# Collaborative Narrative Generation in Persistent Virtual Environments

**Neil Madden** and **Brian Logan**
School of Computer Science
University of Nottingham, UK.
{nem,bsl}@cs.nott.ac.uk

**Abstract**

We describe *witness-narrator agents*, a framework for collaborative narrative generation in persistent virtual environments such as massively multiplayer online role playing games (MMORPGs). Witness-narrator agents allow reports of events in the environment to be published to external audiences so that they can keep up with events within the game world (e.g., via a weblog or SMS messages), or fed back into the environment to embellish and enhance the ongoing experience with new narrative elements derived from participants' own achievements. The framework consists of agent roles for recognising, editing and presenting reports of events to a variety of output media. Techniques for recognising events from an ontology of events are described, as well as how the agent team is coordinated to ensure good coverage of events in a large scale environment.

## Introduction

The last few years has seen the creation of a large number of persistent virtual environments for entertainment, e.g., massively-multiplayer online role-playing games (MMORPGs). These environments provide diverse interactive experiences for very large numbers of simultaneous participants. However, their sheer scale and the activities of other participants makes it difficult to involve players in an overarching narrative experience. One of the main attractions of such environments is the ability to interact with other human players. Such interaction precludes the possibility of an omniscient narrator who 'tells a story' which structures the user's experience, as much of this experience is driven by the (unknowable) thoughts and feelings of other players. Various approaches have been adopted in an attempt to solve this problem, such as guiding players to follow pre-designed storylines (Young 2001), or giving them goals to achieve that advance the storyline, or by having developers ('*dungeon masters*') adapt the narrative to the real-time actions of players. However these solutions can be inflexible, and/or fail to take player interaction into account or do so only at the collective level, for groups of players.

In this paper we investigate a different approach, in which embodied *witness-narrator agents* observe participants' actions in a persistent virtual environment and generate narrative from reports of those actions (Tallyn *et al.* 2005). The generated narrative may be published to external audiences, e.g., via community websites, Internet chatrooms, or mobile telephone text messages, or fed back into the environment in real-time to embellish and enhance the ongoing experience with new narrative elements derived from participants' own achievements. Such in-world narration may enhance the enjoyment of participants, and being talked about is a way of building a reputation and progressing in the community of players. The possibility of appearing in a report (e.g., when doing well in a game) can help to motivate players, and the narrated events can, in turn, influence the participants' future activities thus helping to drive events in the environment. The latter may be particularly important in, e.g., MMORPGs, where the quests and challenges are periodically reset.

A key feature of our approach there is no overarching narrative structure, but rather many strands of narrative which emerge from the interactions of many participants. Participants are free to act within the normal constraints of the virtual environment, and it is the task of the system to synthesise reports of activities within the environment into narrative elements which are relevant to a particular user or group of users. Some narrative strands, e.g., those relating to major conflicts or relating to the 'backstory' of the environment, may be widely shared, while others, e.g., an account of an individual quest, may be only of interest to a single user.

In previous work (Fielding *et al.* 2004a; 2004b), we presented the *reporting agents* framework for generating reports and commentary on events in multiplayer games primarily for an external audience. This work focused on environments with relatively small number of possible interactions between participants, but with a fast pace of action, such as Unreal Tournament. In this paper, we report on work to develop this framework to support the concept of witness-narrator agents. Like reporting agents, witness-narrator agents are embodied in the environment and observe events which can be reported to an external audience. However witness-narrator agents are also capable of presenting reports directly to participants in the virtual world. We describe the application of the framework to the popular *Neverwinter Nights* multiplayer role-playing game[1]. Never-



[1] `http://nwn.bioware.com/`

winter Nights supports around 70 simultaneous players on a single server and allows servers to be linked together to create very large environments. In addition to the much larger scale of both the environment and the number of participants compared to Unreal Tournament, Neverwinter Nights supports a much richer variety of interactions between participants, and a correspondingly richer ontology of events that can occur within the environment. These challenges have required the redesign of the underlying reporting framework and the development of new components for coordinating the team of witness-narrator agents, and for identifying and describing complex interactions occurring in the world.

The remainder of this paper is organised as follows. In the next section we provide an overview of the witness-narrator agents framework and briefly explain how narrative is generated in response to user interaction. We then describe the three basic capabilities provided by agents in the framework, reporting, editing and presenting, and sketch how these capabilities are combined to create different kinds of agent. We then go on to discuss the problems of coordinating teams of agents across large environments, before concluding with a discussion of related approaches and our ongoing work to evaluate the research.

