Controlling Narrative Time in Interactive Storytelling

Julie Porteous, Jonathan Teutenberg, Fred Charles and Marc Cavazza
School of Computing,
Teesside University,
Middlesbrough TS1 3BA,
United Kingdom
{j.porteous,j.teutenberg,f.charles,m.o.cavazza}@tees.ac.uk

ABSTRACT

Narrative time has an important role to play in Interactive Storytelling (IS). The prevailing approach to controlling narrative time has been to use implicit models that allow only limited temporal reasoning about virtual agent behaviour. In contrast, this paper proposes the use of an explicit model of narrative time which provides a control mechanism that enhances narrative generation, orchestration of virtual agents and number of possibilities for the staging of agent actions. This approach can help address a number of problems experienced in IS systems both at the level of execution staging and at the level of narrative generation. Consequently it has a number of advantages: it is more flexible with respect to the staging of virtual agent actions; it reduces the possibility of timing problems in the co-ordination of virtual agents; and it enables more expressive representation of narrative worlds and narrative generative power. Overall it provides a uniform, consistent, principled and rigorous approach to the problem of time in agent-based storytelling. In the paper we demonstrate how this approach to controlling narrative time can be implemented within an IS system and illustrate this using our fully implemented IS system that features virtual agents inspired by Shakespeare's *The Merchant of Venice*. The paper presents results of an experimental evaluation with the system that demonstrates the use of this approach to co-ordinate the actions of virtual agents and to increase narrative generative power.

Categories and Subject Descriptors

H5.1 [**Multimedia Information Systems**]: Artificial, augmented and virtual realities

General Terms

Algorithms

Keywords

Interactive Storytelling, Agents in games and virtual environments, Narrative Modelling, Planning

Cite as: Controlling Narrative Time in Interactive Storytelling, J. Porteous, J. Teutenberg, F. Charles and M. Cavazza, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 449-456.
Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1. INTRODUCTION

Time plays a central role in many aspects of narration [22] both at the story and at the discourse level. Time determines pace, dramatic tension as well as the aesthetic of story visualisation and staging. Existing Interactive Storytelling (IS) systems have emphasised the causal aspects of agents' actions but have not incorporated time in their narrative generation mechanism in a principled fashion.

The prevailing approach in IS has been to use AI planning for narrative generation and empirical solutions for the synchronisation of agents' actions, often arrived at by a process of trial and error or using deliberately underspecified representations that assume uniform execution time of agent actions. These approaches can work well, as demonstrated by a number of successful IS prototype systems, including [2, 27, 19], but they miss an opportunity to use action duration as an element of story presentation at the discourse level.

An alternative approach to controlling narrative time is to extend the representation of narrative actions to reincorporate temporal aspects (such as duration, concurrency, overlap and so on) in the planning process that is used for narrative generation. This would ensure that generated narratives contained explicit information about the timing of agent actions which could be used during the staging of the narrative. While the IS research community has enthusiastically embraced AI planning due to its capability for propagating causality, to date, there has been no use of dedicated temporal planning architectures. Yet these architectures are potentially useful for IS since it is likely that there are narrative situations that require dedicated temporal architectures (ones which are similar to the *temporally expressive* problems documented in the AI planning literature [8]). Applying temporal planning to narrative generation would provide a sound and principled approach to further increase the generative power of IS systems and to expand the range of stories that can be generated.

The use of temporal planning within the process of narrative generation is an approach that neatly re-incorporates aspects of the problem that have tended to be solved by trial and error. Clear benefits of this approach include: (i) it will enable the generation of story and discourse from shared principles; (ii) it will simplify development and production; (iii) it will improve integration of action and motion at the technical level. In addition, we anticipate that the approach will yield the following advantages: (i) help improve system reliability, e.g. by overcoming problems associated with timing and co-ordination of virtual agent actions; (ii) provide a wider range of possibilities for staging and cinematographic aspects of virtual agent actions; and (iii) increase the generative power of the system, i.e. the range of agent situations and narratives that can be generated.

Throughout the paper we illustrate our discussion with examples taken from an interactive narrative that we have developed which features virtual agents and situations inspired by Shakespeare's play *The Merchant of Venice* [24].

The paper is organised as follows. In the next section we consider related work. This is followed in section 3 with discussion of issues related to the explicit temporal representation of actions and narratives. Section 4 gives an overview of our approach to generating temporal narratives. The results of an evaluation using our implemented system are presented in section 5. Section 7 summarises our conclusions.

2. RELATED WORK

2.1 Interactive Storytelling

A number of prototype IS systems have been developed that use AI planning for narrative generation [2, 27, 19]. These systems ignore the staged execution time of agent actions during narrative generation. Instead, they have adopted a range of solutions to the handling of temporal aspects at the staging level. One such approach is the use of *executability conditions* [15] to specify conditions for successful execution of actions [4]. This approach has been used to co-ordinate the actions of virtual agents but its failure to reason about temporal aspects such as staged execution time can make it unreliable. It also requires time-consuming empirical solutions for the actual production of interactive narratives thereby limiting its scalability.

