Applying Planning to Interactive Storytelling: Narrative Control using State Constraints

JULIE PORTEOUS, MARC CAVAZZA and FRED CHARLES
Teesside University

We have seen ten years of the application of AI Planning to the problem of narrative generation in Interactive Storytelling (IS). In that time planning has emerged as the dominant technology and has featured in a number of prototype systems. Nevertheless key issues remain, such as how best to control the shape of the narrative that is generated (e.g. by using *narrative control knowledge*, that is, knowledge about narrative features that enhance user experience) and also how best to provide support for real-time interactive performance in order to scale up to more realistic sized systems. Recent progress in planning technology has opened up new avenues for IS and we have developed a novel approach to narrative generation that builds on this. Our approach is to specify narrative control knowledge for a given story world using state trajectory constraints and then to treat these state constraints as *landmarks* and to use them to decompose narrative generation in order to address scalability issues and the goal of real-time performance in larger story domains. This approach to narrative generation is fully implemented in an Interactive Narrative based on the Merchant of Venice. The contribution of the work lies both in our novel use of state constraints to specify narrative control knowledge for Interactive Storytelling and also our development of an approach to narrative generation that exploits such constraints. In the paper we show how the use of state constraints can provide a unified perspective on important problems faced in IS.

Categories and Subject Descriptors: I.2.8 [**Problem Solving, Control Methods, and Search**]: Plan execution, formation, and generation

General Terms: Algorithms

Additional Key Words and Phrases: Interactive Storytelling, Narrative Modelling, Planning, Agents in games and virtual environments

1. INTRODUCTION

AI Planning technologies have traditionally been associated with problem-solving, as well as tackling real-world complex situations such as space operations [Chien et al. 2010] and forest fire-fighting [Castillo et al. 2006]. With a few exceptions, such as the generation of multimodal presentations [André and Rist 1993], multimedia and visual interfaces have not been perceived as a major application domain for planning. The development of new interactive media such as computer games is progressively changing this situation: since developers identified a commonality of AI problems between robotics and computer games, the latter have become a test

Author's address: Teesside University, Middlesbrough, Tees Valley, TS1 3BA, UK
This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

bed for AI technologies [Laird 2002]. In particular, Interactive Storytelling (IS), one specific topic of Interactive Entertainment research has embraced planning as its core technology.

The aim for IS, is to develop interactive media where the presentation of a narrative, and its evolution, can be influenced in real-time by a user. A central part of this endeavour is the process of narrative generation. The generation of narratives can be seen as a knowledge-based planning problem with associated issues concerning representation and real-time performance. Planning was initially proposed for IS in [Young 2000], and since then it has emerged as the dominant technology for IS prototype systems. A number of factors have contributed to this: one has its roots in the adoption of planning as a technology for virtual agents which was later transferred to reasoning about virtual actors [Geib 1994]; another is the apparent natural fit between narratives and plans which enables narratives to be naturally modelled as a sequence of actions; and another is that plans embed key features including causality among story events which have been shown to be an important factor in user experience [Trabasso and van den Broek 1985].

Over the last ten years planning has featured in a number of IS prototypes such as [Young 2000; Cavazza et al. 2002b; Riedl and Young 2005; Karlsson et al. 2006; Bae and Young 2008]. The tendency has been to use older planning approaches, such as partial order planning [Weld 1994], which has been adapted in a number of ways, for example, to emotions [Aylett et al. 2006], and narrative adaptations [Riedl and Stern 2006]; and Hierarchical Task Network (HTN) planning [Nau et al. 2003], which has been adapted to handle user interaction [Cavazza et al. 2002b; Hoang et al. 2005; Kelly et al. 2007]. Exceptions to this include planning in the style of HSP [Bonet and Geffner 1999], for narrative generation [Pizzi et al. 2007].

This wide scale use of planning in IS has been largely empirical and has led to a number of successful prototypes which have uncovered practical problems, two of which we have explored in this work. One problem is how to *represent* narrative control knowledge – knowledge about such things as narrative pacing, the creation and release of tension and how to shape a generated narrative arc so that it is in line with desired aesthetic principles. The other related problem is how best to *use* this narrative control knowledge to control the process of narrative generation. To tackle such problems, the dominant approach taken in IS research has been to provide further empirical solutions, such as the addition of control modules on top of the planning system. However, our view is that many of the problems faced by IS when using planning for narrative generation can themselves be translated back into planning problems. In particular, the problems of representing and using narrative control knowledge can be seen in terms of the problems of domain representation and planning to satisfy hard and soft constraints on plan trajectories [Gerevini and Long 2005]. In our work we have explored these problems and part of our contribution is demonstrating how state trajectory constraints can be used to specify narrative control knowledge and how this can be used in the process of narrative generation within IS systems.

In earlier work we explored the role of state trajectory constraints to specify narrative control knowledge and their role in the generation of narratives [Porteous et al. 2010]. In this paper we expand considerably on that earlier work in a number of ways. We motivate our approach through an analysis of the requirements of planning systems for narrative generation in IS. We include detailed consideration of the process of analysis and representation of story worlds as planning domains and in particular the representation of narrative control knowledge using state trajectory constraints. In this paper, we also include detailed discussion of the narrative generation algorithm, the process of constraint selection that contributes to the generation of different narrative variants along with detailed discussion of sample narratives that illustrate this process.

The remainder of the paper is organised as follows: we start in section 2 with discussion of the requirements of an IS planner in order to motivate our approach. In section 3, we discuss the representational aspects of our approach: the representation of story worlds as planning domains in general and the use of state constraints in particular. This is followed in section 4 with discussion of the decomposition planning approach for narrative generation that we have developed. In section 5, we present the results of experiments within an Interactive Narrative system that we have developed. In section 6, we discuss closely related work and we finish in section 7 with conclusions and discussion of future work.

2. REQUIREMENTS FOR AI PLANNING IN IS

IS is a very different application domain to those which have traditionally featured in planning such as the benchmark domains used in the series of ICAPS International Planning Competitions or fielded applications such as planning for Mars Rovers [Chien et al. 2010]. Not surprisingly, key requirements for an IS planner differ from those for planners that perform well in these very different domains. Thus we require the planner to be able: to reason about suitably represented narrative knowledge (knowledge-based); to exert real-time control over plan generation; to use plan quality criteria other than optimality; and to support interactivity. To motivate our approach, we will consider each of these requirements.

2.1 Knowledge-Based Planning for IS

An IS planner must be able to reason about suitably represented narrative knowledge, in other words, it must be knowledge-based. In AI, knowledge-based planning describes approaches where available knowledge of a domain is utilised to help efficiently and effectively solve planning problems.

