Intelligent Narrative Technologies: Papers from the 2012 AIIDE Workshop
AAAI Technical Report WS-12-14

A Formal Game for Eliciting Story Structure from Authors

David L. Roberts and **Andrew Hansen**
Computer Science Department
North Carolina State University
{*robertsd@csc,adhansen@*}*ncsu.edu*

Abstract

We address the problem of determining the structure of a set of plot points for an interactive narrative. To do so, we define a formal two-player game where a computer can play with an author to learn the structural representation of the story. This technique will allow for authors unfamiliar, or uncomfortable, with mathematical structures to create the inputs interactive narrative algorithms require. We include the underlying mathematical theory as a foundation of our approach, and characterize it's effectiveness through a series of simulation experiments. Results indicate there is promise in using formal games to aid in authoring interactive narrative structures.

Introduction

Despite notable efforts, many hurdles to broad adoption of interactive stories still remain. Among the challenges are those associated with authoring. Creating interactive narrative experiences requires story authors to provide computational systems with rich enough input to enable interactivity. In fact, Roberts argued that authoring interactive narratives presents seven distinct design challenges (2011). In this paper, we focus on providing computational support to ease one of these challenges: representing story.

We believe that authors are likely to find it easier to query (tacitly known) story structures than to specify them declaratively. In other words, an author is probably much better at looking at a particular story and determining if it is consistent with their desires for an interactive narrative than they are at specifying all the combinations of stories that work. We leverage this assumption to design a technique to enable authors to work with computers to encode complex interactive narrative structures in a non-declarative manner.

To do so, we formulate authoring story structure as a formal two-person game where a computer plays in collaboration with the author to identify (tacitly known) interactive story structure. This process can be used to learn the precedence constraints on plot elements and is suitable for story graphs (Weyhrauch 1997; Nelson and Mateas 2005; Nelson et al. 2006; Roberts et al. 2006) or certain types of plan-based representations (Young 1999; 2001; Riedl and Stern 2006; Magerko 2005). Our insight is that these story representations encode the ordering of plot elements using *partially ordered sets* or “posets,” which can be learned.

We aim for two things. First, our solution should enable authors to encode story structure in a “reasonable” amount of time. We intentionally leave this criteria vague because it is difficult to say how much effort authors will be willing to expend. Second, we want our solution to be simple and intuitive for authors to use without technical expertise. This will enable authors to focus on the creative aspects of making interactive narrative experiences.

For this paper, we have chosen to focus our efforts on simulation in order to characterize the feasibility of the technique. After formally defining the two-person game, we will present empirical results of simulations that indicate the number of game rounds necessary to learn story structures as a function of story size.

Defining Narrative Structures

The representation of narrative structure for interactive narrative has been well studied in the literature. Magerko (2007) discussed characteristics of a broad range of narrative representations, many of which were later surveyed by Roberts and Isbell (2008). In this paper, we focus on two representations: graphs of precedence constraints and plan-based representations. Here, we will first define each of these representations. We will then argue that there are some (restricted) plan-based representations that, from a structural or relational point of view, can be considered the same as plot graphs. In the next section we will define the mathematical background that underlies our approach.

Graph-based Representations

Beginning with Weyhrauch’s work on Search-based Drama Management (1997), a stream of research on Declarative Optimization-based Drama Management (DODM) has been conducted (Nelson and Mateas 2005; Nelson et al. 2006; Roberts et al. 2006; 2007; Roberts 2010; Sullivan, Chen, and Mateas 2008; 2009). In all of that work, story was represented using a *plot graph*.

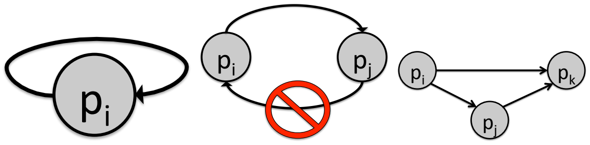
When using DODM, the author specifies the story using abstract plot points, each of which represents some event in the story progression. Additionally, some structure is imposed on these plot sequences. Specifically, the plot points are assigned ordering constraints, so that the drama manager only considers possible sequences that could actually happen; for example, a plot point OPEN SAFE can only happen after both the plot points DISCOVER SAFE and GET SAFE COMBO have already happened. The set of all sequences of plot points form the abstract plot space. Weyhrauch originally specified these ordering constraints by placing all the plot points in a directed acyclic graph (DAG), with the edges specifying a precedence relationship, and therefore an ordering over the plot elements (1997). In the fully-general version of this representation there can also be "or" constraints which state that any one of multiple plot points can satisfy the preconditions of another; however, for the purposes of this paper we have focused on AND constraints only. It is not clear how significant a restriction this is; however, it is the case that many narrative structures in the literature do not make use of or constraints.

