Intelligent Narrative Technologies and Social Believability in Games:
Papers from the AIIDE 2015 Joint Workshop

Merits of a Temporal Modal Logic for Narrative Discourse Generation

Markus Eger, Camille Barot, R. Michael Young
Liquid Narrative Group, Department of Computer Science
North Carolina State University, Raleigh, NC 27695
meger@ncsu.edu, cbarot@ncsu.edu, young@csc.ncsu.edu

Abstract

Just as there exists varied uses for computational models of narrative, there exists a wide variety of languages aimed at representing stories. A number of them have historic roots in automated generation, for which these languages have to be limited in order to make the generation process computationally feasible. Other are focused on story understanding, with close ties to natural language making many reasoning processes computationally intractable. In this paper, we discuss the trade-off between expressivity and computational complexity of the reasoning process and argue that Impulse, a temporal, modal logic provides more expressivity than languages historically associated with story generation, while still affording reasoning capabilities. We show that these properties enable certain aspects of narrative discourse generation by using two examples from different genres, and claim that this generalizes to a broader class of problems.

Introduction

The wide range of past and current research on computational generation and analysis of narratives has resulted in the development of a variety of languages to represent them. It is critical to understand the strengths and limitations of these languages to discuss their applicability to specific tasks. In this paper, we provide a foundation for this discussion by investigating one particular trade-off that has to be considered when choosing a language: expressivity vs. computational cost. To discuss this trade-off in the context of narrative, we make use of the bipartite model proposed by various narratologists (e.g., (Chatman 1980)), which distinguishes between story and discourse. In this model, the story consists of the world in which the narrative happens, along with all its characters, objects, and actions performed. The discourse describes how the narrative is told to an audience, specifying which elements of the story are conveyed, in which order they are conveyed and what medium-specific resources are used to convey them. The same story can be used to create multiple discourses: Movies, texts, or even interactive experiences can convey the story from the point of view of a particular character — what Genette refers to as *focalization* (Genette 1983). The task of creating narrative discourse from an existing story in an automated way is what we call "narrative discourse generation".

This paper presents a discussion of how the choice of a language for story representation is a trade-off between expressivity and computational cost of reasoning about it, and how this affects the task of narrative discourse generation in particular. We argue that Impulse (Eger, Barot, and Young 2015), a story representation language based on a temporal, modal logic, addresses this trade-off and affords reasoning capabilities about stories that extend the range of narrative discourses that can be created from other story representations.

Related Work

Due to the multitude of narrative generation and processing tasks, a wide variety of story representations have been developed. When approached from the vantage point of story generation, the output or internal data structures of generation algorithms are often used as the representation of the story itself. For example, automated planning techniques have been used in a variety of ways to generate stories, and the representation such systems use are that of their underlying planning algorithms. For example, Riedl and Young (2010) proposed a story generation system based on a modified partial-order causal link (POCL) planning algorithm called IPOCL, and the stories generated by this algorithm are represented as IPOCL plans. Logic-based languages are similar to plan-based languages in that they allow a formal representation of how the story world works. Examples range from linear logic (Martens et al. 2014) to modal logic (Barbosa et al. 2014). Depending of the kind of logic that they are built from, these languages afford more or less reasoning capabilities on the generated stories.

Other languages originate not from AI or logic-grounded generation approaches, but from a need to describe existing narratives. This is the case with Story Intention Graphs (SIGs) (Elson and McKeown 2007) and their implementation in the Scheherazade system. This system is built to facilitate the encoding of textual stories in specialized graph structures. The contents of the nodes of this graph structure are drawn from WordNet and VerbNet, and they are ordered temporally with respect to one another, reflecting story time. These nodes can be associated to specific constructs representing character goals and beliefs, along with edges that specify which action causes, fulfills or prevents such a goal or belief.

Once a story is encoded in one of these languages, a computational process can reason about it to generate narrative discourse. For example, Rishes *et al.* (2013) use SIGs to generate different versions of the same story which vary in style, e.g., by telling the story in a more "formal" or "laid-back" way. Similarly, Hoek *et al.* (2014) transform story graphs generated by a multi-agent system to create focalized tellings, i.e. a retelling of the story from the point of view of a particular character. Montfort describes how his interactive fiction system uses focalization to e.g., inform the player when their character refuses to perform a particular action versus when other characters do so (Montfort 2011). Jhala and Young (2010) use stories represented as POCL plans as input to a camera planning system called Darshak to generate movie output for the story. Darshak itself also uses a planner to determine which shot types to use, and uses the story and authorial goals to create the movie.

