Evolutionary Story Sifting over the Log of a Social Simulation*

Pablo Gervás[0000−0003−4906−9837] and Gonzalo Méndez[0000−0001−7659−1482]

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

Abstract. The process of selecting subsets out of a sequence of events on the grounds that told together they constitute an interesting narrative–known as *story sifting*–has become a topic of interest due to its applicability in video games that automatically develop large scale simulations of story worlds. Existing approaches to story sifting operate by matching subsequences of the available events onto patterns of plot considered to be of interest. The present paper proposes a two stage approach that combines a process of matching small strings of events connected by common sense relations–such as asking someone on a date and having them accept, or developing an attachment to someone who has given us a present–and an evolutionary search procedure that explores combinations of this type of paired events into longer sequences that constitute small plot lines about romantic entanglement. This procedure is run over the log of a multi-agent system that simulates a set of characters that develop a set of affective affinities between them as a result of social interactions of a romantic nature.

Keywords: narrative generation · multi-agent simulation · story sifting · evolutionary procedure.

1 Introduction

As the average person goes through their day, they witness a thousand small events and take part in a further set of events. At any point during that time, if called upon, they can very easily isolate a very concise subset of those events as worth telling, and they can build the resulting selection into a story that seems cohesive, makes sense, and appears to have a connecting thread that makes it interesting. This is what any of us does when asked about our day, or what we have been doing, or what is happening back where we came from. In computational terms, these tasks are far from trivial.

* This paper has been partially funded by the projects CANTOR: Automated Composition of Personal Narratives as an aid for Occupational Therapy based on Reminiscence, Grant. No. PID2019-108927RB-I00 (Spanish Ministry of Science and Innovation), the ADARVE (Análisis de Datos de Realidad Virtual para Emergencias Radiológicas) Project funded by the Spanish Consejo de Seguridad Nuclear (CSN).

In practical terms, these tasks involve a number of cognitive operations: (1) identify relations between the events that have happened (usually of causality or intention), (2) identify chains of relations between the events, (3) consider which events in a given chain would be relevant to mention in an interesting story, and (4) consider which chains of events, included together in a given story, give rise to an interesting plot.

The specific challenge of identifying what may constitute an interesting plot ultimately relies on the ability to model the reactions of a reader when reading the corresponding story. The present paper brings together a number of prior achievements in the computational modeling for plot and the dynamics of reading the discourse for a story with an evolutionary algorithm that operates over the set of events from a log for a simulation about social interactions between a set of characters. These social interactions basically involve one character proposing an activity to another character, who decides whether to accept the invitation or not.

2 Previous Work

For the present paper, we will review previous work on social simulations as possible sources from which to extract narratives, story sifting solutions to select story-worthy events from a set of facts, and narrative composition for telling stories about facts that are already known rather than invented for the purpose. Finally, efforts to model computationally the reaction of a reader to a given narrative are reviewed in search of solutions that can provide means for quantifying the relative merit of different candidate narratives.

2.1 Social Simulations as Sources for Narrative Renderings

A multi-agent system (MAS) consists of a set of software entities, the agents, that are autonomous–they can make their own decisions— and interact with their environment and among themselves in terms of cooperation, coordination, negotiation or competition.

The work described in [12] presents a multi-agent planner system that is capable of generating stories taking into account plot coherence and character believability. A similar approach is used in the Sabre system [15] where a centralized planner is used to generate the story. Comme il Faut (CiF) [7] is a knowledge-based system that models the complex interplay between social norms, character desires, cultural background, and social interactions. Based on this information it can be used to support simulations of agents engaging in social settings.

Charade [10] is a multi-agent simulation designed to express relations and interactions between characters in a storyworld based on the existing affinities between the characters, and to model the evolution of these affinities through a given period of time. Based on the model of affinities between them, agents propose activities to other agents, who accept or reject them. The result is a log of interactions and evolutions of affinity levels. An example of a fragment of a log for the Charade sytem is shown in Table 1. These logs are subsequently used to generate episodes within a narrative [1].

```
Megan PROPOSE friend_have_lunch Meredith
Lester PROPOSE friend_chat Robert
Suzette PROPOSE friend_chat Silvy
Betty PROPOSE friend_weekend_out Clark
Meredith PROPOSE mate_watch_tv Lester
Clark REJECT-PROPOSAL friend_weekend_out Betty
Lester REJECT-PROPOSAL mate_watch_tv Meredith
Meredith ACCEPT-PROPOSAL friend_have_lunch Megan
(...)
```

Table 1. An example of a fragment of the log generated by the Charade multi-agent simulation system.