## A Framework for Collaborative Narrative

The witness-narrator agent framework is organised as a society of agents, with different types of user interacting with different types of agent, see Figure 1. We distinguish two main types of user: *participants* in the virtual environment, who are the subject of the narrative; and an external *audience* who are not (currently) embodied in the world but read accounts of the action via some other medium, such as a web page, IRC channel or text messages. Participants interact with *witness-narrator agents* using the standard mechanisms for interacting with NPCs within the game (i.e., menus and text output). Witness-narrator agents are embodied in the environment and observe events in the same way as a human player. Members of the audience interact with *commentator agents* tailored to a specific output channel. Commentator agents are not embodied in the environment and have no first hand knowledge of events in the environment, relying on the witness-narrator agents to provide this information.

The agents function both as an implementation technology, and, more importantly, as an interaction framework which both structures the user's experience of the narrative and allows them to (partially) shape the development of individual narrative strands. The witness-narrator agents' embodiment in the game, and their resulting "first person" view of events explains both the ultimate source of the narrative and makes explicit the limitations on what is knowable about the game. Like human players, their (individual) view of events is limited to the actions of human players, and while they can speculate about the thoughts, feelings or motives, of other players, these remain ultimately opaque.

### Witness-Narrator Agents

The concept of a witness-narrator agent draws on ideas from literary theory, and in particular the notion of 'narrative voices', which describes the various relationships between a narrator and the world being narrated (e.g., (Rimmon-Kenan 1983)). Witness-narrator agents are embodied within the environment (rather than being omniscient) and both observe what is occurring in the environment as well as narrating these events to audiences both inside and outside the environment. The agents are 'witnesses' rather than protagonists, as they do not actively play a part in the activity of the world beyond their presence and the narration they provide (Tallyn *et al.* 2005).

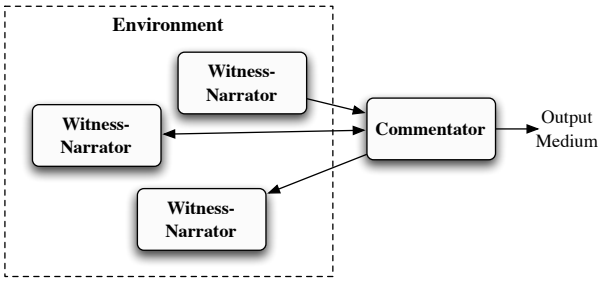


Figure 1: Overall agent framework, showing embodied witness-narrator agents and non-embodied commentator agents.

Embodiment provides an interface to the narrative system which is seamlessly integrated with the virtual environment. Participants can interact directly with the witness-narrator agents in same way as other NPCs. For example a player may approach a witness-narrator agent to request information about current events elsewhere in the environment, or that the agent accompany them as they progress through the game, to share reports of their activities with others. Participants can also interact indirectly with the agents. Being embodied in the environment grants the agents (approximately) the same access to events as a human participant. Participants can therefore determine when they are being observed, and what information an agent is likely to be able to obtain given its position relative to the participant. As a result, players can try to avoid the agents, or can modify their behaviour when around them. For example, participants may wish to keep details of their strategy secret from their opponents in order to preserve an element of surprise. Conversely, players can deliberately try to influence the agents by approaching them, either 'acting up' (e.g., celebrating a victory) or perhaps targeting specific messages at particular individuals in the outside world. We believe that such control over what gets reported is an important part of responsible reporting of events to other participants or an external audience.

### Collaborative Narrative Generation

Users interact with the system by making requests for information about past, present, or future events and by rating the information produced in response to their request. Requests for information are represented within the system as *focus goals*. Focus goals determine which of the events observed by a witness narrator agent are considered 'interesting'. A focus goal consists of four components: a (par-

tial) description of the events expected; the area of the environment in which the events should occur; a time at which the focus goal is active; and an interval specifying how frequently to generate narrative reports. Events which match a focus goal form the basis of *reports*. Multiple events may be summarised in a single report and reports may in turn be aggregated into higher-level reports that summarise a large number of events occurring over an extended period.