Motivation

While many languages exist for representing stories, they are typically either highly expressive, with associated high computational cost when reasoning about stories encoded in them, or provide significantly more efficient inference procedures with an associated reduction in expressive power. We will first describe what we mean by expressivity and by computational complexity of the reasoning process, and then discuss how this trade-off affects different languages. We will illustrate this point with two of the formalisms described in the previous section: SIGs (Elson and McKeown 2007) and IPOCL (Riedl and Young 2010).

Expressivity

A story consists of a multitude of features: Characters, objects, locations, static properties and actions are the most tangible ones, but there are also more abstract concepts like the mental states of the characters at different times, which describe their emotions and goals. For our purpose, we use the word expressivity to describe the abstract measure for how many of these features a language can represent, and in what detail. Of course, this is not a linear metric, since one language may support the representation of character's desires, but not their beliefs, and another vice versa, so none of these languages would be "more expressive" than the other. We claim, however, that it is still a relevant measure on a less granular scale. For example, languages based on natural language are very expressive, since natural language can express complex character mental states and timing relations in addition to the sequence of actions that happens in a story. That is not to say that they are without limitation, though. SIGs, for example, only directly support goals and beliefs of characters as first-class objects for mental models, anything else has to be annotated in natural language content. The expressivity of natural language, however, also comes with a lack of formal semantics, which can result in ambiguity. This is one of the reasons why other languages are more restrictive. Story-centric planning languages, for example, are typically limited to instantaneous actions, without any representation for characters' mental models. IPOCL extends a classical planning formalism with a notion of intentionality, but is not able to represent characters' beliefs or desires, which would be required to actually reason about their intentions.

Computational Complexity of Reasoning

Most tasks that process a story encoded in a language for story representation will need to reason about the story. For example, to summarize this story, a process would need to reason about what parts can be left out because they can be inferred. To explain characters' choices in a story a process would need to reason about possible alternatives that were available to the characters. Some languages are kept deliberately simple to keep that reasoning process (relatively) cheap computationally. Planning, for example, is PSPACE-complete in the worst case, but most problems are actually in NP or even easier in practice (Bäckström and Jonsson 2011), making them quite applicable to real-world problems. Reasoning about the plan itself is also generally easier than generating it in the first place, making these languages even cheaper in terms of computational cost of the reasoning process. On the other hand, languages that utilize natural language are very hard to reason about and can only be reasoned about in limited cases in practice.

The Trade-off

Ideally, a story-representation language would be both very expressive and computationally tractable. However, these two properties are in direct opposition to one another. As features are added to a language, reasoning about it becomes more complex, and therefore also more computationally expensive. In fact, the very reason why less expressive languages are used is because reasoning about very expressive languages is intractable. Referring back to the languages discussed above, SIGs are very expressive, but make it hard to reason about represented content because of their reliance on natural language to provide annotations to describe a range of narrative features. On the other hand, planning languages are limited in their expressivity but reasoning about IPOCL plans is tractable in practice. We claim that there is a need for languages that are more expressive than planning- and related languages but easier to reason about than natural language-based languages, particularly for the task of narrative discourse generation.

Impulse

Impulse (Eger, Barot, and Young 2015) is a story representation language based on a representation of actions and events in Interval Temporal Logic (ITL) (Allen and Ferguson 1994), which it combines with a model of characters' beliefs, desires and intentions (BDI) (Cohen and Levesque 1990). ITL provides the means to encode complex temporal relations between when actions happen, when predicates hold and when characters hold their beliefs, desires and intentions. The BDI model, on the other hand, allows reasoning about several aspects of the characters' mental models.