2.2 Story Sifting

Early work on computational creativity generated literary texts by selecting a subset of lines from an extensive source file [11]. A refinement on this technique that mines sequences of events corresponding to interesting stories from the logs of agent-based simulations has become a line of research known as *story sifting*. James Ryan's PhD thesis [13] outlines how, rather than automatically inventing stories, narrative may emerge from the activity of characters set in motion in a simulated story world, and defines the task of curating such narratives out of simulation logs as story sifting. The Felt story sifting and simulation engine [6] introduced the concept of *story sifting patterns*, which are descriptions of sequences of events that exhibit high potential to be part of interesting narratives. This line of research lead to the development of Winnow [5], a domain-specific language for specifying story sifting patterns that can be run on ongoing simulations to identify event sequences with narrative potential.

2.3 Evolutionary Solutions for Exploring Search Spaces of Plot

Evolutionary solutions rely on fitness functions that measure the quality of the final output with no consideration of how particular individuals might be constructed. This makes them particularly suitable to explore a search space of narratives based on existing work analysing narrative as a product. Examples of the use of evolutionary solutions to validate particular narratives in relation to other candidate drafts are: [14] which relies on an evolutionary approach to identify optimal candidates from an initial population built using knowledge-based heuristics, [4] which explores a search space of combination of plot templates using metrics for story consistency from a semantic point of view, or [8] which identifies combinations of partially ordered graphs of events associated with particular entities to maximise story coherence and story interest.

Of particular relevance for the approach in this paper is the work of [3] which applies an evolutionary search process to identify optimal combinations of plot-relevant units of abstraction–called *axes of interest*–into narratives. The axes of interest have free variables that need to be instantiated with particular characters. The procedure involved separate processes for selecting a set of axes of interest with unbounded variables for the roles of the characters, establishing a relative ordering between them, and creating instantiations of the unbound variables with characters for the story. The instantiations of these variables are chosen to ensure meaningful connections across events in the story, both in terms of causal relations between events and certain characters being involved in related events. An adaptation of this process is used in this paper to sift stories from simulation logs.

3 Evolutionary Story Sifting from the Log of a Simulation

The present paper operates on the log of events generated by the Charade multi-agent simulation system [10] as described in Section 2.1. The set of events in the log is read into a conceptual representation to allow further processing. For the current prototype, the following types of events are considered: `PROPOSE`, `ACCEPT-PROPOSAL` and `REJECT-PROPOSAL`.

3.1 Capturing Plot Relevant Connections in the Representation Format

The present paper bases its approach to story sifting in the identification of plot relevant connections between events in the log. To achieve this, a plot is represented in terms of *plot atoms*, which are abstract descriptions of an event (such as a character proposing an activity to another) that specify how the roles specific to the plot atom (proposer, proposee) are related to the set of characters in a given story. The causal connections between plot atoms in a story is captured in our representation by the concept of *axes of interest* [2]. The axes of interest being considered for the present prototype are shown in Table 2.

Axes of interest are used to parse the sequence of events in a log of the Charade system into subsets of events grouped together by virtue of being instantiations of the plot atoms in a given AoI, appearing in the log in the correct relative order and with the characters involved matching the constraints of the AoI in terms of roles. Such subsets of events are referred to as *plot projections*. Each plot projection shows the plot atoms in the axis of interest that it instantiates, the set of assignments to the roles corresponding to the AoI, and the position in the input discourse in which the corresponding plot atoms appear. An example of the parse of a Charade log into a set of plot projections is shown in Table 3.

Axis of Interest	Activities	Participating characters
ProposalAccepted	ProposeActivity	proposer = **x**, proposee = **y**
	ActivityAccepted	proposer = **x**, proposee = **y**
ProposalRejected	ProposeActivity	**x**, proposee = **y**
	ActivityRejected	**x**, proposee = **y**

Table 2. The two basic Axes of Interest employed by the system. Each axis of interest defines the connection between a proposal event and either an acceptance or a rejection of it, in terms of the co-instantiation of the variables involved (shown in bold in the table).

Plot Element	Arguments	Position
ProposeActivity	[Betty, Clark, friend_weekend_out]	4
ProposedActivityRejected	[Betty, Clark, friend_weekend_out]	6
(...)		
ProposeActivity	[Suzette, Silvy, friend_chat]	3
ProposedActivityAccepted	[Suzette, Silvy, friend_chat]	11
(...)		