Focus goals generated in response to user requests may refer to past or current events, or to future events. For example a participant may ask a witness-narrator agent what their friends or competitors are currently doing in the environment, or about notable events which took place at the current location in the past. Alternatively, if a participant is about to engage in actions which they consider may be of general interest (or of which they want a personal record), they can ask a witness-narrator to follow them and observe their actions. Similarly, audience members may request information about past or current events in the environment, or coverage of anticipated future events, such as a battle or the actions of another participant. In addition, witness-narrator agents are able to autonomously generate focus goals in response to specific events in the environment (commentator agents do not autonomously generate focus goals). Autonomously generated focus goals always refer to current or future events and are always specialisations of existing focus goals. All witness-narrator agents have an *a priori* set of high-level focus goals which can be used as a basis for autonomous goal generation in addition to any user-specified focus goals. For example, a witness-narrator agent which is following a participant, may notice a battle taking place nearby. The agent already has an (*a priori*) focus goal indicating that such a battle is of interest to the system and so will generate a more specific focus goal relating to that particular event (e.g.,, specifying the exact location and expected time duration of the event). Other witness-narrator agents (who are not already engaged) can then be recruited to handle this specific event, which otherwise might be overlooked.

A focus goal generated in response to an audience request, or which cannot be achieved by the witness-narrator agent that generated it (either because it lacks the first hand knowledge necessary to answer the query or because the task it implies is too large for a single agent) are broadcast to all witness-narrator agents, allowing relevant reports to be forwarded to the originating agent. Broadcast focus goals referring to events occurring in the future over a large area or an extended period, or which are likely to be of interest to a wider audience, may give rise to the formation of a team of agents if this is necessary to provide adequate coverage of the events or dissemination of the resulting narrative (as described in more detail below).

Reports are the system's internal descriptions of events and are not presented directly to users. Rather sets of reports which match a focus goal are rendered into *narrative presentations* in one of a number of formats. If the focus goal specifies events in the past or present, this is done immediately. If future events are specified, the narrative may be produced once, e.g., at the end of a specified time period, or periodically, e.g., a daily update to a weblog. Narrative production takes into account the specific constraints of the output channel, e.g., detail present in a weblog may be omitted from SMS messages.[2] Users may 'rate' the narrative produced in terms of its interestingness. Reports which are rated as uninteresting are forgotten by the agents to avoid exhausting system memory. Conversely the most interesting reports are retained, forming a kind of collective memory or 'user generated backstory' of the environment which can be used to satisfy future user requests for information regarding past events.

## Agent Capabilities

Each agent in the witness-narrator framework provides a variety of different *capabilities*. A "capability" is a description of some functionality that an agent offers, such as *reporting*, *editing* or *presenting* reports. Agents communicate their capability descriptions to each other to facilitate team formation in response to broadcast focus goals. A capability is described in the system by a ground atomic formula. Each capability can be specialised by providing parameters that specify particular details. For instance, an agent may advertise that it has a capability to report from a particular region of the environment or to publish to a particular IRC chatroom. Capabilities are used to decide which agents will perform which *roles* in a particular agent team.

Capabilities are implemented as independent components, or *modules*, which can be plugged together in various configurations to create an agent. The details of how these capability-specific modules are plugged together and interact with each other is described in the next section. In this section we describe the three basic capabilities provided by agents in the witness-narrator agent framework: reporting, editing and presenting reports.

### Reporting

The reporting capability is responsible for detecting, recognising and characterising events that occur in the environment. It takes as input one or more focus goals, which describe what events the agent should concentrate on reporting, and produces as output a series of reports of matching events. There are two main sub-tasks involved in reporting: finding interesting events (described in the next section), and then recognising and forming a report of those events. These two tasks are handled independently, with focus goals being used to coordinate the activities of each module. The reporting module constantly monitors incoming percepts from the environment and attempts to classify them according to a low-level *event ontology* that is mostly specific to a particular environment. Once an event has been recognised it is then matched against active focus goals to see if it is of interest. If so, then a report is formed from the event and immediately dispatched to interested agents, otherwise it is discarded.

[2]Narrative presentation by witness narrator agents in the environment is only possible if the agent can interact with the participant, which is why focus goals generated by participants referring to future events are limited to "follow me" type requests.