A form of executability condition is used in the execution management architecture ZOCALO [27] to ensure that actions are executed in legal world states. The system makes some allowance for the time taken for actions to execute (a state of *executing* is maintained) and action effects are not activated until actions have successfully completed. However there is no explicit reasoning about action duration during narrative generation and this could make the system unreliable. For example, this omission may only become apparent during staged execution when an agent arrives too late to co-ordinate with another agent.

The LOGTELL system [19] also features an overall manager of the IS system which is responsible for controlling the staging of a partially-ordered plot output by their IPG generator. The system makes use of temporal logic as a representation for the state of the system, which can be used in particular when authoring the narrative. However no mention is made of its use for resolving the problems of temporal dynamics faced by narrative generation.

HPTS [11] is a system that reasons about time to handle the synchronisation of behavioural agents. Reactive behaviours are described within a runtime environment to handle parallel state machine execution and synchronisation of agents. This approach orchestrates the synchronisation of low level action execution (sometimes referred to as the motion level), such as motion blending and interruption.

An alternative approach is the use of Petri Nets which has been explored to handle the unfolding of story plots and the co-ordination of virtual agent behaviour [3]. However the behaviour of such a system is reactive and only includes deliberation about localised temporal aspects of the problem. Also localised in its approach is the use of cascaded Finite State Machines in SCENEMAKER [14]. This represents an orchestrated approach to temporal and synchronisation issues but its static strategy is rather inflexible and temporal reasoning is at the "microscopic" level not the planning level.

2.2 Research in Automated Planning

On the other hand temporal planning is a very active research topic in the field of AI planning which has generated multiple approaches, targeted specifically at temporal problems. These include logic based planning [1], partial order planning (ZENO [20], VHPOP [28]), hierarchical planning (NONLIN [25], OPLAN [12]), extended state space progression search planning (SAPA [10], SGPLAN [6]) and hybrid planning systems combining features of different temporal planning architectures (TEMPO [8], CRIKEY [7]). Early systems such as ZENO could tackle complex temporal problems but they suffered from performance limitations. More recently systems such as SGPLAN, TEMPO and CRIKEY have overcome efficiency problems to the point where they now have potential for application to IS.

3. REPRESENTING NARRATIVE TIME

IS systems that use planning for narrative generation use a representation of the narrative world that includes information about virtual agent behaviours represented as pre- and post-condition *actions*. These actions detail the way that the agent action is expected to change the state of the narrative world when it is staged in a visual environment. Not only can these actions describe the capabilities of an agent, but they can also describe properties inherent in the process itself – in particular their staged execution time. This notion of execution time may be represented either *explicitly* or *implicitly*: an implicit representation enabling the narrative generator to reason about relative orderings of actions; an explicit representation extending this to enable reasoning about complex temporal interactions[1].

3.1 Narrative Action Representation

In an implicit representation no temporal information is included in the description of agent actions and the assumption is that the effects of actions are instantaneous (the classical STRIPS assumption [13]). In contrast, explicit reasoning about the duration of actions makes it possible to take into account the more sophisticated interplay between the occurrence of actions themselves, not just their consequences. It shows the continuous evolution of the story world over time as actions unfold rather than merely showing actions as their consequences. This explicit *durative* representation provides a means to represent conditions that can be used for agent synchronisation: before an agent is able to start an action (e.g. in order for an agent to start to listen to another agent, they must be within earshot); at the end of the action (e.g. in order for an agent to make a selection between a number of alternatives, they must have reached their decision); or must remain true over the duration of the action as an invariant (e.g. during the time an agent listens to an agent singing they must stay in earshot). The durative action representation also makes it possible to specify which narrative effects occur immediately, as a virtual agent starts to perform an action (e.g. when a virtual agent sings,

[1]We note the correspondence between implicit and explicit models [17] and qualitative and quantitative models [7].

