motives arise as a function of the events during the course of the story. The set of rules is highly constraining, and allows for the construction of only one very specific type of story.

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

TALESPIN [14], a system which told stories about the lives of simple woodland creatures, was based on planning: to create a story, a character is given a goal, and then the plan is developed to solve the goal. TALESPIN introduces character goals as triggers for action. Actions are no longer set off directly by satisfaction of their conditions, an initial goal is set, which is decomposed into subgoals and events. The systems allows the possibility of having more than one problem-solving character in the story (and it introduced separate goal lists for each of them). The validity of a story is established in terms of: existence of a problem, degree of difficulty in solving the problem, and nature or level of problem solved.

Lebowitz's UNIVERSE [10] modelled the generation of scripts for a succession of TV soap opera episodes (a large cast of characters play out multiple, simultaneous, overlapping stories that never end). UNIVERSE is the first storytelling system to devote special attention to the creation of characters. Complex data structures are presented to represent characters, and a simple algorithm is proposed to fill these in partly in an automatic way. But the bulk of characterization is left for the user to do by hand.

UNIVERSE is aimed at exploring extended story generation, a continuing serial rather than a story with a beginning and an end. It is in a first instance intended as a writer's aid, with additional hopes to later develop it into an autonomous storyteller. UNIVERSE first addresses a question of procedure in making up a story over a fictional world: whether the world should be built first and then a plot to take place in it, or whether the plot should drive the construction of the world, with characters, locations and objects being created as needed. Lebowitz declares himself in favour of the first option, which is why UNIVERSE includes facilities for creating characters independently of plot, in contrast to Dehn [3] who favoured the second in her AUTHOR program (which was intended to simulate the author's mind as she makes up a story).

The actual story generation process of UNIVERSE [10] uses plan-like units (plot fragments) to generate plot outlines. Treatment of dialogue and low-level text generation are explicitly postponed to some later stage. Plot fragments provide narrative methods that achieve goals, but the goals considered here are not character goals, but author goals. This is intended to allow the system to lead characters into undertaking actions that they would not have chosen to do as independent agents (to make the story interesting, usually by giving rise to melodramatic conflicts). The system keeps a precedence graph that records how the various pending author goals and plot fragments relate to each other and to events that have been told already. To plan the next stage of the plot, a goal with no missing preconditions is selected and expanded. Search is not depth first, so that the system may switch from expanding goals related with one branch of the story to expanding goals for a totally different one. When selecting plot fragments or characters to use in expansion, priority is given to those that achieve extra goals from among those pending.

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

MEXICA [16] is a computer model designed to study the creative process in writing in terms of the cycle of engagement and reflection [18]. It was designed to generate short stories about the MEXICAS (also wrongly known as Aztecs). MEXICA was a pioneer in that it takes into account emotional links and tensions between the characters as means for driving and evaluating ongoing stories. It relies on certain structures to represent its knowledge: a set of *story actions* (defined in terms of preconditions and post-conditions) and a set of *previous stories* (stated in terms of story actions).

The reflection phase revises the plot so far, mainly checking it for coherence, novelty and interest. The checks for novelty and interest involve comparing the plot so far with that of previous stories. If the story is too similar to some previous one, or if its measure of interest compares badly to previous stories, the system takes action by setting a guideline to be obeyed during engagement. These guidelines are a low level equivalent of author goals, driving which types of action can be chosen from the set of possible candidates. The check for coherence is only carried out over the final version of the story, and it involves inserting into the text actions that convey explicitly either character goals or tensions between the characters that are necessary to understand the story. Unless they are explicitly added during this check, goals and tensions are not included in the discourse.

The Virtual Storyteller [20] introduces a multi-agent approach to story creation where a specific director agent is introduced to look after plot. Each agent has its own knowledge base (representing what it knows about the world) and rules to govern its behaviour. In particular, the director agent has basic knowledge about plot structure (that it must have a beginning, a middle, and a happy end) and exercises control over agent's actions in one of three ways: environmental (introduce new characters and object), motivational (giving characters specific goals), and proscriptive (disallowing a character's intended action). The director has no prescriptive control (it cannot force characters to perform specific actions). Theune et al. report the use of rules to measure issues such as surprise and "impressiveness".