Table 3. A example of a fragment of the parse of a Charade log into a set of plot projections.

3.2 Evolutionary Content Selection based on Plot Projections

By selecting only events which are part of some plot projection we ensure that the set of characters appearing is fairly coherent and the relative order in which things happen makes some sense. To ensure full satisfaction we need to consider a further filter that guarantees full connectivity between all the events present, and which establishes some constraint on the relative order in which they appear. So we need means to identify when a selection of plot projections makes sense as a story. The present paper proposes the use of an evolutionary solution for this task.

Our procedure for identifying interesting sub-sequences of events from a log for the Charade system is defined as an adaptation of the solution by [3] described in Section 2.3. Because our present tasks operates from an input–the log of the Charade system–that already established the set of events, their relative order, and the characters that take part in them, most of the procedure needs to be defined anew. However, we will retain the mechanisms for establishing the relative quality of a given story draft.

Since logs may run for long periods, the search space of all possible sub-sequences of events that can be extracted from a log is very large. To focus on output narratives of an acceptable length, the system takes as input an upper bound on the number of events that can be selected out of a log to build a narrative.

The adapted evolutionary procedure is initialised with a genetic representation in the form of a numerical vector that encodes the choice of which of the plot projections obtained from the parsing of the log to include in the story draft under construction. All the other characteristics of the story come already decided in the log, namely: relative order of events and assignment of characters to play the roles in each one of them.

A population of individual drafts is created by random assignment of values to this genetic representation. Values of 1 indicate the plot projection in the given position is to be included in the story–gene activated–, value of 0 indicates a plot projection to be ommitted–gene deactivated. To focus on output narratives of an acceptable length, the system constructs the initial population with a combination of possible lengths. The upper bound on story size is set to 16. The number of genes activated in the genetic representation vector must take this value into account. Because all the axes of interest currently employed connect a fixed number of two events, the number of activated genes is set to half the desired story size. The set of story sizes explored is currently set to 6, 8, 10 and 12, which correspond to drafts with 3, 4, 5 and 6 projections. With each projection involving two characters, three projections is the minimal set that allows for interactions of at least two different types between two characters or interactions between more than two characters. The upper bound of 12 is set empirically to avoid too large drafts.

Mutation operators are defined so that they deactivate some gene in the representation for every gene that they activate. This ensures that the desired size for the story drafts is respected during evolution.

Cross-over operators are defined to select at random a point in the genetic vector, divide the gene vector for two different drafts at that point, and combine each initial half with the final half of the other draft.

The population is evolved over a desired number of generations. The fitness function applied is described in the following section.

3.3 Metrics on Story Draft Quality

The fitness function for the evolutionary procedure is based on metrics of three types:

- metrics that restrict draft size to manageable proportions
- metrics that measure story cohesion
- metrics the measure character variety

Metrics to control draft length assign score of 100 to drafts under 16 events in length, and 0 otherwise.

The approach in this paper assumes that story drafts will have a higher quality if constraints of a certain type hold between the projections selected to appear in the draft. These constraints specify the relative position in which the events in the selected plot projections appear in the overall sequence of the story draft with respect to other events in which the same characters participate. To provide quantitative measure of these aspects we rely on the measures of correct sequencing of events, and acceptable occurrence of characters sharing roles across AoIs proposed in [3]. For completeness, a brief summary is included here.

A given combination of AoIs, such as for instance Betty repeatedly proposing activities to Clark in spite of his continued rejections, acquires value if the reader can infer some kind of connections between the two AoIs involved by virtue of the relative sequencing and the specific instantiation of the characters. In the example shown, the interest of the particular combination in such a story draft arises from the perseverance exhibited by Betty.

A set of metrics is defined to capture this type of constraints for each combination of AoIs for which relevant connections can be established. The metrics we are considering now are driven by constraints of the type presented in Table 4. In each case, an informative label has been added in the first column to identify the feature that justifies the interest. For each entry marked in bold font in this table, the row immediately following describes role-sharing constraints and the rows after that describes sequencing constraints. The full set of constraints includes further types such as: `RefugeSomewhereElse`, `ChangeOfTarget`, `HappyStreak`, `ReturnInvite` or `CatchOnTheRebound`.

