Automatic Detection of Conflicts in Complex Narrative Structures

SZILAS, Nicolas, ESTUPINAN VESGA, Sergio, RICHLE, Urs Michael

Abstract

The central notion of conflict in drama is well-acknowledged but not properly formalized. Computational models of conflict tend to target one specific type of conflict and consequently lose the global point of view on the story. Using a model of dramatic structure, this article specifies a number of conflict types within a unified model and proposes an algorithm to automatically extract all conflicts within a narrative structure. The algorithm is then tested on a storyworld that shows as many as 31 coexisting conflicts. Finally, a cluster analysis on these conflicts is performed, showing that in the considered case, conflicts can be reduced to three main "conflict groups."

Reference

SZILAS, Nicolas, ESTUPINAN VESGA, Sergio, RICHLE, Urs Michael. Automatic Detection of Conflicts in Complex Narrative Structures. In: *11th International Conference on Interactive Digital Storytelling, ICIDS 2018*. Cham : Springer International Publishing, 2018.

DOI : 10.1007/978-3-030-04028-4_49

Automatic Detection of Conflicts in Complex Narrative Structures

Nicolas Szilas, Sergio Estupiñán and Urs Richle

TECFA, FPSE, University of Geneva, CH 1211 Genève 4, Switzerland
{Nicolas.Szilas, Sergio.Estupinan, Urs.Richle}@unige.ch

Abstract. The central notion of conflict in drama is well-acknowledged but not properly formalized. Computational models of conflict tend to target one specific type of conflict and consequently lose the global point of view on the story. Using a model of dramatic structure, this article specifies a number of conflict types within a unified model and proposes an algorithm to automatically extract all conflicts within a narrative structure. The algorithm is then tested on a storyworld that shows as many as 31 coexisting conflicts. Finally, a cluster analysis on these conflicts is performed, showing that in the considered case, conflicts can be reduced to three main "conflict groups."

Keywords: Interactive Narrative, Computational Models of Narrative, Conflict, Rules

1 Dramatic Conflict: A Proteiform Concept

In dramaturgy and screenwriting handbooks, *conflict*—or an analogous term—is often considered as the core of drama: "Any dramatic situation stems from a conflict between two main directions of effort"[1] ([14], p. 200), "Conflict is at the core of any dramatic work"[1] ([9], p. 32), "All drama is conflict" ([6], p. 24), "Conflict is the basis of drama" ([16], p. 125), "Conflict is the heartbeat of all writing" ([5], p. 178), "A story without a struggle can never be a dramatic story" ([23], p. 143), and so forth. However, what is meant behind the term *conflict* often varies, and despite the common usage of the term, no agreed definition has emerged. Conflict is not just an opposition or contrast of elements (e.g., John is rich and strong, while Mark is poor and weak), but it is related to core actions in the story (e.g., both John and Mark want to marry Elisa). Conflict is usually discussed as a phenomenon occurring at the story or fabula level, contrary to other narrative effects (such as surprise) that may occur at both story and discourse levels.

In addition to definitions, many different characterizations and classifications of conflict have been proposed. First, a distinction between external and internal conflicts is often acknowledged: the former type occurs between a character's goals and

[1] Our translation

resistance from the environment; the latter type occurs within a character and involves his or her internal needs (e.g., [12]). Second, within external conflicts, one can distinguish conflicts with the physical environment (further decomposed into obstacles and complications in [23]) from conflicts between characters (counter intention in [23], p. 146; conflict or opposition with an opponent in [22], p. 94). Third, conflict has been related to the moral values of characters and to the ethical dimension of the story as a whole [22]. The notion of conflict extends beyond drama and applies to narrative in general. For example, the six-actant model [7] contains a subject and an opponent. The paradox model proposed by Nichols [13] concerns narrative in general. The early myth model proposed by Levi Strauss is fundamentally based on a set of contradictions [10] (the conflict in this theory has not been attached to specific story elements).

A first observation following this overview is the wide variety of terminology that covers either different names of the conflict concept or specific cases of conflict—for example, *obstacle*, *complication*, *struggle*, *dilemma*, *paradox*, *opposition*, and *contradiction*. A second observation is the full heterogeneity of theories and classifications regarding conflict. Conflict thus appears as a proteiform concept that is not well-defined, raising the question of whether this concept should be used at all for formal and computational models of narrative and drama.