Comme il Faut (CiF) [13] is a knowledge-based system that models the interplay between social norm, social interactions, character desires, and cultural background. The underlying model of social interaction covers a range of aspects, from cultural static knowledge relevant to social interaction to fleeting desires of characters, with models for intervening factors like social exchanges, microtheories for significant concepts (such as friendship), and set of complex rules capturing likely behaviours of characters when faced with particular social circumstances.

Stella [11] performs story generation by traversing a conceptual space of partial world states based on narrative aspects. World states are generated as the result of non-deterministic interaction between characters and their environment. This generation is narrative agnostic, and an additional level built on top of the world evolution chooses the most promising ones in terms of their narrative features. Stella makes use of objective curves representing these features and selects world states whose characteristics match the ones represented by these curves. Stella is aligned to the current approach in the sense that simulation is also the base for generation. Stella, however, does not address characters' interactions as a key feature in the creative process.

In general, approaches to Interactive Storytelling have some degree of simulation as conceived in this work [1, 2, 12]. While every approach models the problem of story generation in a specific way, there exist some degree of similarity in the way they perform, namely by chaining sequential states that are driven or selected by an implicit or explicit model of plot quality.

3 A Model of Character Affinity

When running intelligent agents in simulations, and specially when they are in the form of intelligent virtual agents within virtual environments, some authors

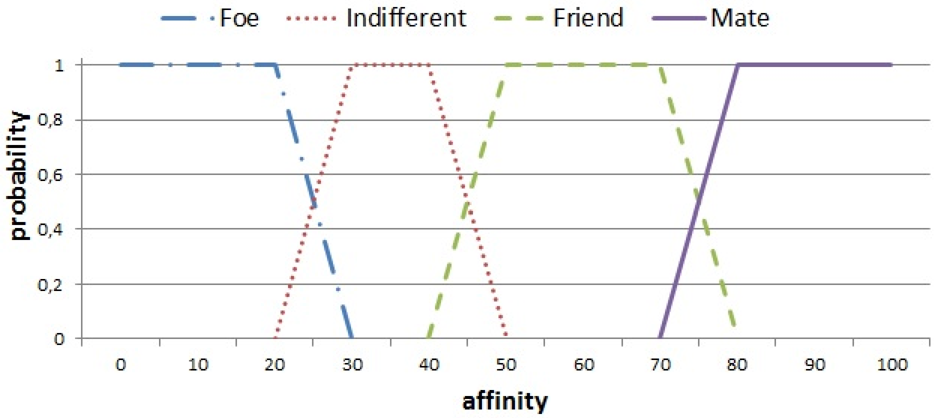

Fig. 1. Model of affinity

report the impossibility to run but a few of them at the same time [7, 6], since all the artificial intelligence involved in making them intelligent implies a high computational cost. One of our concerns when designing the current model has been for it to be as light-weight and cost-effective as possible, so its combination with other artefacts to create intelligent characters with personality traits, emotions and complex decision making maintains a low computational cost.

One of the most relevant research works on the subject is *Thespian* [19], the social behaviour framework used in [7]. In this work, the authors describe the use of an affinity factor to model social interaction which affects how characters can behave with each other. In this case, affinity is affected by other factors, such as social obligations and characters goals.

The first approach we used was to model affinity as a set of symbolic values that would be subsequently used to reason about the character's actions. The advantage of this approach is that it is easier to understand and reason about what is happening in the simulation. However, it is more difficult to operate with these values, a certain semantic has to be added to the code to understand how these values change and, on the long run, symbolic reasoning tends to be slow when combined with other processes.

Therefore, we have opted for a numeric representation that allows us to use common arithmetic operators to modify the degree of affinity between characters. The main drawback of this approach is that it is more difficult to calibrate the model and interpret what is happening in the simulation. To reduce this drawback, we have opted for a representation similar to the fuzzy concepts proposed in [22], as shown in Fig. 1, an approach that has already been used by other authors to model cognitive architectures [6, 4].

We have modelled four levels of affinity according to four different kinds of affinity: foe (no affinity), indifferent (slight affinity), friend (medium affinity) and mate (high affinity). These four levels of affinity overlap on their limits, which allows for relationships not to change constantly when moving around the limits of two different levels. Therefore, the change from *indifferent* to *friend*, takes place when the affinity value is 70, and changing from *friend* to *indifferent* is done with an affinity value of 50.

