Distributed Interactive Narrative Planning System

Joe Winegarden and **R. Michael Young**
Department of Computer Science
North Carolina State University
jjwinega@ncsu.edu, young@csc.ncsu.edu

Abstract

(Riedl, Saretto, & Young 2003) describes narrative mediation, a plan-based technique for addressing human user actions that conflict with the story structure maintained by computer-controlled agents within interactive narrative environments. This technique is used to generate real-time system responses to unanticipated user actions while balancing user freedom with story coherence. Narrative mediation has previously only been used for a single plan representing an entire story. A high amount of computation is required to monitor user actions and, in response, replan as appropriate to avoid a threat to the story's consistency. Toward the goal of maintaining plan consistency, use of a single planner is limited in that it can process only a single user's actions. To handle multiple users, distributing the planning burden across several planners requires collaborative communication to maintain global consistency. We propose a method that extends Riedl, Saretto and Young's mediation techniques to allow for mediation of exceptional user actions within a narrative environment consisting of many users.

1. Introduction

Research in interactive narratives has received growing attention in recent years. As noted by Riedl (2003), story is the main aspect of narrative; this is apparent in movies and video games, as well as being the motivation for automated narrative generation. The Mimesis system, described by Riedl, Saretto and Young (2003), allows for a human user to interact with animated agents to play out a story within a virtual world. They have devised a technique of maintaining narrative coherence while allowing a user the freedom to perform any action she desires, including those actions which are not part of the narrative plan. This technique, called narrative mediation, takes advantage of the partiality of knowledge offered by a plan-based model of narrative to detect and react to exceptional user actions that threaten the coherence of a story. When an exceptional action is detected, the system must decide whether to accept the action's effects and work them into the narrative structure, or to replace the effects with an alternate set that is both consistent with the causal structure of the story and maintains a sense of coherence within the narrative structure by being similar in nature of execution. In the former case the system must regenerate the unexecuted portion of the plan structure to include the effects of the action and to reestablish any broken causal connections. This replanning can be computationally expensive when a broken causal link is the first in a chain of events that are critical to the story structure. See Section 2.2 for a more detailed description of narrative mediation.

Earlier work by Riedl, Saretto and Young, leveraging this narrative mediation technique to address exceptional user actions, is only concerned with a single plan representing an entire story. Due to the high amount of computation required to monitor user actions and to replan when necessary due to a threat to the consistency of a plan, use of a single planner-mediator pair is limited in that it can handle only a single user in the face of maintaining plan consistency. To address this issue and allow for mediation with many users and within many narrative plans, we propose to divide the planning and mediation load among several different servers. By dividing the planning in this distributed manner, the amount of computation needed by the planner at each server is reduced to a level that increases the feasibility of addressing user actions in real time.

The greatest challenge in creating a Distributed Interactive Narrative Planning System (DINPS) for the open domain is to maintain global consistency among all of the plans, especially when reacting to unplanned, externally imposed human user actions that cause asynchronous local deviations in the plans. We define maintaining global consistency as preventing a condition, asserted as the effect of any plan step, from being negated, unless the negation can be reversed by reasserting the condition through replanning. In the single user/plan case, when none of the conditions of the plan are threatened by an action of one of its actors or the system response to this action, the mediator can assume that allowing this action will not render the plan inconsistent. However, if multiple plans exist within a shared space and this plan overlaps with any of the other plans through the use of shared objects, then allowing this action could pose a threat to the consistency of one of these overlapping plans and therefore poses a threat to system as a whole. To avoid this threat to global consistency, some manner of collaborative communication is needed among the planners and/or

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

mediators. We propose a method that extends Riedl, Saretto and Young's mediation techniques and allows for mediation of exceptional user actions within a narrative environment consisting of many users and plans.

2. Background

2.1 The Mimesis Architecture

The Mimesis system is a distributed architecture for building, monitoring, and adapting interactive narratives. The system couples a virtual world with a set of intelligent narrative-creating components, separating the work of user interaction monitoring from that of story planning (Young & Riedl 2003). The three components of the Mimesis system that are most relevant to our discussion are the Planner, the Execution Monitor and the Narrative Mediator (see Figure 1). The Planner, responsible for creating a plan that details the actions to be enacted in portraying a story within a virtual world, dynamically updates this plan when a user's actions conflict with current or future steps in the unfolding story. The Execution Monitor accepts a plan from the planner, sending each of the steps in their partial order to other components in the system so they may be executed. The Narrative Mediator receives the original plan from the Planner and examines it for all possible points within the plan where a user's action would threaten the causal constraints for the rest of the plan. It then constructs a mediation policy for each such event, including constructing a new plan for each entry. This policy is sent to the client containing the virtual world, and the client alerts the mediator each time an intervention or accommodation occurs, allowing the mediator to send the client a pre-computed, replacement policy and plan.