Plan-based Representations

In other work, AI planning languages like ADL (Pednault 1987) or PDDL (McDermott 1998) were used to encode story (Magerko 2006; Young 1999; 2001; Riedl and Stern 2006; Barber and Kudenko 2007). In those settings, a structure that can be considered similar to a graph-based representation is used. From a high level, its components are similar: a set of plan-operators that represent narrative events and a set of preconditions and effects for each operator.

Consider the safe opening example from above. In the DAG-based plot graph representation, the plot event OPEN SAFE would have two incoming edges, one from the DISCOVER SAFE plot point and one from the GET SAFE COMBO plot point. These edges would indicate that both the discover and get combo events must occur before the safe is opened. To encode the same relationship in a plan-based representation, each of these plot events would be modeled as plan operators. The OPEN SAFE operator would have two preconditions: *safe located* and *combo discovered*. Additionally, the DISCOVER SAFE operator would have a *safe located* effect and the GET SAFE COMBO operator wold have a *combo discovered* effect. Thus, in order for the OPEN SAFE plot event to occur, the effects of both the GET SAFE COMBO and DISCOVER SAFE plot event are needed. Similar to the graph-based representation, there can be multiple operators with effects that satisfy the preconditions of another operator; however, for the purposes of this paper we focus on situations where only one operator's effects can satisfy (one or all of) the preconditions of another operator.

Certain planning domains can be considered the same as a precedence graph. To convert from a plan-based representation to a graph, we use the following construction. Assume we have a set of plan operators, effects, and preconditions. 1) Create a graph with vertices that represent plan operators; 2) For every ordered-pair of vertices $(v_i, v_j), j \neq i$, if one or more of the effects of v_i satisfy one or more of the preconditions of v_j, add (v_i, v_j) as an edge. To convert from a graph to a plan-based representation, we do the following. Assume we have a graph $G = (V, E)$. First, instantiate a plan operator for each $v_i \in V$. Next, for each $(v_i, v_j) \in E$ add $e_{i,j}$ to the effects of v_i and $p_{i,j}$ to the preconditions of v_j.



(a) Reflexive (b) Antisymmetric (c) Transitive

Figure 1: A graphical representation of the reflexive, antisymmetric, and transitive properties of a partial order.

Mathematical Underpinnings

Thus far we have introduced two common narrative representations from the literature on interactive narrative and argued how they can be considered the same for the purposes of learning structure. Here, we will depart from a discussion of narrative and introduce some formal definitions and notation that will be used in our technique.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set consisting of n elements and $R_P \subseteq P \times P$ be a binary relation defined over P. If R_P satisfies the following three criteria, it is said to be a partial order over P, and the ordered pair (P, R_P) is said to be a partially ordered set, or "poset".

Definition 1 (Reflexivity). A relation R_P on a set P is said to be *reflexive* if for every $p \in P, (p, p) \in R_P$.

Definition 2 (Antisymmetry). A relation R_P on a set P is said to be *antisymmetric* if for every $p_i, p_j \in P$ we have $(p_i, p_j) \in R_P$ and $(p_j, p_i) \in R_P$ implies that $p_i = p_j$.

Definition 3 (Transitivity). A relation R_P on a set P is said to be *transitive* if for every $p_i, p_j, p_k \in P, i \neq j \neq k$ we have $(p_i, p_j) \in R_P$ and $(p_j, p_k) \in R_P$ then $(p_i, p_k) \in R_P$.