An additional aspect of affinity is that it is not symmetrical. Given two characters, their mutual affinity is likely to have different values and it may even be situated in different levels, with the exception of mates: *character A* considers *character B* as its mate only if *character B* considers *character A* as its mate, too. However, if they are not mates, *character A* may think *character B* is a friend, while *character B* may think *character A* is a foe.

There are two ways in which the affinity value can change. The first one is by lack of interaction, in which case the affinity value moves towards the indifferent level. The second one is through interactions among characters, that obey a few simple rules. There is a set of interactions that is appropriate for each affinity level, so when dealing with a friend a character may only propose to carry out friend actions, but not mate actions. In addition, characters ignore proposals that do not correspond to their perceived affinity level, and receiving such proposals may penalise the affinity with the character proposing them. The exception to this rule are foes, who carry out what they intend to do irrespective of what the other character may want. When receiving a proposal, a character may decide to either accept or reject it. If the proposal is accepted, both characters increase their mutual affinity. If it is rejected, the proposer will penalise its affinity with the receiver. Actions for the same level of affinity have different impact on it. For example, a romantic dinner has a higher effect on affinity than watching tv together. Similarly, the negative effect of rejecting an invitation is opposite to the positive effect of accepting it.

4 Implementation of the Model

The described model has been implemented by means of a multi-agent system developed using JADE[1].

The main objectives of the implementation were: to test the model apart from other factors such as the environment in which the story takes place or the personality traits and emotional state of the characters, which cause them to make different decisions in the same situation; and to implement the model as independent as possible from the domain of the story, so it can be easily used to generate different kinds of stories.

The system consists of two types of agents: a Director Agent, which is in charge of setting up the execution environment and creating the characters; and Character Agents, one for each character of the story, which are the ones that interact to generate the story. In the current case, the story consists of a set of interactions that make the affinity between characters change accordingly.

The information that the Director Agent needs to set up the execution environment is written in a text file that contains the number and names of the

[1] http://jade.tilab.com

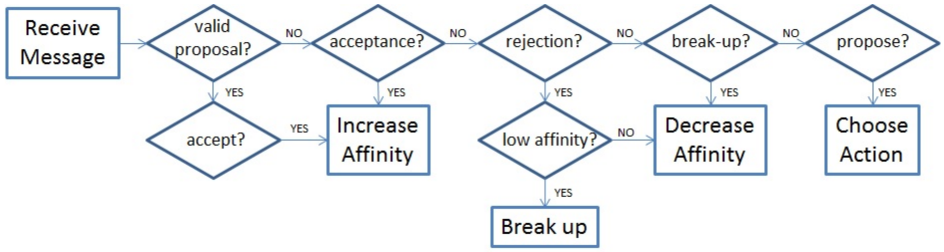

Fig. 2. Interaction protocol for mate characters

characters that have to be created. Subsequently, the information needed to configure each character is also included in a text file that currently contains the name and gender of the character, the name and affinity with its mate, a list of friend names and affinities and a list of foe names and affinities.

Each Character Agent is endowed with three different behaviours: one to interact with its mate, another one to interact with its friends and the last one to interact with its foes. Each behaviour is independent from the others, and they can all be added, blocked and removed dynamically to keep the system as lightweight as possible. These behaviours run the interaction protocols needed to implement and test the affinity model. The information needed by these behaviours, mainly the actions that characters can perform when executing them, along with the degree in which these actions affect the affinity between characters, is also stored in text files, so it is easy to add and remove actions and modify their influence on the affinity without having to change the code and recompile the system. This also means that, as far as the affinity model is concerned, actions have no semantic apart from their influence on the affinity value.

In Fig. 2 we can see how the MateBehaviour works. When a character receives a message from its mate, it checks whether it is a proposal to do something together or not. If it is, it may accept it, in which case it increases its affinity with its mate, or decline. In both cases, the decision is notified to its mate. When a character receives an acceptance, it increases its affinity with its mate, whereas if is a rejection, it checks its affinity with its mate and, if it is already below a given threshold, it will decide to break up with its mate. When an agent receives a break-up notification, it decreases its affinity with its mate and decides whether to make it its friend or its foe. If none of this has happened, the character may then decide to propose its mate to do some activity together.