The current Mimesis implementation allows only a single Planner, Mediator and Execution Manager, along with a single user, to interact with a single virtual world. We propose to update this architecture, allowing many users to interact within a much larger story world containing several virtual worlds. Each set of Planner, Mediator and Execution Manager may span across several virtual worlds. Our vision is that each of the separate plans within this distributed megaworld will overlap, forming related subplots within the metastory. To allow such overlap, we must ensure that no conflicts arise among the subplans. Conflict avoidance can be ensured by requiring collaborative communication among the Planners.

2.2 Narrative Mediation

Narrative mediation is a process that occurs when a user action is detected. User actions can be classified into three possible categories: constituent, consistent or exceptional. A constituent action is one which exactly follows the story plan, mapping its causal structure directly to a step in the plan as expected to be executed by a user. A consistent action is one which is not constituent but does not threaten the causal structure of the plan and can therefore be allowed to execute without modifying the plan. Exceptional actions are those which threaten the causality of a plan, and if allowed to execute, will render the plan unexecutable and force the system to replan. It is the exceptional actions that the original process of narrative mediation aims to address.

The process of mediation involves applying one of two techniques when exceptional actions are detected; they are intervention to prevent the action, and accommodation to allow it (Riedl, Saretto, & Young 2003). The former technique does not require replanning at the local level because the response generated by the system does not threaten the remaining portion of the story plan. Consequently, the consistency of the local plan in the single planner environment is not threatened. However, in a distributed story planning environment where the individual plans may overlap, the intervention response may threaten the consistency of one or more other plans and therefore global consistency as well. Likewise, the actions previously considered consistent in a single plan may affect global consistency and therefore must be addressed. On the other hand, allowing the action via accommodation and then replanning without the negated condition becomes even more threatening to global consistency. If this same condition is required to be true by the constraints of some other plan within the story realm, then this negation threatens to cause replanning at the global level as well, which can be very computationally expensive. While it may still be possible to reestablish the problematic condition by performing global replanning, it is entirely possible that such a condition might not be able to be reestablished at all, resulting in an inconsistent and therefore unsound plan.

3. Related Work

One concentration within collaborative agent research is the area of leveraging Grosz and Kraus's (1996; 1998) SharedPlans formalization, which has been applied to design a variety of collaborative agent systems. Rich and Sidner's (1998) COLLAGEN is a simple air-travel application that applies SharedPlans theory to the design of an intelligent human-computer communication interface. By constructing a Partial SharedPlan between the agent and the human, COLLAGEN is able to interpret user input discourse, filling in the details of the plan as information is gained from and by the user, without requiring any understanding of natural language.

Not surprisingly, Grosz, Hunsberger and Kraus (1999) have continued to further their own research by using SharedPlans in the construction of three different types of systems: GigAgent, WebTrader, and DIAL. GigAgent is a system that allows for mixed groups of human and computer agents collaborating in teams towards a common goal. Examples of this system's application are scheduling a musical gig (hence the name) or performing computer systems administration activities. The wrapper software of this system handles the actual process of creating and maintaining a SharedPlan for the human user, implicitly allowing for collaboration with the software agents. WebTrader, an e-commerce application, allows intelligent buyer/seller agents to fulfill the requirements of a respective enterprise through collaboration with other buyer/seller agents, representing other enterprises as well as individual humans. In WebTrader, Grosz, Kraus and Hunsberger allow for SharedPlans to be formed among agents working towards a common