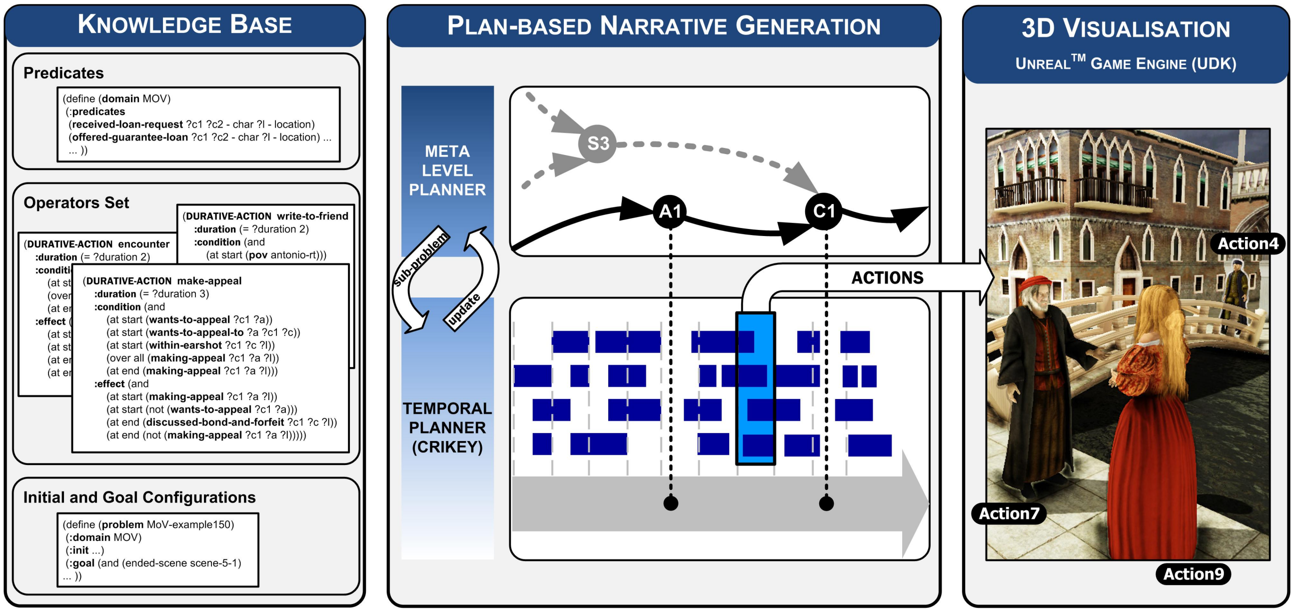

Figure 1: System Architecture: input is a domain model (knowledge base) represented temporally; the plan-based generator builds the narrative incrementally by decomposing the problem into a series of sub-problems which are then tackled in turn using a temporal planner; for 3D visualisation, the temporal narrative actions output by the planner map to UnrealScript action descriptions.

the sound starts immediately), and which are delayed until the agent finishes the action (e.g. an agent spends time persuading another agent, the effect of having been persuaded is activated at the end).

An illustration of the need for temporal reasoning is provided by act III scene ii of our *Merchant of Venice* system. In the scene there are specific narrative actions that require an informed decision by an agent. These actions must unfold whilst the agent acquires additional information through other actions (e.g. conversations). One such action is the selection of a casket by a character, Bassanio, in an attempt to win the hand in marriage of another character, the wealthy heiress Portia. A durative representation of the action is[2]:

```
(:durative-action select-casket
    :parameters (?c - char ?ca - casket ?l - location)
    :duration (= ?duration 4)
    :condition (and ....
              (over all (selecting ?c ?l))
              (at end (selecting ?c ?l))
              (at end (decided-to-select ?c ?ca)))
    :effect (and
              (at start (selecting ?c ?l))
              (at end (selected ?c ?ca ?l))
              (at end (not (selecting ?c ?l)))))
```

This illustrates the temporal properties of the action where deliberation lasts for the duration of the action (over all the character is *selecting*) but this must be finalised for the action to end when post-conditions are activated.

A non-durative version of this narrative action is cumbersome and does not capture the unfolding of agent deliberation over time. This may prevent the action from being synchronised with other agent actions or being interrupted (either by other agents or users in an interactive setting). In addition, deliberation has dramatic value in terms of staging and understandability: it enables the spectator to see agents' decision processes and the factors that influence them.

3.2 Narrative Representation

Temporal narrative plans include information about the time each agent action is scheduled to start and the expected duration of each action. The following example:

```
0.001: (select-casket bassanio lead casket-room) [4.00]
0.002: (give-hint-in-song portia casket-room) [3.00]
0.003: (listen-to-song bassanio casket-room lead) [3.00]
```

is a representative example of the paradigm, showing the start time on the left of the action name and the duration on the right. This example occurs in act III scene ii of the *Merchant of Venice* where one of the characters, Bassanio, is deliberating about the selection of a casket whilst simultaneously acquiring information from hints that are given to him in song by another character. The temporal aspect of the action, namely the character's decision process (deliberation) can now be staged as an important element of discourse, as it incorporates important information on the relation between characters. Also, it allows for interference by other agents (or the user) thereby supporting further narrative generation

In contrast, capturing this in a non-temporal narrative is problematic since there is no way to specify start times and duration of actions. Actions can be left partially ordered (either generated by a partial order planner [28] or by lifting a partially ordered narrative from a totally ordered one [26]) but the required overlap between actions cannot be captured without explicit representation of time.

[2]We chose PDDL3.0 [16] because of its expressive power and since it is a standard action description compatible with multiple planning approaches.