A story encoded in Impulse consists of 6 parts:

- A time basis T, which is a set of intervals
- An object type hierarchy, which consists of subset definitions of a root set O that contains all objects
- An action type hierarchy, which consists of subset definitions of a root set $Actions$, that contains all actions
- A set of action properties P, which is a set of functions, mapping from an action to either an object or an interval
- A set of action instances Σ that make up the story
- A set of Impulse sentences Ψ that encode additional information about the story

The object type hierarchy always has to define the two subsets $A \subseteq O$, which contains all actors, and $L \subseteq O$, which contains all locations. Other subsets can be defined as needed for a given story. For example, a medieval setting may have a subset $Weapons \subseteq O$, with other subsets $Swords \subseteq Weapons$ and $Bows \subseteq Weapons$. Each of these sets then contains the actual objects, which are constants in the predicate logic sense. Similarly, the action type hierarchy defines how actions that can happen in the story world are related to one another. For example, the medieval setting could have an action type $attack \subseteq Actions$, with subtypes $shootBow \subseteq attack$ and $swingSword \subseteq attack$. This hierarchy is used to define the action properties. In our example, all $attack$ actions might have a common property $\mathrm{target} : attack \mapsto A$, but only the $shootBow$ action has a property $\mathrm{projectile} : shootBow \mapsto Arrows$. The set Σ contains all actions that actually happen in the story, and is used to define the values of the action properties for those. We could, for example, have an action $s_1 \in shootBow$, with $\mathrm{projectile}(s_1) = bodkin$. Finally, the set Ψ contains Impulse sentences, that is ITL with a BDI model, that describe how the story world operates. These sentences are typically in one of two forms: *Facts* are conjunctions of ground atomic literals that describe what is true, or false, in the world, while *rules* are implications where both the antecedent and the consequent are conjunctions of ground atomic literals. Rules describe how the world changes in response to what happens in it. As a special case of rules, *action effect definitions* describe what happens when an action is performed in a particular circumstance. For example, the $shootBow$ action will have different effects when the actor has arrows versus when they do not. These could be written as:

$$\begin{gathered}\forall s \in shootBow\ \mathrm{occurs}(s) \land \\ \mathrm{has}(\mathrm{actor}(s), \mathrm{projectile}(s), \mathrm{pre1}(s)) \\ \rightarrow \\ \mathrm{shotfired}(\mathrm{location}(s), \mathrm{eff1}(s))\end{gathered}$$

$$\begin{gathered}\forall s \in shootBow\ \mathrm{occurs}(s) \land \\ \neg\, \mathrm{has}(\mathrm{actor}(s), \mathrm{projectile}(s), \mathrm{pre1}(s)) \\ \rightarrow \\ \mathrm{confused}(\mathrm{actor}(s), \mathrm{eff2}(s))\end{gathered}$$

This means that if the $shootBow$-action occurs, and the actor has the projectile that should be used for the shot, a shot will be fired. However, if the action occurs when the actor does not have the projectile, the second sentence applies and the result is that the actor becomes confused. Note that the functions $\mathrm{pre1}, \mathrm{eff1}$ and $\mathrm{eff2}$ refer to times that a particular precondition or effect of the action holds, which depend on the instantiation of the action.

Impulse's use of Interval Temporal Logic allows for the representation of complex timing relations between predicates and actions using intervals, while the use of a BDI model allows for the representation of characters' beliefs, desires and intentions. We claim that this makes Impulse very expressive, yet that its formal semantics can be used to reason about the stories it is used to describe. As for its computational cost, Impulse is an extension of predicate logic, which is generally undecidable in theory (Turing 1936), (Church 1936). Yet in practice predicate logic can be reasoned on, as is evidenced by the fact that multiple theorem provers have been implemented (Fitting 2012), and we claim that the same holds for Impulse. Levesque also argues that the reasoning that is actually done by humans is tractable in logic (Levesque 1988). Impulse thus holds a middle ground between planning- and natural language-based languages, since reasoning about it is possible in practice, even if computationally expensive in the worst case.

Narrative Discourse Generation

The importance of the trade-off between expressivity and computational complexity of reasoning is particularly salient for the task of narrative discourse generation. We define this task as generating a discourse with particular authorial goals for a given story, that is encoded in a language for story representation. The authorial goals define how the author wants to convey the story to the audience, for example by defining what the audience should know at which point in time, or the perspective from which the story is told. We will present two examples from very different genres, fables and murder mysteries, and compare the suitability of different languages for the task. Again, we will refer to SIGs as a representative for languages that encode a story utilizing natural language databases like WordNet, and to IPOCL as a representative for languages that only have basic support for actions, objects and ordering. We suggest, however, that our point also holds for other languages in these two classes.