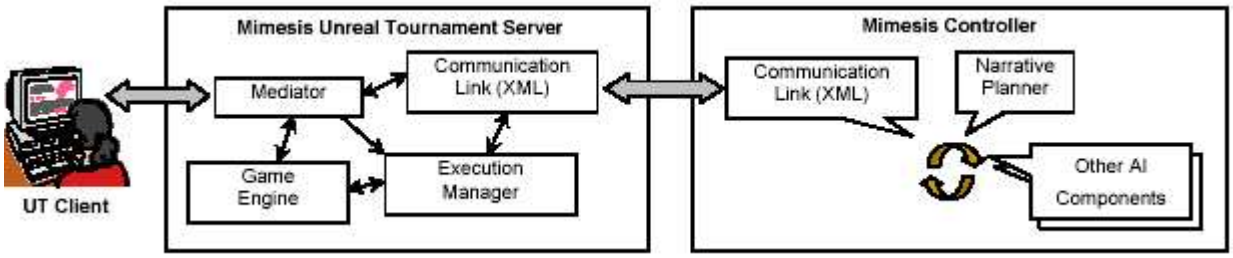

Figure 1: The Mimesis System Architecture

goal, allowing these agents to resolve conflicts and compromise on any conflicting interests so they can achieve the main goal. DIAL is a system that allows people to more effectively communicate with a computer agent by reducing their communication burden through the use of SharedPlans.

Another area of collaborative agent research is that of a distributed teamwork implementation of the theory of joint intentions (Cohen & Levesque 1991). In constructing STEAM, Tambe (1997) builds up a hierarchy of joint intentions, also using performance monitoring and decision-theoretic communication selectivity. Throughout this work Tambe attempts to maintain complete compatibility with SharedPlans theory, making design decisions with respect to joint intentions in a manner that parallels Grosz and Kraus. Unlike much of Grosz and Kraus's theoretical formalization work at the time, however, Tambe's STEAM is a fully implemented model of teamwork with applications in several complex domains.

Durfee and Lesser (1991) offer up a different perspective on distributed planning. They combine the abilities of four different types of tasks within a distributed environment: task-sharing, result-sharing, planning, and organizational structuring. Durfee and Lesser term this unified framework, created from this combination, Partial Global Planning (PGP). They have applied PGP to the domain of distributed sensor networks where communication of the large amounts of information sensed by each node is needed for the nodes to collaborate in forming a global view of the system. Each node in this system is assigned an authority weighting factor, and each PGP sent out by a node is assigned a rating value based on its importance. The product of these two factors, compared with a node's local PGPs, tells a node whether to accept a received PGP and conform to it, or reject the PGP. A node rejects new information when it believes that the information it already has is more current.

Several researchers investigate the idea of applying plan merging techniques to multiagent planning to allow agents to achieve their goals more efficiently. De Weerdt et al. (de Weerdt *et al.* 2001) treat the effects of each step in a plan as a resource that can be used by another agent, allowing the other agent to drop some of its steps and therefore become more efficient. As a result, the agents become dependent on each other. Cox and Durfee (Cox & Durfee 2003) coined the term synergy to describe this sharing of agent action effects to merge plans. However, in later work they propose a method to reduce the amount of agent dependency created by plan merging while still taking advantage of the efficiency gains (Cox & Durfee).

One major difference between our DINPS and the other collaborative systems we've mentioned is that our plans lack a common goal and, especially, that there are users interacting within these plans. Particularly in SharedPlans, the agents in the system are each constructing their own plans with a common goal in mind, and the purpose of their individual plans is to work towards achieving this goal. Although we do, in a sense, have a common goal of global consistency, this goal lies more at a meta-level rather than in the actual planning process. Rather than the planners planning out their own roles in achieving the common goal, each planner has its own separate goals, and the goals can dynamically change to coincide with user actions. We can not rely on intentions because there is no commitment by the planners themselves to the group as a whole. We also believe that if we were to apply plan merging techniques to create efficiency in execution of these plans, then it might make the mediation process even less efficient as a result. It is precisely the overlap in plans that causes potential conflicts and replanning when exceptional user actions occur. Thus, we need a new method of distributed collaboration, described in the following sections, to achieve this global consistency among more than a single planner.

4. System Configuration

Our Distributed Interactive Narrative Planning System (DINPS) generates several separate, but possibly interrelated narrative plans, and enacts these narrative plans within a virtual world consisting of several, distributed subworlds. The Longbow planner created by Young (1994) is used to create, and recreate, the plans. A *Story Server* (SS) component is in charge of managing the communication of each of the steps of a narrative plan to a subworld for execution. SSs also maintain a list of human *Users* that are to take part as actors in the plan, to allow for communication with the *World Servers* (WSs) in which the users are interacting. We allow for narrative mediation of exceptional user actions, and ensure that all plans within the system remain consistent when mediation occurs within any of the plans. A mediation policy is created on a per-user basis for each possible action that the particular user could perform within a subworld. A *Mediator* component is assigned to construct