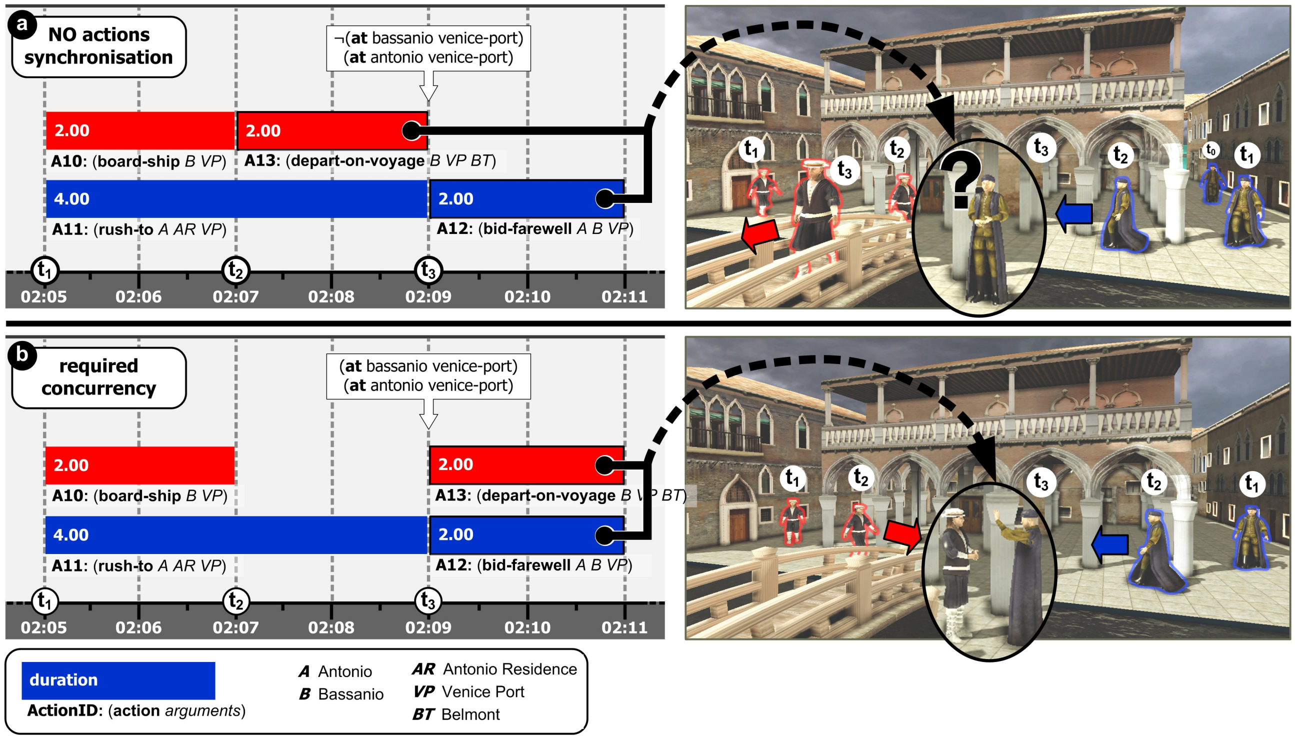

Figure 2: *Merchant of Venice* example illustrating the role of reasoning about staged execution time: (a) staging failure when narrative actions are not synchronised; (b) successful staging when reasoning about staged execution time during narrative planning identifies required concurrencies between actions.

4. NARRATIVE GENERATION

Generation of narratives that feature concurrent durative agent actions requires a planning architecture that can reason about explicit temporal information. Research in AI planning has led to the development of a number of dedicated temporal planning architectures (discussed in section 2). Recent, hybrid temporal planners such as TEMPO and CRIKEY have managed to overcome the performance limitations of earlier partial order planners and the incompleteness experienced by the extended space progression planners. Since our motivation includes being able to generate narratives that feature overlapping concurrent agent actions, we have chosen to use the CRIKEY system of Coles et al[7] in our implemented narrative generator. The system will use CRIKEY in combination with narrative structuring information since, without such information, the planner could end up generating sparse narratives or even no narrative at all [23]. The generator will use the information to guide CRIKEY towards the generation of narratives that are sufficiently rich and in keeping with the narrative genre.

The narrative structuring information represents key narrative situations that can be used like intermediate goals to guide the planner. After [21], we refer to these situations as *constraints* but they have also been described in the literature as *author goals* [23] and are similar to the notion of landmarks [18] that have featured in AI planning. The constraints for a narrative world are represented as a partially ordered set of predicates – a declarative representation which separates this information from action descriptions and may help facilitate its specification and maintenance.

Our implementation is based on the decomposition approach of [21]. This can be summarised as follows: use an input set of constraints to decompose the process of narrative generation into a sequence of sub-problems; generate a narrative for each decomposed sub-problem; and then assemble the final narrative by composition of the sequence of narratives. This approach implements a higher level of representation, where the constraints enable reasoning about narrative at the meta-level. The constraints can also be re-combined for different total orderings (as used in our experiments, see section 6).

Our contribution has been to extend their approach to handle temporal reasoning. These extensions were possible because of fundamental properties of the system that enabled the control program to be integrated with different base planners. An overview of the architecture of our implemented experimental system is shown in figure 1. The input is represented using the representation language PDDL3.0 which permits both implicit and explicit representations of the narrative domain to be input to the system. The control mechanism uses the input constraints to decompose the problem and then sends decomposed sub-problems to CRIKEY. As narrative actions are received from CRIKEY by the control mechanism they are sent to a visualisation module. The switch to temporal planning provides a direct route to mapping between planning actions and their visualisation through the transfer of PDDL3.0 temporal parameters to animation control structures (UnrealScript action descriptions).

5. QUALITATIVE EVALUATION

The objective of our evaluation was to provide data for the systematic assessment of system performance and behaviour. Here we evaluate the approach qualitatively, with reference to sample *Merchant of Venice* narratives generated by our system and shown in figures 2 and 3. These narratives provide answers to some key questions about our approach to controlling narrative time, namely: (1) can our approach help to avoid timing problems as agent actions are staged? (2) does our approach provide a mechanism to exploit information about the staging of agent actions? (3) does our approach to explicit temporal representation and reasoning increase the generative power of the system?