When reporting on an event, there are a number of basic questions that need to be answered by the report: *What* happened? *Where* did it happen? *When* did it happen? *Who* was involved? *How* did it happen? and *Why* did it happen? Some of these questions are easier to answer than others. We assume that the *where*, *who*, and *when* questions are trivially observable from the environment. *What* happened is largely a domain-specific question, as the types of events that can occur will vary to some degree from environment to environment. For this reason, we separate this aspect into an *event ontology* that can be developed independently of the rest of the framework. The ontology is represented in OWL-DL, allowing flexibility in describing how low-level events combine to describe higher-level events (for instance, individual aggressive actions form part of a larger battle involving many participants). The higher levels of the ontology are largely independent of the particular environment, and most of the agent architecture refers only to these levels. This approach is similar to that used in classical approaches to plan recognition, e.g., (Kautz 1987), where observations are matched against an event hierarchy to find a set of candidate plans that could explain the observed actions. *How* or *why* an event happened is largely the concern of the editing capability, discussed below.

## Editing

The reporting capability produces reports of low-level events (for instance, individual actions of participants). The primary responsibility of the editing capability is to collate and edit low-level reports from multiple agents in the environment. In particular, editing involves combining low-level reports into higher level reports of events taking place in a wider area. For instance, multiple reports of combat between individuals in a particular region may indicate a large-scale battle occurring between two or more teams of participants. These high-level event descriptions are encoded into the upper-level event ontology (described in the previous section). Specific heuristic rules are implemented to recognise these higher level events from multiple lower level events. These rules are implemented using generic descriptions from the upper-level ontology, allowing them to be re-used in similar environments. For example, a high-level event rule may be concerned with instances of "combat", whereas a particular environment may have an ontology describing particular weapons and types of combat unique to that environment (for instance, spells or futuristic weaponry). The editors abstract from these details using the subsumption relationship in the ontology, while preserving those details in the reports that are sent to presenters.

To describe *how* an event occurred, the order and causal relationships between events are explicitly recorded. For instance, if one participant attacks another and the victim subsequently dies, then this causal relationship is recorded by specific rules. These rules are again described using only concepts from the upper-level ontology that are common to many environments.

## Presenting

The presenting capability is the primary interface between the witness narrator agent framework and the users. It is responsible both for formatting reports for presentation via some output medium and allowing user to rate the resulting narrative, as well as allowing users to specify which events they are interested in.

Narrative presentation of reports consists of three main stages:

1. *Content determination* decides which events to include in a presentation and which details of those events.
2. *Narrative generation* converts these declarative event descriptions into a prose narrative at an appropriate level of detail.
3. *Output formatting* formats the prose narrative for a particular output medium (such as HTML, an Atom newsfeed, or an IRC message).

At present, each of these stages is performed using simple mechanisms, as the main focus of our work is on the collaborative aspects of narrative generation, rather than on producing a polished final narrative.

The high-level reports produced as a result of editing are declarative structures describing a particular event. Each structure contains fields describing what happened, where, when, and who was involved. Each event description may also have a number of sub-events which describe in finer detail how the event unfolded. In general then, an event description forms a tree structure, with leaves representing the lowest-level details of what happened (for instance, movements and actions of individuals) while the root of the tree represents a high-level overview of the event.

In the content determination phase, the presenting capability first matches received reports against the focus goal for which a narrative is being created. Once events have been selected, the level of detail appropriate for the narrative is determined. For example, output to a weblog may involve using the entire event structure in the report, whereas a SMS message will require much less detail. The level of detail is specified as a simple limit in the depth to which an event description is traversed to extract information. Narrative generation currently makes use of a relatively simple text template scheme. The tree of event descriptions is traversed to successive depths (up to the depth limit) and each level is matched against a number of rules which extract relevant information and plug it into pre-designed text templates. To avoid overly repetitive text, some variation is allowed in choosing words to describe the entities that are referred to in the event description. After this narrative prose has been constructed, a final phase adds appropriate formatting for a particular output medium. For instance, one output formatter wraps the prose in an Atom XML news entry description which can then be published to a variety of web publishing platforms using a standard Atom publishing API. Figure  shows an example of the prose output produced by the presenter capability.

The presenting capability also forms the interface for the generation of focus goals in response to user requests and for

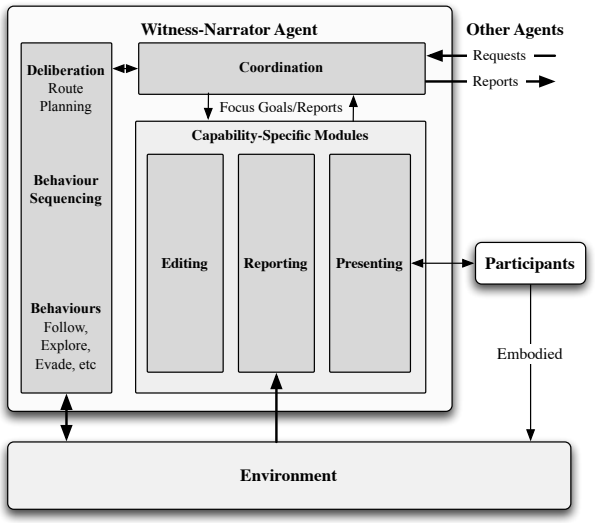


Figure 2: Witness-narrator agent architecture.