Perseverance	ProposalRejected	ProposalRejected
	proposer = proposer	proposee = proposee
SuccessAfterFailure	ProposalRejected	ProposalAccepted
	proposer = proposer	proposee = proposee
HappyStreak	ProposalAccepted	ProposalAccepted
	proposer = proposer	proposee = proposee

Table 4. Example of constraints.

Finally, a metric has been introduced that scores 100 to drafts that have between 2 and 4 characters, 50 for drafts of 2 characters or characters above 4 but with less characters than the number of events in the draft, and 0 otherwise. This compensates the tendency of the more demanding metrics on story cohesion to force the drafts in the population towards stories with only two characters in them, and focuses the results on stories about specific characters.

A number of combinations of these metrics have been tested to use as fitness functions. They are discussed in the following section.

4 Discussion

The discussion includes a quantitative comparison with a number of baselines for the fitness metrics and a qualitative discussion of relations with prior approaches.

4.1 Comparative Evaluation

Any solutions provided by the system must be tested against some objective function. Such an objective function should require that the draft tell a story about a particular set of characters (the same set of characters recur throughout the draft) and that all events in the draft be tied in to some other event in the draft by some link of relevance.

These requirements allow us to establish some baselines for our fitness function. We consider the following alternatives for the fitness functions:

- percentage of projections in the draft that share at least one character with some other
- percentage of projections in the draft that are connected by consistency constraints
- ratio of size of largest subset of projections completely connected together by consistency constraints to overall number of projections
- ratio of size of largest subset of projections completely connected together by consistency constraints to overall number of projections weighted with percentage of applicable constraints that are actually satisfied

These metrics provide a progressive scoring, so that drafts where the sequencing constraints are not met are scored relative to how far they need to be modified for the constraints to be met. This allows mutations that modify the sequence in the right direction to be scored progressively higher, allowing evolution to converge towards optimal solutions.

Table 5 shows results for different versions of the system, each configured to run with a different fitness function. For each configuration the averages for a set of 6 runs are presented. The features shown include: maximum score found in population, number of generations required to converge to that top score, minimum score found in population, number of generations required to converge to that top score, length of draft in number of events, number of distinct characters that appear in the draft, average of the number of times that characters are mentioned.

The configuration of the evolutionary process was kept the same across the runs: initial population of 20 individuals, evolution over 20 generations, selection by accumulated fitness of a population of 20 for the next generation, and the mutation and cross-over operators described in Section 3.2.

Metric	Max	conv	Min	conv	Length	Chars	Slots/char
every aoi shares variable	100.0	2.0	100.0	10.7	10.0	4.0	5.0
every aoi in constraint	100.0	7.0	100.0	17.0	10.0	3.8	5.2
all aois connected by constraints	100.0	5.5	100.0	15.3	10.0	3.7	5.5
all aois connected by constraints (weighted)	100.0	5.0	100.0	15.8	10.0	3.3	5.7

Table 5. Results for different metrics.

Several observations can be made from these results. Although all the fitness functions produce populations that reach the top score, as the complexity of the metrics increases, the number of generations required for the scores to converge increases. The point of convergence for the minimum scores provides an indication of the relative difficulty involved in reaching the top score for all individuals in the population. This is relevant because it guarantees a complete population of quality drafts rather than a single one. There is clearly a preferred size of 10 events per draft that apparently maximises all of these fitness functions. This is understandable because it allows two pairs of characters, with two projections connected together for each pair, and at least one additional projection to connect one character from one pair to a character of the other. For larger drafts it is probably much more difficult to find instances of projections in the simulation that satisfy a sufficient number of constraints. The number of characters progressively decreases, as the events in the drafts tend to focus on a smaller set of characters that are the protagonists of the story. The average number of appearances in the draft per character also increases for the more demanding metrics.

In terms of efficiency, the system has been tested with various configurations of the evolutionary parameters to identify the choice leading to best performance. Table 6 shows best average scores over six runs with different values for population size and number of generations, while all other parameters were kept unchanged. These results show that the algorithm is sensitive to increase both in the size of the population and the number of generations. They also show that slightly higher gains in performance are obtained by the increase in the size of the population. As the increase in number of generations carries a significant overhead in increased execution times, we settle for a configuration of population size of 20 and 30 generations.

Table 7 shows an example of system output expressed in terms of the internal representation format, together with the corresponding story rendered as text automatically by the system using basic templates for each of the actions involved. The evolutionary solution was run for 30 generations, with a population size of 20 individuals. The final score for this draft is 100 /100.