In the points where the characters should make a decision, such as whether to accept a proposal to do something or not, a random probabilistic decision has been made in order to be able to test the implementation of the affinity model by itself, without the interference of other processes. Thus, for example, the probability of accepting a proposal of the character's mate has been empirically established in 0.6. The reason for this value in our experiments is that it is high enough for couples to remain fairly stable, but it is low enough to keep things happening, so that stories don't turn boring.

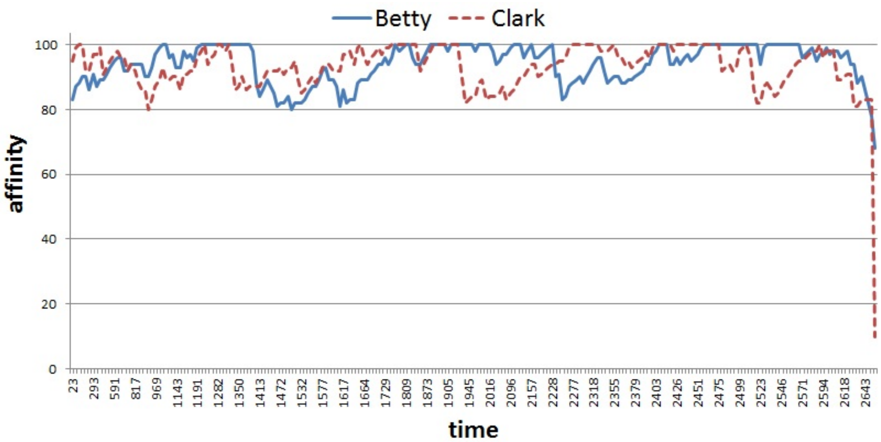

Fig. 3. Evolution of the affinity between characters Betty and Clark

Running the implemented system with 15 characters (8 females and 7 males forming 7 couples), we have chosen the couple formed by two of them, Betty and Clark, to show the evolution of their affinity over time, as shown in Fig. 3. The image shows how their affinity varies between 80 and 100 over the execution, but both affinities evolve separately (although they are not completely independent). In general, affinity increases at a lower speed than it decreases. This is due to two causes: the heavier impact that rejections have on the affinity than acceptances; and the fact that, if no other action is taken, affinity slowly fades over time, which affects the overall decreasing speed. The most remarkable fact that can be appreciated in the image is how, at the end of the execution, the affinity between both characters falls dramatically due to the final break-up of the couple. This break-up is caused by Clark's rejections of several of Betty's proposals, which in turn causes Betty's decision to put an end to the relationship, once the affinity level has gone below the threshold of mate affinity.

5 System Test

Although a more exhaustive testing is still needed, we have run some tests with different configurations in order to check empirically that the model is consistent and that it allows us to generate different stories by just modifying the thresholds that are used for the characters to make decisions.

In Figs. 4 and 5 we can observe the results of two of these tests, where we show three of the most representative affinities in each of the generated stories. In both tests, the initial situation was the same: 15 characters, 14 mate relationships (i.e. 7 couples), 28 friend relationships (with all characters having at least one friend) and 4 foe relationships (only two characters had foes).

The graphics displayed in Figs. 4 and 5 show how the affinities between two given characters change over time. They have been selected because they

