

Interactive Narrative Intervention Alibis through Domain Revision

Justus Robertson and **R. Michael Young**

Liquid Narrative Group
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
jjrobert@ncsu.edu, young@csc.ncsu.edu

Abstract

Interactive narrative systems produce branching story experiences for a human user using an interactive world. A class of interactive narrative systems, called strong story systems, manage a user's experience by manipulating the interactive world and its characters according to a formal story model. In these systems, a human user may place the world into a state such that the formal story model can no longer control interaction. One solution to this problem, called intervention, is to exchange the undesirable outcomes of a player's action for a set that do not violate the story model. However, the player may become aware that their intended action is being intervened against by a context-sensitive, meta-narrative process. In this paper we describe a method of ensuring game world alibis for interventions through domain modification of world mechanics.

Introduction

The challenge of interactive narrative is to create an interesting or structured story experience that changes and adapts to the actions a human user takes as they interact with the storytelling system. One approach to this problem, called strong story systems (Riedl and Bulitko 2013), control interactive experiences with a well-defined central story model. Automated strong story systems generate their story models according to constraints set by a human author and in response to or anticipation of actions a user can perform. However, sometimes an automated strong story system will not be able to generate a system response to all user actions that satisfy its pre-defined author constraints. When this happens, the system can make one of several choices: don't present the harmful action as a possibility to the player, throw out the authorial constraints and replace them with an alternative set that the system can satisfy (Riedl et al. 2008), or allow the player to take action but modify the action's outcome so that it does not contradict the author's goals (Riedl, Saretto, and Young 2003).

Not presenting or not allowing the course of action to the user may be straightforward to implement in most cases. However, doing so may make it apparent to the user that some courses of action, available in other circumstances,

are now unavailable for no apparent reason. This realization could have an adverse effect on aspects of the user's experience, like decreasing the user's sense of agency or autonomy (Mawhorter et al. 2014). The second method, exchanging the initial set of constraints set by the author on the story for a new set, is effective but requires more work on the part of the author to specify sets of constraints that are each effective at allowing the system to generate interesting experiences. Furthermore, it could be that none of the sets of constraints are met by the world state to which the user has navigated. In this case, the method is no longer an option. The third method, called intervention, requires an author to create special operators called failure modes that match the action the user is trying to perform but with slightly different outcomes that do not violate the story model. This method may also lead to the user realizing that in some cases their action is executed normally, but in other cases it behaves according to the intervention failure mode outcome for no apparent in-world reason. Like the first method, this may lead to adverse effects on agency and autonomy.

In this paper we present a method of providing *alibis* for interventions. An alibi is a shift in the game world, imperceptible to the user, that ensures a logical explanation for why the system behaves a certain way. In our work, we provide alibis for interventions by rewriting author-specified axioms on the fly to provide explanations for failure mode behavior in terms of replicatable world mechanics. By rewriting world mechanics whenever an intervention is performed, the system provides an in-world context for the alternate outcome that always differs from the original action. Furthermore, the system maintains a model of what world mechanics the user has observed and only performs interventions if the rewritten rule has not yet been observed.

Related Work

This paper presents a method of ensuring interventions in interactive narratives can be explained away by world mechanics. Mimesis (Young et al. 2004) is a plan-based interactive narrative system that uses interventions as one of two strategies, described in Riedl, Saretto, and Young (2003), to merge autonomous player behavior with an automatically generated plot. One expected downside to the intervention method used in Mimesis is its negative impact on *invisibility* (Roberts and Isbell 2008), specifically, when the meta-

narrative system used to select failure mode interventions becomes apparent to the player.

We ensure interventions can be explained away by providing world-mechanic alibis for intervention behavior. Sunshine-Hill and Badler (2010) used alibis to explain away randomly generated NPC behavior once a player begins paying attention to the character by transitioning from a random behavior generator to a behavior pattern that explains their current activity in the context of a sequence of actions that accomplish some task. Robertson and Young (2014) use alibis in the context of an interactive narrative system to widen the space of accommodations the system can find and Li et al. (2014) presents a system that generates NPC alibis and communicates them to the player using natural language.

We implement our method in a plan-based experience management system by selectively rewriting conditional effects stored in domain operators. Porteous et al. (2015) has also applied the technique of domain revision to interactive narrative, generating new planning operators that create more pathways through an interactive narrative space by procedurally creating antonyms of existing actions.

World Mechanic Alibis

The system we describe here provides in-world alibis for interventions by changing aspects of the world mechanics that the player has not observed to allow the player to take an action that otherwise would derail the interactive narrative system. This can be accomplished while remaining consistent from the player’s perspective if the system tracks what game rules the player has observed and only changes aspects of the game mechanics that the player has not yet observed.