The Fox and the Crow

The Fox and the Crow is one of Aesop's fables, where a cunning Fox tricks a gullible Crow into singing so that a piece of cheese that she had in her beak would fall out and the Fox could get it. The full text of the fable, as cited by Elson (2012), is given in Figure 1. As previously mentioned, the perspective of the character used for the telling of the story is one of the properties of narrative discourse, called focalization. An example of the same story, as told from the point of view of the Crow, is presented in Figure 2. Generating this telling in an automated way poses the challenge of accurately representing the beliefs and thought process of the Crow. Most planning-based languages, such as Riedl and Young's IPOCL, do not formally distinguish characters beliefs from other world states. Story Intention Graphs are better suited for this task, but since they do not provide a way to represent action outcomes in a general way, the reasoning of the Crow at the end, that if she had not cawed, she would still have her cheese, would require reasoning about the natural language contents of the nodes if it was to be done with SIGs[1]. Impulse, on the other hand can represent the Crow's beliefs over time, as well as reason about the effects of an action that did not happen. One action that happens in the story is the cawing of the Crow, which is encoded as an action type $caw \subseteq Actions$. The effect this action has is encoded using these Impulse sentences:

$$\forall s \in caw \ \mathrm{occurs}(s) \\ \rightarrow \\ \mathrm{openBeak}(\mathrm{actor}(s), \mathrm{eff1}(s)) \wedge \mathrm{sound}(\mathrm{eff2}(s))$$

$$\forall t, t_1 \in T \ \forall a \in A \ \forall i \in Items \\ \mathrm{openBeak}(a, t) \wedge \mathrm{has}(a, i, t) \wedge t : t_1 \\ \rightarrow \\ \neg \, \mathrm{has}(a, i, t_1)$$

$$\forall s \in caw \ B_{Crow}(t) \, (\mathrm{occurs}(s) \wedge \mathrm{actor}(s) = Crow \\ \rightarrow \\ B_{\mathrm{listener}(s)} \, \mathrm{sweetvoice}(Crow, \mathrm{eff1}(s)))$$

This means that, if the caw-action occurs, the actor's beak will be open, and a sound is heard. The second sentence states that if someone's beak is open, and they have some item, they no longer have that item (because it is dropped)[2]. Finally, the third sentence states that the Crow believes that, if she caws, any listener will believe that she has a sweet voice. This third sentence is a direct cause of the cawing of the Crow, and should therefore be mentioned when the story is told from the point of view of the Crow. The first two sentences, on the other hand, describe why the cheese was dropped. By following the reasoning chain backwards from when the cheese drops to the fact that the caw action occurred in the story we can determine that the cawing was the cause of the cheese dropping, and by doing forward reasoning on an alternate world, where it did not happen, we can generate the last assertion of the Crow "Had I not cawed, I would still have my cheese!"[3].

[1]However, once we have this focalized telling it is possible to encode it in a SIG, including the Crow's reasoning about what would have happened had she not cawed.

[2]Splitting this up into two sentences allows us to provide other actions that make an actor open their beak, with the same side effect of an item dropping "automatically."

[3]For additional plausibility, this whole reasoning could be done within the beliefs of the Crow, i.e. "because the cheese dropped, she now believes that cawing results in her opening her beak, and that results in the cheese dropping, so now the Crow beliefs that had she not cawed, she would still have the cheese"

A Crow was sitting on a branch of a tree with a piece of cheese in her beak when a Fox observed her and set his wits to work to discover some way of getting the cheese.

Coming and standing under the tree he looked up and said, "What a noble bird I see above me! Her beauty is without equal, the hue of her plumage exquisite. If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds."

The Crow was hugely flattered by this, and just to show the Fox that she could sing she gave a loud caw.

Down came the cheese,of course, and the Fox, snatching it up, said, "You have a voice, madam, I see: what you want is wits."

Figure 1: "The Fox and the Crow," as cited by (Elson 2012)

I was sitting on a branch of a tree with a piece of cheese in my beak.
A fox came and stood under the tree, and looked up to me and said "What a noble bird I see above me! Her beauty is without equal, the hue of her plumage exquisite. If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds."

I was hugely flattered by this, but also taken aback that the Fox did not believe just how sweet my voice was. I knew that if he heard me, I could convince him otherwise, so I gave a loud caw. What I forgot was that the cheese was still in my beak and that it would fall when I gave the loud caw.

When I gave a loud caw, the cheese dropped from my beak, and the Fox, snatching it up, said, "You have a voice, madam, I see: what you want is wits." Had I not cawed, I would still have my cheese!