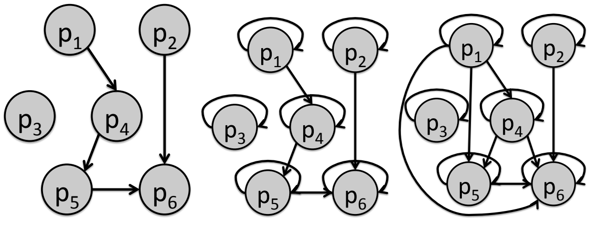
These three characteristics can be represented graphically as well. Let $G = (P, R_P)$ be a graph that consists of vertices representing the elements of P and edges connecting the vertices that are related by R_P. If R_P is a partial order over P then every vertex $p \in P$ will have a directed edge to itself (reflexive, Figure 1(a)), there will be no directed cycles (antisymmetric, Figure 1(b)), and there will be a direct edge between any two verticies for which there is also a directed path a length greater than two (transitive, Figure 1(c)).

Narrative Precedence is a Partial Order

In order to leverage our understanding of partial orders to learn narrative structure, we first must argue that the plot structure encoded by plot graphs are partial orders. To do so, we first define an "immediate predecessor relation."

Definition 4 (Immediate Predecessor). Given a set P and partial order R_P, A relation $I_P \subseteq P \times P$ is said to be an *immediate predecessor relation* if $\forall (p_i, p_j) \in R_P$ we have $(p_i, p_j) \in I_P$ only if $\nexists p_k \in P$ such that $(p_i, p_k) \in R_P$ and $(p_k, p_j) \in R_P$.

The immediate predecessor relation is not itself a partial order—it is not reflexive, antisymmetric, or transitive; however, as Preparata and Yeh point out, in a sense it "contains the same information as a partial order" (1973). They use the



(a) Predecessor (b) Reflexive (c) Partial Order

Figure 2: An immediate predecessor relation (a), the relation closed under reflexivity (b), and the associated partial order relation closed under transitivity as well (c). Note the topological orderings of vertices are the same in all cases.

following construction to convert an immediate predecessor relation to a partial order relation. Assume we begin with the immediate predecessor relation in Figure 2(a).

(1) Add "loops" to the immediate predecessor relation (close it under reflexivity). See Figure 2(b).

(2) For any three vertices $p_i, p_j, p_k \in P$, if the edges (p_i, p_j) and (p_j, p_k) are in the graph, then add the edge (p_i, p_k) to the graph. See Figure 2(c).

Note it is important to make sure that step (1) is performed before step (2) or the final relation may be incorrect. A similar process can be conducted in reverse to convert from a partial order to an immediate predecessor relation.

This process is important for the success of our approach for two reasons. First, the game we describe in the next section will result in a partial order. And second, the graph of precedence constraints (or equivalent plan-based representation) is an immediate successor relation. It is important to note that an immediate successor relation does not imply that a valid ordering of plot points must have immediately successive vertices adjacent. An immediate successor relation merely removes some of the redundant information in a partial order that is due to reflexivity and/or transitivity. In other words, if two plot events p_i and p_j are in the immediate successor relation, any number of plot events can occur between p_i and p_j when a story occurs provided their ordering is *topologically consistent* with the graph-based representation of the immediate successor relation.

The Game

We formalize the problem of learning the structure of an interactive narrative as a two-person collaborative game between Alice (the author) and Bob (the builder). We assume Alice knows what is acceptable, but can't easily write down all possibilities. Further, we assume that Bob can ask Alice if a story is consistent with her desires. Initially, Bob starts with a clean slate, having no belief about which story events can occur in what order. With each question Bob asks and answer he receives from Alice, Bob updates his belief about the story structure. Based on his current beliefs, Bob then asks another question. *Etc.*

We need three additional definitions:

Definition 5 (Linear Order). A relation R_P on a set P is said to be a *linear order* if $\nexists\, i, j$ where $i \neq j$ such that $(p_i, p_j) \notin R_P$ and $(p_j, p_i) \notin R_P$.

A linear order is a partial order where every element in the ground set is comparable. A particular realization of an interactive story results in a linear order of plot events.

Definition 6 (Linear Extension). Let (P, R_P) be a poset. $R^*_P \subset P \times P$ is a *linear extension* of R_P exactly when R^*_P is a linear order and $\forall (p_i, p_j) \in R_P$ we have $(p_i, p_j) \in R^*_P$.

In other words, a linear extension of a poset is a linear order that is topologically consistent with the partial order.