One well known approach is HTN planning [Nau et al. 2003] which has been popular in IS because narrative knowledge can be encoded in the network decompositions [Hoang et al. 2005; Kelly et al. 2007]. However, control knowledge can be difficult to specify and maintain precisely because it is embedded [Cavazza et al. 2002a]. Indeed, we would argue for declarative specification of control knowledge on the basis of the well known principle that advises us to "separate functionality from implementation" [Pressman 2009] because such a declarative specification can be argued to have advantages in terms of ease of understanding, conciseness of expression, modularity and ease of validation.

An alternate approach to knowledge-based planning is to augment the domain model with a declarative specification of constraints on the properties of solution plans. This approach is similar to the specification and use of search control knowledge seen in TLplan and TALPlanner [Bacchus and Kabanza 2000; Kvarnström

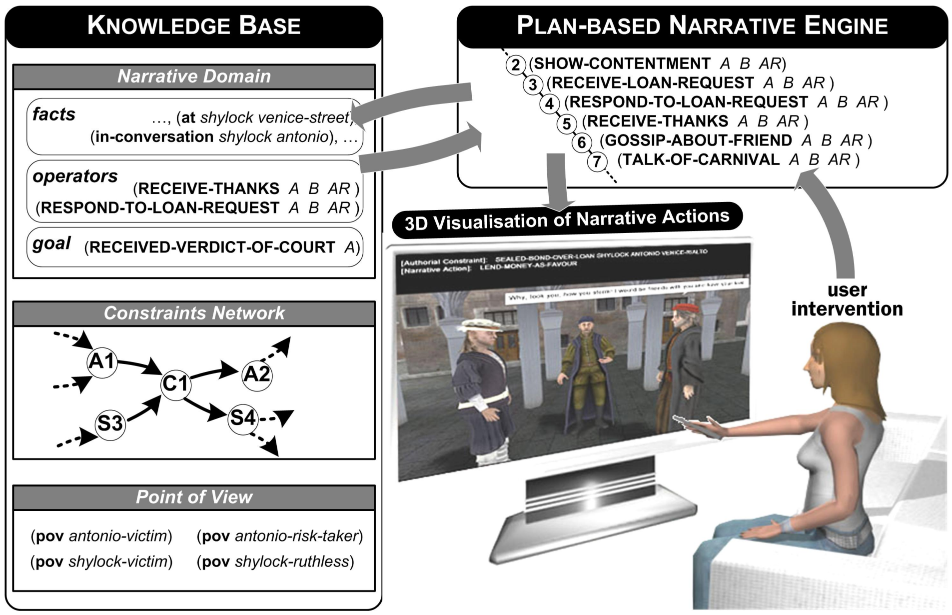

Fig. 1. Architecture of a typical IS system. Narrative actions are passed sequentially from the generation engine to the visualisation module where they activate character animations. Results of user interaction are fed back to the generator, triggering updating and re-planning as required.

and Doherty 2001]. Such constraints can be strong (they must be achieved) or soft (they are desired) and the task for the planner is to satisfy the strong constraints and as many of the soft constraints as it can. One way to represent the information about desired plan properties is to use the extensions provided in the representation language PDDL3.0 [Gerevini and Long 2005]. We observed that a sub-set of the PDDL3.0 state trajectory constraints provide a neat mechanism for the representation of narrative knowledge for IS applications since we can use constraints to represent situations that are important to the development of the narrative. Our approach to narrative generation is built on these observations and uses state constraints to specify narrative control knowledge (see section 4).

2.2 Real time planning for IS

An IS planner should be able to perform in real-time. This is because story generation operates by nature in a dynamic environment, where the story world is modified by user interventions, or, depending on the IS paradigm implemented, the interaction between the various autonomous characters. For our purposes, real-time is based on the response time to user interaction as discussed in [Pizzi et al. 2007]. For practical purposes this means that the IS engine should respond within 1500ms.

Figure 1 shows the architecture of a typical IS system. It features a narrative engine which continuously generates and passes the next narrative action to a visualisation engine which manages presentation of the action to the user (the current state-of-the-art imposes that these visual contents are computer generated). In the absence of user intervention the sequence of narrative actions are generated around a baseline plot with variations resulting from the situation of virtual world objects and characters. With user intervention the narrative situation can change and this will be fed back to the generation engine with subsequent updating of the current world state. In this way, user intervention can result in different evolutions of the narrative. One question is how to ensure real-time performance: whether to use real-time search techniques as in [Pizzi et al. 2007]) or simply re-plan in response to user intervention. We have adopted the latter approach and the results of a series of experiments show that our narrative generator is able to perform within the required time frame (performance of our system is discussed further in section 5.2).

2.3 Plan criteria for IS

Development of early classical planners, such as STRIPS [Fikes and Nilsson 1971], was driven by a desire to generate optimal plans for problem domains such as the blocks world and logistics. Over time the optimality criteria was relaxed for some planners, such as FF [Hoffmann and Nebel 2001], in a trade-off between quality and efficiency and the object became finding a solution that satisfied some set of adequacy criteria: any plan will do but more adequate plans are preferred.

However in IS, plan optimality is not necessary or even desirable. It is the combination of the plan trajectory[1] and sequence of narrative actions that accounts for the narrative experience, and the semantics of the intervening actions play a dominant role. Indeed, the criteria for plans in IS concerns the trajectory of the plan and how far that fits with the trajectory shape of desired narrative plans for that domain – where narrative plans are desired if they conform to narrative control knowledge. Hence, we have focussed on providing mechanisms to specify such criteria and enable the narrative generator to use this to shape the narrative trajectory. Since our approach is to specify narrative control knowledge using constraints, assessment of the quality of generated narrative plans is therefore assessed in terms of how far they satisfy the selected constraints.

2.4 Interactive Planning in IS

A key requirement for an IS narrative generation engine is that it should support interactivity in order to allow the user to influence the presentation of a narrative and its evolution, as shown in figure 1. The interaction can take many forms (for instance, a user might play the role of a character [Cavazza et al. 2009] or might physically interact with objects in the story world [Cavazza et al. 2002b]). The result is that the output narrative must reflect user changes to the story world.

One of the strengths of a plan-based approach to narrative generation is that its generativity supports interactivity since it provides the ability to re-plan. In addition the development of our approach to narrative generation (to be discussed in section 4) was driven by the need to support interactivity and real-time performance. Our approach is to decompose the problem of narrative generation into a sequence of sub-problems which then enables a planner to tackle them in order and generate the plan for the next sub-problem as needed. This means that when user

[1] By plan (narrative) trajectory we refer to the shape of the narrative in terms of the states (story world situations) that it passes through.

interaction causes changes to the story world only the next sub-problem need be re-planned rather than the whole plan, with consequent planning time reduction.