Figure 2: Manually authored version of "The Fox and the Crow" retold from the point of view of the Crow

Murder mystery

Our second example is a murder mystery story. Todorov argued that such a story actually consists of two stories: one about the crime and one about the investigation (Todorov 1977). However, a telling of the story in the style called *whodunit* omits details of the murder that would identify the culprit and focuses on describing the investigation process. The murderer is then only revealed at the end of the story, when the detective finally catches them. When presenting a murder mystery in this style it is therefore essential to be able to reason about what the detective believes at which point in time, to be able to convey the same information to the audience as is conveyed to the detective as the narrative unfolds. At the end of the story, though, the detective and the audience should have sufficient information to make it plausible beyond reasonable doubt that the person the detective caught is indeed the murderer, and other suspects are innocent. The process of generating the investigation story therefore consists of conveying information to the detective so that they can identify the murderer, which can be viewed as an in-world narrative discourse generation problem about the murder story.

To illustrate this point, we will now present a very simple murder mystery story. The story revolves around $Victor$, who is murdered in his mansion by $Mario$. The detective $Diana$ is investigating the case who finds $Isabel$ in the garden shed of the mansion and suspects her of the murder. The resolution of the case comes with $Diana$'s realization that it was raining at the time of and after the murder, but there were no footprints in the muddy garden, so $Isabel$ could not possible have ended up in the garden shed after the murder. The investigation could then proceed (and be conveyed to the audience) in the following order:

1. $Victor$ is murdered, but it is not told by whom.
2. $Diana$ starts investigating in and around the house.
3. $Diana$ finds $Isabel$ in the garden shed.
4. $Diana$ realizes that it is raining, and that walking through the garden would have left footprints.
5. $Diana$ concludes that $Isabel$ is innocent.
6. $Diana$ finds footprints in another part of the grounds, which lead her to find the real culprit, $Mario$.

The two key realizations in step 4 are the start time of the rain relative to the murder and what effect walking through the garden during the rain would have had. Generating this from an encoding of the murder part of the story therefore hinges on being able to represent and reason about these two things. Additionally, at each step it is necessary to be able to know what the beliefs of the detective (and therefore also the audience) are, because that is what drives the investigation process.

To encode this as an IPOCL plan, one could introduce a step that represents "the rain is starting" that is ordered before the murder step, which has an effect of "the garden is muddy". Walking from the mansion to the garden shed then has a conditional effect that, if the garden is muddy, footprints will be left behind. A system can then reason about which ordering constraints are important to be discovered by the detective, and told to the audience in this way:

1. Drop a subset of ordering constraints from the plan
2. For each actor a other than the murderer:
   (a) Replace the actor in the murder action with a
   (b) Check if the plan is consistent

If a consistent plan can be found this way for any subset of dropped ordering constraints, (some of) these constraints were essential for the requirement that the murderer is uniquely identifiable.

In a SIG encoding of a story the events that happen are totally ordered in the timeline layer. Since each node contains a natural language representation of its event without any associated formal semantics, it is not possible to reason about what would have happened had the events been ordered differently. In fact, as far as SIGs are concerned, the sequence of events "$Victor$ is murdered by $Victor$; $Victor$ goes to the garden shed" would be perfectly fine.

Finally, Impulse is based on Interval Temporal Logic, which is inherently suited for expressing durations and orderings of events and states. For example, the fact that rain makes the garden muddy can simply be expressed by [4]

$$\forall t \ \mathrm{raining}(t) \rightarrow \mathrm{muddy}(t)$$

It is also possible to express that the $walk$-action has an additional effect when it is muddy:

$$\forall s \in walk \ \mathrm{occurs}(s) \\ \rightarrow \\ \mathrm{at}(\mathrm{actor}(s), \mathrm{to}(s), \mathrm{eff1}(s))$$

$$\forall s \in walk \ \mathrm{occurs}(s) \wedge \mathrm{muddy}(\mathrm{pre1}(s)) \\ \rightarrow \\ \mathrm{footprints}(\mathrm{eff2}(s))$$

The first sentence just states that the $walk$-action has the effect of the actor being at the location they walked to, while the second sentence expresses the additional effect when it is muddy, which is footprints being left behind.