However, the notion of conflict is so central that it cannot be ignored, and several interactive digital storytelling systems have implemented various models of conflict. We have summarized the computational models of conflict in Table 1, matching each model with its narratological counterpart.

Table 1. Various types of conflicts and their computational models.

Conflict in narrative theories	Conflict in computational model
Internal moral conflict [12]	IDtension: goals, tasks, and values [18] Conflict of goals and values [2] Dilemma generation models [3, 8] Moral dilemmas [15]
Inner (nonmoral) conflict [12]	Conflict within one character's plans [24]
External conflict: obstacle [9, 23]	IDtension: obstacle [18] Plan failure [4]
External conflict: personal conflicts [12], counter intention [23], conflict with antagonist [5]， intercharacter conflict [22, 23]	Conflict between two characters' plans [24]
External conflict: social dilemma	GADIN: betrayal, sacrifice, greater good, take down, favor [1] Generation of dilemma [3]
Paradox [13]	Dramatic situations [19]

Each computational approach tends to focus on one aspect of conflict. For example, models of moral conflicts [2, 18] cover internal moral conflicts, not intercharacter conflicts. Conversely, Ware and Young's planning-based model of conflict considers conflicts between plans and therefore cannot account for conflicts between achievement goals and moral goals or values. GADIN [1] is one of the most complete models of conflict (as dilemma), but it covers only situations in which a social relationship is at stake (friends and enemies), putting aside internal conflicts. Conversely, the dilemma generation model [3] does not deal easily with intercharacter conflicts. The paradox-based model [19] focuses on internal conflict in general.

Ultimately, the variety of conflict-based computational models reflects the broadness and fuzziness of the original concept of conflict. To fully embrace the concept of conflict without reducing it to one specific expression, this article adopts an existing structural model of dramatic situation [21] and attempts to demonstrate that this model can accurately cover the variety of conflicts found in literature and computer implementations. In the next section, after the model's main principles are summarized, each type of conflict it can cover is systematically defined and characterized. In the process, the model is extended with new types of structural elements. In Section 3, the model is implemented, along with an algorithm that extracts existing conflicts from a given narrative structure. Section 4 provides a practical illustration of the algorithm on one story. The practical results of this experiment draw new lines of investigation regarding groups of conflicts to deal with the complexity that characterizes authentic storyworlds.

2 Conflicts and Their Modelling

2.1 A Unified Model of Conflict

We based this research on a model of a dramatic situation that was successfully used to manually analyze stories [21]. This model is based on goal-task structures that describe dramatic situations in terms of a relational network that consists of six types of *nodes*: goals, tasks, obstacles, side-effect, characters, and character sets. These elements are connected via different types of weighted and oriented *relations*. For example, the fact that a task enables a goal to be reached is represented by a *reaching* relation from the task to the goal, with weight "1." A given arrangement of nodes and relations creates a *situation*, formally described as a well-formed[2] and oriented graph. A situation may contain a *dramatic cycle*, which is a pattern that corresponds to conflicts. It is formally defined as follows:

- A dramatic cycle is a subgraph of a situation graph that forms a cycle, regardless of the relation's direction.
- Within a dramatic cycle, two nodes named the *start node* and the *end node* are such as there are two distinct paths from the start node to the end node, with strengths of opposite signs. The *strength* of a path is defined as the product of the weights of all relations it contains.

[2] Each relation needs to connect to nodes of the type specified for this type of relation.

