Author in the Loop: Using Mixed-Initiative Planning to Improve Interactive Narrative

James M. Thomas
Liquid Narrative Group
Department of Computer Science
Box 7535, NC State University
Raleigh, NC 27695 USA
jmthoma5@ncsu.edu

R. Michael Young
Liquid Narrative Group
Department of Computer Science
Box 7535, NC State University
Raleigh, NC 27695 USA
young@csc.ncsu.edu

Abstract

This paper describes a foundation for an interface to allow non-technical human authors to collaborate with an automated planning system to design interactive narrative. Drawing from research in advisable and mixed-initiative planning, a domain metatheory is presented that allows for qualitative elaborations of narrative domains. The authors describe a graphical user interface that exploits this metatheory to specify authorial preferences. Specific constructs related to interactive narrative are considered to demonstrate how the preferences of the human author may be used to define and control the possible user experiences of an interactive narrative.

Introduction

"Interactive narrative" describes the stories that develop within virtual worlds in which human users interact with one or more computer controlled agents. The most well known examples of interactive narrative are computer games, but also included are intelligent tutoring systems, embodied conversational agents, virtual environments, and training simulators. A persistent challenge for such systems is the narrative paradox: "how to reconcile the needs of the user who is now potentially a participant rather than a spectator with the idea of narrative coherence." (Aylett 2000).

Few systems attempt to reconcile these goals dynamically at run-time. Those favoring strong plot coherence often restrict the depth of the computer-controlled characters, and/or the human user's available palette of interactions with these characters, reducing character believability. Systems with interesting and believable characters often lack any automated mechanism to coerce these 'emergent' bots to meaningfully contribute to a story. Although many useful and commercially successful systems have been built with these limitations, none has yet fully met the goal of *Hamlet on the Holodeck* (Murray 1998, Cavazza et. al, 2000).

One approach for the balancing of these competing goals is the Mimesis system (Riedl, Saretto, and Young 2003). Their algorithm generates plans for actions of story world characters based on hierarchical task decompositions and discrete causal requirements. Although Mimesis simultaneously solves for plot coherence and character believability, the authors acknowledge (Riedl and Young 2004) that a primary limitation is the lack of a search space heuristic that would allow the system to judge the relative "goodness" of one plan over another. In other words, there is no mechanism to ensure that particular narrative qualities such as "suspense", "surprise" or "romance" will be produced in resulting plans.

One might attempt to define a generalized heuristic function in terms of universally accepted narrative ideals, but most planners lack a sufficiently powerful model to make associations between such generalized ideals and the semantics of a specific problem domain and plan space. Also, no set of heuristics has yet been identified that guarantees "good" narrative even when applied by skilled and motivated humans. As author Somerset Maugham quipped, "There are three rules for writing the novel. Unfortunately, no one knows what they are".

An alternative approach is to involve the human author in defining heuristic functions for each interactive narrative based on that author's preferences of setting and plot. For the system to capture these preferences and report them to the planner, it must have an integrated understanding of the definitions of actions and entities in the problem domain (the setting) and the effects that the constraints on those actions have in defining the topology of the plan space (plot experiences). A reasonable approach for gaining that understanding is to keep the author "in the loop" throughout the plan construction process. This paper describes the foundations for the design of such a collaborative authoring environment for interactive narrative. The first stage of this environment is being implemented as part of the Zócalo system of planning services at North Carolina State University (NCSU).

Planning For Interactive Narrative

Planning for interactive narrative offers special challenges and opportunities. Even in systems that do not attempt the dynamic generation of narrative structure, it is difficult to maintain clear knowledge or control of what can happen at run-time. Two examples from a sub-genre of narrative, massively multiplayer online games, illustrate this point. An early such game (1985) was Lucasfilm's Habitat. One of the first game-wide campaigns planned inside Habitat was a treasure hunt called the "D'nalsi Island Adventure". Habitat designers Morningstar and Farmer recall:

> It took us hours to design, weeks to build (including a 100-region island), and days to coordinate the actors involved. It was designed much like the puzzles in an adventure game. We thought it would occupy our players for days. In fact, the puzzle was solved in about 8 hours by a person who had figured out the critical clue in the first 15 minutes. Many of the players hadn't even had a chance to get into the game. The result was that one person had had a wonderful experience, dozens of others were left bewildered, and a huge investment in design and setup time had been consumed in an eyeblink. (Benedikt 1990)

This lack of predictability was not simply an artifact of the times. In January of 2006, Jeff Kaplan, a lead designer of World of Warcraft (the most popular on-line game to date at the time of the article with 5.5 million subscribers) was interviewed by the New York Times. In the interview he was asked how long it would take until the top boss in one realm of the game (Ahn'Qiraj) would be defeated:

> My estimates are in the one-to-two-month range, but my expectation is that it could happen today. I've learned that as soon as something is in the game, you have to expect that it's going to be beaten. (Shiesel, 2006)

In essence, the designers of this type of interactive narrative have given up on predicting the run-time possibilities of their work. Instead, these Game-Masters (Louchart and Aylett 2003) are kept busy developing the next episodic installment of narrative while tweaking its predecessor to accommodate the unforeseen actions of users.