3. REPRESENTING STORY WORLDS AS PLANNING DOMAINS

In our work, we have followed a popular approach in IS in which the modelling of a baseline classical plot is a first step towards interactive narrative. The underlying principle being as follows: by modelling all the baseline actions of a story world for the default linear narrative as planning operators, it is possible to produce multiple variants of a narrative when the planning domain changes, either initially or at run-time (this approach is not unlike the Remediation hypothesis [Bolter and Grusin 1999]). One condition for this approach to be successful is that the representation of default actions (those from the original linear story) must be made generic enough to represent more than the default context: for instance, an action corresponding to bearing bad news should be described in sufficiently fine-grained detail to cover different types of news and different sorts of pre- and post-conditions. An additional benefit of this initial modelling approach is that it allows basic testing of the overall system, through its ability to re-generate the default story in the absence of changing initial conditions (or lack of dynamic changes). This hypothesis has been illustrated in a number of interactive narrative systems including the Madame Bovary system of [Cavazza et al. 2009].

Once a baseline plot has been modelled using a combinatorial formalism, such as PDDL3.0, it then becomes possible to transform the linear plot into a non-linear one. We can do this by identifying different standpoints and ranges of actions for different characters and then modelling these character behaviours (note that the perspective is at the narrative rather than the character level so narratives are generated to satisfy individual characters' goals and constraints within the context of the overall plot). As soon as we do this we are departing from the baseline plot and entering the speculative realm of narrative generation.

One way to manage these character behaviours is via character Point of View (PoV), a concept we introduced to describe a character's perspective (or a particular standpoint) on an overall plot through which a story can be told [Porteous et al. 2010]. PoV is an important concept which can help preserve genre consistency by: generating narrative variants that don't revolutionise the story; providing a means to study the nature of the plot. This fits well with the study of classics such as The Merchant of Venice or Madame Bovary [Cavazza et al. 2009]. Hence, we use PoV as a test case for the use of constraints to represent IS narrative control knowledge since it constitutes a representative IS problem.

In the remainder of the paper we will use examples from an interactive narrative that we have developed based on Shakespeare's Merchant of Venice, a play which rests on the opposition between two central characters: Antonio, a wealthy Christian merchant and Shylock, a Jewish moneylender, against the backdrop of XVIth century Venice, which is characterised by trade and prosperity, but also by racial and religious discrimination[2].

[2]We were inspired by the recent screen adaptation of the play [Radford 2004]. Controversy has surrounded the play throughout its history but modern interpretations, culminating in Radford's adaptation, have offered a more sympathetic treatment of Shylock.

Narrative Condition	Predicate
Shylock and Antonio have sealed a bond over the loan of 3,000 ducats	*(sealed-bond-over-loan shylock antonio)*
Shylock has responded to the news that his daughter Jessica has eloped	*(responded-to-news-of-elopement shylock)*
At the end of the trial, Antonio and Shylock have received the verdict of the court	*(received-verdict-of-court antonio shylock)*

Fig. 2. Example predicates that represent narrative conditions on Merchant of Venice characters.

To illustrate the concept of PoV, following the analysis of the Merchant of Venice in [Hinely 1980], we can identify the standpoint and values of Antonio and Shylock and the range of actions permissible within these. Shylock sees himself as a victim of discrimination and later as a victim of Antonio's refusal to abide by the (contractual) law he wants to see enforced. Within this standpoint, his behaviour could range from revenge to conciliation. Antonio's values may not be diametrically opposed to Shylock's [Hinely 1980] but his standpoint is that of the ruling class, despite the contradictions that follow such as his need for Shylock's assistance. Within this standpoint his behaviour can range from carelessness (mistreating Shylock, accepting the bond) to conciliation. This behaviour is equivalent to selecting actions from relevant semantic categories throughout a portion of the narrative (plan).

The Merchant of Venice domain model that we have developed concentrates on a central element of the play, a bond between Antonio and Shylock, by which the latter agrees to lend 3000 ducats to the former without interest, but if he fails to repay the loan then the penalty would be "one pound of (his) flesh". Following Hinely's analysis of the play [Hinely 1980], we refer to this as the "pound-of-flesh" sub-plot. The domain model consists of roughly 200 narrative actions which are grounded prior to narrative generation (by making legal substitutions of constants for variables) which results in a set of approximately 1500 ground actions. There are roughly 150 constraints in the domain model of which an average of 15 are selected for use in narrative generation. In the rest of this section we discuss the development of the model. Note that we undertook the task of creating this domain model ourselves but our comments about the process of representing story worlds as planning domains apply to authors and story creators in general.

3.1 Story World Predicates

Given our approach, modelling a story world as a planning domain starts from analysis of a baseline plot from which a model of a narrative domain can be constructed based upon the actions and attributes of the main characters. Characters' attributes, such as their location, activities and allegiance can be represented as the predicates of the planning domain. These predicates can be thought of as describing the condition of characters and story world states. Once they have been identified, the predicates can then be used to specify goals for the narrative. The main actions, those that modify characters' attributes, can be represented as planning operators.

As an example consider the following Merchant of Venice predicates: the signing of a bond between Antonio and Shylock, by which the latter agrees to lend 3000 ducats to the former without interest; the response by Shylock to the news that

Standpoint (PoV)	Predicate
Shylock sees himself as a victim of discrimination and later as a victim of Antonio's refusal to abide by the (contractual) law he wants to see enforced.	*(point-of-view shylock-victim)*
Shylock has a history of cruel treatment by Antonio and is intent on revenge	*(point-of-view shylock-ruthless)*
Antonio sees himself as a loyal friend who is a victim of Shylocks insistence on enforcing the contractual law over the defaulted loan	*(point-of-view antonio-victim)*
Antonio sees himself as a member of the ruling class, pleasure seeking and unconcerned with risk	*(point-of-view antonio-risk-taker)*

Fig. 3. Example of predicates that represent different character PoV for the Merchant of Venice.

his daughter has eloped; and the receipt, by Antonio and Shylock, of the verdict of the court over the case of Antonio's default on the loan. These are represented as predicates in the domain model as shown in figure 2.

Also represented as predicates in the domain model are the different character PoV, or standpoints. As an illustration, the different PoV for Shylock and Antonio that were discussed earlier are shown represented as predicates in figure 3.

3.2 Story World Operators

The representation of characters' PoV aims at producing proper story variants rather than simply different presentations of the same story. Our working hypothesis is that a given narrative action (such as a contract, a betrayal, a challenge, and so on ...) can be represented differently depending on the perspective of each character taking part in that action. In other words, a PoV consists of a character's representation defined from the perspective of the overall plot, not just of the character's role independent of any other. The PoV also implements the naive concept of a given character's standpoint on a set of events, although in an a priori rather than a posteriori fashion. This is achieved by defining different representations for the same narrative action depending on the PoV, which in turn requires, for instance, different sets of pre (resp. -post) conditions. With such representations, narrative generation will adopt a given character's PoV for the selection of the actual narrative action, thus resulting in story variants according to the PoV. In addition, these variants will respond differently to real-time modifications of the narrative domains such as those introduced by user interaction.