As with IPOCL plans, a system can use this representation to deduce what the detective needs to know to uniquely identify the murderer. As before, by replacing the actor that performs the murder action with every other actor in the story, and deriving a contradiction, the system can deduce what the detective has to know. For example, the line of reasoning when assuming $Isabel$ to be the murderer would be: Since there are no footprints in the garden, either no walk occurred, or it wasn't muddy. Since $Isabel$ is at the garden shed, a walk must have occurred (assuming her initial location was in the house). Therefore, it must not have been muddy at the time of the walk. However, since it started raining before the murder happened, the walk must also have happened before the murder. But then $Isabel$ would not have been at the murder site when the murder happened, which contradicts her being the murderer. A system can then start changing time intervals that occur in this line of reasoning, like the duration of the rain, and try to find a consistent story. Like before, having the rain start after the murder leads to a consistent story, and therefore this information must be discovered by the detective and conveyed to the audience.

Simply being able to reason about what needs to be discovered by the detective is not enough, though. The investigation process is driven by what the detective beliefs over time, and this is used to determine which further beliefs he needs to form to find the murderer. To be able to do that a

[4] This is actually a simplified representation that says "while it is raining, it will be muddy". Impulse could also express more complex relations like "5 minutes after it starts raining it will start being muddy, and will stay so until the sun comes out for more than an hour".

representation of the belief state of the detective is necessary. IPOCL plans do not have the capabilities to represent actor beliefs, although a system could track actor beliefs over time separately, and use IPOCL states for each of them. In Impulse beliefs and their durations are directly representable. When telling the detective story above, a system must then ensure that at each time only the facts that are in the detective's belief state are told to the audience. SIGs can represent actor beliefs, but their contents consist of natural language nodes without formal semantics. It is, however, possible to deduce which beliefs are actualized at each point in time in the story world, if they have been encoded in SIGs priorly. Therefore, SIGs could be used for this stage of the narrative discourse generation process, if the knowledge about action effects and character beliefs is generated in another way, e.g. utilizing IPOCL plans or Impulse.

Limitations and Future Work

As we discussed in the previous sections, Impulse provides a trade-off between expressivity and computational complexity of the reasoning process that is necessary for tasks such as narrative discourse generation. This does not however come without limitations. Predicate logic is undecidable in theory, and the addition of time and mental models does nothing to change that. Even on problems that are decidable, the additional expressivity increases computational complexity. Limiting the sentences to facts and rules, as described, reduces the computational complexity for practical applications, by using forward chaining. We acknowledge that determining temporal consistency is still undecidable in the general case (Bresolin et al. 2008), but it is tractable for practical applications, like the ones discussed above.

Another limitation of Impulse is that it does not use any linguistic model, and therefore its representation of stories is not fit for human consumption. In the case of Impulse, it is actually by design rather than an oversight. By not being tied to any particular medium of narrative, Impulse can be used in a variety of situations, and can be augmented by a media model for the desired output mode. For example, a text realizer can be used to translate Impulse stories into text, while a camera planning system could do the same for a movie. Since the underlying representation is medium-independent, it is also possible to combine the two, with some parts of the story told in text, and video clips for others.

Another point of concern is the acquisition and the modeling of the stories themselves in Impulse. For SIGs, for example, one of the driving goals was to create a user-friendly tool that allows easy annotation of existing stories through a visual graph-based representation. Impulse sentences, on the other hand, may seem challenging to write. However, it is possible to automatically convert IPOCL plans to Impulse, even though these would not use the full capabilities of the temporal representation and BDI model. Likewise, it is possible to convert SIGs to Impulse, which would result in some loss of the subtlety of a natural language representation. However, this baseline Impulse representation could be improved in both cases by using the additional capabilities that Impulse provides. For example, an Impulse story created from a IPOCL plan could be extended through the use of a micro-theory of beliefs and perceptions to automatically generate beliefs for the actors, for example by using actor-action and actor-actor co-location to build up actor beliefs about the state of the world. On the other hand, an Impulse translation of a SIG could be augmented with definitions of action effects that can be used to verify the story consistency, by checking if every action really has the effects it should have, and is only performed in situations where it could be performed. Such an encoding could also be used to reason about potential alternative stories.

Conclusion

We discussed the trade-off between expressivity and computational cost of the reasoning process that languages for story representation have to make. Generally speaking, more expressive languages are also more computationally expensive to reason about than less expressive languages. In particular, it is not currently feasible to reason about story content represented purely in natural language. Even relatively simple languages like those based on planning are hard to reason about in theory. Problems encountered in practice, however, are simple enough to be solved in reasonable time. Impulse, which combines Interval Temporal Logic as used by Allen and Ferguson with a BDI model like the one described by Cohen and Levesque, is more expressive than planning-based languages, but also more computationally expensive to reason about.