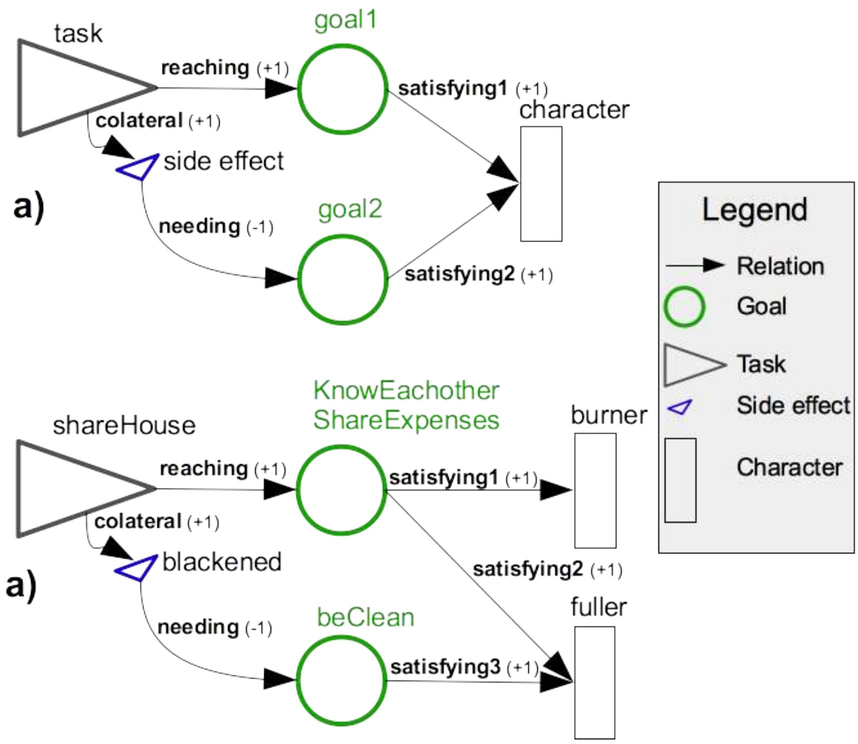

Fig. 1. An example of a dramatic cycle with the generic model (a) and a corresponding specific example (b), describing the Aesop fable "The Charcoal-Burner & the Fuller" (see text).

This formal characterization of conflict expresses the fact that an element in the story has contradictory consequences. Fig. 1a provides a generic example of a dramatic cycle with generic elements. In this example, a character may perform a task that, on the one hand, reaches his or her goal and, on the other hand, threatens another of his or her goals (via a side effect). Fig. 1b provides a specific example that instantiates the generic example. Many other specific examples can be found elsewhere in literature [20, 21]. A key feature of this model is that it does not specify the type of start and end nodes in the dramatic cycle.

From a global perspective, one can analyze the model according to three levels:

- The metamodel: it encompasses the notions of nodes and relations. At this level, the general notion of conflict is defined as a dramatic cycle.
- The narrative model: it specifies a given set of node types and relation types.
- The story model: given a certain narrative model, it describes one specific story or storyworld.

In this article, while keeping the metamodel unchanged, we introduce new types of relations, thereby extending the narrative model to cover more cases of conflicts.

2.2 Exercises in conflict type

Let us now consider different topologies of dramatic cycles that correspond to different types of conflicts.

The paradox: the end node is a goal. If, in a dramatic cycle, the two opposite paths converge toward a goal, it means that the start node (typically a task) both leads to the achievement of the goal and prevents this achievement. This situation is frequent in narratives and has been described in detail by Nichols [13].

The internal conflict: the end node is a character and connected via two satisfying relations. If two paths of opposite signs lead to the same character via two different goals, on one hand a beneficial goal is achieved but, on the other hand, another beneficial goal is impaired. This corresponds to an internal conflict [9]. If one goal is an achievement goal and the other is a moral goal, then we have a moral dilemma or conflict [2, 18].

The intercharacter (social) conflict: the end node is a set and connected via two belonging relations. In this case, a task in the structure satisfies one character but not another character. Because the two characters belong to the same set (one can always define a general set of all characters), a conflict arises at the level of the social group (the set). Depending on the type of belonging relation, several subtypes of conflicts may occur; for example, if both characters are friends or brothers, this constitutes a stronger conflict than if the characters are just unrelated human beings. The strength of a conflict may be modeled by changing the weight of the relation (this possibility is not explored further in this article). Depending on which character has the possibility to act on the structure, the conflict can be either a betrayal (the character can act positively for himself or herself but negatively for the other) or a sacrifice (the character can act negatively for himself or herself but positively for the other), according to the wording of the GADIN system [1].

The internalized social conflict: the end node is a character, and one incoming relation is a mattering relation. Formally speaking, when a social conflict (see above) arises, the protagonists themselves do not really care about the conflict. The social conflict is processed only at the global level of the story; it is understood by the viewer but not empathetically through the characters. In this article, we introduce a new relation in the narrative model—the "mattering" relation—from a set to a character. This means not only that the character belongs to a group but also that it matters to him or her. In that case, a harmful event to the group is harmful to the character as well.

The authoritative conflict: the end node is a character, and one incoming relation is a domination relation. The above belonging relation enables a friendship to be modeled as a reciprocal relation that corresponds to the internalized social conflict. But in other cases, the conflict is between a character's goal and another character's goal, and the latter character is dominant. For example, a character may do something that is good on a personal level but bad for his or her boss, which creates conflict. Therefore, we introduce in this article a new relation called the *domination* relation, from the dominant to the dominated, which enables a conflict to be modeled on the basis of personal interest and social obligation.

