A Model of Character Affinity for Agent-Based Story 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* [11] 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* [15] 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 [4]. 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. The particular murderer and victim depended on the character traits specified as input. 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. Personality characteristics are explicitly represented but marked as "not to be described in output".

TALESPIN [9], 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. The systems allows the possibility of having more than one problem-solving character in the story. 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 [6] 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. It is aimed at exploring extended story generation, a continuing serial rather than a story with a beginning and an end. 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).

MEXICA [10] is a computer model designed to study the creative process in writing in terms of the cycle of engagement and reflection [12]. 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. 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. 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.

The Virtual Storyteller [14] introduces a multi-agent approach to story creation where a director agent is introduced to look after plot. Each agent has its own knowledge base and rules to govern its behaviour. In particular, the director agent has basic knowledge about plot structure 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).

Comme il Faut (CiF) [8] 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 rules capturing likely behaviours of characters when faced with particular social circumstances.

Stella [7] 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.

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 report the impossibility to run but a few of them at the same time [3, 2], 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* [13], the social behaviour framework used in [3]. 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

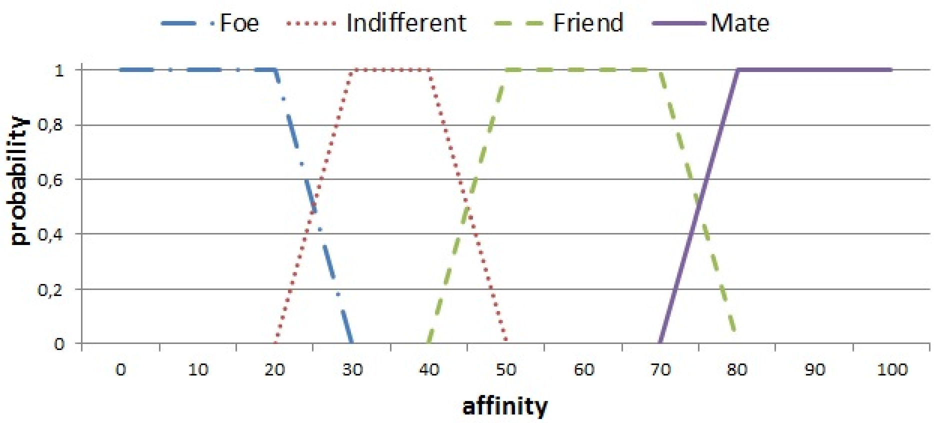

Fig. 1. Model of affinity

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 [16], as shown in Fig. 1, an approach that has already been used by other authors to model cognitive architectures [2, 1].

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 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

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

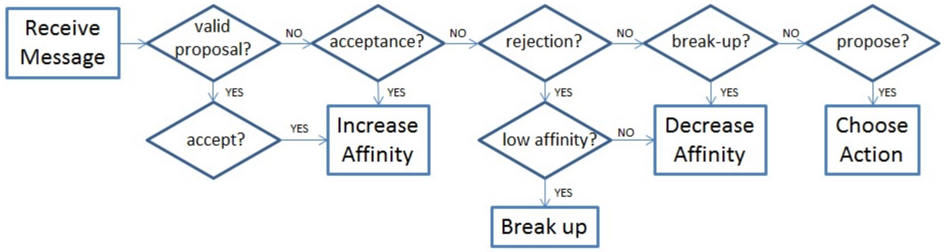

Fig. 2. Interaction protocol for mate characters

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.

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 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 to that generates character-based stories.

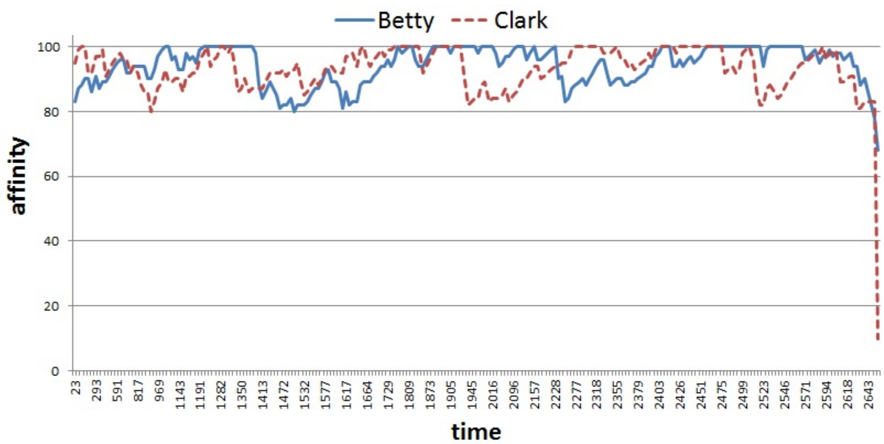

Fig. 3. Evolution of the affinity between characters Betty and Clark

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. 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.

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 so that unexpected events can happen to make the plot more interesting.

6 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 this paper with the generator presented in [5], 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 [2] to model emotions, so that it can be easily integrated with the present model and the implementation can maintain a low computational cost.

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.

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. M. S. El-Nasr, J. Yen, and T. R. Ioerger. Flame - fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, 3(3):219–257, 2000.
2. R. Imbert and A. de Antonio. An emotional architecture for virtual characters. In *International Conference on Virtual Storytelling*, pages 63–72, 2005.
3. W. L. Johnson and A. Valente. Tactical language and culture training systems: Using artificial intelligence to teach foreign languages and cultures. In *AAAI*, pages 1632–1639, 2008.
4. S. Klein, J. F. Aeschliman, D. Balsiger, S. L. Converse, C. Court, M. Foster, R. Lao, J. D. Oakley, and J. Smith. Automatic novel writing: A status report. Technical Report 186, Computer Science Dept., The University of Wisconsin, December 1973.
5. I. M. Laclaustra, J. L. Ledesma, G. Méndez, and P. Gervás. 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.
6. M. Lebowitz. Storytelling as Planning and Learning. *Poetics*, 14:483–502, 1985.
7. C. León and P. Gervás. Creativity in story generation from the ground up: Non-deterministic simulation driven by narrative. In *5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, 06/2014 2014.
8. J. McCoy, M. Treanor, B. Samuel, A. A. Reed, M. Mateas, and N. Wardrip-Fruin. Social story worlds with comme il faut. *IEEE Trans. Comput. Intellig. and AI in Games*, 6(2):97–112, 2014.
9. J. R. Meehan. Tale-spin, an interactive program that writes stories. In *Proc. of the Fifth International Joint Conference on Artificial Intelligence*, pages 91–98, 1977.
10. R. Pérez y Pérez. *MEXICA: A Computer Model of Creativity in Writing.* PhD thesis, The University of Sussex, 1999.
11. G. Polti. *The Thirty-Six Dramatic Situations.* 1917.
12. M. Sharples. *How We Write.* Routledge, 1999.
13. M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: Modeling socially normative behavior in a decision-theoretic framework. In *Intelligent Virtual Agents*, volume 4133 of *Lecture Notes in Computer Science*, pages 369–382. Springer, 2006.
14. M. Theune, E. Faas, A. Nijholt, and D. Heylen. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, pages 204–215, 2003.
15. J. W. von Goethe. *Elective Affinities / Kindred by Choice.* 1809.
16. L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1):149–184, 1983.