Definition 7 (Realizer). A set $\mathcal{R}$ of linear extensions of (P, R_P) is a *realizer* of (P, R_P) if $\cap \mathcal{R} = (P, R_P)$.

When the intersection of a set of linear extensions is the partial order, it is a realizer of the partial order.

The two-person game is played with a poset (P, R_P) defined over story events P. The partial order relation R_P that defines the structure of the story space is known tacitly to Alice (*i.e.*, she can query it, but can't generate it) but not to Bob. Both players know the ground set of elements P. The game proceeds in three rounds that are repeated until Bob knows he has learned the structure.

Round 1) Bob asks Alice if a particular ordering of plot events is consistent with her desires for the story. Formally, Bob presents Alice with the tuple $(E_i, \widetilde{R}^*_{E_i})$, where $E_i \subseteq P$, $|E_i| \leq n \leq |P|$, and $\widetilde{R}^*_{E_i} \subseteq E_i \times E_i$ is a linear order of the elements of E_i. n is a constant determined before the game begins.

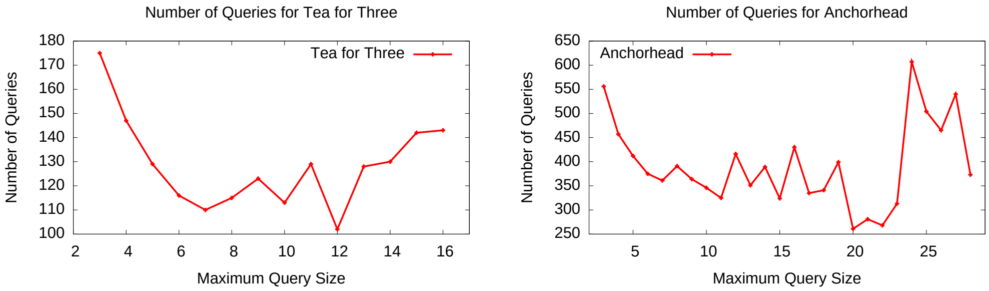
Round 2) Alice receives Bob's query and determines if the ordering is consistent with the story she has in mind. Formally, she responds by confirming that $(E_i, \widetilde{R}^*_{E_i})$ is a linear extension of the subposet of (P, R_P) induced by $\widetilde{R}^*_{E_i}$, or if not, by providing Bob with $(E_i, R^*_{E_i})$ which is a linear extension of the subposet of (P, R_P) induced by $R^*_{E_i}$.

Round 3) Bob updates his understanding of the partial order. If he believes he knows it, the game terminates. Otherwise, it returns to Round (1).

Bob seeks to reconstruct $(P, R_P) = \cup_i (E_i, \cap \mathcal{R}_{E_i})$.

For Bob's part in the game, he must select queries to send to Alice that result in as much information as possible with each round. In our approach, Bob has two phases to his query generation. In the first phase, Bob seeks to learn something about each plot event (element of P). In the second phase, Bob refines his knowledge about plot events for which he is uncertain after the first phase. Bob stores the partial order relation in an adjacency matrix M where each element can be one of four values: no information ($\emptyset$), definite edge (e), definite no edge (n), or possible edge ($?$).

To illustrate these rules, consider the example from Figure 2(a). Initially, M would be a 6×6 matrix with each entry equal to $\emptyset$. For the sake of this example, suppose Bob can construct queries of at most three plot points. He asks Alice, is p_4, p_1, p_3 a reasonable story? In this case, Alice would say no, and send Bob back a different ordering. Let's



(a) Tea for Three

(b) Anchorhead

Figure 3: Results for the *Tea for Three* and *Anchorhead* story domains. The x axis indicates the maximum query size (n) and the y axis indicates the number of queries needed to learn the story poset.

say p_1, p_3, p_4. Bob would learn six things from this information: 1) $M(4, 3) = n$, 2) $M(3, 1) = n$, 3) $M(4, 1) = n$, 4) $M(1, 3) = ?$, 5) $M(3, 4) = ?$, and 6) $M(1, 4) = ?$. Note that despite Alice's reordering of the plot elements, Bob has not yet learned that an edge exists between p_1 and p_4.