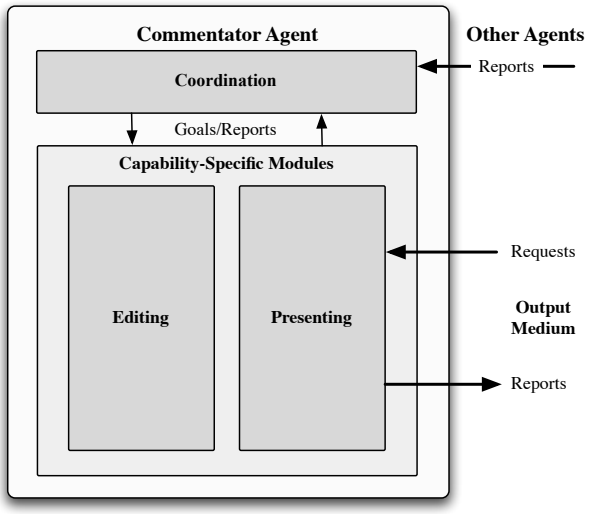


Figure 3: External commentator agent architecture.

rating the resulting narrative presentation. How users communicate requests is specific to each output medium. For instance, a weblog presenter may provide a form where members of the audience can submit details of types of events they are interested in receiving reports about. Similarly, participants can approach a witness-narrator agent and engage it in conversation. A simple menu-based dialogue is conducted in which the user can request reports of, e.g., particular (past) events, or receive the agent's personal take on recent events that it has witnessed directly.

## Agent Architecture

The basic architecture of the agents in the witness-narrator framework is organised into a number of layers, and is implemented in a variant of AgentSpeak(L) using the Jason interpreter (Bordini, Hübner, & Vieira 2005).

The top-most layer of the architecture deals with deliberative processes, and in particular with coordination with other agents. Deliberation involves deciding which focus goals to adopt, planning how to achieve these goals, and handling communication and coordination with other agents. This top-most layer is generic to each agent and implements the overall coordination strategies and protocols used by the agent society as a whole. Once this layer has adopted a goal and formed a plan for how to achieve it, it passes this information down to the capability specific modules. These layers are not generically implemented, but are instead divided into a number of capability-specific modules:

- A *reporting* module takes care of event detection and recognition. This module can spot events from low-level perception data and record important details as events happen.
- An *editing* module that is capable of combining reports from multiple sources, assessing accuracy, and carrying out higher level event recognition.
- A *presenting* module that is responsible for communicating reports to particular output media.

The modules are implemented as collections of Jason plans and rules. For example, the reporting component contains a set of plans and rules for detecting and recognising events occurring in the game world. These modules are combined in various configurations to create individual agents. There are two basic agent configurations currently used in the witness-narrator framework: embodied witness-narrator agents, shown in Figure 2, and non-embodied commentator agents, shown in Figure 3.

Embodied agents have an additional component that deals with movement in the world. This component consists of several layers. The top-most layer deals with deliberation and route-planning: deciding where the agent will be at what times, and how to get there. This layer acts as an extension of the coordination layer, and is called on to decide if it is feasible to commit to a particular focus goal (i.e., if the agent is able to reach the area specified in the focus goal at the required times). Once a route plan has been adopted it is passed down to a *behaviour sequencing* layer for execution. This layer determines which behaviours should be performed in what order to carry out the plan. The behaviours themselves are semi-autonomous processes that interact directly with the environment to control steering and other effects. These behaviours make use of the primitive actions permitted by the environment. A number of basic steering behaviours have been implemented, adapted from (Buckland 2005):

- **Explore**: the agent wanders around the environment in search of events. This is the default behaviour if there are no more specific tasks to accomplish, and is implemented as a random walk within the current area of the environment.
- **Follow**: the agent tracks a particular player, in the expectation that they will do something interesting. It is as-