The above "exercises" show that the goal-task model is well-suited to modeling conflict in general, and they can be used to propose a general characterization that encompasses the variety of cases covered by the different and distinct computational models. Note that the model describes conflicts at the syntactic level and does not suggest any universal types of goals or values for describing conflicts (see [8] for an alternative approach). Conflicts are considered as specific to a given story, and their specification is left to the author, including which values are at stake in a moral conflict.

3 Implementation

3.1 Structures and algorithms

The goal-task model for describing different types of conflict has been implemented as a set of Java classes to describe the various components of the model, as shown in the UML class diagram in Figure 2. A structure is composed of nodes and relations abstracted as elements. A path is composed of elements, and two classes are added for conflict calculation based on the paths Impact and Conflict, as shown below. Not described in this paper, this object-oriented model is part of a larger system aimed at the dynamic generation of stories for interactive digital storytelling.

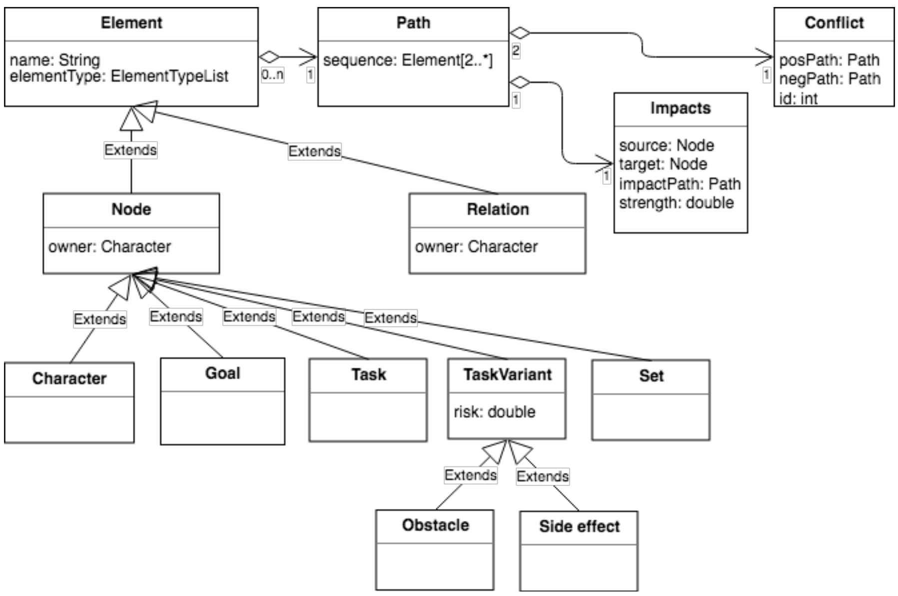

Fig. 2. UML class diagram of the narrative model. The lower left part describes the elements in the structures, while the three other classes are used to reason about conflicts (see text).

In addition to the model, an algorithm for detecting conflicts (dramatic cycles) in the structure has been implemented in a rule engine—namely, Drools. Conflicts are detected via three rules. The first two rules calculate the impact, if any, between each pair of nodes in the structure. There is an impact from node *A* to node *B* if there exists a path from *A* to *B*.

The first rule initializes the impact by stating that if node *A* is connected to node *B* via relation *r*, then *A* impacts *B*:

```
rule "impacts initialization"
when
       $A: Node()
       $B: Node()
       $r: Relation( source==$A , target==$B )
       not Impacts( source==$A, target==$B )
   then
       insert( new Impacts( $A , $r , $B , $r.getWeight() ) );
end
```

The first part of the rule, the left-hand side, defines the triggering condition. In the above case, it means that if there exists a node (assigned to $A), if there exists a node (assigned to $B), if there exists a relation between these nodes (assigned to $r), and if there exist no impacts between these two nodes, then one should insert an impact between these nodes.