At design time, the interactive narrative author defines two types of world mechanics: static and dynamic. Static rules can never be removed by the system, they always dictate how the world operates. A dynamic rule can be removed from the system if it leads to an unwanted state and the player has not observed it hold in the world.

Our implementation of world mechanic alibis is a plan-based system where dynamic rules are implemented as conditional effects of plan actions.

Planning

The system we define is plan-based and uses a planner to construct linear narratives which consist of ordered actions for character to take within the story world. Plan actions are defined by a set of templates called operators.

Definition 1 (Operator). A planning *operator* is an action template that describes a state transition. Planning operators are represented $\langle \rho, \epsilon \rangle$ where ρ , the set of *preconditions*, is a set of atomic formulae that must hold true for the action to be performed, and ϵ , the set of *effects*, is a set of atomic formulae that become true after the action and *conditional effects*.

Our system uses conditional effects, which are syntactic sugar for specifying two operators at once.

Definition 2 (Conditional Effect). A *conditional effect* is a precondition-effect pair specified in a planning operator’s list of effects. Conditional effects are represented

$\langle c_\rho, c_\epsilon \rangle$ where c_ρ , the set of *preconditions*, is a set of atomic formulae that must hold true for the conditional effect to be triggered, and c_ϵ , the set of *effects*, is a set of atomic formulae that become true when the conditional effect holds.

A conditional effect c_ϵ is only added to an action’s effect list if the action’s set of preconditions, ρ , and set of conditional effect preconditions, c_ρ , hold in the current state.

A planning problem is comprised of two state descriptions and a set of operators.

Definition 3 (Planning Problem). A *planning problem* describes an initial and a goal state, which are conjunctions of ground atomic formula representing a start and end state for the world, and a set of operators. Planning problems are represented $\langle I, G, O \rangle$ where I is the initial state description, G is the goal state description, and O is the set of operators.

From the planning problem, a planner searches through sequences of fully ground operators called actions.

Definition 4 (Action). An *action* is a ground instantiation of an operator. Actions are represented by the triple $\langle \rho, \epsilon, \beta \rangle$, where ρ and ϵ are preconditions and effects from the corresponding operator, and β is a set of binding constraints on free variables in ρ and ϵ . The set of preconditions of a particular action a are referred to as ρ_a and the set of effect of a are referred to as ϵ_a .

Definition 5 (Plan). A *plan* is a sequence of actions that represents the solution to a planning problem.

A plan, when executed from the initial state specified by its planning problem, will produce the problem’s goal state.

Mediation

Mediation is a plan-based process that transforms a planning problem into a branching tree structure by analyzing dependency relationships between actions in an initial plan produced by a planner on the planning problem.

Definition 6 (Dependency). A *dependency* is a relationship between two plan actions and an atom in a plan P . A dependency is represented $\langle a_i, e, a_j \rangle$ where a_i and a_j are the two actions, e is the atom, e is an effect of a_i , e is a precondition of a_j , a_i is ordered before a_j in P , and for all a_k ordered between a_i and a_j in P , $e \notin \epsilon_{a_k}$. The set of all dependencies between actions in some plan P is referred to as D_P and the atom of dependency d is referred to as e_d .

A dependency is any world state literal needed for the execution of a plan action. It is an interval between the last action that establishes the literal to the action that lists the literal as a precondition. Any time the player has an opportunity to reverse a plan dependency with one of their actions, we say the action is exceptional.

Definition 7 (Exceptional Action). An *exceptional action* is any action a_e a player can perform in plan P over the interval of dependency $\langle a_i, e, a_j \rangle$, where $a_i, a_j \in P$, such that $\neg e \in \epsilon_{a_e}$.

A player-performed action is exceptional if it breaks a dependency in the current story plan by reversing an atomic formula listed as a precondition of a future plan action. The process of responding to an exceptional action by replanning is called accommodation.

Definition 8 (Accommodation). Given a conjunction of ground atomic formulae S that represents a world state after an exceptional action a_e performed by the player during execution of story plan P that is a solution to planning problem $\langle I, G, O \rangle$, an accommodation of a_e is a new plan P' that solves planning problem $\langle S, G, O \rangle$.

Iteratively accommodating possible exceptional actions is a method used by some interactive narrative systems to generate tree data structures that represent a branching story (Riedl and Young 2006), called a mediation tree.

Definition 9 (Mediation Tree). A *mediation tree* is a tree whose nodes are plans and edges are exceptional actions. Each of a node's outgoing edges represent a possible exceptional action in its plan. Each of a node's children is an accommodation plan for the exceptional action edge that connects the two nodes.