As an illustration consider an asymmetric action that features in our Merchant of Venice Interactive Narrative: the loan of three thousand ducats, by Shylock to Antonio. The two characters have different roles in this transaction: Shylock is the lender of the money and Antonio is the borrower. When the different PoV are taken into account this results in four ways of representing this action as shown in figure 4. These actions all share one effect: that Antonio and Shylock have sealed a bond over the loan of money but they differ with respect to their other effects and any necessary enabling conditions, such as PoV. For example, when the PoV is Antonio as risk taker then he pays no heed to the consequences but when his PoV is victim then an effect of sealing the bond is that he is aware of the risks.

PoV	Action
(pov antonio-risk-taker)	Antonio, carefree risk taker, borrows money with no heed to the consequences
(pov antonio-victim)	Antonio, a loyal friend, borrows money from Shylock, fully aware of the risks
(pov shylock-victim)	Shylock, a patient victim, extends a favour to Antonio by lending him money
(pov shylock-ruthless)	Shylock, intent on revenge, lends money to Antonio anticipating the forfeit

Fig. 4. Different asymetric actions that achieve the same goal depending on character PoV.

Figure 5 shows these actions represented using PDDL3.0. They differ with respect to their pre-conditions which include information such as PoV and some effects (post-conditions) but they share the effect of sealing the bond over the loan.

```
(:action borrow-money-confident-repay antonio shylock venice-rialto) ...
    :precondition    (and (pov antonio-risk-taker) ...)
    :effect          (and (sealed-bond-over-loan shylock antonio)
                     (unconcerned-over-forfeit antonio) ... ))

(:action borrow-money-wary-of-risk antonio shylock venice-rialto) ...
    :precondition    (and (pov antonio-victim) ...)
    :effect          (and (sealed-bond-over-loan shylock antonio)
                     (concerned-over-forfeit antonio) ...))

(lend-money-as-favour shylock antonio venice-rialto) ...
    :precondition    (and (pov shylock-victim) ... )
    :effect          (and (sealed-bond-over-loan shylock antonio) ...))

(:action (lend-money-intent-on-revenge shylock antonio venice-rialto) ...
    :precondition    (and (pov shylock-ruthless) ...)
    :effect          (and (sealed-bond-over-loan shylock antonio) ...))
```

Fig. 5. Example representations of asymetric actions that depend on different character PoV.

3.3 Representing Narrative Control Knowledge as Constraints

The approach we have taken is to use PDDL3.0 constraints [Gerevini and Long 2005] to represent narrative knowledge. The constraints can be viewed as key components of the plot structure, representing desirable conditions that could feature in a narrative variant: desirable in the sense that making them true will cause the selection of operators that enrich the narrative, increase pace, heighten suspense and so on. Hence the constraint information constitutes a meta-level of representation for the plot: in terms of contents it can be assimilated to invariants which have to hold true for all well-formed narratives (meaning consistent with the baseline plot, despite constituting a variant). This also provides a way to address a recurrent problem in IS which is to control the level of variation around the baseline.

Constraints are identified through analysis of a baseline plot by the domain author (or content creator), in a similar manner to the narrative actions, only at a more

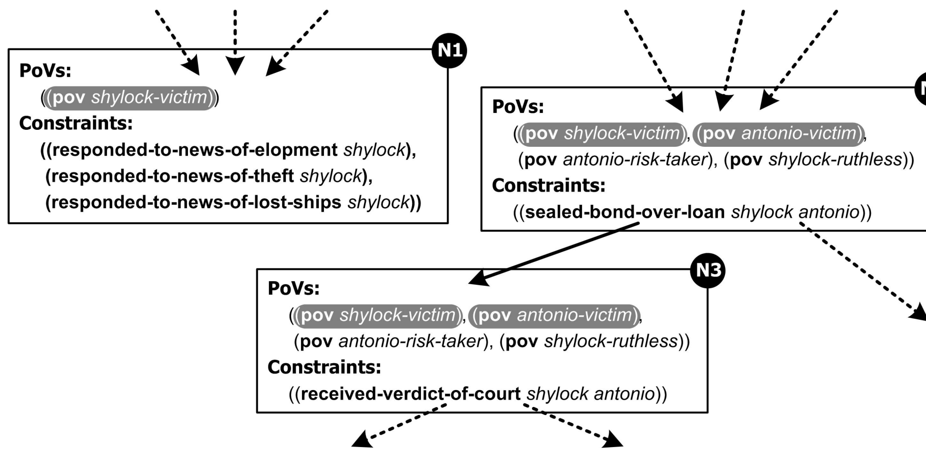

Fig. 6. A fragment of the Constraints Graph for our Merchant of Venice Interactive Narrative. The nodes show constrained facts and relevant PoV information. The arrows indicate temporal ordering: node N2 is ordered before N3 but N1 is unordered with respect to the other nodes.

abstract and declarative level. Any story world predicates that are determined to be important – in the sense that making them true will cause the selection of operators that enrich the narrative, increase pace, heighten suspense, enhance user experience and so on – are included as constraints within the domain model (for example, *(sealed-bond-over-loan shylock antonio)* that was discussed earlier).

We have used the PDDL3.0 modal operators *sometime-before* and *sometime* to represent IS knowledge since they enable us to represent important conditions that may feature in a narrative along with any important temporal orders. In particular, if a constrained predicate can occur at any time in the narrative then we represent it using the *sometime* modal operator but if relative order is important then we represent it using the *sometime-before* modal operator. In addition, we also include information about which PoV a constraint is relevant to, since it may not be relevant to all PoV's. For example, it makes sense in the context of the pound-of-flesh subplot for the bond to have been sealed between Shylock and Antonio before they have received the verdict of the court. If we also suppose that this constraint is relevant for PoV *antonio-victim* or *shylock-victim* then it could be specified as:

(sometime-before (and (or (pov antonio-victim) (pov shylock-victim)) (received-verdict-of-court antonio shylock)) (and (or (pov antonio-victim) (pov shylock-victim)) (sealed-bond-over-loan shylock antonio)))

For the constraint that Shylock has responded to the news of his daughter's elopement there is no such restriction on ordering so this could be specified using the *sometime* modal operator, with accompanying information that this is relevant only in the context of the PoV *shylock-victim*.

In some situations we may also wish to include a number of facts at a constraint and leave it to the control mechanism to select one for a particular narrative variant at run-time (precisely how the fact is selected is discussed in section 4.1). This is useful since it allows for variation in output narratives depending on the constrained fact which is selected. As an example, suppose that we wish to specify a choice of facts at a constraint which are unordered with respect to any other constraints and that are relevant for PoV *shylock-victim*. If the facts are *(responded-to-news-of-elopement shylock)*, *(responded-to-news-of-elopement shylock)* and *(responded-to-news-of-elopement shylock)* then they could be specified as:

(sometime (or (and (pov shylock-victim) (responded-to-news-of-elopement shylock)) (and (pov shylock-victim) (responded-to-news-of-elopement shylock)) (and (pov shylock-victim) (responded-to-news-of-elopement shylock)))

These constraints and the order between them form a graph, a fragment of which is represented in figure 6. It shows the constrained facts, their relative ordering and PoV information for constraint selection. The constrained fact at N1, has a single associated PoV *shylock-victim* whereas nodes N2 and N3 are also associated with *antonio-victim*. This information is used by the control mechanism to determine whether the node is relevant to the current narrative variant. Since node N1 has the single PoV *shylock-victim* it is only relevant when the narrative is told from this PoV, whereas nodes N2 and N3 are relevant for variants told from either PoV.