Once Bob has eliminated all $\emptyset$ elements from M, he enters the second phase of query construction. At this point, all of the entries in M are either n or $?$, so Bob seeks to eliminate the $?$s. Bob selects a pair of vertices for which the entry is $?$. To illustrate, let's take $M(1, 4)$ from the above example. Bob reverses the order and includes that in his query, optionally including additional vertices provided they are known not to be incident on p_1 or p_4. In this case, he would query Alice with p_4, p_1. If Alice returns yes, Bob knows $M(1, 4) = n$. If Alice returns p_1, p_4, Bob knows $M(1, 4) = e$. Bob will know the poset once M contains only n or e.

Results

We note that the problem of determining the poset has some known hardness properties. In particular, this game we propose is actually related to the problem of determining the "dimension of a poset" which is the minimum cardinality of a realizer (Trotter 2012). Ideally, Bob would finish the game in the number of rounds equal to the dimension of the poset; however, the problem of testing if a poset's dimension is less than or equal to t for $t \geq 3$ is known to be NP-complete (Yannakakis 1982). For perspective, consider that there are $O(n!)$ partial orders of a set with n elements, and that exhaustive search would be impossible for any reasonably-sized story domain. Thus, our goal is that the game ends quickly enough to be feasible for human authors to play.

Thus, our results here are intended to empirically characterize how realistic it is to use this method to determine story structure. Our goal is not a new theoretical result, but to illustrate how many queries Bob would have to ask Alice, the author, under various structural conditions. To accomplish this, we conducted a number of experiments on posets of varying size, as well as on two posets obtained from stories published in the literature on interactive narrative. In all cases, we simulated Alice using the computer. Human subjects experiments are a topic for future work.

In order to simulate Alice, we used the following process. When Alice receives a query from Bob that is not a linear extension, she rearranges Bob's query by examining each element of the query in order. Suppose Bob asks p_i, p_j, p_k, Alice will check connections from p_i to p_j, p_i to p_k, and p_j to p_k in that order. If Alice detects any pairwise problems, those two elements are swapped (*e.g.*, if p_i cannot precede p_k, then they are swapped and the new query is p_k, p_j, p_i). The entire process is repeated to make sure the swap didn't cause any problems. It is important to note that this process can inadvertently confuse Bob. Because Alice has swapped p_i and p_k, it may appear to Bob (in certain circumstances) that p_j also precedes p_i even if it is not the case. Fortunately, as we will see below, this is very rare. Further, in these rare cases what Bob learns will contain an extra precedence constraint between two plot events for which the order doesn't actually matter, not an incorrect order. Thus, the effect of failures on the resulting interactive story should be minimal.

Narrative Experiments

To begin with, we selected two stories from the literature on interactive narrative that have previously been encoded as plot points with precedence constraints. Specifically, we examined the interactive fiction *Tea for Three* first studied by Weyhrauch (1997), and the interactive fiction Anchorhead first studied by Nelson and Mateas (2005). The version of Tea for Three we used had 16 plot points with 11 precedence constraints. The version of Anchorhead we used had 28 plot points and 30 precedence constraints.

Figures 3(a) & 3(b) contain the results for Tea for Three and Anchorhead respectively. Of interest are two things. First is the overall number of queries required to learn the partial order over the plot points. In the worst case for Tea for Three 175 queries were needed; in the best case, only 102 queries. In the larger Anchorhead story, Bob required significantly more queries, ranging from 607 in the worst case to 261 in the best case. Second, the minimum number of queries occurred with query size somewhere in the middle in both cases, rather than at the extremes. Lastly, we point out that answering 261 queries, while potentially tedious, is a very reasonable task for an author to perform. As we will see below, the task is easier for shorter stories.

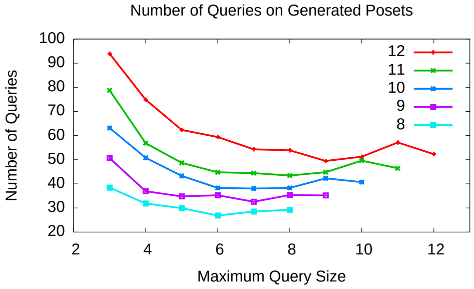


Figure 4: The average number of queries required to learn posets of size eight, nine, 10, 11, and 12.