This example has been selected on the basis of the perseverance of Silvy in the face of Mary's rejections, the decision of Mary to take refuge from Silvy in John, the appearance of Meredith to support Mary, and John's final change of mind later in rejecting Mary's proposals

	10	**20**	**50**
10	95.0	100.0	98.3
	76.7	84.3	74.7
	357.3	742.7	1049.7
20	100.0	100.0	100.0
	95.0	96.7	100.0
	743.8	1174.3	2660.8
30	100.0	100.0	
	98.3	100.0	
	984.2	2170.8	

Table 6. Best maximum and minimum scores (over 100) and execution times (in milliseconds) averaged over 6 runs on the same log.

PR-0-ProposeActivity-100	friend_play_tennis / Mary / Silvy
PR-0-ProposedActivityRejected-101	friend _play _tennis / Mary / Silvy
PA-0-ProposeActivity-113	friend _go _out / John / Mary
PA-0-ProposedActivityAccepted-117	friend _go _out / John / Mary
PA-1-ProposeActivity-532	friend _play _tennis / Mary / Meredith
PA-1-ProposedActivityAccepted-533	friend _play _tennis / Mary / Meredith
PA-2-ProposeActivity-596	friend _serious _talk / Mary / Meredith
PA-2-ProposedActivityAccepted-597	friend _serious _talk / Mary / Meredith
PR-1-ProposeActivity-691	friend _day _out / Mary / Silvy
PR-1-ProposedActivityRejected-692	friend _day _out / Mary / Silvy
PR-2-ProposeActivity-752	friend _serious _talk / John / Mary
PR-2-ProposedActivityRejected-753	friend _serious _talk / John / Mary

Silvy proposes to Mary to play tennis as friends. Mary rejects Silvy's invitation to play tennis as friends. Mary proposes to John to go out as friends. John accepts Mary's invitation to go out as friends. Meredith proposes to Mary to play tennis as friends. Mary accepts Meredith's invitation to play tennis as friends. Meredith proposes to Mary to serious talk as friends. Mary accepts Meredith's invitation to serious talk as friends. Silvy proposes to Mary to day out as friends. Mary rejects Silvy's invitation to day out as friends. Mary proposes to John to serious talk as friends. John rejects Mary's invitation to serious talk as friends.

Table 7. Example of story draft obtained by story sifting from a Charade log followed by the automated template-based rendering of the story

In general terms, it is important to point out that the quality of the stories that can be sifted out of a story log is constrained by the interest of the events in the log used as input. A possible way of taking this into account would be to develop an additional set of metrics to measure the interest of the events in the whole log, and to consider that as a baseline in the sense that stories sifted from the log cannot add interest other than by intelligent selection.

4.2 Relationship with Prior Work

The original work on the Charade simulation system [10] considered outputs of the simulation in terms of threads for specific characters. That is, a pair of related characters would be chosen, and an extract of all the events from the log in which the two characters were involved would be considered. This provided a span of events too long to be considered a real story. The original work explored the possible interest of these spans in terms of the evolution of the affinity value for relation between the chosen characters.

The Felt story sifting and simulation engine [6] used story sifting patterns that allowed it to represent combinations of events such as a repetition of betrayals, and parse a log to identify sequences of actions that matched them. The patterns used by Felt are expressed in a linear logic programming language that allows them to represent more complex combinations than those captured in the constraints used here. These patterns allow the user to perform very detailed searches. The solution proposed in this paper, combines a simpler set of constraints with the power of evolutionary approaches to search.

Now that Large Language Models (LLMs) are starting to be used to perform tasks that were previously carried out using other techniques, we have also tried to shed some light on their performance for story sifting [9]. We have tested ChatGPT using different prompts in order for it to process the original log used in this contribution and we found several limitations: it cannot use log files as input and has a restricted capacity for input data during conversations; it tends to forget instructions quickly, making it challenging to work with large amounts of data; it also overlooks sequential inputs, focusing more on the latest batch of information; when summarizing a set of events, ChatGPT's response is limited to a concise summary without the ability to select specific subsets based on narrative qualities; ChatGPT tends to introduce its own content, which can make it difficult to confine the output to the provided input data; ChatGPT's criteria for story sifting operations are unclear, generic, and challenging to influence for obtaining results aligned with different criteria or specific domains. As a result, our impression is that there is still room for improvement in relation to LLM-based story sifting, and consequently there is still need to keep on using and researching on other techniques for this task.