5.1 Avoiding Timing Problems

Failure to reason explicitly about temporal aspects of the IS domain at the point of narrative generation can cause problems that only become apparent when the virtual agent actions are staged. This can manifest itself both in real-time failure of the system and failure at the "production" level which it may be possible to repair through ad hoc local solutions. For example, if action duration isn't reasoned about during narrative generation then an agent may fail to meet up with another agent because they arrive too late, after the other agent has already left.

A scene from our *Merchant of Venice* system, shown in figure 2, illustrates how this situation can arise. In this scene one character, Antonio, is endeavouring to reach another character, Bassanio, in time to bid him farewell before he departs to sea. In principle, it is possible to generate a narrative for this scenario without reasoning about the staged execution time of the actions and then to use executability conditions (as used in [4]) to try to synchronise agents by testing that conditions for successful execution of agent actions hold. In this example the actions for Antonio are to rush to the port and then bid farewell to Bassanio as he leaves; the actions for Bassanio are to board the boat and then depart on his voyage. The first action for Bassanio has him boarding the ship and since this is independent of the first action for Antonio, rushing to the port, they can be staged and visualised in a concurrent fashion (which also gives interesting opportunities for exploration of automated camera control). The executability conditions for Bassanio's next action, departing aboard ship, do not mention anything about Antonio's location. Hence the action can start being visualised irrespective of the actual on-stage localisation of Antonio. Depending on how long Antonio takes to arrive at the port, it can happen that this doesn't occur until Bassanio has completely departed from the port, making it impossible for Antonio's final action, that of bidding his friend farewell, to be executed in the visual environment. This situation is depicted in figure 2.

How would explicit reasoning about time at the point of narrative generation mean such situations were avoided? The critical consequence of reasoning about the staged execution time of these agent actions is the recognition of the requirement that Bassanio must still be at the port when Antonio bids farewell to him, in other words that these actions are staged at the same time. This is shown in figure 2: the narrative generator has considered the duration of the actions, identified the required concurrency between them and forced them to overlap.

5.2 Providing Information for Staging

Our use of an explicit model of time results in generated narratives that include scheduled start times for each agent action and their duration, precisely the information that can be utilised for staging actions in different ways.

Act I scene (iii) of the *Merchant of Venice* provides an illustration of the generation of this staging information. The narrative for this scene (figure 3) shows the scheduled actions for the characters named Antonio, Bassanio and Shylock. The start of the narrative includes actions which bring them together on the Rialto ready to discuss the loan of a sum of money and subsequently seal a bond committing them to this arrangement. The red line drawn through the narrative in figure 3 shows the point at which this scene begins in the original play – opening with Bassanio and Shylock in conversation on the Rialto and continuing with the arrival of Antonio who joins them in conversation. This use of scene changes in classical theatre can be seen as a "tweak" which enables characters to appear at different locations as and when needed with no need to reason about their actions during the elapsed time (this tweaking of time has also been used in IS systems to avoid reasoning about agent actions whilst they are "off-screen" [21]).

However, in IS the objective is to provide different possible directions for the narrative and if there is a possibility that agent actions may need to be staged then they must be reasoned about. In our *Merchant of Venice* example, this means that earlier portions of the narrative (i.e. those before the start of the original scene from the play) need to be reasoned about during narrative generation. Consequently, the narrative in figure 3 also includes agent actions for the time before they enter into conversation. This allows for multiple ways of staging these actions, for example, focussing on one agent and their actions and motivations prior to the conversation, rather than cutting directly to them.

5.3 Generative Power

There are narratives that can only be generated with an explicit temporal approach. The scene depicted in figure 3 where the character Bassanio is enquiring about a loan and Shylock is simultaneously listening can be used to illustrate this. The action of Bassanio enquiring about the loan requires that Shylock listens to Bassanio for the whole of the enquiry. The action can be represented as:

```
(:durative-action listen-to-enquiry
  :parameters (?c1 ?c2 - char ?l - location)
  :duration (= ?duration 2)
  :condition (and
     (at start (at ?c1 ?l))  ...
     (over all (listening-to-enquiry ?c1 ?c2 ?l))
     (at end (listening-to-enquiry ?c1 ?c2 ?l)))
  :effect (and
     (at start (listening-to-enquiry ?c1 ?c2 ?l)) ...
     (at end (not (listening-to-enquiry ?c1 ?c2 ?l)))))
```

which captures the ongoing nature of the listening process with the condition *(listening-to-enquiry ?c1 ?c2 ?l)* that is activated at the start of the action and is maintained over the duration. However, in a non-durative version of this action, time would be compressed[3] and this condition would not be made true. This is problematic since the action of

[3]A compressed version of a durative action can be formed by setting the effects of the action to be the result of applying the start effects followed by the end effects and then setting