such a policy using a modified form of the Longbow planner. To coordinate the communication between components in the system, each of the SSs, WSs, and Mediators register themselves with a *Global Controller* (GC). The GC then broadcasts the entire list of WSs to the other components to allow for direct communication. This list is pruned by each component to meet its respective needs, and we assume that the locations and "connected" nature of the WSs remains constant throughout execution, although it may be possible for the environment to dynamically change if allowed to do so. For example, the WSs only need to know about other WSs that are spatially nearby in the virtual domain because Users can only pass to and from these areas. However, plans may span multiple subworlds, even those that are completely spatially separated. Therefore, we specify that a hash table of all of the predicates (objects, locations, etc.) within all of the worlds is kept by the GC. This table contains a list of all SSs whose plans use a particular predicate in at least one of their steps. By using this table, each SS is able to generate a list of only those other SSs that could affect the consistency of its plan, and whose plans could be affected by its own policy decisions. In this manner we are able to avoid having to broadcast all assertions globally, making it more feasible for the necessary communication to occur in real time.

4.1 Initial Plan Creation

Our DINPS consists of several different components working together to unite several different substories into a single coherent story. Each Story Server consults its Planner to construct a narrative plan with a given planning context (set of preconditions, goals, operators, etc). We assume that each Planner is sound and therefore all plans produced by it are locally consistent. To simplify, we also assume that the Planners all have access to a library with a common set of operators, as well as common access to all objects and agent-actors within all of the subworlds. This common, global access will most likely lead to overlapping story plans, which in turn is very likely to lead to conflicts and inconsistencies between the individual plans. It is our goal to maintain both local and global consistency across all of the plans in order to obtain a coherent story line. By consistency we mean that there is no causal link in any of the plans that is threatened by the effects of any step in any of the plans during a given time interval. Consistency must first be established during the initial creation of plans, and then must be maintained throughout execution, even in the event of exceptional user actions.

Once all of the SSs have received their narrative plans from their respective Planners, they must communicate with each other through the use of the global hash table to ensure that no causal link in any plan is threatened by any plan step in another narrative. If a conflict exists, then the conflicting SSs (the "offender" and the "victim") must negotiate to alleviate the conflict. The negotiation process includes the following strategies (in no relative order):

1. offender replans to remove the offending step
2. victim replans to reestablish the broken causal link via the addition of new steps or the removal of the step containing the nullified condition
3. perform a temporal shift in either or both plan(s) such that the nullified condition is reestablished after the offensive step, or such that the step containing the threatened condition happens before the offensive step
4. a third party Story Server's plan contains a step that reestablishes the condition, so the conflicting parties need not alter their plans, only set temporal parameters

Once all conflicts have been resolved, global consistency is achieved and the plans are ready to be executed. However, to ensure that the system is able to react to user actions in a timely manner and maintain this consistency, we first create the mediation policy tables before beginning execution of the plans. This creation process is described in the next section.

4.2 Initial Mediation Policy Construction

Assuming that the initial setup is truly consistent, if the actions represented by the plan steps were the only actions to occur in the worlds, then there would not be any problems maintaining consistency. However, by allowing human users to asynchronously interact with the stories and the environments, we face the threat of having our precious balance ruined by user actions that are not planned for within the story plan structure. Herein lies the need for narrative mediation techniques to deal with such exceptional user actions when they occur.

Narrative mediation during execution in its simplest form at the local level is a lookup in a predefined table of possible user actions. In the case of many users, a single global table for all possible user actions across all of the subworlds could be so large as to make both the construction of the table and the lookup procedures intractable. Thus, we need to distribute the mediation policy information across multiple Mediators.

To speed up the system's reaction time when an exceptional action is detected, we precompute a mediation policy table (MPT) per User before execution begins. An MPT consists of three main fields per entry: a description of the action in question, the interval during which it occurs and the response to be enacted. In the case of accommodation, the table also contains a replacement plan to be sent to the client for use in place of its current plan. See Figure 2 for an example MPT[1]. Each of the entries in an MPT represents an automatic response that the system will take in the event that a user performs the given action some time in the future. Such responses may alter the current state of the world, as well as future steps in any plans in which the user is a participant, both of which may cause an inconsistency in the global plan structure. Therefore we would like our system to be able to generate these potential responses such that the amount of potential conflict with the current plan structure is minimized.