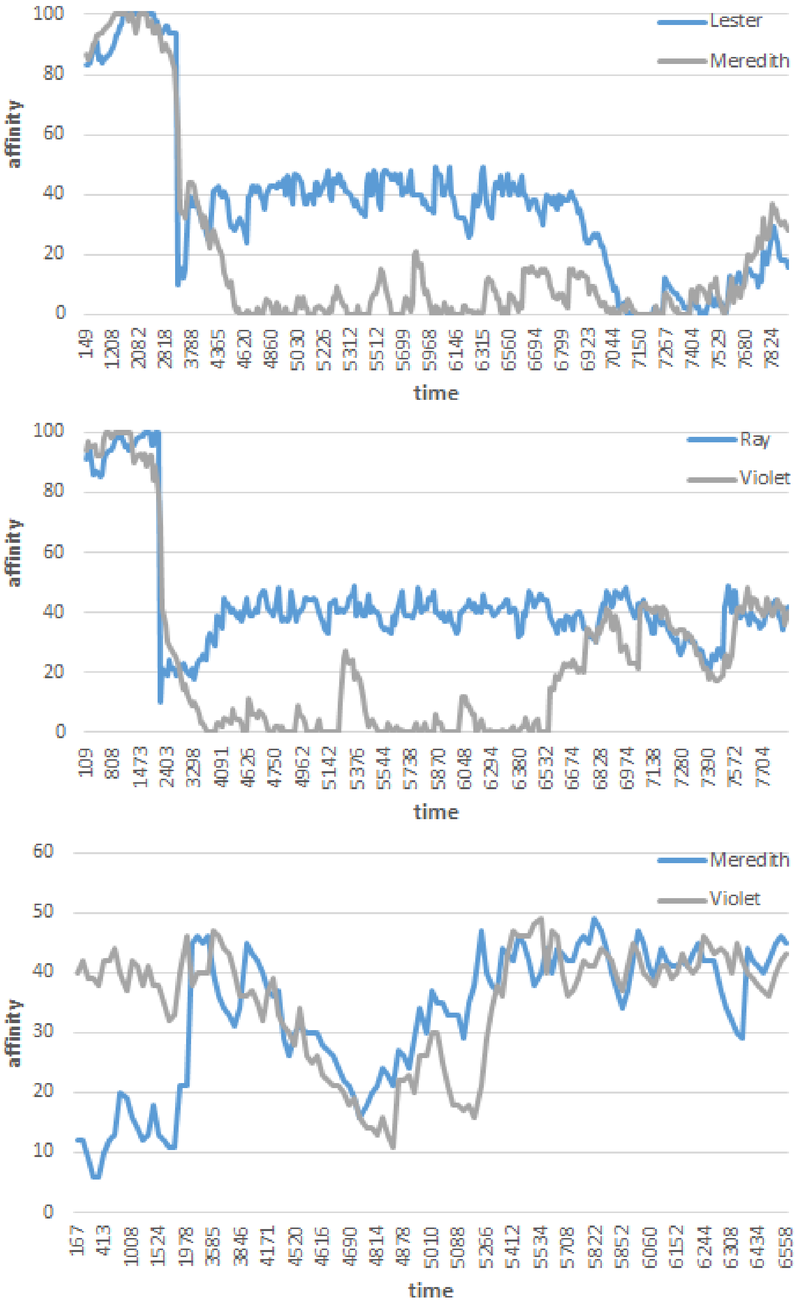

Fig. 4. Evolution of the affinity with a high threshold for foe interaction

Fig. 5. Evolution of the affinity with a low threshold for foe interaction

represent relationships that have persisted over time and have a high degree of variability, so they are likely to give rise to interesting stories.

One of the aspects that has drawn our attention is that the affinities between two characters tend to evolve in parallel, and although at some points they may diverge, they keep on moving in the same direction and they eventually converge, showing different values but a similar tendency.

In Fig. 4 we can see the result of running the system with high thresholds for foe interaction (0.75 over 1) which means there is only a probability of 0.25 that a character will decide to act against a foe or to respond to an act from a foe. Higher values for these thresholds (i.e. less probability for action) often produce stories with little or no interaction among foes. In general, the stories we have obtained produce some stable relationships, a few relationships that change from friend to foe and back again, and a fair amount of relationships that fade into indifference over time.

The first two graphics displayed in Fig. 4 show a similar evolution of the affinity value. In both cases, one of the character's affinity value decreases enough for the character to decide to break up the relationship and, as a response, their mate decides to consider them as foes. At this point, the values of the affinities start diverging, and one of the affinities moves to the lower range of foe affinity, while the other moves to the upper values. In both cases, the affinities evolve separately, showing the same tendency but with separated values. As it can be seen, the affinities eventually converge and start showing the same evolution.

The third graphic shows a different starting situation, since Meredith considers Violet as an enemy, while Violet is initially indifferent about Meredith. When Meredith acts against Violet, she immediately responds in the same way and considers Meredith as a foe, but using a higher affinity value as a starting point. As in the other cases, the affinity values eventually converge and start evolving in parallel.