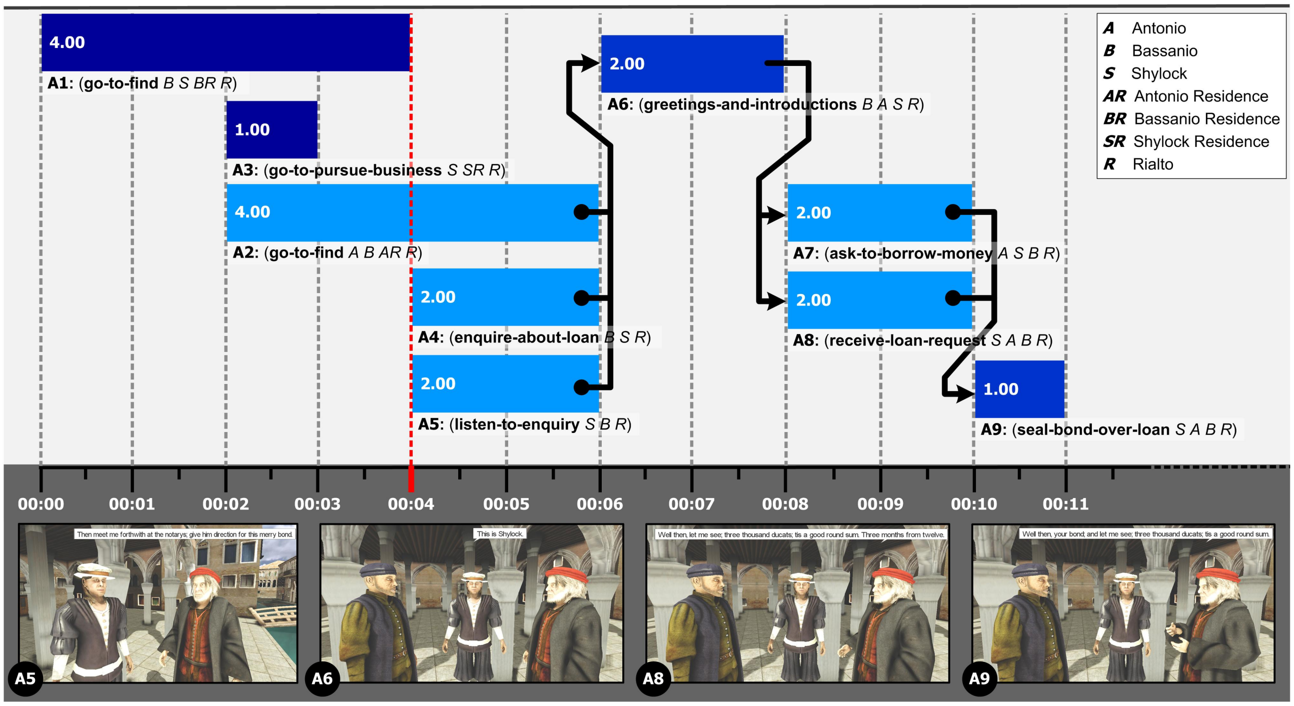

Figure 3: Example *Merchant of Venice* Narrative with overlapping durative actions: multiple possibilities for staging are introduced by temporal reasoning before the start of the scene in the original play (red line).

Bassanio enquiring about a loan requires Shylock to be listening (even in a compressed version this would remain as a pre-condition). The only way to handle scenarios such as this would be somewhat clumsy and would involve coercing the conversational exchange to take place at a given stage.

This example demonstrates the increased generative power of a temporal approach: narratives can be generated that require interactions over the duration of actions and these cannot be generated by compressed versions of the same actions. This is discussed further in the next section.

6. RESULTS

As demonstrated in the previous section, there are narratives which can only be properly generated and staged using narrative actions which have duration. Here we assess how this could affect real-world IS narrative generation problems, by examining the capacity these representations have for generating narratives for the different sub-problems that result from applying our decomposition approach in our experimental *Merchant of Venice* domain. These experiments focussed on narrative generation and consequently were performed off-line, without visualisation. The inclusion of staging would not significantly alter these results, and if anything, temporal planning would be less adversely affected given that the resolution of temporal factors is handled prior to visualisation.

In the course of one run of the IS system, user interaction could force the story to enter a broad range of unforeseeable world states. To simulate this, we generated a set of 20 potential initial states of the narrative domain by sampling randomly from the set of facts that are relevant to the different story sub-problems (where a fact is relevant if it can appear in a causal chain for achieving the sub-problem). A typical example of one of the randomly generated initial states contains the following facts:

```
(at bassanio venice-rialto)
(at antonio venice-street )
(decided bassanio lead-casket)
(enquired-about-loan bassanio shylock antonio)
```

In addition to facts specifying virtual agents' initial locations, in this state Bassanio has decided to choose the correct casket prior to travelling to Belmont, and has already discussed potential loans with Shylock. It should be noted that spurious facts, such as *(decided antonio gold-casket)* are never included in the generated initial states, as they are not deemed to be relevant facts (i.e. in this case, Antonio is not a suitor, and therefore has no reason for selecting caskets).

For each of the initial states, two narrative plans were generated, built up from 10 decomposed sub-problems. The first of these narratives was constructed using non-durative agent actions, and the second with durative ones. A cumulative count of the number of sub-problems successfully achieved was kept for each run. If one approach failed to achieve a sub-problem its state was changed to that reached by the other approach, and the system was then permitted to continue narrative generation from that point. This strategy was adopted in order to avoid unfairly penalising an approach for failing to achieve a sub-problem especially early in the narrative. Figure 4 shows the mean rate of sub-problem achievement for durative and non-durative actions. The solid lines indicate the mean number of sub-

the action pre-conditions to be the start conditions of the durative action along with all end conditions and invariants that are not achieved by the start effects [7].