The initial construction of an MPT by a Mediator is essentially the same as that in (Riedl, Saretto, & Young 2003).

[1]Note that we have left out the replacement plan for the accomodation entries in this table for the sake of brevity.

As in the single User case, a Mediator examines all possible points within a plan where a User's action could threaten the causal constraints for the rest of the plan. The Mediator constructs a MPT with an entry for each such event, including a new plan to replace the unexecuted portion of the existing plan in the event that a particular entry is enacted.

When choosing the policy for a particular entry in a MPT, a Mediator generates all possible options for both intervention and accommodation, with a proposed replacement plan for each option, and stores this full search tree in case the policy needs to be changed in the future due to a potential conflict. Initially, a Mediator examines the options and chooses one that avoids conflict with the local plan and enters it into the MPT. This decision process is essentially the same as that used by (Riedl, Saretto, & Young 2003).

At this point we have three options. We can begin execution and, in real time, deal with any conflicts that arise when a mediation response is enacted, having the Mediators try to resolve any potential conflicts before they occur by altering their policies. Secondly, we can try to modify all of the policies before execution begins to minimize the number of potential conflicts between current plan structures and given mediation responses. Finally, we can use a hybrid approach of the previous two options, assuming that the number of conflicts is reduced below some threshold before execution begins, and this number is actively reduced by the Mediators while the Story Servers are executing their plans via a refinement process.

4.3 Exception Handling

Once the initial setup of plans and mediation policy tables has been completed, the Story Servers begin sending out the steps in their respective plans for execution on the World Servers. During this execution period, the actions of each of the Users are monitored by an Execution Monitor. When an exceptional action is detected by a WS, it performs a lookup in the Mediation Policy Table for the given User and executes a response as specified by the corresponding entry in the MPT. The replacement plan for this response is also sent to the respective Story Server to be integrated into its plan. At this point, the consistency of the system as a whole needs to be reconsidered because both the response action and any new steps in the replacement plan might have caused an (current or future) inconsistency in the plan structure of any of the subplans. Hopefully, any conflict that might have arisen at this point has already been resolved during the MPT creation and/or refinement process. In the case that a conflict does exist, we restore the consistency in the system through communication between the Story Servers that are in conflict using the process described in Section 4.1.

5. Example

In this section we describe an example of the distributed planning environment described in this paper. The system consists of three separate Story Servers P1, P2 and P3, each with its own story plan. Story plans S1, S2 and S3 are shown in Figures 3, 4 and 5, respectively. In these figures, actions are indicated by gray rectangles and are assigned unique letters for reference. Solid arrows between actions indicate causal links between one action's effects and another action's preconditions. Text labels are used to show the conditions for each causal link. To avoid unnecessary complexity within the figures, we omit any links between the initial conditions and their occurrences as preconditions for the actions. Any condition that does not have a link explicitly pointing to it is assumed to be an initial condition. Temporal ordering is maintained in a rough left-to-right manner, loosely following the alphabetic ordering of the step labels.

	Action	Interval	Response
1	Shoot(?char, Robber, ?gun)	<0, A>	Interv.
2	Search-contents(?char, bag)	<0, A>	Accomm.
3	Pick-up(?char, book)	<0, A>	Accomm.
4	Shoot(?char, Accomplic, ?gun)	<0, A>	Accomm.
5	Pick-up(?char!=Robber, gun1)	<A, B>	Interv.
6	Shoot(?char, Robber, ?gun)	<A, B>	Interv.
7	Pick-up(?char, book)	<A, B>	Accomm.
	...		

Figure 2: Sample Mediation Policy Table for P1

The three stories share three locations within the global story world: a bank, a coffee shop and a post office. Objects within these locations are shared among the plans, regardless of whether a particular object is used within a given plan. If a user acting within one of the plans picks up a book in the coffee shop, this action may be unrelated to the current plan and the action can be considered consistent for this plan. However, if that same book is required for a character action in the near future of one of the other plans, the user has committed an exceptional action at the global level. All three of the stories share this type of potential conflict due to the spatial overlap of the stories. Two of them are also related in that S2's goal condition (*in letter book*) is required as a precondition to step M in S3; any user action that causes S2 to fail also causes S3 to fail.