The second rule propagates the impact by applying a transitivity rule: if *A* impacts *B* and *B* impacts *C*, then *A* impacts *C*:

```
rule "impacts propagation"
   when
       $A: Node()
       $B: Node()
       $C: Node()
       $impAB: Impacts( source==$A , target==$B , $strengthAB:strength ,
                 $pathAB:ImpactPath )
       $impBC: Impacts( source==$B , target==$C , $strengthBC:strength ,
                 $pathBC:ImpactPath , $pathBC.disjoint($pathAB )
       not Impacts( source == $A , target == $nB , ImpactPath.identical( new
                 Path( $pathAB , $pathBC ) ) )
   then
       Path $pathAC = new Path( $imp12.getImpactPath() , $imp23.getImpactPath() );
       Insert( new Impacts( $A , $C , $path , $impAB.getStrength() *
                 $impBC.getStrength() ) );
end
```

An additional condition was added to the transitivity: the second path should not contain an element in the first one, except the one that connects them. It prevents obtaining paths that repeat some edges (in graph theory terms, the positive and negative paths in a dramatic cycle are *trails*).

Finally, the third rule calculates the conflicts by identifying two nodes where there are two ways in which the first one impacts the second one, and these two paths are of opposite strength.

```
rule "conflict detection"
   when
       $source: Node()
       $target: Node()
       $posImpact: Impacts( source==$source , target==$target ,
                 strength > 0 , $posPath:impactPath)

       $negImpact: Impacts( source==$source , target==$target ,
                 strength < 0 , !$path2.crosses($posPath) )
   Then
       Insert( new Conflict( $posImpact.getPath(), $negImpact.getPath() );
end
```

These three rules implement the definition of conflict in Section 2.1 in a compact, human-readable way.

3.2 The storyworld

In the context of this article, a storyworld denotes a set of instantiated nodes and relations that constitute a specific story structure. It is not a story but it may produce many different stories, depending on the execution, though this process is not considered in this article. We decided to start from a storyworld already available, one designed for a narrative engine under development. From a methodological point of view, the key point in this study is that this storyworld has a size sufficient to exhibit an authentic application of the conflict-detection algorithm, in contrast to a "toy problem," for which a small number of solutions are already known.

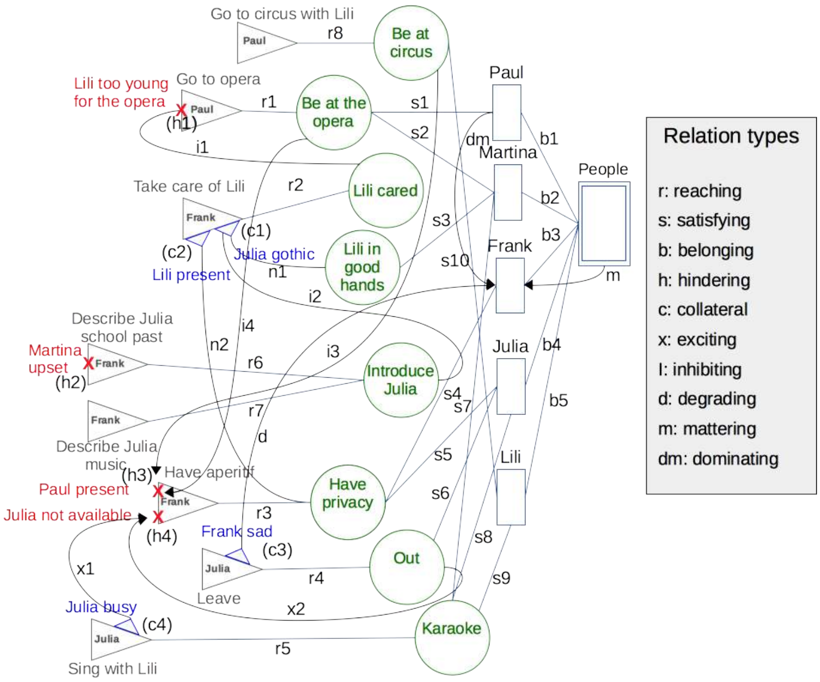

Fig. 3. Graphical representation of the narrative structure used for experimentation (see Fig. 1 for legend; red crosses are obstacles, and double rectangles are character sets). When an obstacle is placed on a task, it represents, in a compact way, a hindering link from the obstacle to the task—the relation is mentioned in parenthesis. Similarly, side effects (blue triangles) are represented directly on the task, and the collateral relation name is mentioned in parentheses.

The storyworld was written in collaboration with a professional author; it is depicted in Figure 3 and may be summarized as follows:

Frank has just moved into his new apartment. He has invited Julia for the evening. Julia is a gothic girl, and she sings in a gothic band. Frank is in love with her, but Julia does not know it yet. He has the plan to declare his love to her this evening, because the day after, Julia is leaving for three months for a concert tour. But this same evening, Frank's parents, Paul and Martina, have got tickets for the opera. They come to

Frank's apartment and bring his little sister, Lili. They want him to take care of Lili during the opera, which spoils the romantic meeting with Julia. In addition, Frank's mother has a problem with Julia's style and would not leave her daughter with her during the evening. Frank may therefore want to introduce Julia better to her mother. The complicated situation may lead Julia to decide to leave the apartment. But Lili could also want to play karaoke with her or go to the circus with her father.