This tree can be pre-computed or expanded dynamically as a player interacts in a story world. However, if the planner cannot find a plan from the state following an exceptional action to the planning problem's goal state an accommodation cannot be made and a new branch cannot be added to the interactive story data structure. In this case, some other method must be used to reconcile the player's action with the narrative plan. One possible response, called intervention, would be for the system to exchange the set of effects of the player's exceptional action for a second set that is not exceptional, called a failure mode.

Definition 10 (Failure Mode). Given some operator $o = \langle \rho_o, \epsilon_o \rangle$, a failure mode of o , $o' = \langle \rho_o, \epsilon_{o'} \rangle$, is a second operator with the same set of preconditions as o but a different set of effects.

Definition 11 (Intervention). Given an exceptional action a_e , a system intervention replaces a_e with failure mode a'_e such that $\forall d \in D_P, \neg e_d \notin \epsilon_{a'_e}$.

Intervention is the process of exchanging the effects of the player's exceptional action for an alternate set that does not violate any dependencies. However, one intuitive problem with intervention is that there is no in-world reason why the failure mode effects happen in some situations but the normal effects in others. The driving force behind an intervention is whether an exceptional action has occurred, which depends on the system's current plan. The system plan exists outside the story world, so it might become apparent to the player by observing the special intervention reactions that the system is manipulating the world. In the next section we present a method of providing in-world alibis for interventions by ensuring that world-mechanics are altered to create an in-world context for triggered failure modes.

Operator Revision

At design time, our system allows the domain author to specify special, dynamic conditional effects.

Definition 12 (Dynamic Conditional Effect). A *dynamic conditional effect* is a conditional effect, $\langle d_\rho, d_\epsilon \rangle$, that is allowed to be removed from the set of effects ϵ_o of its operator o in the planning problem.

Dynamic conditional effects allow a domain author to specify two alternate behaviors for a domain per dynamic conditional effect: the base case where the dynamic conditional effect is erased and the more complex case where it remains and plays a role in the operator's behavior. This allows a domain author to over-specify a domain's mechanics with dynamic conditional effects and have the mediation system selectively remove rules if they ever lead to unwanted world states. This process is called operator revision intervention.

Definition 13 (Operator Revision Intervention). Given an exceptional action a_e with effect $\neg e \in d_\epsilon$ of dynamic conditional effect pair $d \in \epsilon_{a_e}$ that breaks some dependency $\langle a_i, e, a_j \rangle \in D_P$ of the current plan P that solves planning problem $\langle I, G, O \rangle$ where $o \in O$ is the operator a_e is instantiated from, remove d from ϵ_o .

By removing the dynamic conditional effect directly from the domain operator used to generate plans, all future instances of actions created from the operator will respond in the same way as the current intervention. In this way the world remembers the interventions it has performed and never jumps back and forth between a base and failure mode case. However, this does not yet account for what the player has observed in the domain. To ensure that the world's mechanics are consistent from the player's perspective we need to model what domain rules the player has observed.

Knowledge Microtheory

Similar to creating character alibis for accommodation (Robertson and Young 2014) the system here needs a working theory of what the player has observed so it never erases a dynamic conditional effect from the domain if the player knows it exists. A microtheory (Guha and Lenat 1993) is a group of statements about a topic that make special assumptions or simplifications and can be integrated with a more complete theory at a later date.

We label our microtheory *DomainKnowledgeMt*. To specify that some axiom A belongs to *DomainKnowledgeMt*,

$$ist(\text{DomainKnowledgeMt}, A) \quad (1)$$

where *ist* stands for 'is true in', *DomainKnowledgeMt* is the context where the axiom holds, and A is the axiom. There are three parts to our microtheory. We apply our microtheory to some world state s in the linearly ordered set of world states S that have occurred in the game world. We say an action $a \in s$ if a is the action that transformed s to its successor s' in S . First, a character observes every object or entity at their current location.

$$\begin{aligned} ist(\text{DomainKnowledgeMt}, \forall xyz \text{Character}(x) \wedge At(x, z) \\ \wedge At(y, z) \rightarrow \text{Observes}(x, y)) \end{aligned} \quad (2)$$

If one character observes another character as they perform an action, the first character also observes the action.

$$\begin{aligned} ist(\text{DomainKnowledgeMt}, \forall xyz \text{Character}(x) \wedge \text{Actor}(y, z) \\ \wedge \text{Action}(z) \wedge \text{Observes}(x, y) \rightarrow \text{Observes}(x, z)) \end{aligned} \quad (3)$$

Finally, if an action is observed then so are all of the action's enabled effects.

$$\begin{aligned} &ist(\text{DomainKnowledgeMt}, \forall xyz \text{Character}(x) \wedge \text{Action}(y) \\ &\quad \wedge \text{Effect}(z, y) \wedge \text{Observes}(x, y) \rightarrow \text{Observes}(x, z)) \end{aligned} \quad (4)$$

Note that if a conditional effect's preconditions are not met, the conditional effects will not be listed in the action's effect list, so a dynamic conditional effect is only observed when its preconditions are met.

