Governing Narrative Events With Institutional Norms

Matt Thompson[1], Julian Padget[1], and Steve Battle[2]

1 **University of Bath**
Bath, United Kingdom
{mrt32,masjap}@bath.ac.uk
2 **Sysemia Ltd**
Bristol, United Kingdom
steve.battle@sysemia.co.uk

Abstract

A narrative world can be viewed as a form of society in which characters follow a set of social norms whose collective function is to guide the characters through (the creation of) a story arc and reach some conclusion. By modelling the rules of a narrative using norms, we can govern the actions of agents that act out the characters in a story. Agents are given sets of permitted actions and obligations to fulfil based on their and the story's current situation. However, the decision to conform to these expectations is ultimately left to the agent. This means that the characters have control over fine-grained elements of the story, resulting in a more flexible and dynamic narrative experience. This would allow the creator of an interactive narrative to specify only the general structure of a story, leaving the details to the agents. We illustrate a particular realisation of this vision using a formalization of Propp's morphology in a normative social framework, with belief-desire-intention agents playing the characters.

1998 ACM Subject Classification I.2.11 Distributed Artificial Intelligence

Keywords and phrases institutions, norms, narrative, agents

Digital Object Identifier 10.4230/OASIcs.CMN.2015.142

1 Introduction

A satisfying narrative must be more than just a series of interactions between character agents. There is a need for some underlying structure to these interactions. Additionally, agents are not a natural way to model events such as off-screen occurrences or scene introductions from a narrator.

Simulating a narrative using intelligent agents as characters offers many advantages. Each agent can be programmed to behave in certain idiosyncratic ways, based on a psychological or behavioural model. A common approach to add narrative structure to an agent-based simulation is to implement a drama manager, as in Mateas and Sterns' *Façade* [9].

This presents a problem: if the agents are being governed by a drama manager, to what extent are they autonomous? Do they still have some degree of 'free will' to carry out their own individual actions, in accordance with their personalities?

Other approaches to balancing authorial control with player or character agency include the use of director agents [8], reincorporation of player actions back into the narrative [15] and mediation to prevent narrative-breaking actions [12].

In this paper we present an approach to regulating narrative structure while still allowing agents some degree of autonomy. The narrative world is described and managed using an institutional model.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 142–151

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

An institutional model can be thought of as a model of society. By specifying a set of social norms, certain agent behaviours can be encouraged or discouraged according to the needs of that society. Institutions have been used to simulate the workings of auctions [3], vehicle convoys [1] and crowd movement [7]. All these applications are similar in that they all involve intelligent agents working together in a social environment.

The advantages of using institutions to govern agents' behaviours is that they still allow the agents some autonomy in their actions. The rules of a society are implied, and while adherence to these rules is encouraged, it is possible for them to be broken (often incurring a penalty). This makes them ideal for regimenting the actions of characters in a narrative. In order to have a narrative that is satisfying and consistent with a certain story world, some kind of structure is needed. However, if this narrative is to be interactive, the characters within the narrative need some degree of freedom in their actions. They need the ability to bend or break the rules of the storyworld at times, in order to surprise the player. Institutions make this possible for the agents to do. However, as with breaking the rules of any society, diverging from the norm may bring penalties and hardship upon the deviating agent.

In order to describe a narrative using an institution, we use Vladimir Propp's formalism of Russian folktales, from "The Morphology of the Folktale" [10].

2 Propp's Morphology of the Folktale

Propp's seminal work "The Morphology of the Folktale" [10], though first published in 1928, is still a widely-used formalism for researchers and game designers looking to generate narratives procedurally. Propp identifies recurring characters and motifs in Russian folklore, distilling them down to a concise syntax with which to describe stories.

In this formalism, characters have *roles*, such as *hero*, *villain*, *dispatcher*, *false hero*, and more. Characters performing a certain role are able to perform a subset of *story functions*, which are actions that make the narrative progress. For example, the *dispatcher* might send the *hero* on a quest, or the *victim* may issue an *interdiction* to the *villain*, which is then *violated*.

Propp defines a total of 31 distinct story functions, some of which can have subtle variations from story to story. Each function is given a number and symbol in order to create a succinct way of describing entire stories. Examples of such functions are:

- One of the members of a family absents himself from home: *absentation*.
- An interdiction is addressed to the hero: *interdiction*.
- The victim submits to deception and thereby unwittingly helps his enemy: *complicity*.
- The villain causes harm or injury to a member of the family: *villainy*.