S1 is a slightly modified version of the bank robbery story from (Riedl, Saretto, & Young 2003). In this story, a Robber steals money from the bank, and then a bank Guard shoots him as he tries to escape. The story depends on the Robber finding a gun in a bag in the coffee shop and using this gun to threaten a Teller at the bank, receiving the money as a result. In S2, an Accomplice receives a letter at the post office. This letter contains information regarding the whereabouts of one of America's Most Wanted criminals, information which the Accomplice cannot allow to fall into the hands of law enforcement. When a Detective walks into the coffee shop and interrogates the Accomplice about his whereabouts the previous night, the Accomplice stashes the letter in a book so that the Detective will not have access to it. This same book has a role in S3 as well. An ordinary Citizen sits down in the coffee shop to relax and happens to pick up the same book that the Accomplice left behind, with the Most Wanted letter still inside. To her surprise, the Citizen discovers the letter, reads it, and realizes that the information in the letter refers to the same criminal she saw recently on a "Reward Offered" poster in the post office. She turns the letter in to the authorities (the Mailman), receives a healthy reward, and

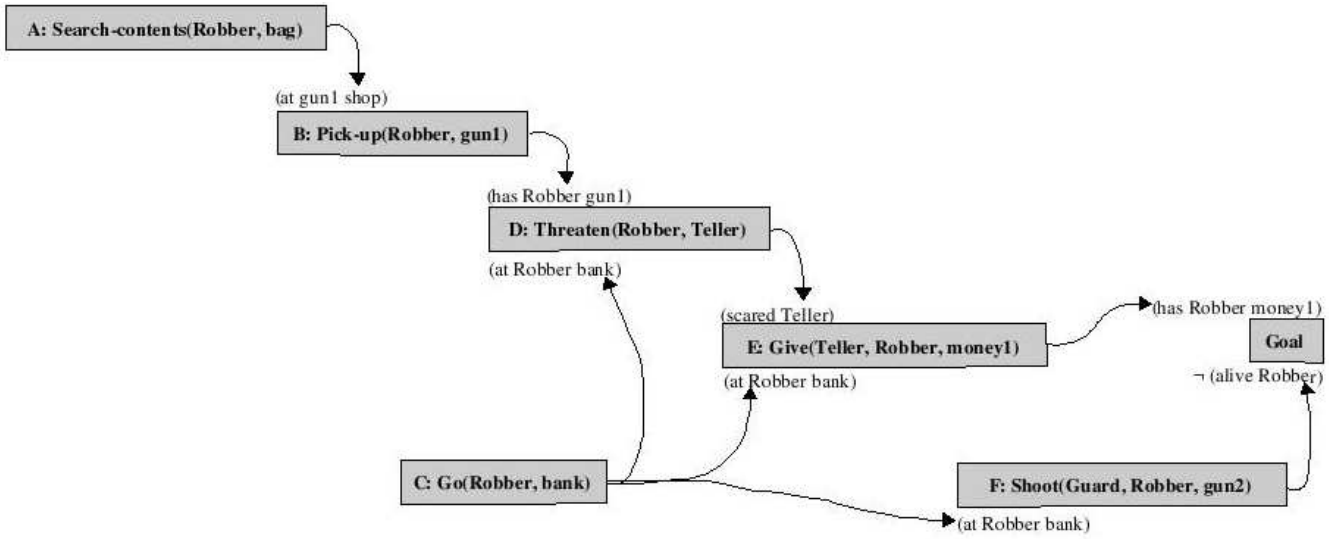

Figure 3: Narrative Story Plan S1 - "Bank Robbery"