Despite the difficulty of constructing shrink-wrapped games that solve the narrative paradox, designers continue to make the attempt. Marc Laidlaw, lead designer of the commercially successful, narrative-rich game Half Life 2 for Valve Entertainment, asserted the importance of plot in at the Austin Game Writer's Conference in late 2005:

> Game designers should be in love with plot. It's the engine of the story. It's the core tech, that little mathematical nub that everyone can point at and make the subject of proofs and axioms and corollaries. None of these technical considerations serve us very well when we start arguing about meaning, but if you get plot right, then meaning inevitably follows.
>
> I like to talk about plot because it can be directly implemented in a game. Plot is the sum of cause and effect. Whether the cause is something in the player's control, or something the designers force upon the player, it's measurable. It can map directly to gameplay decisions, and lead to an outcome or outcomes that are equally tangible.
>
> A thorough discussion of plot gives you everything you need to build your story, and your game. (Laidlaw 2005)

Automated planning manages "the sum of cause and effect" that defines plot. Beyond its desirability to a few risk-takers in the game industry, ensuring that a coherent narrative achieves particular goals is an important requirement for educational and training applications. The task for planning systems in interactive narrative reaches well beyond finding a single complete and consistent plan. Authors are interested in understanding how unplanned user actions may affect story goals. This in turn raises issues about the variability of narrative experiences that are possible with each construction and how those possibilities shift as authors make changes. Compared to other plan authors, those building interactive narrative are probably more likely to work with the planner to make incremental refinements to the planning problem through multiple iterations.

Including Domain Knowledge in Planning

Traditional automated planners are not designed specifically to facilitate iterative collaboration with the plan author. Research into collaborative planning methodologies has generally been referred to as advisable or mixed-initiative planning. Advisable planning (Myers 1996b) attempts to shape the behavior of the planner by adding additional information to the definition of the planning problem prior to invocation of the planners. Mixed-initiative planners allow for the iterative and incremental construction of the plan with both the user and the planner capable of proposing or initiating requests to change aspects of the problem or solution. Thus, advisable planning is effectively a special case of mixed-initiative planning where the initiative is first taken by the human, then by the planning system. "Configurable" planning (Nau 2005) is the combination of domain-independent planning engines with higher-level abstractions like hierarchical task networks that capture and exploit domain knowledge. Each of these research threads has application toward collaborative planning of interactive narrative.

Interactive Narrative as a Planning Domain

Much of the motivation for configurable planners is based on the gulf between the real world and restrictive experimental domains descended from "blocks world. Where Nau's "configurable" planners represent an architectural middle ground, interactive narrative represents a domain of similarly intermediate complexity between the "blocks world" and the real world.

Because interactive narrative takes place in a virtual world, its domains are both fully knowable and fully malleable. An advantage for planning research is that these domains may be amended or contracted to suit the requirements of the planning problem. In fact, the plan author may be responsible not only for the domain representation, but also may be involved in the construction of the domain itself. As interactive narrative planning is a component within this larger creative process, there are possibilities and requirements for experimentation and exploration than are not found working with real world domains. This affords researchers the freedom to investigate intricate relationships between the domain, its representation, the planning problem and the resulting plan spaces. Integrating these concepts into an authoring tool can benefit both the interactive narrative and the planning research communities.

Mixed-Initiative Planning Research

Mixed-initiative techniques have long been associated with several prominent planning research projects. Ferguson and Allen (1998) have studied extensively aspects of mixed-initiative in their TRIPS and TRAINS projects In their estimation "far more attention needs to be paid to the gap between the abilities of automated reasoners and the needs of human decision makers (Allen and Ferguson 2002). The systems Allen and Ferguson have built rely on human-computer interfaces based on natural human dialog. Their focus is on building a dialog system intermediary between the human plan author and group of back-end agents. A key challenge they have addressed is the mapping of individual communicative utterances of the user to the most appropriate plan editing action. They bias this intention recognition toward those candidates suggested by recency and those that will minimize plan churn. Another challenge they have addressed is the resolution of ambiguities about the scope of an intended change. Is the requested change to be performed on the problem goal or the proposed solution? Is the solution to be modified, extended or rejected? To perform these types of reasoning, the authors employ a collaborative interaction model compatible with the SharedPlans formalism of Grosz and Kraus (1996) and realized as an inter-agent communication protocol. The application of Allen and Ferguson's work to interactive narrative is limited by two factors. First, they rely on a domain representation assumed to be complete and accurate, where these are very much in flux during the authoring process of interactive narrative. Second, much of their focus is on the interpretation of spoken natural language statements about plans and plan goals in order to make the appropriate changes to the plans, where interactive narrative inputs are likely to be text with formally constrained syntax and semantics. A key contribution of their work that can help interactive narrative is modeling the problem solving state at multiple levels of abstraction, from a high-level hierarchy of objectives, to a compact summary of a class of possible concrete solutions, to the intermediate world states of particular solutions (Ferguson and Allen 2002).