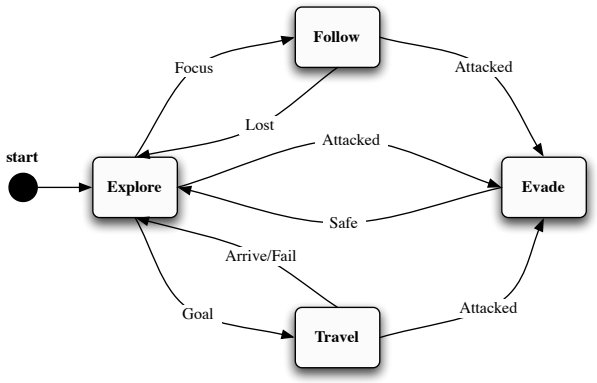


Figure 4: Embodied agent movement behaviour state machine.

sumed that interesting events typically occur around players, so it is useful to be able to follow a player.

- **Travel**: the agent navigates to a particular location, avoiding obstacles.
- **Evade**: if the agent is attacked or told to go away, it should take steps to avoid the participant.

Only a single behaviour is active at a time, and the scheduler uses a simple finite state machine, depicted in Figure 4, to determine the current active behaviour. The transitions between behaviour states are triggered by external events and goals. For instance, if the agent is attacked, it enters the *evade* state and flees from the attacker. Once the agent is safe again, then it reverts to the default *explore* state. The *travel* and *follow* states are entered according to the plan being executed. In addition, the scheduler can be interrupted and returned to the start state if a new plan is to be executed.

Each embodied agent is also provided with an *avatar* which is the in-world representation of the agent. The avatar is provided by the virtual environment (in this case, the *Neverwinter Nights* game environment), and is controlled by the witness-narrator agent sending commands to the game engine to perform actions on behalf of the agent, and to retrieve information about what the agent can currently sense of the environment. The avatar of the agents appears as a small 'gnome' character in bright purple and green clothing. This ensures that the agents are easily recogniseable (no other creatures or players have the same appearance), and also there small stature helps to minimise the interference caused by their presence (i.e., they are less likely to obscure the view or get in the way).

## Agent Coordination

In order to generate compelling narrative from a large-scale environment it is necessary to ensure adequate coverage of events occurring within all areas of the game world, and to adapt to changing conditions to ensure that each event is covered by a sufficient number of agents. This is the function of the coordination layer in the architecture, which draws on ideas from the literature on teamwork (Cohen & Levesque 1991; Scerri *et al.* 2004).

### Team Formation

A team of agents is formed to handle each broadcast focus goal which specifies future events. Each focus goal requires that some team of agents commit to it. A single team can take on multiple focus goals, but typically a new team will be created for each focus goal. Individual agents can belong to multiple teams, and teams are hierarchically structured with members of a subteam also being members of a larger team. Initially, there is a single team that includes all agents in the system and attends to some general focus goals (such as reporting on all deaths that occur anywhere in the environment).

The agent that generated the focus goal is known as the coordinator, and is responsible for recruiting agents to work towards the goal, and for ongoing coordination of the agent team. This is the case even if the originating agent is not itself able to contribute to the team. For instance, if an agent is committed to covering some event but notices another interesting event en route, then it will attempt to recruit other agents to cover the goal while still carrying on to its original destination. This ensures that all noticed events are covered (if possible) while avoiding an agent having to drop a commitment.

The initial team formation phase involves determining which agents are available to work on the goal and what capabilities they can offer. To achieve this we use a Contract Net based protocol. Firstly, the coordinator broadcasts a general call for participation, including the details of the focus goal. Each agent must then decide if it can commit to the goal and whether to make a bid to be on the team. In its bid, an agent includes a list of its capability descriptions along with the times at which it is available to work for the team.

Agents determine if they are available to work on a particular focus goal using a simple goal arbitration scheme. Each agent considers only its position within the environment (if it is embodied) over time. An embodied agent keeps track of the locations it is committed to being in and during which time intervals, and uses this information to determine if a new focus goal is compatible with its existing commitments. If an agent can attend to a focus goal at any time when that goal is active, then it will submit a proposal to join the team, including information on when it is available and what capabilities it can offer (presenting, editing, reporting). An agent may commit to as many focus goals in whatever combination of roles that it believes it can achieve.

### Role Assignment

Once all bids have been received (or the proposal deadline is reached), the coordinator then moves to assigning *roles* to team members. To do this, it generates a set of role requirements consisting of a particular capability pattern that an agent must perform, along with an *ideal* number of agents required for that particular role. Role requirements are patterns which can be matched against capability descriptions to determine if a particular agent is suitable for a particular role.