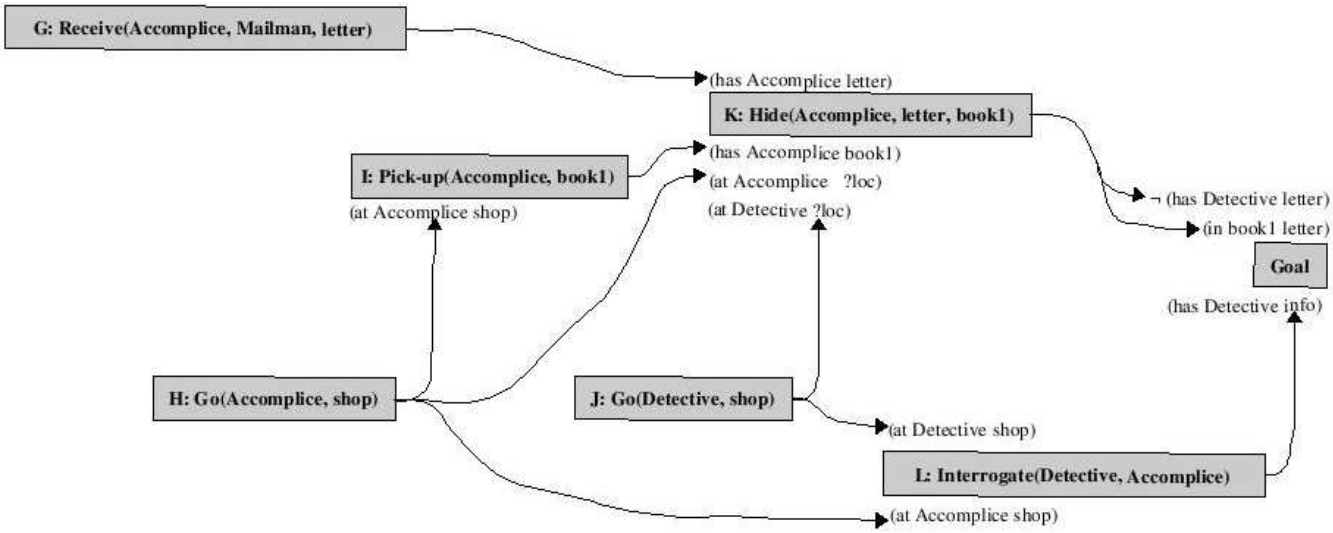

Figure 4: Narrative Story Plan S2 - "The Letter"

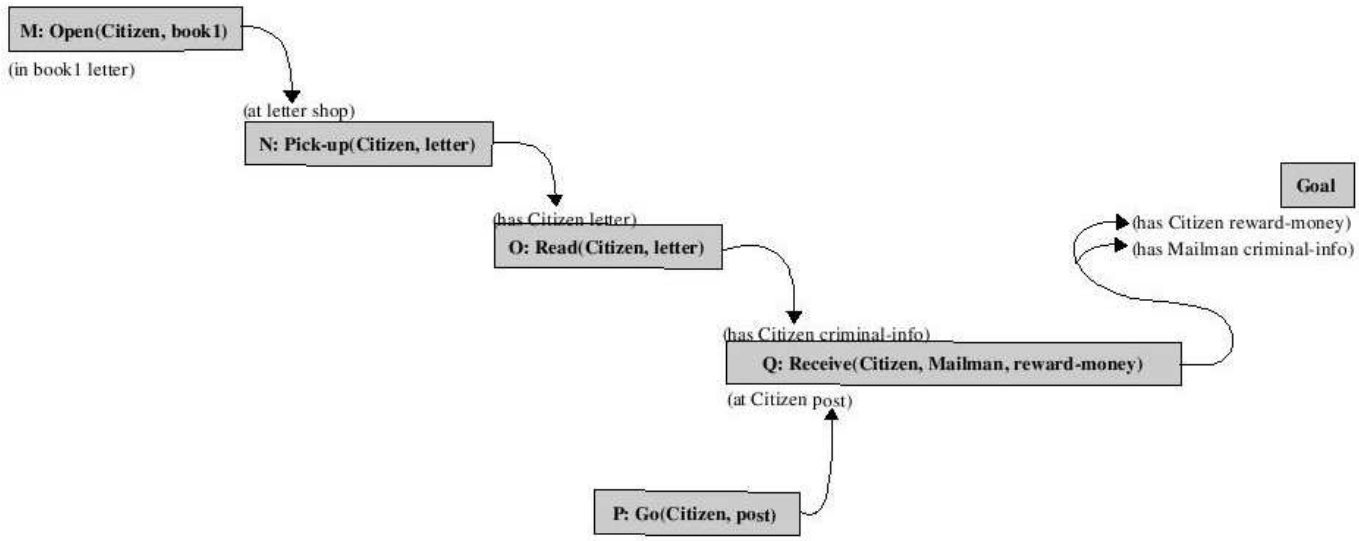

Figure 5: Narrative Story Plan S3 - "Lucky Day"

lives happily-ever-after.

Once P1 finishes constructing S1, it communicates the objects to which it is bound with P2 and P3. Likewise, P2 and P3 generate their own lists of objects in their respective plans and communicate their list with the group.