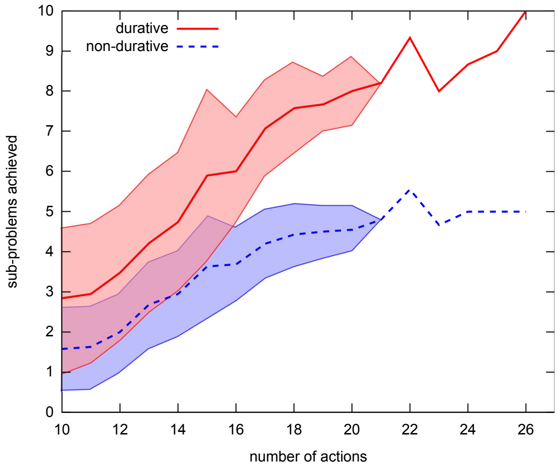

Figure 4: Count of the number of successfully generated narratives for decomposed sub-problems (with and without durative actions). Corridors show one standard deviation. (See text for further details.)

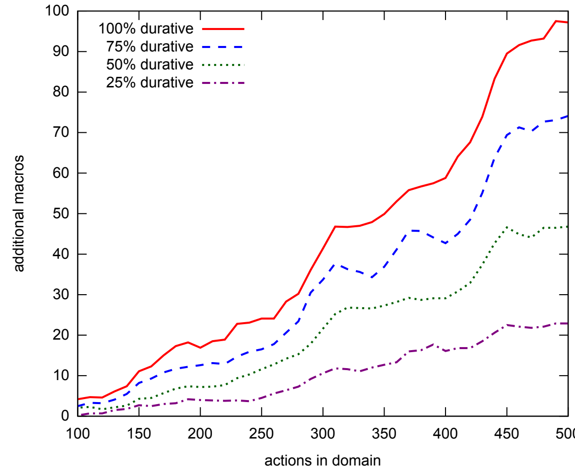

Figure 5: Increase in generative power resulting from the use of durative action representation. Lines show the increase in potential macros depending on domain size and percentage of durative actions.

problems achieved at each point in a narrative, and the corridor around each shows one standard deviation.

It is immediately clear that in this real-world example of an IS problem, generativity issues can have a significant effect on its execution. The graph shows results on output narratives of more than 10 actions, since narratives shorter than this are deemed too brief to be meaningful. For narratives of increasing length there is a clear difference in the number of sub-problems that can be achieved with the use of a temporal approach. Each failed sub-problem represents a point at which a real-world IS system must either sacrifice logical consistency of the narrative, or apply hand-crafted repair rules that jeopardise its scalability and reliability.

In addition to quantifying the expected rate of failure to achieve constraints after arbitrary user interaction, we also want to quantify the increase in generative power that temporal representations provide. As a measure of generative power, we consider the potential for non-trivial interactions between narrative actions of a domain. The simplest examples of these interactions can be seen in producer-consumer relationships between agent actions, such as when a condition that is added by one action is then deleted by another; or when a fact that is deleted by an action is then replaced by another. In IS, these sorts of interaction appear, for example, in conversations between characters, or when movement between locations is performed. An illustration is provided by the agent action *(board-ship bassanio venice-port)* that covers the movement of Bassanio from the port and interacts with actions that move Bassanio to the port (the "producers" in the relationship). Similar interactions occur between conversational actions, which feature agents entering and exiting the conversation through different actions. Most importantly, actions that do *not* interact in this way provide no scope for the generation of novel interesting narrative situations (similar to the idioms described in [5]).

The identification of these macros is performed in a phase of static domain analysis [9]. For the macros considered here the macro action sequence must be valid (i.e. the preconditions for each action are not violated by prior actions) and the post-conditions of the macro as a whole must differ from the union of its parts.

As a measure of these interesting narrative situations or idioms we counted the number of additional macro actions (i.e. all sets of actions with non-trivial interactions) that result as a consequence of using an explicit temporal representation. We created a set of test domains to measure the presence of macro actions with varying numbers of durative actions. The domain objects and facts were the same as those in our *Merchant of Venice* domain. The number of actions in each domain was similar to that used in the previous evaluation – between 100 and 500. These actions were randomly generated from the domain facts, and had the same number of pre- and post- conditions as those found in the *Merchant of Venice* IS domain. Figure 5 shows the number of additional macro operators present when 25%, 50%, 75% or 100% of the agent actions in the domain were defined as durative actions, and the remainder were compressed, non-durative versions of them (as described in section 5.3).

The results show that the fundamental nature of the durative representation of actions gives rise to a significant increase in the number of possible interesting interactions. For a 500 action domain, almost 100 additional macro actions were seen to appear from the switch to a pure temporal representation – each of which is a new, potential situation or idiom. When moving to domains with larger sets of actions (e.g. planning for the entire *Merchant of Venice* rather than the sub-plot used to illustrate this paper), the number of additional macros relative to the number of actions can be seen to grow at a super-linear rate. As seen in figure 5, applying the durative representation to only a subset of a domain can still realise this increase in generative power.