4. A DECOMPOSITION PLANNING APPROACH TO NARRATIVE GENERATION

A number of approaches have been proposed for generating plans in the presence of strong and soft constraints, including MIPS-xxl [Edelkamp et al. 2006] and SG-Plan5 [Hsu et al. 2006]. Their focus has been on generating plans that satisfy sets of preferences and given that this is computationally expensive, the practice has been to seek to generate reasonably preferred plans. Rather than adopting one of these approaches we have placed our emphasis on planning to satisfy constraints representing narrative control knowledge as well as supporting real-time performance within an interactive system. A key factor for this is the role of user interaction in IS systems and the high likelihood of the need to re-plan.

Hence the approach we have taken is to use the constraints to decompose the problem of generating a narrative into a sequence of sub-problems, where each sub-problem has a constraint, selected from the constraints graph, as its goal. A strength of this decomposition approach for IS is that since the plan is generated incrementally, effort is not wasted when user interaction forces the system to re-plan. When our narrative generation engine is integrated within an IS system (as shown in figure 1), narratives are generated for each sub-problem in turn, operators are sent one at a time to the visualisation engine and the generator waits for a response from the visualisation engine before continuing. Hence, a complete narrative (plan) is not output in the traditional sense although it can be constructed by composing the narratives (plans) for each of the individual sub-problems.

4.1 Constraint Selection for Narrative Variants

An important part of our decomposition approach is the selection of constraints for the generation of different narrative variants. This selection is handled by the function *select_constraint(C, PoV)* which is shown in line 3 of algorithm 1. It takes as input the set of constraints, C, from the domain model which form a graph $G = (N, E)$ and where nodes, N, are obtained from the arguments of the *sometime* and *sometime-before* modal operators and the edges, E, are pairs (a, b) whose orders correspond to those in the *sometime-before* modal operators. The nodes, $n \in N$,

Algorithm 1: A Decomposition Planning approach to Narrative Generation

```
    input  : F, I, G, O, PoV, C
1   repeat
2       c = select_constraint(C, PoV) ;
3       call planner with: O, I, c  → P';
4       if planner found a solution P' then
5           for p ∈ P' do
6               if visualise(p) → replan then
7                   update I, G, PoV as required;
8                   if c is constraint for current PoV then
9                       goto line 3;
10                  else
11                      break ;
12      mark c as visited;
13  until all c ∈ C have been visited;
14  call planner with: O, I and conjunctive goal G → P' ;
15  if planner found a solution P' then
16      if visualise(P') → replan then
17          update I, G, PoV as required;
18          goto line 14;
```

are sets containing either single facts or disjunctions of facts from which a single fact can be chosen (as discussed below). It also takes as input, PoV, a singleton set containing the currently adopted PoV fact. The output of the function is a single constrained fact, c, which forms the goal of the next decomposed sub-problem.

The function considers only those nodes in C that have not been visited. It starts by selecting the set of nodes that are: (i) the *earliest* unvisited nodes in the graph, that is nodes, $n \in N$, with no direct predecessor nodes $(n', n) \in E$; and (ii) where the PoV fact is *relevant* to the currently adopted PoV (this is true if $PoV \cap n \neq \emptyset$). From this set, a single node is selected: in our experiments we have selected the node arbitrarily but this could be extended to monitor narrative variants over time and select constraints to ensure varied presentation to users. If the node is a singleton then that fact is returned, otherwise, it contains a disjunction of facts (as in the example given at the end of section 3.3) from which one fact is selected: either arbitrarily or using a user model in order to promote variation.

4.2 Narrative Generation using Decomposition Planning

The narrative generation algorithm is shown in algorithm 1. The input includes a standard planning framework composed of: F, a set of facts that can be used to describe the story world; I, an initial situation of the story world such that $I \subseteq F$; G, a goal condition such that $G \subseteq F$; and O, a ground set of operators (representing narrative actions in the story domain) each with an Add, Delete, and Precondition list. The input also includes the constraints graph, C, and the PoV that is adopted for the start of the narrative.

The main part of the algorithm (lines 1-13) is a loop that steps through each constraint in C in turn, starting from the earliest and continuing until all the nodes have been visited. In line 2, the function $select\_constraint(C, PoV)$ selects the next constraint for the planner to generate the narrative towards (the selection is described in section 4.1). Once this fact c has been selected a new sub-problem is formed with c as the goal condition, the operator set O and the current initial state I. A base planner is then called with this sub-problem (line 3). For this any propositional planner would be suitable and in our experiments we used FF-v2.3 [Hoffmann and Nebel 2001]. If a solution plan P' is returned the algorithm then steps through each operator in turn and sends it to the graphics engine to be visualised (line 6). If *visualise* returns *replan* (for example, this would occur if the user changed the state of the story world or requested a change of PoV) then the state of the world is updated accordingly, ready to re-plan. The nature of the re-planning depends on whether the constraint c is a constraint for the currently adopted PoV – if it is then the portion of narrative with constraint c as the goal is re-planned. In all other situations (either the whole narrative segment has been visualised or a change to PoV necessitates moving to the next applicable constraint) the algorithm goes to line 12. Here, constraint c is marked as visited and control loops back to line 1. Once all constraints have been visited the final conjunctive goal is tackled, with the same provision for re-planning in the event of user interaction.

A detailed example for our Merchant of Venice Interactive Narrative which illustrates narrative generation using this algorithm can be found in section 5.1.

5. RESULTS

We have argued that the use of constraints provides a mechanism to both represent narrative knowledge and guide narrative generation within an interactive system. In this section, we present experimental results that support this argument, along with discussion of run-time performance statistics. We will focus solely on the generative aspects of our approach since interactive aspects are best presented "live" rather than emulated and also because generation supports interactivity. The results are presented through analysis of a selection of sample narratives taken from our Merchant of Venice Interactive Narrative system. This is the prevailing approach in IS research (e.g. [Riedl 2009; Bae and Young 2008]) since plan optimality is not relevant and consequently such metrics are not applicable.

Our Merchant of Venice Interactive Narrative has the usual IS system architecture (as shown in figure 1). The narrative engine features an implementation of the algorithm described in the previous section. In the visualisation module, story visualisation is based on the Unreal Tournament™game engine, which supports staging and character animation. Narrative actions that are produced by the planner are passed sequentially to the visualisation module and activate Unreal scripts controlling the different character animations.