Rich and Sidner (1998) also leverage discourse interpretation and SharedPlans in COLLAGEN. COLLAGEN, like TRIPS and TRAINS, is a few steps beyond the immediate challenges of authoring interactive narrative. COLLAGEN constrains search through a detailed model of interaction history. This includes intentional structure (partial SharedPlans), linguistic structure (hierarchical groupings of actions into segments), and attentional structure (a "focus stack" of segments). This model is used to generate context-dependent natural language formulations from which the user may choose. Rich and Sidner believe that, in contrast to the weakly structured interaction histories in most interactive systems, the interaction history in COLLAGEN "reflects the user's problem solving process". This idea of the system asserting informed choices of actions to the plan author could be used to guide the authors of interactive narrative toward decisions that have the best utility relative to their goals.

Tate, Dalton, Levine (1998) introduced the <I-N-OVA> Model (for Issues, Nodes, Orderings / Variables / Auxiliary) abstraction to allow for "plans to be manipulated and used separately from the environments in which they are generated." In Tate's system, the user and planning system work to refine the sets of constraints under which the planner must operate. Amant, et. al., (2001) have built mixed initiative interfaces for plan visualization and navigation. Blythe, et. al. (2001) have investigated representing plan structures in ways interpretable by humans as business processes. These systems focus on mapping plan representations to natural language correlates within the domain.

Advisable Planning

The idea of an advisable problem solving system goes back as far as John McCarthy's proposed "Advice Taker" program (McCarthy 1959). McCarthy's first example problem for a 'program with common sense' was a planning problem. Advice continued to have a prominent role in research into automating common sense including projects such as Cyc (Lenat 1995). Meanwhile, as the field of automated planning developed specialized knowledge representations and reasoning methods it became separated from McCarthy's more general strain of commonsense reasoning work. However, Myers and her colleagues (Myers 1996b), have recently investigated the application of user-supplied advice within the context of modern planning techniques. Myers' advisable planner employs a model where abstract advice specifications provided by the user are compiled into a language of constraints common to traditional planning algorithms. Myers distinguishes between three "idioms" of advice. Task advice identifies the goals and actions to be included in solution. Strategic Advice recommends how goals and actions are to be accomplished relative to parameter values. "Evaluational" Advice puts constraints on some metric defined for the overall plan (e.g., resource usage, execution time or solution quality).

In Myers' work, the advice an author gives the planner is grounded in a domain metatheory, an abstract representation independent of underlying planning technologies. A domain metatheory is intended to enhance user directability of the planning process, aid in the generation of qualitatively different plans, and aid plan summarization. Myers (2000a) proposes a model built on three constructs: *roles*, *features*, and *measures*. *Roles* describe the function of an object within an operator, *features* are attributes that differentiate operators, and *measures* are partial orderings of features with respect to some criterion.

For example, the feature "**Air**" might be associated with the operator "AirMail(loc1, loc2, item)" while the feature "**Land**" might be associated with "BicycleMessage(loc1, loc2, item)". Related features may be grouped into feature categories, e.g., **Transport-Media** could be a category containing both **Air** and **Land**.

A *measure* is an ordering (possibly partial) of features within a feature category. For example, the measure **AFFORDABILITY** might be defined over feature category **Transport-Media** as to rank **Land** higher than **Air**, where the measure **COMFORT** might be defined over feature category **Transport-Media** as to rank **Air** over **Land**. A role-fill specifies explicit object instances or constraints over a set of instances relative to a given operator role.

Measures may be extended to describe object instances within the domain through the assignment of measure values. For example, if the measure **AFFORDABILITY** has measure values defined as (**Cheap**, **Moderate**, **Expensive**) the object instance *Lear Jet* would have the **AFFORDABILITY** measure value *Expensive* while the object instance *Subway* would have the **AFFORDABILITY** measure value *Cheap*.