7. CONCLUSIONS

In this paper we presented the case for the use of an explicit approach to controlling narrative time in IS. This approach involves extensions to the representation of agent actions to include their staged execution time. It also includes a shift to planning architectures that can schedule agent actions with required concurrency. The approach is applicable to a wide variety of different genres: those where timing or pace play a role, those where staging needs to be explored and those where story and discourse may have complex relationships. Overall the approach provides a uniform, consistent, principled and rigorous approach to the problem of time in agent-based storytelling

Our evaluation clearly demonstrated the advantages of a temporal IS approach: at the level of staging, it has been shown to overcome problems of timing of agent actions and provides a mechanism to exploit information about the staging of agent actions; and at the level of narrative generation, it has been shown to increase the generative power of the system. In addition the principled nature of the approach will be advantageous in system production since it removes the time consuming search for empirical solutions.

8. ACKNOWLEDGMENTS

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

9. REFERENCES

[1] J. Allen. Planning as Temporal Reasoning. In *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1991.

[2] R. Aylett, J. Dias, and A. Paiva. An affectively-driven planner for synthetic characters. In *Proc. of 16th Int. Conf. on Automated Planning and Scheduling*, 2006.

[3] D. Balas, C. Blom, A. Abonji, and J. Gemrot. Hierarchical Petri Nets for Story Plots Featuring Virtual Humans. In *Proc. of the 4th AI and Interactive Digital Entertainment Conference*, 2008.

[4] M. Cavazza, F. Charles, and S. Mead. Generation of Humorous Situations in Cartoons through Plan-based Formalisations. In *Proc. of the ACM CHI Workshop on Humour Modelling in the Interface*, 2003.

[5] F. Charles and M. Cavazza. Exploring the scalability of character-based storytelling. In *Proc. of 3rd Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, pages 872–879, NY, USA, 2004.

[6] Y. Chen, B. Wah, and C. Hsu. Temporal Planning using Subgoal Partitioning and Resolution in SGPlan. *Journal of Artificial Intelligence Research*, 26:323–369, 2006.

[7] A. I. Coles, M. Fox, K. Halsey, D. Long, and A. Smith. Managing concurrency in temporal planning using planner-scheduler interaction. *Artificial Intelligence*, pages 1–44, 2009.

[8] W. Cushing, S. Kambhampati, Mausam, and D. Weld. When is Temporal Planning *Really* Temporal. In *Proc. of the 20th Int. Joint Conf. on AI (IJCAI)*, 2007.

[9] C. Dawson and L. Siklossy. The Role of Preprocessing in Problem Solving Systems "An Ounce of Reflection is Worth a Pound of Backtracking". In *Proc. of the 5th Int. Joint Conference on AI (IJCAI)*, 1977.

[10] M. Do and S. Kambhampati. SAPA: A Multi-objective Metric Temporal Planner. *Journal of Artificial Intelligence Research*, 20:155–194, 2003.

[11] S. Donikian. HPTS: A Behaviour Modelling Language for Autonomous Agents. In *Proc. of the 5th Int. Conference on Autonomous Agents*, 2001.

[12] B. Drabble and A. Tate. The Use of Optimistic and Pessimistic Resource Profiles to Inform Search in an Activity Based Planner. In *Proc. of the 2nd. Conf. on AI Planning Systems (AIPS-94)*, 1994.

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

[14] P. Gebhard, M. Kipp, M. Klesen, and T. Ritt. Authoring Scenes for Adaptive, Interactive Performances. In *Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.

[15] C. W. Geib and B. Webber. A Consequence of Incorporating Intentions in Means-End Planning. In *Working Notes – AAAI Spring Symposium Series: Foundations of Automatic Planning: The Classical Approach and Beyond*, 1993.

[16] A. Gerevini and D. Long. BNF Description of PDDL3.0. Technical report, 2005. http://www.cs.yale.edu/homes/dvm/papers/pddl-bnf.pdf.

[17] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[18] J. Hoffmann, J. Porteous, and L. Sebastia. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research (JAIR)*, 22:215–278, 2004.

[19] B. Karlsson, A. Ciarlini, B. Feijó, and A. Furtado. Applying a Plan-Recognition/Plan-Generation Paradigm to Interactive Storytelling. In *ICAPS Workshop on AI Planning for Computer Games and Synthetic Characters*, 2006.

[20] J. Penberthy and D. Weld. Temporal Planning with Continuous Change. In *Proc. of the 12th Nat. Conf. on AI (AAAI-94)*, 1994.

[21] J. Porteous, M. Cavazza, and F. Charles. Narrative generation through characters' point of view. In *Proc. of 9th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2010)*, 2010.

[22] P. Ricoeur. *Time and Narrative, Volume 1*. University Of Chicago Press, 1990.

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

[24] W. Shakespeare. *The Merchant of Venice*. Penguin Popular Classics (New Edition), 2007.

[25] A. Tate. Generating Project Networks. In *Proc. of the Int. Joint Conf. on AI (IJCAI-77)*, 1977.

[26] M. Veloso, M. Pérez, and J. Carbonell. Nonlinear Planning with Parallel Resource Allocation. In *Proc. of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, 1990.

[27] T. Vernieri. *A Web Services Approach to Generating and Using Plans in Configurable Execution Environments (M.S. thesis)*. PhD thesis, North Carolina State University, 2006.

[28] H. Younes and R. Simmons. VHPOP: Versatile Heuristic Partial Order Planner. *Journal of Artificial Intelligence Research*, 20:405–430, 2003.