Each of these functions can vary to a great degree. For example, the *villainy* function can be realised as one of 19 distinct forms of villainous deed, including *the villain abducts a person*, *the villain seizes the daylight*, and *the villain makes a threat of cannibalism*.

These functions are enacted by characters following certain roles. Each role (or *dramatis personae* in Propp's definition) has a *sphere of action* consisting of the functions that they are able to perform at any point in the story. Propp defines seven roles that have distict spheres of action: *villain*, *donor*, *helper*, *princess*, *dispatcher*, *hero*, and *false hero*.

In a typical story, one story function will follow another as the tale progresses in a sequential series of cause and effect. However, Propp's formalism also allows for simultaneous story functions to occur at once.

2.1 Example: A Punch and Judy show

Consider the classic British-Italian "Punch and Judy" puppet show often seen at English seaside resorts. The "Punch and Judy" world is a very simple and consistent narrative domain, in which simplistic characters act out predictable sequences of events. The key features of a Punch and Judy show include:

- The show is introduced by a clown named "Joey".
- Punch beats and kills his child, and then his wife Judy.
- There is a scene where Punch chases a monkey or cat.
- A policeman tries to arrest Punch, but is instead killed by him.
- Joey asks Punch to look after some sausages in one scene. Shortly after Joey leaves, a crocodile appears and eats them.
- Punch, the lead character, beats and kills almost every other character by the end of each scene. Only Joey and sometimes the monkey or cat avoid this fate.
- The show sometimes ends with an encounter between Punch and the Devil, which Punch wins.

Despite this harrowing combination of narrative elements, Punch and Judy is considered a *farce* due to the over-the-top violence and simplicity of its world. It is usually performed as a puppet show for children, who are encouraged to cheer or boo the puppets.

The common elements of Punch and Judy are easily described in terms of Propp's story functions. Using the example where Joey asks Punch to guard some sausages, the appropriate story functions are:

1. Joey tells Punch to look after the sausages (*interdiction*).
2. Joey has some reservations, but decides to trust Punch (*complicity*).
3. Joey gives the sausages to Punch (*provision or receipt of a magical agent*).
4. Joey leaves the stage (*absentation*).
5. A crocodile enters the stage and eats the sausages (*violation*).
6. Punch fights with the crocodile (*struggle*).
7. Joey returns to find that the sausages are gone (*return*).

In order to better model the Punch and Judy world in terms of Propp functions, we have allowed some flexibility of the roles that each agent assumes. At points Punch is the hero, at other times he is the villain. Sometimes Joey is the hero, but he can also be a donor (a character who gives an object to the hero). The crocodile is a villain, but other characters are all certainly victims (since they are all obliged to be killed by Punch as part of the Punch and Judy story world).

One novel aspect of managing these Propp functions with an institutional model is that the agents' roles *can* be flexible. If the audience cheers on Judy as she hits Punch, why not fulfil their desires and make her the hero, and Punch the victim? This is what we aim to achieve with our approach: a story world where certain rules do hold, but are flexible enough to be broken if the player or audience wills it.

3 Institutions for narrative regulation

3.1 Institutions and norms

Early examples of institutional models suggest their application to the regulation of systems involving multiple actors. Noriega's "fish market" thesis describes the application of an agent-mediated institution for regulating a fish market auction scenario [3], checking the

validity of agent actions and addressing the issue of agent accountability in an auction environment. Rodriguez [13], and later Vázquez-Salceda [16], refine and extend Noriega's implementation of agent-mediated institutions.

However, it is Cliffe's approach of using Answer Set Programming (ASP) to specify institutions that we use here [4]. We define an institution in terms of *deontic logic*, specifying the permissions and obligations that act upon agents at any particular point in the story.

This approach alone is not enough, however. In order to effectively model a narrative using an institution and ASP, we must use a formalism for narrative that specifies which events and actions occur at certain points in the narrative. We achieve this by translating Propp's formalism of Russian folktales [10] into actions that agents are permitted or obliged to perform.

3.2 Describing institutions with deontic logic

We describe our institution using *deontic logic*, defining our model in terms of *fluents*, *events*, *powers*, *permissions* and *obligations*.