This storyworld contains 30 nodes and 42 relations. It is obviously complex to read in Fig. 3, but we estimate that its execution via a proper narrative engine would enable approximately 15–20 minutes of gameplay, which would be comparable with *Façade* [11] or *Nothing for Dinner* [25]. When this scenario was designed, three dramatic cycles were intentionally introduced:

- On one hand, if Frank takes care of Lili, he will please his father; on the other hand, he will not have privacy with Julia. (C1)
- On one hand, if Frank takes care of Lili, Martina will be happy to go to the opera, but she is unsatisfied with the idea of leaving her daughter with Julia. (C2)
- On one hand, Julia is happy to sing karaoke with Lili, but on the other hand, this will prevent her from having privacy with Frank. (C3)

Nothing was known regarding the number of additional dramatic cycles that may be present in the structure.

3.3 Results

The execution of the above algorithm on the structure depicted in Fig. 3 generated 31 conflicts. Among these 31 conflicts, we found 16 intercharacter conflicts, 7 internalized social conflicts, 1 authoritative conflict, 3 both internalized social and authoritative conflicts, 3 internal conflicts, and 1 paradox. For each example, we manually produced a text description of the conflict. We observed that each of the 31 conflicts made sense, though some of them might seem slightly far-fetched. Table 2 shows one example of each conflict category, with a hand-written, plain-text explanation of the conflict.

Table 2. Extract of conflicts in the example scenario, with plain-text descriptions.

Type	Algorithm's output	Plain-text description
Paradox	`[takeCareLili, r2, liliCared, i1, liliTooYoung, h, goToOpera, r1, opera, i4, paulPresent, h3, haveAperitif, r3, havePrivacy]` **vs** `[takeCareLili, c2, liliPresent, n2, havePrivacy]`	On the one hand, if Frank takes care of Lili, his father will go to the opera and not prevent Frank and Julia from having privacy; on the other hand, Lili will be around, and this will prevent them from having privacy.
Internal	`[out, s6, Julia]` **vs** `[out, x2, juliaNotAvailable, h4, haveAperitif, r3, havePrivacy, s5, Julia]`	Julia is satisfied to be out, but this prevents her from having an aperitif with her friend.

Interchara cter	[out, s6, Julia, b4, people] **vs** [out, x2, juliaNotAvailable, h4, haveAperitif, r3, havePrivacy, s4, Frank, b3, people]	For Julia, being out is good, but for Frank, it is not, because it prevents them from having an aperitif together.
Internalized social	[out, s6, Julia, b4, people, m, Frank] **vs** [out, x2, juliaNotAvailable, h4, haveAperitif, r3, havePrivacy, s4, Frank]	On the one hand, Frank understands that being out is good for Julia; on the other hand, it prevents them from having an aperitif together.
Authoritative	[takeCareLili, r2, liliCared, i1, liliTooYoung, h, goToOpera, r1, opera, s1, Paul, dm1, Frank] **vs** [takeCareLili, c2, liliPresent, n2, havePrivacy, s4, Frank]	On the one hand, if Frank takes care of Lili, his father will be satisfied to go to the opera, so Frank feels obliged to do that. On the other hand, this would prevent him from having a private aperitif with Julia, because they will have to take care of his sister, Lili.

These results show that the number of conflicts formally found in the storyworld is much higher than the number of conflicts initially conceived by the creators of the story (see the three conflicts in Section 3.2). This main finding is discussed in the next section. In addition, the results show a high discrepancy in the number of conflicts in each category, and the intercharacter conflict is the most represented.

3.4 Discussion

The formalization of conflict as dramatic cycles has shown that a storyworld, even not a very elaborate one such as that represented in Fig. 3, exhibits many conflicts. Each of these conflicts makes sense in the narrative context, and most of them were discovered by the automatic analyses, as they were not designed a priori. This contrasts with the classical view in dramaturgy (the current inspiration of most screenwriting practices), which considers one or a few central dramatic situations that drive the plot [9, 14, 17]. According to the present study, drama is rather made of a constellation of specific and interwoven conflicting situations, which results in one or a few emerging global conflicts. This sheds a new light on the nature of drama that needs to be understood in terms of a complex system of conflicts.