To illustrate our point, we have described the task of narrative discourse generation, which consists of creating a telling of a story that satisfies author-defined goals. We showed two examples where a logic-based language like Impulse provides the necessary expressivity for the task, while still being feasible to reason about. The first example is a telling of Aesop's fable "The Fox and the Crow" from the point of view of the Crow which requires a model of the Crow's beliefs, and what she believes about the effects of her actions. The second example is a simple murder mystery story, where we draw parallels between conveying information to the audience and the detective learning about the murder. We argue that to uniquely identify the murderer, the detective and the audience have to reason about what effects actions have or could have had and the order they happen in. Furthermore, a system would have to keep track of the detective's belief about the world.

We conclude that Impulse offers more expressivity than plan-based and similar languages, but unlike languages based on natural language, it is still feasible to reason about stories it is used to encode. While this property proved to be important for narrative discourse generation, we believe that it would also benefit many other narrative-related tasks.

Acknowledgements

This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

References

Allen, J. F., and Ferguson, G. 1994. Actions and events in interval temporal logic. *Journal of logic and computation* 4(5):531–579.

Bäckström, C., and Jonsson, P. 2011. All PSPACE-complete planning problems are equal but some are more equal than others. In *Fourth Annual Symposium on Combinatorial Search*.

Barbosa, S. D. J.; Lima, E. S. d.; Furtado, A. L.; and Feijò, B. 2014. Generation and Dramatization of Detective Stories. *SBC Journal on Interactive Systems* 5(2):39–52.

Bresolin, D.; Della Monica, D.; Goranko, V.; Montanari, A.; and Sciavicco, G. 2008. Decidable and undecidable fragments of halpern and shohams interval temporal logic: towards a complete classification. In *Logic for Programming, Artificial Intelligence, and Reasoning*, 590–604. Springer.

Chatman, S. B. 1980. *Story and discourse: Narrative structure in fiction and film*. Cornell University Press.

Church, A. 1936. A note on the entscheidungsproblem. *The journal of symbolic logic* 1(01):40–41.

Cohen, P. R., and Levesque, H. J. 1990. Intention is choice with commitment. *Artificial intelligence* 42(2):213–261.

Eger, M.; Barot, C.; and Young, R. M. 2015. Impulse: A formal characterization of story. In *Proc. of the Sixth International Workshop on Computational Models of Narrative*.

Elson, D. K., and McKeown, K. R. 2007. A platform for symbolically encoding human narratives. In *Proc. of the AAAI Fall Symposium on Intelligent Narrative Technologies*.

Elson, D. K. 2012. *Modeling narrative discourse*. Ph.D. Dissertation, Columbia University.

Fitting, M. 2012. *First-order logic and automated theorem proving*. Springer Science & Business Media.

Genette, G. 1983. *Narrative discourse: An essay in method*. Cornell University Press.

Hoek, M.; Theune, M.; and Linssen, J. 2014. Generating game narratives with focalization and flashbacks. In *Proceedings of the AIIDE 2014 Workshop on Games and Natural Language Processing*, 9–14. AAAI Press.

Jhala, A., and Young, R. M. 2010. Cinematic visual discourse: Representation, generation, and evaluation. *IEEE Transactions on Computational Intelligence and AI in Games* 2(2):69–81.

Levesque, H. J. 1988. Logic and the complexity of reasoning. *Journal of Philosophical Logic* 17(4):355–389.

Martens, C.; Ferreira, J. F.; Bosser, A.-G.; and Cavazza, M. 2014. Generative Story Worlds as Linear Logic Programs. In *Proceedings of (INT)7 - Intelligent Narrative Technologies*.

Montfort, N. 2011. Curveship's automatic narrative style. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, 211–218. ACM.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Rishes, E.; Lukin, S. M.; Elson, D. K.; and Walker, M. A. 2013. Generating different story tellings from semantic representations of narrative. In *Interactive Storytelling*. Springer. 192–204.

Todorov, T. 1977. The poetics of prose. *Paris: Ithaca*.

Turing, A. M. 1936. On computable numbers, with an application to the entscheidungsproblem. *J. of Math* 58(345-363):5.