3.2.1 Fluents

Fluents are properties that may or may not hold true at some instant in time. *Institutional events* are able to *initiate* or *terminate* fluents at points in time. A fluent could describe whether a character is currently on stage, the current scene of a story, or whether or not the character is happy at that moment in time.

Domain fluents ($\mathcal{D}$) describe domain-specific properties that can hold at a certain point in time. In the Punch and Judy domain, these can be whether or not an agent is on stage, or their role in the narrative (equation 1).

Institutional fluents consist of *institutional powers*, *permissions* and *obligations*.

$$\mathcal{D} = \{\texttt{onstage, hero, villain, victim, donor, item}\} \tag{1}$$

An **institutional power** ($\mathcal{W}$) describes whether or an agent, and by extension the action they have taken, has the authority to meaningfully generate an institutional event. Using Propp as an example, a *violated interdiction* can only occur after an *interdiction* has taken place. Therefore, the institution would not be empowered to generate a *violated interdiction* institutional event if the prior *interdiction* has not yet taken place.

Institutional powers describe what events the institution is capable of bringing about. As institutional events represent Propp's story functions in our model, the institution should only be capable of generating events if they fit in the right place in the narrative. For example, a *violation* can take place only after an *interdiction* event has occurred. Punch can only violate Joey's request to guard the sausages after the request itself has happened. Equation 2 shows a list of possible empowerments, essentially a list of institutional events.

$$\begin{aligned}\mathcal{W} = \{&\texttt{pow(introduction), pow(interdiction), pow(give), pow(absentation),}\\ &\texttt{pow(violation), pow(return)}\}\end{aligned} \tag{2}$$

Permissions ($\mathcal{P}$) are external actions that agents are permitted to do at a certain instant in time. These can be thought of as the set of *socially permitted* actions available to an agent. While it is possible for an agent to perform other actions, societal norms usually prevent them from doing so.

For example, it would not make sense in the world of Punch and Judy if Punch were to give the sausages to the Policeman. It is always Joey who gives the sausages to Punch. Also,

it would be strange if Joey were to do this in the middle of a scene where Punch and Judy are arguing. We make sure agents' actions are governed so as to allow them only a certain subset of permitted actions at any one time. Equation 3 shows a list of permission fluents.

$$\mathcal{P} = \{\texttt{perm(leavestage), perm(enterstage), perm(die), perm(kill),} \\ \texttt{perm(hit), perm(give), perm(fight)}\} \tag{3}$$

Obligations ($\mathcal{O}$) are actions that agents *should* do before a certain deadline. If the action is not performed in time, a *violation event* is triggered, which may result in a penalty being incurred. While an agent may be obliged to perform an action, it is entirely their choice whether or not they actually do so. They must weigh up whether or not pursuing other courses of action is worth suffering the penalty that an unfulfilled obligation brings.

Anybody who has seen a Punch and Judy show knows that at some point Joey tells Punch to guard some sausages, before disappearing offstage. Joey's departure is modelled in the institution as the *absentation* event. It could be said that Joey has an obligation to leave the stage as part of the *absentation* event, otherwise the story function is violated. Equation 4 shows how this would be described in the institution.

$$\mathcal{O} = \{\text{obl}(\texttt{leavestage, absentation, viol(absentation)})\} \tag{4}$$

3.2.2 Events

Cliffe's model specifies three types of **event**: *external events* (or 'observed events', $\mathcal{E}_{obs}$), *institutional events* ($\mathcal{E}_{instevent}$) and *violation events* ($\mathcal{E}_{viol}$). *External events* are observed to have happened in the agents' environment, which can *generate institutional events* which act only within the institional model, *initiating* or *terminating* fluents, permissions, obligations or institutional powers. An external event could be an agent leaving the stage, an agent hitting another, or an agent dying. Internal events include narrative events such as scene changes, or the triggering of Propp story functions such as *absentation* or *interdiction* (described in Section 2).