More precisely, one individual conflict is often derived into several other related conflicts. For example, take the second conflict in Table 2, in which Julia is conflicted because she wants not only to leave because of Martina's attitude but also to have a drink with her friend. This is related to the third conflict in Table 2, in which Frank and Julia are conflicted about this same action. If one considers the authored conflict presented above (C1) and the 31 conflicts, several appear related to it; for example, there is a similar conflict with Julia (she, too, wants privacy), the internalization of the conflict within Frank (via the mattering relation), and the similar conflict motivated by the father's authority.

4 Further Analyses of the Results

These observations suggest that behind the diversity of conflicts, a few of the main conflicts may be extracted in order to conciliate with the general idea that a story should be built around a few main core ideas. Therefore, we statistically processed the 31 found conflicts to determine if it was possible to group them automatically into a small number of meaningful clusters.

To numerically process the conflicts, the first step was to define a distance measurement between two conflicts. This distance was defined as the average of the distance between two positive paths and two negative paths:

$$dist(C_1, C_2) = \frac{1}{2}[\Delta(C_1.posPath, C_2.posPath) + \Delta(C_1.negPath, C_2.negPath)]$$

Next, the distance Δ between two paths was calculated as the distance between the two sets of elements in each path, which could be simply calculated as the size of the symmetric difference[3] between the two sets.

Once the distance measurement was defined, we could establish a similarity matrix that gathered the distances of each pair of conflicts. Next, we applied the gap statistic method with the k-mean clustering algorithm to obtain automatically the optimal number of clusters. The 31 conflicts could be grouped into three clusters. Finally, we applied the partition around medoids (PAM) algorithm to our dataset, producing the three clusters represented in Fig. 4: the first one, in red, gathered 14 conflicts around Frank having privacy with Julia; the second one, in green, concerned Martina and her issue with Julia's style (7 conflicts); the third, in blue, one concerned consequences of taking care of Lili (10 conflicts).

In Fig. 4, one can see that the clusters group similar conflicts, meaning that the storyworld displayed in Fig. 3 can be expressed by taking only a few of its 31 conflicts—for example, one per cluster. In addition, one may notice that some conflicts are very similar (some dots are almost superimposed in Fig. 4), while others in the same cluster are different (e.g., see cluster 20 in blue).

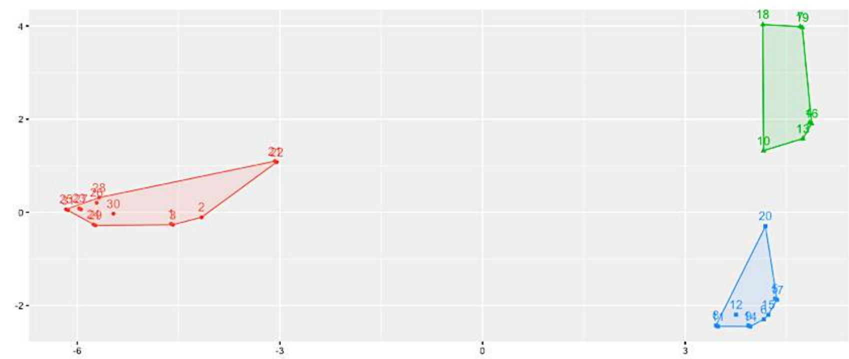

Fig. 4. Cluster plot with three clusters grouping the 21 detected conflicts.

[3] The symmetric difference between two sets is the union of the two sets, without the intersection.

5 Conclusion and Future Applications

Conflict is an umbrella term but fundamental to understanding the story dynamics in drama. In this article, based on a general computational model of conflict, a theoretical categorization of conflict types was proposed. Next, this approach was empirically tested on a formally described storyworld of average complexity. The main result was that there was not one or two conflicts in a story but a constellation of conflicts; this result shows the complexity behind any nonelementary story. By applying a statistical analysis of extracted conflicts, we could also extract a small number of clusters, showing that behind the diversity of conflicts, main topics could be extracted.