Consistent Operator Revision

With the knowledge microtheory we can track what dynamic conditional effects have been observed by the player and only remove them from domain operators if the microtheory says they have not been observed in the past, or their observation can be explained away by an alternate plan.

Example

For an example, imagine a noir world where a detective, Thursday, has a confrontation with a mob leader, Big Boy, in a warehouse. Thursday has a pistol and in the warehouse is a faulty grenade. The goal state is for Big Boy to be deceased and Thursday to be alive. The 'take-object' operator used to take an object from a location has a dynamic conditional effect that when a faulty grenade is taken, it explodes in the hands of the taker and ends their life.

If the player as Thursday shoots Big Boy with the pistol, the goal state is satisfied and the game is over. If the player doesn't act, the system directs Big Boy to take the faulty grenade which explodes according to its dynamic conditional effect and satisfies the goal state. If the player takes the grenade, the dynamic conditional effect of faulty grenades blowing up is removed from the take action which preserves the player's life and allows the world to continue.

If the game were to continue then all future faulty grenades would not explode when taken and could be differentiated from regular grenades by another behavior, like not exploding when thrown. This is different from defining the same behavior with a failure mode intervention operator because the failure mode behavior would not persist and faulty grenades would explode whenever the system wanted a character dead but would not explode whenever it wanted a character to live.

Future Work

We plan to test whether players can detect repeated failure mode interventions and measure how they impact play experience. We expect, as others in the literature have pointed out, that repeated failure mode interventions will make the system's goals transparent to the player which will have a detrimental effect on aspects of their experience, like agency and autonomy. We plan to compare these observations against the same system using domain revision interventions. We expect that intervening with domain revisions will preserve the player's feelings of agency and autonomy by providing in-world rules that govern system intervention.

Conclusion

Operator revision is one way to structure interventions such that the system creates a world-rule alibi for what happened. This addresses the transparency issue of previous intervention methods, where repeated system interventions might lead to the player becoming aware of the experience management system interfering with the world. We plan to validate our method by comparing player experience when playing in a system with regular intervention verses revision.

References

- Guha, R. V., and Lenat, D. B. 1993. Cyc: A Midterm Report. *Readings in Knowledge Acquisition and Learning* 839–866.
- Li, B.; Thakkar, M.; Wang, Y.; and Riedl, M. O. 2014. Data-Driven Alibi Story Telling for Social Believability. In *Social Believability in Games*.
- Mawhorter, P.; Mateas, M.; Wardrip-Fruin, N.; and Jhala, A. 2014. Towards a Theory of Choice Poetics. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.
- Porteous, J.; Lindsay, A.; Read, J.; Truran, M.; and Cavazza, M. 2015. Automated Extension of Narrative Planning Domains with Antonymic Operators. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1547–1555. International Foundation for Autonomous Agents and Multiagent Systems.
- Riedl, M., and Bulitko, V. 2013. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine* 34(1):67–77.
- Riedl, M. O., and Young, R. M. 2006. From Linear Story Generation to Branching Story Graphs. *Computer Graphics and Applications* 26(3):23–31.
- Riedl, M. O.; Stern, A.; Dini, D. M.; and Alderman, J. M. 2008. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on Systems Science and Applications* 4(2):23–42.
- Riedl, M.; Saretto, C. J.; and Young, R. M. 2003. Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 741–748.
- Roberts, D. L., and Isbell, C. L. 2008. A Survey and Qualitative Analysis of Recent Advances in Drama Management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4(2):61–75.
- Robertson, J., and Young, R. M. 2014. Finding Schrödinger's Gun. In *Artificial Intelligence and Interactive Digital Entertainment*, 153–159.
- Sunshine-Hill, B., and Badler, N. I. 2010. Perceptually Realistic Behavior through Alibi Generation. In *Artificial Intelligence and Interactive Digital Entertainment*, 83–88.
- Young, R. M.; Riedl, M. O.; Branly, M.; Jhala, A.; Martin, R. J.; and Saretto, C. J. 2004. An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments. *Journal of Game Development* 1(1):51–70.