5.1 Example: Generating Narrative Variants

A tendency in IS research has been to follow the convention of classical theatre by modelling in detail fragments of novels or plays and pacing the animation and staging the action to reflect the real-time unfolding of the action. A consequence of

Fig. 7. A single generated narrative for the pound-of-flesh sub-plot with (pov antonio-risk-taker). Key constraints selected by the generator are highlighted (A1, C1, A2, A3, A4, C2, C3, A5, C4) and are preceded by the sequences of narrative actions selected to achieve them. This variant shows a carefree Antonio who confidently borrows money and continually dismisses personal risks even when brought to trial for defaulting on the loan. It ends with Antonio celebrating his release.

this approach is that key actions can be staged with minimal description, resulting in a whole play being condensed rather than including dialogue commensurate with the complexity of the play. We depart from this approach with our "Merchant of Venice" Interactive Narrative and aim to generate complete sub-plots that span the entire play. As an illustration, consider figures 7 and 8 which represent two narrative variants: one obtained by generating the pound-of-flesh sub-plot from the PoV of Antonio and the other for the PoV of Shylock. Both variants span the entire play and share specific situations which constitute the backbone of the pound-of-flesh plot. Represented as constraints these are: C1, the bond be sealed; C2, the bond is forfeit; and C3, the dispute ends in the high court.

There is a marked difference in the content of these narratives and different PoV places emphasis on specific actions. In particular, the narrative following Antonio's PoV (figure 7), emphasises the reasons for the loan: the relation between Antonio and Bassanio (operators 2-7), and the associated risk-taking (operators 12-15). Conversely, according to Shylock's PoV (figure 8), it is the relationship between Antonio and Shylock which is prominent, in particular with a history of persecution and humiliation (operators 7-8), and how Shylock sees the loan as an offer of friendship (operator 14). After the bond has been sealed, the narrative for Antonio's PoV continues to feature risk-taking and his relationship with Bassanio (operators 19-20; 26). On the other hand, Shylock's narrative describes further suffering with his daughter fleeing his house, adding to his victim status and justifying his future insistence on enforcing the bond (operators 19; 24-25). The pound-of-flesh sub-plot continues with the forfeit of the bond (Antonio is unable to repay the loan), after which, in the courtroom, the PoVs find their most salient expression in clearly reflecting Antonio's contempt for Shylock (operator 33) and Shylock's desire for justice (operators 28-29). This example shows how it is possible to generate different narrative variants whilst remaining true to the baseline plot and without user interaction. The variants manage to retain plot information and consistency whilst also shedding light on character motivation. They also show how it is possible to use PoV to present a more sympathetic treatment of character attitude.

Lets consider how these narratives were generated. For Antonio the constraints selected for this variant are labelled A1 to A5 and for Shylock they are labelled S1 to S5. There are also some constraints common to both characters, labelled C1 to C3. The constraints are key components of plot structure as they form the backbone of the plot outline. For example, if we look at the narrative for Antonio, the plot outline consists of the following constraint sequence: $A1 < C1 < A2 < A3 < A4 < C2 < C3 < A5 < C4$. Each of these constraints forms the goal of a separate sub-problem, so the goal of the first problem is A1 *(offered-guarantee-loan antonio bassanio antonio-residence)*, the goal of the next sub-problem is the constraint C1 and so on. These sub-problems must be tackled in order since they aren't independent: the conditions that are true at the end of one sub-problem become the initial conditions for the next sub-problem.

In these examples we can observe a number of sources of narrative variation. One results from selection between alternative constraints. For example, constraint node N1 in figure 6 includes three constraints. For some variants, such as the Shylock narrative in figure 8, the constraint *(responded-to-news-of-elopement shylock)* labelled

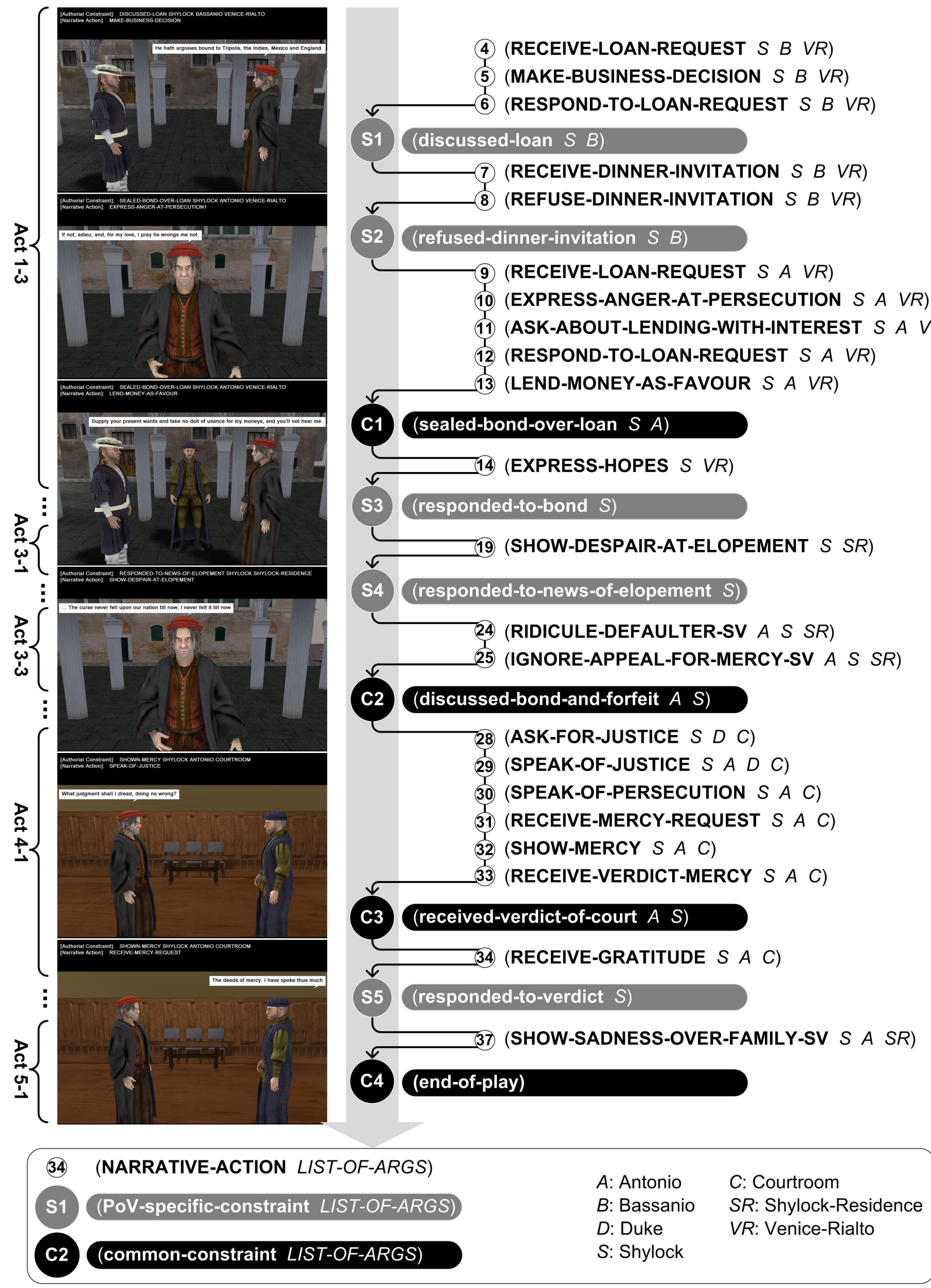

Fig. 8. A single generated narrative for the pound-of-flesh sub-plot with (pov shylock-victim). Key constraints selected by the generator are highlighted (S1, S2, C1, S3, S4, C2, C3, S5, C4) and are preceded by the sequences of narrative actions selected to achieve them. This variant shows Shylock lending money in friendship and showing mercy when Antonio has defaulted on the loan. It ends with Shylock receiving gratitude but showing sadness at the elopement of his daughter.