Team formation is approached on a best-effort basis. The only hard requirement is that at least one agent must commit


Figure 5: Screenshot of *Neverwinter Nights* showing witness-narrator agent.

to each of the three role types (reporter, editor, presenter). The coordinator is responsible for initial role assignment, and also for on-going coordination of the team, such as arranging cover for agents that become unavailable, or recruiting new agents that become available over time.

The coordinator agent tries to assign agents to roles to ensure the greatest possible coverage of the focus goal (measured by time at which agents are available), up to the ideal number of agents specified in the role requirement. Once roles have been assigned, each agent is informed of its expected task by a message including the specific role information and the times at which the agent is expected to commit to the role. At this stage, each agent must recheck its commitments (in case they have changed) and can either confirm the commitment (perhaps with a slightly altered schedule) or can refuse (in which case, the coordinator will attempt to reassign the role). Once the final role assignments have been agreed, the coordinator broadcasts the information to all members of the team so that they know who is responsible for what.

## Related Work

A variety of approaches to interactive narrative have been developed, ranging from systems with an overarching plot structure, to character-based systems where the narrative emerges from interactions between agents, and combinations in between. Of these, our work is closest to the character-based systems, such as Improv (Perlin & Goldberg 1996) where narrative is generated as a result of interactions between complex characters, or the hybrid approach adopted in the Mission Rehearsal Exercise Project (Swartout *et al.* 2001) which combines pre-scripted actors with complex agent characters that make decisions based on an emotion model. However, the witness-narrator approach differs in that the agents are primarily passive witnesses rather than protagonists in the narrated world, and it is the actions of human players that are being narrated rather than a story generated from the interactions of agents.

**Dragon slain in Etum Castle District.**

*An ancient dragon was slain in Etum Castle District within the last hour. Lance Bannon, a powerful mage, delivered the fatal blow by casting a fireball at the dragon.*

*It all started when Jim Fingers, a young rogue, attacked the dragon with a sword. The ancient dragon slashed Jim Fingers with its talons. Lance Bannon, a powerful wizard, cast invisibility. Oliver Ranger, a fighter, stabbed the dragon with a dagger. The ancient dragon cast a fireball at Jim Fingers. Lance Bannon cast a fireball at the dragon. Finally, the ancient dragon died.*

Figure 6: Example prose output of the presenter capability for a simple event report.

Our work also has similarities to commentary systems that have been developed for multiplayer games, such as the TTM module for Unreal Tournament 2003[3] which can deliver a running commentary to an IRC channel while a game is in progress, or the commentary agents developed for the RoboCup soccer simulation (André *et al.* 2000). However these commentary systems are omniscient in that they are provided with global overall knowledge about the current state of the game. From this information they then infer the current interesting action and form prose text describing these events. The witness-narrator approach we describe here instead relies on embodied agents with a limited view of the environment. We believe this approach is both more scalable to large environments and also provides participants with much more control over what gets reported and how it is presented.

## Summary and Discussion

We have described witness-narrator agents, a framework for narrative generation in persistent virtual environments and its implementation as a multi-agent system, see Figure 5. Our approach is distinguished by two key factors. First, there is no single overarching narrative, but rather the narrative consists of many strands. Some narrative strands, e.g., those relating to major conflicts or relating to the 'backstory' of the environment, may be widely shared, while others, e.g., an account of an individual quest, may be only of interest to a single user. Second, the generation of narrative is collaborative, in that users have both direct and indirect control over which events are ultimately narrated. In particular, users have control over:

- which events are performed — each participant determines which events are potentially observable
- which events are observed — participants have negative control over which events get reported, in that they can ask the agents to go away, or simply avoid them
- which events get reported — users have positive control in determining which events the agents will search out and

[3] `http://www.planetunreal.com/ttm/`

report

- which events are remembered and hence can form part of future narratives — users can rate reports, which determines which reports (and hence which actions/events) become part of the "official" collective narrative or backstory of the environment.

More generally, one role of narrative is to develop a sense of shared values and experience, e.g., "what it means to be a player of *Neverwinter Nights*". In this case, the function of the narrative is not just to entertain, but to help foster a sense of community, through participation in and shared experience of the narrative. We believe the witness-narrator agent framework can be seen as a first step towards this goal for the domain of persistent virtual worlds.