In all three cases, we can see how the affinity values tend to separate from the lowest limit, due to the fact that, since the threshold for action is set to a high value, the characters don't act too much against their foes, so there is a chance for the affinity values to increase and for the relationships to evolve in different ways.

In contrast to the described graphics, the ones displayed in Fig. 5 show what happens when the threshold for action and reaction to foe events are set to lower values (in this case, 0.4 to initiate an action and 0.5 to react to another character's action). The probability of acting against a foe is considerably higher than in the former case, and the graphics show how, in similar initial situations, the affinity values present much less variation and tend to be much closer to the lower limit of the foe affinity values.

However, as it can be seen in the first of the three graphics, it is still possible to obtain stable relationships in different ranges of the affinity spectrum, as long as they do not fall into the foe category, from which it is difficult to get out due to the high activity imposed by the thresholds. This behaviour is useful when our aim is to generate stories that do not have a happy ending.

6 Discussion

There is room for improvement in the current model. A number of design decisions have been taken in order to prove the plausibility of the hypotheses, but these can be further developed.

One of the most relevant aspects of the current system is the description of the domain, namely the number of actions that can take place among the characters, the values for the parameters, and so on. The current prototype proposes a straightforward implementation meant to serve as placeholder for a richer definition of the domain. More specifically, the interaction protocols used to drive the calculation of the affinities has been defined by hand, being driven only by common sense intuition. While this well serves the purpose of deploying a working prototype, following a model of affinities grounded on a psychological model is necessary for the conclusions to have a general validity.

From the point of view of creativity, two important aspects are worth the discussion: the evaluation process and the exploration and comparison of different evolution sequences for affinity and their relation with the quality of the generated narratives. Wiggins proposes a model in which every creative artifact –story in this case– is a member of a more general set of artifact, and the overall objective of creativity is to find the best ones [21]. From this perspective, it makes sense to compare the output of different executions in terms of the perceived quality of the generated narratives. While the design of a full evaluation procedure is still planned as part of the future work, the current work makes it possible to highlight some particular points to be addressed in this evaluation.

First, the evaluation must focus on the emotional aspects of the stories and not on a general value of a narrative. It would not make sense to assess other aspects as literary quality, for instance. While this might sound trivial, it is crucial to acknowledge that isolating particular aspects when evaluating a narrative is not simple. As humans, we tend to provide overall ratings that are not properly bound to any concrete parameter of stories.

Second, if we assume the previous point solved, we would have to go the other way around and discover the influence of the evolution of the affinity on the overall perception of the story, which is exactly the general objective for this kind of creative systems.

7 Conclusions

In the previous sections we have described a model to express characters relations based on their affinity, and we have shown how this model has been implemented using a multi-agent system that generates character-based plots.

We have run the implemented system with up to 15 characters and the results show that the possible interactions are rich enough to generate a high variety of stories, and the variability between executions is such that, for the same initial situation, the output story is always different. For this reason, it is necessary to develop evaluation mechanisms that allow us to decide when a story is good or when we need to keep on trying to generate a story that meets certain quality standards.

In addition, it is possible to change most of the information needed to generate the stories through configuration files, which makes it easy to produce new stories with different situations almost effortlessly. However, these situations are still generated by hand, so for non-trivial set-ups, an automated way to check and correct the consistency of the initial situations would be useful.

In particular, we have seen that the model can be configured in such a way that it keeps relationships stable, but it allows enough flexibility for unexpected events to happen and make the plot more interesting, and it can also be tuned so that we can easily obtain a story with the desired type of ending (either happy or not).

8 Future Work

The model can be further improved. More relations can be included in the system and a more refined selection of them can be tried and evaluated. The results on how the selection of features affects the complexity and the number of generated stories can shed light on what the set of relevant aspects of affinity are.

There is still much work to be done in order to generate stories that are not only based on character relationships. We will start by integrating the work described in [15] (which this paper extends) with the generator presented in [9], which will allow us to situate characters within a map and a context, giving us the chance to generate interactions only when proximity makes them possible.

We intend to endow characters with personality traits and emotions, in order to complement the affinity model and give characters the possibility to make decisions in a more cognitive way. We plan to use an approach similar to the one described in [6] to model emotions, so that it can be easily integrated with the present model and the implementation can maintain a low computational cost. This work line is specially relevant, since it will allow characters to make consistent decisions according to the personality and state of mind, instead of random ones, and it will also allow them to behave in a different way from each other.