	Shylock		Antonio	
	victim	ruthless	victim	risk-taker
Average response time to visualisation engine	1.20	1.19	1.04	1.00
Total operators sent to visualisation engine	41	42	45	43
Average duration of complete 3D story	550.35	554.42	560.25	558.36
Percentage of constraints satisfied	100%	100%	100%	100%

Fig. 9. Summary of run-time Performance Results

S5, will be selected but for other variants one of the other constraints will be chosen. The selection of different constraints forces the planner to search for narrative actions, resulting in different narrative variants. Another source of variation results from the selection of asymmetric narrative actions on the basis of character PoV. For example, the domain model includes a number of asymmetric actions that all result in the loan being arranged between Shylock and Antonio. In figure 7 we can see that the action *(borrow-money-confident-repay antonio shylock venice-street)* has been selected, in keeping with a PoV of *(pov antonio-risk-taker)* that has been adopted for this narrative. For a different PoV, such as *(pov shylock-victim)* shown in figure 8, then a different asymmetric narrative action, *(lend-money-extend-favour shylock antonio venice-rialto)* is selected to achieve this same constraint.

5.2 Run-time Performance

Our Merchant of Venice system features a narrative generator, an implementation of the algorithm from section 4, which is integrated with a visualisation engine in an architecture as shown in figure 1. In this section, we discuss key run time performance statistics of the system which show that our system is able to perform within the desired response time that was discussed in section 2.2. The table below summarises these performance statistics. All times are in seconds and all measurements were taken on a 2.26GHZ machine with 4GB of RAM.

The average response time to the visualisation engine was within the upper bound on 1500ms. For each PoV, the generation engine produced narrative variants which spanned the whole of the Merchant of Venice and contained 40+ operators for which the narrative actions corresponding to these operators were then staged in the 3D visual environment. The average duration of the complete 3D story (i.e. the presentation of the narrative to the user) was approximately 9 minutes. For all test cases, the system generated narratives that satisfied all the constraints.

The system response time is acceptable: it can generate sub-problem plans in response to the visualisation engine within the time limit. The following factors contribute to this acceptable level of performance: our decomposition approach means that the important system response time is for each decomposed sub-problem and not the whole narrative; also, the system only has to do “incremental” re-planning i.e. re-planning of sub-problems, which these results show to be within the acceptable range. To determine the performance of the system on larger problems we would need to increase the size of the domain model (the size of our Merchant of Venice interactive narrative was given in section 3). This is a topic for future work and we hypothesise that decomposition will still yield performance gains.

6. RELATED WORK

Our use of state constraints to specify narrative control knowledge is similar to Riedl's notion of *author goals* [Riedl 2009]. He observed that without information about narrative structure planners can generate sparse plans or even no plan at all. Riedl extended his partial order planner to plan with the inclusion of author goals: a process of "complexifying" the planning process. This was the first use of explicit constraints in narrative paths and our approach can be viewed as a dynamic extension of it with our development of a mechanism that dynamically handles constraint selection for narrative variants at run-time (see section 4). Also, our approach supports interactivity via a forward state space planning approach and uses a standard representation language, PDDL3.0, for domain modelling.

Our approach is inspired by earlier collaborative work by one of this paper's authors (J. Porteous), on the identification and use of *landmarks* [Hoffmann et al. 2004] to decompose the planning problem into a sequence of sub-problems. The approach presented here differs in that instead of landmarks we use narrative conditions, expressed as constraints, to decompose the problem. In addition the narrative generator is itself integrated within an interactive IS system.

Although our approach decomposes narrative generation into a series of sub-problems it differs from the types of HTN decomposition that have featured in other IS systems such as [Cavazza et al. 2002b] and [Riedl and Stern 2006]. There, the decomposition is of a hierarchy of compound tasks, with action effects only allowed to be associated with non-compound tasks. There is no notion of hierarchy in our decomposition, simply sub-division into a series of smaller problems all at the same "level".

A number of planners have been developed which can reason about constraints, including SGPlan5 [Hsu et al. 2006] and MIPS-XXL [Edelkamp et al. 2006]. Motivated by IS requirements, we have taken a different approach and use the constraints to decompose the planning problem so that the narrative is produced incrementally, ready for visualisation within the IS system.

7. CONCLUSION

In this paper, we have presented a novel approach to plan-based IS which is tailored to IS requirements. It is built on recent developments in planning technology such as the use of landmarks and the move to the representation of constraints on properties of the plan itself, not just on the final goal conditions. Our approach is novel in IS terms since it embeds declarative control through the use of constraints. This is in contrast to other IS approaches where separate narrative control mechanisms have been added on top of the planner itself. In addition, the constraint-based approach that we have used provides a unified framework in which dynamic aspects linked to IS such as Pace, Point of View and Discourse effects can be represented and reasoned about. Again, this is in contrast to other IS approaches where ad hoc mechanisms have been required for each different IS aspect.

The technology we have developed allows us to declaratively specify narrative control knowledge and use this to control trajectory. This opens up the possibility of new modes for narrative control and for authoring of IS story worlds (there is a clear need for authoring support in IS [Pizzi and Cavazza 2008]). For example, our declarative approach to the specification of narrative knowledge may be more "author friendly" than other approaches. In future work, we intend to explore the extent to which our approach assists authors in content creation.

In addition, we intend to develop graphical tools that will assist story authors and creators in the process of story world specification. This will facilitate the creation of larger narrative domain models which will enable us to investigate further how our approach scales up. It will also enable us to explore the formalisation as constraint problems of other IS phenomena.

ACKNOWLEDGMENTS

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

REFERENCES

André, E. and Rist, T. 1993. The Design of Illustrated Documents as a Planning Task. In *Intelligent Multimedia Interfaces*, M. Maybury, Ed. AAAI Press.