Table 1: The query size that resulted in the fewest queries to learn the poset. The last column indicates the ratio of the query size to the size of the ground set of the poset.

Set Size	Query Size	Num Queries	Ratio
4	4	6.57	100.0
5	4	10.8	80.0
6	5	15.36	83.3
7	5	21.95	71.4
8	6	26.86	75.0
9	7	32.58	77.8
10	7	38.05	70.0
11	8	43.45	72.7
12	9	49.49	75.0
16	12	102	75.0
28	20	261	71.4

Table 2: The percentage of correctly learned partial order relations based on ground set size for all query sizes.

Ground Set Size	Accuracy
4	100.0
5	100.0
6	100.0
7	100.0
8	99.45
9	99.45
10	99.48
11	99.51
12	99.60

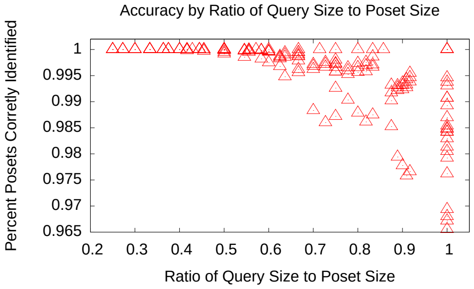


Figure 5: The percentage of posets correctly identified based on the ratio of the query size to the size of the poset. In general, the smaller the query size the fewer the errors.

Simulation Experiments

To further characterize effectiveness, as well as complexity, we conduced simulation experiments on generated posets. We ran experiments on ground sets with cardinality from four to 12. For posets of size four through seven, we exhaustively generated all possibilities. For posets of size eight or greater, we randomly generated 100,000 posets with varying numbers of edges. The data reported below is an average of performance across all posets. Additionally, we examined query sizes from three up to the full size of the ground set.

To begin with, we look at the number of queries required to learn the partial order relation. Figure 4 contains data from five of the generated data sets. Each curve represents the average number of queries required for queries of the given maximum length. Of interest in this plot is the distance between the curves and their relative shapes. First, the distance between the plots does appear to be growing as the size of the ground set increases. This trend is further supported when the data from the narratives is examined in comparison. While we do not have any complexity results for this approach yet, it would not be surprising to learn this algorithm is super-linear. Second, the curves are all decreasing and then increasing, indicating there is an ideal query size to learn poset structure quickly. To further examine this trend, we identified the query size associated with the lowest average number of queries. These data are presented in Table 1. Of particular interest here is the trend towards a ratio of query size to ground set size between 0.7 and 0.75. Note that the last two entries in this table are for the larger narrative posets discussed in the previous section, indicating that this trend seems to continue even as poset sizes increase.

Because our algorithm has no theoretical guarantees, we also examined the accuracy as a function of the percent of correctly learned partial order relations for different poset sizes. Our algorithm will fail in cases where Alice's choice of a new linear extension results in some ambiguity. We did identify cases where our algorithm failed to find the partial order; however, these cases were rare. Table 2 contains the average accuracy across all query sizes for each of the sets of posets. Of interest in this table are two things: the relative decline in the accuracy as the size of the poset grows and the fact that the average accuracy is above 99.45% in all cases. This second fact is encouraging; however, it does not reveal the entire story. Figure 5 contains additional accuracy data. It shows the accuracy as a function of the ratio of the query size to the size of the ground. What is interesting to note here is that as the ratio approaches 1.0, the accuracy tends to decline. In the "sweet" spot for efficiency around 0.75 discussed above, the majority of the results cluster above 99.5% accuracy. Again, this is very encouraging.

Future Work

This paper describes some preliminary work in developing a technique to aid authors in specifying narrative structure. There is still a lot of work that can be done in this area. Most notably, it is very important as we move forward to work with authors to ensure our technologies are appropriate for their needs. We plan to develop a graphical user interface for this technique that will enable us to share it with authors and collect human-subjects data.

In conducing human-subjects experiments, however, we will introduce some complexity that our technique does not currently address. First, we currently do not model disjunctive precedence constraints—something originally included in the formalism presented by Weyhrauch (1997). Further, our approach is not currently error-tolerant. If authors make mistakes, or if multiple authors give us conflicting linear extensions, our approach will surely fail. We hope to address these issues in future development of our technique.