Next, P1 forms its mediation policy table. At first, P1 forms its MPT based on its local plan only. A partial MPT for P1's local plan is given in Figure 2. Each entry in this table is an exceptional user action that might occur during the execution of S1. The entries indicate the exceptional actions, the interval during which each is considered, and the type of system response to each action. (The replacement plan for each response is not shown here.) For example, entry 1 states that if any character tries to shoot the Robber with some gun between the start of the plan and the time at which the Robber searches the contents of the bag, then the system should intervene. Entry 4 states that the system should accommodate any attempt to shoot the Accomplice character in this same interval. This response for entry 4 is perfectly acceptable in the single-planner case because the Accomplice character does not have a role in S1, so shooting him would not threaten to break any conditions. However, if P1 were to allow the Accomplice to be shot while S2 is executing in parallel, then the consistency of the latter plan would be ruined. By looking in the list of objects used by P2, P1 realizes that P2 uses *Accomplice* and finds out that (*alive Accomplice*) is required by S2. Therefore, allowing the Accomplice to be shot at any time before the completion of action L in S2 threatens global consistency. Thus, P1 updates the response in entry 4 of Figure 2 to be intervention instead of accommodation, given that it is able to execute some failure mode in place of the exceptional *Shoot* action to maintain the global constraints.

Entry 3 in Figure 2 is another example of an accommodation that is acceptable at the local level but produces a conflict globally. If P1 allows a user to pick up the book before the Accomplice is able to do so, then the causal structure of S2 will be ruined, and as a result the structure of S3 will be ruined as well since it depends on the outcome of S2. However, suppose that P1 does not have any failure modes in its operator library that can both take the place of $Pick\text{-}up$ and maintain its local constraints. If P1 selects some failure mode that violates its local constraints then it must reconstruct its plan to resatisfy its constraints. On the other hand, if P1 chooses to accommodate the action, then P2 must replan. Thus, P1 and P2 must decide who replans using the criteria in Section 4.1.

6. Summary

Previously, use of the technique of narrative mediation to address exceptional user actions in interactive narratives has only been concerned with a single plan representing an entire story and a single user interacting within that story. We have extended this technique to allow for multiple planners and multiple users by distributing the planning burden across several planners. This new distribution requires collaborative communication to maintain global consistency among the narrative plans, both during the initial plan creation and as exceptions arise during execution. Many challenges lie ahead to allow for the system to be able to respond to such exceptions in real time while still maintaining global consistency. When resolving a conflict, several heuristic measures will need to be developed to aid the decision process on which planner should replan to resolve the conflict. The creation of a central hash table for objects used in each plan could result in stale information if not constantly monitored.

7. Acknowledgements

This work has been supported by National Science Foundation CAREER award 0092586.

References

Cohen, P. R., and Levesque, H. J. 1991. Teamwork. Technical Report 504, Menlo Park, CA.

Cox, J. S., and Durfee, E. H. Exploiting synergy while maintaining agent autonomy.

Cox, J. S., and Durfee, E. H. 2003. Discovering and exploiting synergy between hierarchical planning agents. In *AAMAS '03*.

de Weerdt, M.; Bos, A.; Tonino, H.; and Witteveen, C. 2001. A resource logic for multi-agent plan merging.

Durfee, E. H., and Lesser, V. R. 1991. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics* 21(5):1167–1183.

Grosz, B. J., and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86(2):269–357.

Grosz, B., and Kraus, S. 1998. The evolution of shared-plans.

Grosz, B.; Hunsberger, L.; and Kraus, S. 1999. Planning and acting together. *The AI Magazine* 20(4):23–34.

Hunsberger, L. 1999. Making shared plans more concise and easier to reason about. In Müller, J.; Singh, M. P.; and Rao, A. S., eds., *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555, 81–98. Springer-Verlag: Heidelberg, Germany.

Rich, C., and Sidner, C. L. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3-4):315–350.

Riedl, M.; Saretto, C.; and Young, R. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*.

Riedl, M. 2003. Actor conference: character-focused narrative planning. Technical Report 03-000, North Carolina State University Liquid Narrative Group.

Tambe, M. 1997. Towards flexible teamwork. *Journal of Artificial Intelligence Research* 7:83–124.

Young, R., and Riedl, M. 2003. Towards an architecture for intelligent control of narrative in interactive virtual worlds.

Young, R. 1994. A developer's guide to the longbow discourse planning system. Technical Report 94-4, University of Pittsburgh Intelligent Systems Program.