The framework as described in this paper is implemented and can produce simple narratives from *Neverwinter Nights*. Our current work is focused on improving the event recognition capabilities of the framework and evaluating the work from both a technical point of view, as well as planning a more extensive study to investigate how people react to the presence of witness-narrator agents, and how the narratives themselves are perceived. We also hope to evaluate the framework using other environments, such as for simulation and training. This should be relatively straightforward due to the use of a replaceable low-level event ontology that can be adapted to new environments while the majority of the implementation remains unaltered.

The current implementation has a number of limitations which we plan to address in future work. An area of particular interest is determining *why* an event occurred, which we are addressing using techniques from the plan recognition (Carberry 2001; Albrecht, Zukerman, & Nicholson 1998) and explanation literature. We hope this *"motive recognition"* capability will enhance the relatively factual narratives currently produced with more speculative information about possible motives for a participant's actions. Motives are recognised based on past actions (for instance, "revenge" is recognised based on the history of aggressive encounters between players), and also by attempting to infer the current goals and plan of a participant (e.g., a player may be attempting to complete a particular quest). In addition, the current presentation layer can be improved by incorporating more sophisticated techniques from natural language generation, e.g., (Reiter & Dale 1997).

## References


Albrecht, D. W.; Zukerman, I.; and Nicholson, A. E. 1998. Bayesian models for keyhole plan recognition in an adventure game. *User Modelling and User-Adapted Interaction* 8(1–2):5–47.

André, E.; Binsted, K.; Tanaka-Ishii, K.; Luke, S.; Herzog, G.; and Rist, T. 2000. Three RoboCup simulation league commentator systems. *AI Magazine* 21(1):57–66.

Bordini, R. H.; Hübner, J. F.; and Vieira, R. 2005. Jason and the golden fleece of agent-oriented programming. In Bordini, R. H.; Dastani, M.; Dix, J.; and Seghrouchni, A. E. F., eds., *Multi-Agent Programming: Languages, Platforms and Applications*. Springer-Verlag. chapter 1, 3–37.

Buckland, M. 2005. *Programming Game AI by Example*. Texas, USA: Wordware Publishing.

Carberry, S. 2001. Techniques for plan recognition. *User Modeling and User-Adapted Interaction* 11(1–2):31–48.

Cohen, P. R., and Levesque, H. J. 1991. Teamwork. *Noûs* 25(4):487–512. Special Issue on Cognitive Science and Artificial Intelligence.

Fielding, D.; Fraser, M.; Logan, B.; and Benford, S. 2004a. Extending game participation with embodied reporting agents. In *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2004)*. Singapore: ACM Press.

Fielding, D.; Fraser, M.; Logan, B.; and Benford, S. 2004b. Reporters, editors and presentors: Using embodied agents to report on online computer games. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, 1530–1531.

Kautz, H. A. 1987. *A Formal Theory of Plan Recognition*. Ph.D. Dissertation, University of Rochester, Rochester, NY. Tech. Report TR 215.

Perlin, K., and Goldberg, A. 1996. Improv: A system for scripting interactive actors in virtual worlds. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, 205–216. New York, NY, USA: ACM Press.

Reiter, E., and Dale, R. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3(1):57–87.

Rimmon-Kenan, S. 1983. *Narrative Fiction: Contemporary Poetics*. Routledge.

Scerri, P.; Xu, Y.; Liao, E.; Lai, J.; and Sycara, K. 2004. Scaling teamwork to very large teams. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS '04)*, 888–895.

Swartout, W.; Hill, R.; Gratch, J.; Johnson, W. L.; Kryakakis, C.; LaBore, C.; Lindheim, R.; Marsella, S.; Miraglia, D.; Moore, B.; Morie, J.; Rickel, J.; Thiebaux, M.; Tuch, L.; Whitney, R.; and Douglas, J. 2001. Toward the Holodeck: Integrating graphics, sound, character and story. In *Proceedings of the 5th International Conference on Autonomous Agents*.

Tallyn, E.; Koleva, B.; Logan, B.; Fielding, D.; Benford, S.; Gelmini, G.; and Madden, N. 2005. Embodied reporting agents as an approach to creating narratives from live virtual worlds. In *Lecture Notes in Computer Science, Proceedings of Virtual Storytelling 2005*. Strasbourg, France: Springer.

Young, R. M. 2001. An overview of the Mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. In *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*. Stanford, CA: AAAI Press.