Lastly, there are deep mathematical roots in our technique. As this is preliminary work, we have not made any effort to connect our approach to the body of literature on dimension theory or partial orders, especially complexity results. We plan to do so in the future.

Conclusion

In this paper, we have defined a formal two-person game for helping authors specify the structural relations between plot events in interactive narratives. We have defined the mathematical theories behind our approach, and argued for the applicability of it to both graph-based and plan-based representations of interactive narratives. We have also presented a set of simulation experiments that demonstrate the accuracy and feasibility of our approach.

Based on our results, we conclude that there is promise in our technique. Even in cases of relatively large sets of plot events, authors will need only to answer a few hundred queries from our algorithm which can be done at their leisure. We further conclude that despite our algorithm not guaranteeing to find the exact partial order, it is extremely accurate and fails in rare cases. Therefore, we believe that with some refinement and interface development, this may prove to be a valuable authoring aid for those seeking to create interactive narratives.

References

Barber, H., and Kudenko, D. 2007. Dynamic Generation of Dilemma-based Interactive Narratives. In *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment conference (AIIDE07)*.

Magerko, B. 2005. Story Representation and Interactive Drama. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05)*.

Magerko, B. 2006. *Player Modeling in the Interactive Drama Architecture*. Ph.D. Dissertation, Department of Computer Science and Engineering, University of Michigan.

Magerko, B. 2007. A comparative analysis of story representations for interactive narrative systems. In *Proceedings of the 3rd Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE 07)*.

McDermott, D. 1998. The planning domain definition language manual. Technical Report 1165 CVC 98-003, Yale Computer Science.

Nelson, M. J., and Mateas, M. 2005. Search-Based Drama Management in the Interactive Fiction Anchorhead. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05)*.

Nelson, M. J.; Roberts, D. L.; Isbell, C. L.; and Mateas, M. 2006. Reinforcement Learning for Declarative Optimization-Based Drama Management. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*.

Pednault, E. 1987. Formulating multi-agent dynamic-world problems in the classical planning framework. In Georgeff, M., and Lansky, A., eds., *Reasoning about actions and plans*. Morgan Kaufmann. 47–82.

Preparata, F. P., and Yeh, R. T. 1973. *Introduction to Discrete Structures for Computer Science and Engineering*. Addison-Wesley.

Riedl, M. O., and Stern, A. 2006. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. In *In Proceedings of the Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06)*.

Roberts, D. L., and Isbell, C. L. 2008. A Survey and Qualitative Analysis of Recent Advances in Drama Management. *Int. Transactions on Systems Science and Applications, Issue on Agent Based Systems for Human Learning* 3(1):61–75.

Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; Mateas, M.; and Littman, M. L. 2006. Targeting Specific Distributions of Trajectories in MDPs. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI06)*.

Roberts, D. L.; Bhat, S.; Clair, K. S.; and Isbell, C. L. 2007. Authorial Idioms for Target Distributions in TTD-MDPs. In *Proc. of the 22nd Conf. on Artificial Intelligence (AAAI07)*.

Roberts, D. L. 2010. *Computational Approaches for Reasoning About and Shaping Player Experiences in Interactive Narratives*. Ph.D. Dissertation, Georgia Tech.

Roberts, D. L. 2011. Seven design challenges for fully-realized experience management. In *Proceedings of the Fourth Workshop on Intelligent Narrative Technologies (INT4)*.

Sullivan, A.; Chen, S.; and Mateas, M. 2008. Integrating Drama Management into an Adventure Game. In *Proc. of the Fourth Conf. on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 08)*. AAAI Press.

Sullivan, A.; Chen, S.; and Mateas, M. 2009. From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience. In *Proc. of the 2009 AAAI Spring Symposium on Intelligent Narrative Technologies II*.

Trotter, W. T. 2012. Personal email correspondence.

Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, School of Computer Science.

Yannakakis, M. 1982. On the complexity of the partial order dimension problem. *SIAM Journal on Algebraic and Discrete Methods* 3:351–358.

Young, R. M. 1999. Notes on the Use of Plan Structures in the Creation of Interactive Plot. In *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence*.

Young, R. M. 2001. An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment. In *Working Notes of the AAAI Spring Symp. on AI and Interactive Entertainment*.