Aylett, R., Dias, J., and Paiva, A. 2006. An Affectively Driven Planner for Synthetic Characters. In *Proc. of 16th Int. Conf. on Automated Planning and Scheduling (ICAPS 2006)*, D. Long, S. F. Smith, D. Borrajo, and L. McCluskey, Eds. Cumbria, UK, 2–10.

Bacchus, F. and Kabanza, F. 2000. Using Temporal Logics to express search control knowledge for planning. *Artificial Intelligence 116*, 123–191.

Bae, B. and Young, R. 2008. A Use of Flashback and Foreshadowing for Surprise Arousal in Narrative Using a Plan-Based Approach. In *Proc. of 1st Int. Conf. on Interactive Digital Storytelling (ICIDS 2008)*, U. Spierling and N. Szilas, Eds. Erfurt, Germany, 26–29.

Bolter, J. D. and Grusin, R. 1999. *Remediation: Understanding New Media.* Cambridge (Massachusetts), MIT Press.

Bonet, B. and Geffner, H. 1999. Planning as heuristic search: New results. In *Proceedings of the European Conference on Planning (ECP)*.

Castillo, L., Fdez.-Olivares, J., García-Pérez, O., and Palao, F. 2006. Bringing Users and Planning Technology together. Experiences in SIADEX. In *Proc. of the 16th Int. Conf. on Automated Planning and Scheduling (ICAPS 2006)*. Lake District, England.

Cavazza, M., Charles, F., and Mead, S. 2002a. Emergent Situations in Interactive Storytelling. *ACM Symposium on Applied Computing (SAC 2002)*, 1080–1085.

Cavazza, M., Charles, F., and Mead, S. J. 2002b. Character-based interactive storytelling. *IEEE Intelligent Systems 17,* 4, 17–24.

Cavazza, M., Pizzi, D., Charles, F., Vogt, T., and André, E. 2009. Emotional Input for Character-based Interactive Storytelling. In *Proc. of modal 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*. Budapest, Hungary, 313–320.

Chien, S., Tran, D., Rabideau, G., Schaffer, S., Mandl, D., and Frye, S. 2010. Timeline-Based Space Operations Scheduling with External Constraints. In *Proc. of the 20th Int. Conf. on Automated Planning and Scheduling (ICAPS 2010)*.

Edelkamp, S., Jabbar, S., and Nazih, M. 2006. Large-Scale Optimal PDDL3 Planning with MIPS-XXL. In *Proc. of 5th Int. Planning Competition*. Part of the 16th Int. Conf. on Automated Planning and Scheduling (ICAPS 2006), Lake District, England.

Fikes, R. and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence 2*, 189–208.

Geib, C. W. 1994. The Intentional Planning System: ItPlanS. In *Proceedings of the 2nd Int. Conf. on Artificial Intelligence Planning Systems (AIPS-94)*. AAAI Press, Chicago, USA.

Gerevini, A. and Long, D. 2005. Plan Constraints and Preferences in PDDL3. Tech. rep., Department of Electronics for Automation, University of Brescia, Italy. http://www.cs.yale.edu/homes/dvm/papers/pddl-ipc5.pdf.

HINELY, J. L. 1980. Bond Priorities in The Merchant of Venice. *Studies in English Literature, 1500-1900 20,* 2, 217–239.

HOANG, H., LEE-URBAN, S., AND MUNOZ-AVILA, H. 2005. Hierarchical Plan Representations for Encoding Strategic Game AI. In *Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-05).* AAAI Press.

HOFFMANN, J. AND NEBEL, B. 2001. The FF Planning System: Fast Plan Generation through Heuristic Search. *Journal of AI Research 14*, 253–302.

HOFFMANN, J., PORTEOUS, J., AND SEBASTIA, L. 2004. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research (JAIR) 22*, 215–278.

HSU, C., WAH, B., HUANG, R., AND CHEN, Y. 2006. New Features in SGPlan for handling Preferences and Constraints in PDDL3.0. In *Proc. of 5th Int. Planning Competition.* Part of the 16th Int. Conf. on Automated Planning and Scheduling (ICAPS 2006).

KARLSSON, B., CIARLINI, A., FEIJÓ, B., AND FURTADO, A. 2006. Applying a plan-recognition/plan-generation paradigm to interactive storytelling. In *In Workshop on AI Planning for Computer Games and Synthetic Characters.*

KELLY, J. P., BOTEA, A., AND KOENIG, S. 2007. Planning with Hierarchical Task Networks in Video Games. In *Proceedings of the ICAPS-07 Workshop on Planning in Games.*

KVARNSTRÖM, J. AND DOHERTY, P. 2001. TALPLANNER: A Temporal LogicBased Forward-Chaining Planner. *Annals of Mathematics and Artificial Intelligence 30*, 119–169.

LAIRD, J. E. 2002. Research in Human-Level AI using Computer Games. *Commun. ACM 45,* 1.

NAU, D., AU, T., ILGHAMI, O., KUTER, U., MURDOCK, J., WU, D., AND YAMAN, F. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research (JAIR) 20*, 379–404.

PIZZI, D. AND CAVAZZA, M. 2008. From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description. In *Proc. 1st Joint Conf. on Interactive Digital Storytelling.*

PIZZI, D., CHARLES, F., LUGRIN, J., AND CAVAZZA, M. 2007. Interactive Storytelling with Literary Feelings. In *Proc. of 2nd Int. Conf. on Affective Computing and Intelligent Interaction (ACII).*

PORTEOUS, J., CAVAZZA, M., AND CHARLES, F. 2010. Narrative generation through characters' point of view. In *Proc. of 9th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2010).* IFAAMAS, Toronto, Canada.

PRESSMAN, R. S. 2009. *Software Engineering: A Practitioner's Approach.* McGraw-Hill.

RADFORD, M. 2004. MGM Home Ent. (Europe) Ltd. The Merchant of Venice (film adaptation).

RIEDL, M. 2009. Incorporating Authorial Intent into Generative Narrative Systems. In *Proc. of AAAI Spring Symposium on Intelligent Narrative Technologies.* Palo Alto, California.

RIEDL, M. AND STERN, A. 2006. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. In *Proceedings of the 3rd Int. Conf. on Technologies for Interactive Digital Entertainment (TIDSE-06)*, S. Göbel, R. Malkewitz, and I. Iurgel, Eds. Elsevier.

RIEDL, M. AND YOUNG, R. 2005. Open-world planning for story generation. In *Proc. of the Int. Joint Conf. on Artificial Intelligence IJCAI 2005).*

TRABASSO, T. AND VAN DEN BROEK, P. 1985. Causal Thinking and the Representation of Narrative Events. *Journal of Memory and Language 24*, 612–630.

WELD, D. 1994. An introduction to least commitment planning. *AI magazine 15,* 4, 27–61.

YOUNG, R. M. 2000. Creating Interactive Narrative Structures: The Potential for AI Approaches. In *AAAI Spring Symposium in Artificial Intelligence and Entertainment.* AAAI Press.

Received March 2010; June 2010; accepted July 2010