Violation events occur when an agent has failed to fulfil an obligation before the specified deadline. These can be implemented in the form of a penalty, by decreasing an agent's health, for example.

$$\mathcal{E}_{obs} = \{\texttt{startshow, leavestage, enterstage, die, give,} \\ \texttt{harmed, hit, fight, kill, escape}\} \tag{5}$$

$$\mathcal{E}_{instact} = \{\texttt{introduction, interdiction, give, absentation,} \\ \texttt{violation, return, struggle, defeat, complicity,} \\ \texttt{victory, escape}\} \tag{6}$$

$$\mathcal{E}_{viol} = \{\texttt{viol(introduction), viol(interdiction), viol(give),} \\ \texttt{viol(absentation), viol(violation), viol(return),} \\ \texttt{viol(struggle), viol(defeat), viol(complicity)} \\ \texttt{viol(victory), viol(escape)}\} \tag{7}$$

3.2.3 Event Generation and Consequences

An **event generation** function, $\mathcal{G}$, describes how events (usually external) can generate other (usually institutional) events. For example, if an agent leaves the stage while the

$$\mathcal{G}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \mathit{tellprotect}(\texttt{donor, villain, item}) \rangle \rightarrow \{\mathit{interdiction}\} \quad (8)$$

$$\langle \{\mathit{interdiction}\}, \mathit{agree}(\texttt{villain}) \rangle \rightarrow \{\mathit{complicity}\} \quad (9)$$

$$\langle \emptyset, \mathit{give}(\texttt{donor, villain, item}) \rangle \rightarrow \{\mathit{receipt}\} \quad (10)$$

$$\langle \{\mathit{interdiction}\}, \mathit{leavestage}(\texttt{donor}) \rangle \rightarrow \{\mathit{absentation}\} \quad (11)$$

$$\langle \{\mathit{interdiction}\}, \mathit{harmed}(\texttt{item}) \rangle \rightarrow \{\mathit{violation}\} \quad (12)$$

$$\langle \{\mathit{interdiction}, \mathit{absentation}\}, \mathit{enterstage}(\texttt{donor}),\ \text{onstage}(\texttt{villain}) \rangle \rightarrow \{\mathit{return}\} \quad (13)$$

$$\langle \emptyset, \mathit{hit}(\texttt{donor, villain}) \rangle \rightarrow \{\mathit{struggle}\} \quad (14)$$

$$\mathcal{C}^{\uparrow}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \texttt{receipt} \rangle \rightarrow \{\text{perm}(\texttt{leavestage(donor)})\} \quad (15)$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{violation} \rangle \rightarrow \{\text{perm}(\texttt{enterstage(dispatcher)})\} \quad (16)$$

$$\langle \{\mathit{active}(\mathit{absentation}), \mathit{active}(\mathit{violation})\}, \texttt{return} \rangle \rightarrow \{\text{perm}(\texttt{hit(donor, villain)})\} \quad (17)$$

$$\mathcal{C}^{\downarrow}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \texttt{interdiction} \rangle \rightarrow \{\text{perm}(\texttt{give(donor, villain, item)})\} \quad (18)$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{absentation} \rangle \rightarrow \{\text{perm}(\texttt{leavestage(donor)})\} \quad (19)$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{violation} \rangle \rightarrow \{\mathit{active}(\mathit{interdiction})\} \quad (20)$$

$$\langle \{\mathit{active}(\mathit{absentation}), \mathit{active}(\mathit{violation})\}, \texttt{return} \rangle \rightarrow \{\mathit{active}(\mathit{absentation})\} \quad (21)$$

Figure 1 Generation and consequence rules for Punch and Judy.

interdiction event holds, they trigger the *leavestage* event. This combination generates the *absentation* institutional event (equation 11).

Event generation functions follow a ⟨`preconditions`⟩ → {`postconditions`} format, where the preconditions are a set of fluents that hold at that time and an event that has occurred, and the postconditions are the events that are generated. They are generally used to generate internal, institutional events from external events.

Consider the Punch and Judy scenario described in Section 2.1. There are seven institutional events (story functions) that occur during this scene: *interdiction*, *complicity*, *receipt* (from Propp's *receipt of a magical agent*) *absentation*, *violation*, *struggle*, *return*. These institutional events are all generated by external events. The *interdiction* is generated when Joey tells Punch to protect the sausages. Punch agreeing amounts to *complicity*. Joey *gives* punch the sausages (*receipt*), then leaves the stage (*absentation*). The crocodile eating the sausages is a *violation* of Punch's oath, the agents fight (*struggle*), then Joey enters the stage again (*return*).