Finally, since we are capable of generating a high variety of different stories, we need to develop a mechanism to evaluate these stories in order to discard those that lack interest and to refine the generation mechanism so that less non-interesting stories are generated. The first hypotheses have been pinpointed (as show in Section 6), but a concrete computational model is still to be created.

Acknowledgements

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

References

1. Aylett, R.S., Louchart, S., Dias, J., Paiva, A., Vala, M.: Fearnot!: An experiment in emergent narrative. In: 5th International Working Conference on Intelligent Virtual Agents. pp. 305–316. LNCS, Springer-Verlag (September 2005)
2. Cavazza, M., Charles, F., Mead, S.: Planning characters' behaviour in interactive storytelling. J. of Visualization and Computer Animation 13, 121–131 (2002)
3. Dehn, N.: Story generation after tale-spin. In: In Proceedings of the International Joint Conference on Artificial Intelligence. pp. 16–18 (1981)
4. El-Nasr, M.S., Yen, J., Ioerger, T.R.: Flame - fuzzy logic adaptive model of emotions. Autonomous Agents and Multi-agent systems 3(3), 219–257 (2000)
5. von Goethe, J.W.: Elective Affinities / Kindred by Choice (1809)
6. Imbert, R., de Antonio, A.: An emotional architecture for virtual characters. In: International Conference on Virtual Storytelling. pp. 63–72 (2005)
7. Johnson, W.L., Valente, A.: Tactical language and culture training systems: Using artificial intelligence to teach foreign languages and cultures. In: AAAI. pp. 1632–1639 (2008)
8. Klein, S., Aeschliman, J.F., Balsiger, D., Converse, S.L., Court, C., Foster, M., Lao, R., Oakley, J.D., Smith, J.: Automatic novel writing: A status report. Tech. Rep. 186, Computer Science Dept., The University of Wisconsin (December 1973)
9. Laclaustra, I.M., Ledesma, J.L., Méndez, G., Gervás, P.: Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In: 5th International Conference on Computational Creativity. Ljubljana, Slovenia (2014)
10. Lebowitz, M.: Storytelling as Planning and Learning. Poetics 14, 483–502 (1985)
11. León, C., Gervás, P.: Creativity in story generation from the ground up: Nondeterministic simulation driven by narrative. In: 5th International Conference on Computational Creativity, ICCC 2014. Ljubljana, Slovenia (06/2014 2014)
12. Mateas, M., Stern, A.: Structuring content in the Faade interactive drama architecture. In: Proceedings of AIIDE. pp. 93–98 (2005)
13. McCoy, J., Treanor, M., Samuel, B., Reed, A.A., Mateas, M., Wardrip-Fruin, N.: Social story worlds with comme il faut. IEEE Trans. Comput. Intellig. and AI in Games 6(2), 97–112 (2014)
14. Meehan, J.R.: Tale-spin, an interactive program that writes stories. In: Proc. of the Fifth International Joint Conference on Artificial Intelligence. pp. 91–98 (1977)
15. Méndez, G., Gervás, P., León, C.: A model of character affinity for agent-based story generation. In: 9th International Conference on Knowledge, Information and Creativity Support Systems. Limassol, Cyprus (11/2014 2014)
16. Pérez y Pérez, R.: MEXICA: A Computer Model of Creativity in Writing. Ph.D. thesis, The University of Sussex (1999)
17. Polti, G.: The Thirty-Six Dramatic Situations (1917)
18. Sharples, M.: How We Write. Routledge (1999)
19. Si, M., Marsella, S.C., Pynadath, D.V.: Thespian: Modeling socially normative behavior in a decision-theoretic framework. In: Intelligent Virtual Agents, Lecture Notes in Computer Science, vol. 4133, pp. 369–382. Springer (2006)
20. Theune, M., Faas, E., Nijholt, A., Heylen, D.: The virtual storyteller: Story creation by intelligent agents. In: Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference. pp. 204–215 (2003)
21. Wiggins, G.: A preliminary framework for description, analysis and comparison of creative systems. Knowledge-Based Systems 19(7) (2006)
22. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers & Mathematics with Applications 9(1), 149–184 (1983)