5 Conclusions

Story sifting has become an area of interest for its potential to automatically develop narratives emerging from large-scale simulations of story worlds. In this paper, we have proposed a two-stage approach that combines matching small strings of events connected by common sense relations with an evolutionary search procedure to explore combinations of paired events into longer sequences that form small plot lines about romantic entanglement. By running this procedure over a multi-agent system that simulates characters and their social interactions of a romantic nature, the resulting set of narrative plots show both coherence and an interesting chaining of events. The order of events is maintained as it was generated in the original simulation. Any cases of altered chronology, such as flashbacks or flashforwards, will be addressed in future research. To improve the set of proposed metrics, potential extensions to cover additional features of the Charade system will also be explored. The consideration of affinities between characters as a relevant aspect to the perceived quality of stories will be explicitly examined. Furthermore, the process outlined in this paper for generating stories can be combined with a method for producing multi-plotline stories, as described in [4].

References

1. Concepción, E., Gervás, P., Méndez, G.: Ines: A reconstruction of the Charade storytelling system using the Afanasyev framework. In: Proc. of the Ninth International Conference on Computational Creativity, Salamanca, Spain. pp. 48–55 (2018)

2. Gervás, P.: Generating a search space of acceptable narrative plots. In: 10th International Conference on Computational Creativity (ICCC 2019). UNC Charlotte, North Carolina, USA (07/2019 2019)
3. Gervás, P.: Evolutionary stitching of plot units with character threads. In: WIVACE 2022 XVI International Workshop on Artificial Life and Evolutionary Computation (09/2022 2022)
4. Gervás, P., Concepción, E., Méndez, G.: Evolutionary construction of stories that combine several plot lines. In: 11th International Conference, EvoMUSART 2022. Springer, Springer, Madrid, Spain (04/2022 2022)
5. Kreminski, M., Dickinson, M., Mateas, M.: Winnow: A domain-specific language for incremental story sifting. Proc. of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment **17**(1), 156–163 (Oct 2021)
6. Kreminski, M., Dickinson, M., Wardrip-Fruin, N.: Felt: A simple story sifter. In: Interactive Storytelling: 12th International Conference on Interactive Digital Storytelling, ICIDS 2019, Little Cottonwood Canyon, UT, USA, November 19–22, 2019, Proceedings. p. 267–281. Springer-Verlag, Berlin, Heidelberg (2019)
7. McCoy, J., Treanor, M., Samuel, B., Reed, A.A., Mateas, M., Wardrip-Fruin, N.: Social story worlds with comme il faut. IEEE Transactions on Computational Intelligence and AI in Games **6**, 97–112 (2014)
8. McIntyre, N., Lapata, M.: Plot induction and evolutionary search for story generation. In: Proc. of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1562–1572. Association for Computational Linguistics, Uppsala, Sweden (Jul 2010)
9. Méndez, G., Gervás, P.: Using chatgpt for story sifting in narrative generation. In: 14th International Conference on Computational Creativity. Waterloo, Ontario, Canada (06/2023 2023)
10. Méndez, G., Gervás, P., León, C.: On the Use of Character Affinities for Story Plot Generation, Advances in Intelligent Systems and Computing, vol. 416, chap. 15, pp. 211–225. Springer (02/2016 2016)
11. Montfort, N., Fedorova, N.: Small-scale systems and computational creativity. In: International conference on computational creativity. vol. 82 (2012)
12. Riedl, M.O., Young, R.M.: An intent-driven planner for multi-agent story generation. In: Autonomous Agents and Multiagent Systems, International Joint Conference on. vol. 2, pp. 186–193. IEEE Computer Society (2004)
13. Ryan, J.: Curating Simulated Storyworlds. Ph.D. thesis, University of California Santa Cruz, CA, USa (12 2018)
14. de Silva Garza, A.G., y Pérez, R.P.: Towards evolutionary story generation. In: Colton, S., Ventura, D., Lavrac, N., Cook, M. (eds.) Proceedings of the Fifth International Conference on Computational Creativity, ICCC 2014, Ljubljana, Slovenia, June 10-13, 2014. pp. 332–335. computationalcreativity.net (2014)
15. Ware, S.G., Siler, C.: The Sabre Narrative Planner: Multi-Agent Coordination with Intentions and Beliefs. In: Proc.of the 20th International Conference on Autonomous Agents and MultiAgent Systems. p. 1698–1700. AAMAS '21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2021)