It is desirable that these story function occur in this sequence in order for a satisfying narrative to emerge. Agents may decide to perform actions that diverge from this set of events, but the institution is guiding them towards the most fitting outcome for a *Punch and Judy* world. For this reason, a currently active story function can be the precondition for event generation. For example, the *receipt* event may only be triggered if an agent externally performs a *give* action **and** if the *complicity* event currently holds (equation 10). Examples of event generation function for this scenario, complete with preconditions, are listed in equations 8 to 14 in Figure 1.

Consequences consist of fluents, permissions and obligations that are *initiated* ($\mathcal{C}^{\uparrow}$) or *terminated* ($\mathcal{C}^{\downarrow}$) by institutional events. For example, the institutional event *give* could initiate the donor agent's permission to leave the stage, triggering the *absentation* event (equation 11). When the *interdiction* event is currently active and a *violation* event occurs, the interdiction event is terminated (20). Equations 15 to 21 in Figure 1 describe the initiation and termination of fluents in the Punch and Judy sausages scenario detailed in Section 2.1.

4 Regimenting agent actions with institutions

4.1 Institutions and multi-agent systems

Belief-Desire-Intention (BDI) agents' behaviour can be governed by running an institution manager in their environment, observing all agent actions and events. Given a set of observed events over time, such a manager can infer what permissions, obligations and institutitional powers hold at any given time.

The institution manager updates each agents' percepts to change their permissions and obligations. At each instant in time, the institution manager works out what an agent is permitted or obliged to do, then updates the agent's percepts (beliefs about the environment) with the set of permissions and obligations that hold at that time. It is up to the agent whether or not they act on these percepts.

As part of the BDI architecture of agents, an agent has beliefs about themselves, other agents and their environment. They also have goals that they desire to carry out (desires) and goals they intend to carry out next or are carrying out (intentions). The permissions and obligations that an agent receives from the institution manager only affect their beliefs: they believe that the norms of their world put certain expectations on them. These beliefs may or may not affect the plans that the agent desires or intends to carry out.

4.2 Describing institutions with InstAL and ASP

Answer Set Programming (ASP) [2] is a method of programming by specifying the requirements that a solution must fulfil. A specification of the constraints and rules of a problem are written and then queried, producing solutions in the form of answer sets.

Each line of an ASP program is a *rule*, which is a constraint that narrows down the set of solutions when queried. Rules consist of two parts: a head literal (l) and a body (B), separated with a left arrow: $l \leftarrow B$. If every literal in the body evaluates to $true$, then the head literal is also true.

Specifying our institution in ASP allows us to reason about the effects of events occuring over time. Given an institutional model and a sequence of events as input, the output would be the set of norms in the form of *permissions* and *obligations* that hold at certain instants in time.

To describe our institutional model, we use InstAL [4], a domain specific language for describing institutions that compiles to AnsProlog, a declarative programming language for Answer Set Programming (ASP) [2]. instAL's semantics are based upon the Situation Calculus [11] and the Event Calculus [6]. It is used to describe how external events generate institutional events, which can then initiate or terminate fluents that hold at certain instants in time. These fluents can include the permissions and obligations that describe what an agent is permitted or obligated to do at specific points in time.

Returning to the scenario in Section 2.1, if an agent with the role of *donor* leaves the stage, it generates the *absentation* Propp story function in the institution:

```
leaveStage(X) generates intAbsentation(X) if role(X, dispatcher),
    activeTrope(interdiction);
```

The *absentation* institutional event gives the crocodile permission to enter the stage if there are any sausages on the stage. It also terminates the permission of the absented agent to leave the stage, as they have already done so:

```
intAbsentation(X) initiates perm(enterStage(croc)) if objStage(sausages)
    ;
intAbsentation(X) terminates onStage(X), perm(leaveStage(X));
```

InstAL rules like those shown above are compiled into AnsProlog ASP rules describing which fluents hold at certain points in time. Once the InstAL model is compiled to AnsProlog, we use the *clingo* answer set solver [5] to ground the logical variables, and 'solve' queries by finding all permissions and obligations that apply to any agents, given a sequence of events as the query input. The agents' percepts are then updated with their permitted and obliged actions from that moment in time onwards.

Listing 1 shows how the sausages scenario would be described in ASP, for the first two events of the scene. Starting with an initial set of fluents that hold at $t0$, only fluents that have been initiated and not terminated hold at the next instant.