In addition to the theoretical contribution of this work, there are several ways the proposed algorithm could support story generation. First, in case the narrative engine requires the author to enter the conflicts by hand, the automatic extraction of conflicts could help the author to define which ones should be expressed during the dynamic unfolding of the story. The clustering might help the author to pick conflicts that are different from each other. Second, a more sophisticated narrative engine could drive the story according to all conflicts present in a structure, not only the ones identified by the author. Given the number of conflicts, this approach may greatly increase the variability of the generated stories. Finally, it would be possible to use natural-language generation techniques to produce a text description of conflicts, which could then feed the generated story itself, with a character or a narrator explicitly referring to a dramatic conflict.

Acknowledgements

This research has been supported by the Swiss National Science Foundation under grant No. 159605 (Fine-grained Evaluation of the Interactive Narrative Experience).

References

1. Barber, H., Kudenko, D.: Dynamic generation of dilemma-based interactive narratives. Proc. Third Conf. on Artificial Intelligence and Interactive Digital Entertainment – AIIDE. pp. 2–7 AAAI Press, Menlo Park, CA (2007).
2. Battaglino, C., Damiano, R., Torino, U.: A Character Model with Moral Emotions : Preliminary Evaluation. In: Finlayson, M.A., Meister, J.C., and Bruneau, E.G. (eds.) 5th Workshop on Computational Models of Narrative (CMN'14). pp. 24–41 OASICS (2014).
3. Benabbou, A., Lourdeaux, D., Lenne, D.: Dynamic generation of dilemmas in virtual learning environments for non-technical skills training. In: Wang, Y., Howard, N., Widrow, B., Plataniotis, Y., and Zadeh, L.A. (eds.) IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC). pp. 231–234 IEEE (2016).
4. Cavazza, M., Charles, F., Mead, S.J.: Generation of Humorous Situations in

Cartoons through Plan-based Formalisations. In: Nijholt, A., Stock, O., Dix, A., and Morkes, J. (eds.) ACM CHI-2003 Workshop: Humor Modeling in the Interface. University of Twente, Twente, The Netherlands (2003).
5. Egri, L.: The art of dramatic writing. Simon & Shuster, New York (1946).
6. Field, S.: Screenplay – The Foundations of Screenwriting. Dell Publishing, New York (1984).
7. Greimas, A.J.: Sémantique structurale. Presses universitaires de France, Paris (1966).
8. Harmon, S.: An Expressive Dilemma Generation Model for Players and Artificial Agents. The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16). pp. 176–182 (2016).
9. Lavandier, Y.: La dramaturgie. Le clown et l'enfant, Cergy, France (1997).
10. Lévi-Strauss, C.: Anthropologie Structurale. Plon, Paris (1958).
11. Mateas, M., Stern, A.: Façade, http://games.softpedia.com/get/Freeware-Games/Facade.shtml.
12. McKee, R.: Story: Substance, Structure, Style, and the Principles of Screenwriting. Harper Collins, New York (1997).
13. Nichols, B.: Ideology and the image. Indiana University Press, Bloomington, IN (1981).
14. Polti, G.: Les trente-six situations dramatiques. Mercure de France, Paris (1903).
15. Saillenfest, A.: Modélisation Cognitive de la Pertinence Narrative en vue de l'Évaluation et de la Génération de Récits. Télécom ParisTech (2015).
16. Seger, L.: Making a Good Script Great. Samuel French Trade, Hollywood, CA (1989).
17. Souriau, E.: Les deux cent mille Situations dramatiques. Flammarion, Paris (1950).
18. Szilas, N.: A Computational Model of an Intelligent Narrator for Interactive Narratives. Appl. Artif. Intell. 21, 8, 753–801 (2007).
19. Szilas, N.: Modeling and representing dramatic situations as paradoxical structures. Digit. Scholarsh. Humanit. 32, 2, (2017).
20. Szilas, N.: Structural Analysis of the Aesop's Fables, http://tecfalabs.unige.ch/mediawiki-narrative/index.php/Aesop's_Fables.
21. Szilas, N., Estupiñán, S., Richle, U.: Qualifying and quantifying interestingness in dramatic situations. In: Nack, F. and Gordon, A.S. (eds.) 9th International Conference on International Digital Storytelling (ICIDS 2016). pp. 336–347 Springer, Cham (2016).
22. Truby, J.: Anatomy of story. Faber and Faber, New York, NY (2007).
23. Vale, E.: The technique of screenplay writing. Grosset & Dunlap, New York (1973).
24. Ware, S.G., Young, R.M.: CPOCL: A Narrative Planner Supporting Conflict. Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. pp. 97–102 AAAI Press, Palo Alto, CA (2011).
25. Nothing For Dinner, http://nothingfordinner.org.