Listing 1 Sausages scenario in ASP

```
holdsat(perm(tellprotect(dispatcher, villain, item), t0).
occurred(tellprotect(dispatcher, villain, item), t0).
initiated(active(interdiction), t1).
initiated(perm(give(donor, villain, item)), t1).
terminated(tellprotect(dispatcher, villain, item), t1).
holdsat(perm(give(donor, villain, item)), t1).
holdsat(active(interdiction), t1).
occurred(give(donor, villain, item), t1).
initiated(active(receipt), t2).
initiated(perm(leavestage(donor)), t2).
terminated(perm(give(donor, villain, item)), t2).
holdsat(active(interdiction), t2).
holdsat(active(receipt), t2).
holdsat(perm(leavestage(donor)), t2).
```

4.3 Adding agent percepts from ASP solutions

With every event that occurs in the narrative, a query consisting of all events so far is sent to the solver. Its output tells us what permissions and obligations hold for certain agents at the next instant. These permissions and obligations are added to the agents' belief bases as percepts. The agents' plans are carried out based on these permissions and obligations.

For example, in the scene where Joey gives the sausages to Punch, Punch may see that he has permission to eat the sausages, drop them, fight the crocodile, run away (leave the stage)

or shout for help at the crocodile or audience. His obligation for the scene, in accordance with the Punch and Judy narrative world, is to either eat the sausages himself, or let the crocodile sausages. This ends Propp's *interdiction* story function with a *violation* function. Note that his obligation is not to guard the sausages as asked to by Joey. While Joey's entrustment of the sausages is an obligation of sorts, Punch's only true obligations are to the narrative.

We have a prototype system where the agents choose their actions based on their emotional state. Before carrying out a potentially narrative-altering plan, each agent appeals to the audience for encouragement. They do this by turning to the audience and announcing their intentions. The audience then cheers or boos the character, which affects their emotional state, which is based on Russell's [14] circumplex model of emotion. In this model, a person's emotion is determined by three variables: Valence (positivity), Arousal and Dominance.

Depending on the action planned, a cheer or boo from the audience will raise or lower an agent's valence, arousal or dominance level. This changes the agents' motivation to select a certain permitted action to carry out as part of their plan.

In the above example, a depressed Punch may decide to violate his obligations by not eating the sausages and instead leave the stage with them. Alternatively, a furious Punch would viciously attack the crocodile, not allowing him to eat the sausages. This also violates the norms of the narrative world. However, for most emotional states the norms are observed by either Punch eating the sausages or letting the crocodile eat them.

5 Conclusion

With our approach to interactive narrative generation, we regiment the rules of the story domain using an institutional model. This model describes what each agent is permitted and obliged to do at any point in the story. Institutional regimentation of agents acting out a story using story-world norms allows much more flexibility than if the world's rules were strictly enforced. The deontic language of permissions and obligations allows the agents to act out small details of the narrative, while guiding them into an underlying narrative structure.

References

1 Vincent Baines and Julian Padget. A situational awareness approach to intelligent vehicle agents. In Michael Behrisch and Melanie Weber, editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 77–103. Springer International Publishing, 2015.

2 Chitta Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

3 Pablo Cayetano Noriega Blanco-Vigil. *Agent mediated auctions: the fishmarket metaphor*. PhD thesis, Universitat Autònoma de Barcelona, 1998.

4 Owen Cliffe, Marina De Vos, and Julian Padget. Specifying and reasoning about multiple institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 67–85. Springer, 2007.

5 Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

6 Robert Kowalski and Marek Sergot. A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer, 1989.

7 JeeHang Lee, Tingting Li, and Julian Padget. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013.

8 Seung Y Lee, Bradford W Mott, and James C Lester. Learning director agent strategies: An inductive framework for modeling director agents. In *Intelligent Narrative Technologies*, 2011.

9 Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*, pages 4–8, 2003.

10 Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 2010.

11 Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy*, 27:359–380, 1991.

12 Justus Robertson and Robert Michael Young. Modelling character knowledge in plan-based interactive narrative to extend accomodative mediation. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

13 Juan A Rodriguez-Aguilar et al. *On the design and construction of Agent-mediated Institutions*. PhD thesis, Universidad Autónoma de Barcelona, 2001.

14 James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

15 Zach Tomaszewski. On the use of reincorporation in interactive drama. In *Intelligent Narrative Technologies*, 2011.

16 Javier Vázquez-Salceda. The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework. *AI Communications*, 16(3):209–212, 2003.