Strategic advice is specified through the metatheoretic elements of Activities, Roles, Role-Fills and Measures, which in turn are simpler and closer to the natural cognitive models employed by human experts than the lower level planning constructs of goals, operators, variables and bindings. Strategic advice consists of prescriptions and restrictions of roles, fillers (a.k.a. role-fills), relative to specific activities. This advice takes two forms. Role Advice designates which object-role specifications (role-fills) are required or restricted in specific activities. For example, a role template of "Stay in <Accommodation> while vacationing in <Location> might be instantiated as "Stay in 3-star hotels while vacationing in Scotland" where the role of <Location> is filled by "Scotland" and the filler "3-star hotel" is prescribed for the role <Accommodation> (Myers 1996b). This is an example of an target activity with a feature of Vacation. In contrast, Method Advice operates at a higher level, as it prescribes or prohibits the use of specific activities within the plan.

Advice For Qualitative Differences

Once the planner becomes knowledgeable of the advice associations of its elements, it is possible to direct it toward solutions that have particular qualities relative to that advice. Many planners can generate different plans for the same problem, but extracting and summarizing the meaning of those differences is difficult. Furthermore, the particular differences of interest will vary from user to user and task to task. A deeper problem is the assumed accuracy and completeness of the domain and problem representations. Because much of Myers' work has been situated in the application of planning military operations for the real world, the domain representation is often seen to be incomplete or imperfect by the human experts who use the system because they have experience and knowledge over a vast number of real world exceptions. Therefore, considerable effort is devoted to eliciting more complete descriptions of the domain and problem representations.

A goal for advisable planning systems is that they create novel plans that are qualitatively different from one another (Myers and Lee 1999) a goal that is especially relevant for interactive narrative. To achieve this goal, the plan author nominates a subset of measures from the domain metatheory to serve as criteria for evaluating chosen properties of plans. Myers introduces an evaluation function that maps feature measures into categories on which measurements normalized over the interval [0, 1] can be applied. A set of k evaluation criteria thus define a k-dimensional space in which the Euclidean distances can be measured between the locations of each plan relative to each of these dimensions as measured by the evaluation functions. Myers' recent work (Myers 2005) uses the metatheory to summarize plan content, and uses a type hierarchy to reason about differences based on which objects are bound to different features of the plan.

As Myers moved toward a mixed-initiative model in which the user makes many of the decisions necessary to create the final plan, a new problem was introduced. At some points in the creation of a complex plan there may be hundreds or thousands of unresolved issues. The system must rank these decisions based on importance so that the user has a chance to complete the plan. As many as five different methods for this type of prioritization were considered and three were implemented in a system called PASSAT (Wolverton 2004). The exploratory nature of interactive narrative construction is likely to produce similarly complex plan spaces. The prioritization methods pioneered in PASSAT would be useful in making optimal use of the finite attention of human narrative authors.

Domain Elaboration Framework

To leverage the results of advisable and mixed-initiative planning, this paper introduces DEF, the Domain Elaboration Framework. DEF is an adaptation of Myers' domain metatheory that allows authors to add detail to classical planning domains to enable expressive problem definition and reasoning about plans.

The basis of DEF is a STRIPS-stye (Fikes and Nilsson 1971) planning domain characterized by objects, conditions and operators. More formally, an object symbol provides a unique name for an entity in the world. All object instances are predefined by the plan author. A condition is a conjunction of function-free literals composed of a unique name identifying a relation and a set of placeholder variable terms or object instances. These terms are also referred to as condition parameters. An operator is defined by the set of literals stating the preconditions that must hold before it can be invoked, the set of literals stating the effects that will hold following its invocation, and a parameter list that may be applied to designate variables in these sets of literals.

Where the metatheory introduced by Myers relies on roles, role-fills, features, and measures, DEF uses an alternate grammar of **types**, **dimensions**, **weights**, and **measurements**. A type is a symbolic name of a node in a global hierarchy of author-defined types with a unique root node named "***anyThing***". Every operator, parameter, and object instance is required to have at least one associated type. Although type can be seen as an implicit concept in Myers' original metatheory. it is not until her recent work (Myers 2005) that one can find an explicit representation of type. In the example of the move operator whose *loc1* parameter was assigned the role of origin the type might be inferred to be location. It would seem obvious to a human author of the move operator that the *loc1* should only bind to objects of type location, but without explicit constraints a planner could just as easily fill the origin role with a cat, a cake, or a comb.

Because every parameter of an operator or condition and every object has a **type** associated with it, the type hierarchy can be used to guide the planner in assuring that the authorial intentions for bindings are maintained. In fact, an interactive narrative creation tool built on the DEF framework could communicate type constraints on parameters and objects through extending the set of preconditions for each operator and for the initial and goal states.

For example, in the case of the *loc1* parameter within the move operator the type constraint isalocation(*loc1*) could be added to the operator's list of preconditions, and when object instances of type location are created, corresponding isalocation(*newobject*) conditions could be added to the initial state of the planning problem. Some planners allow the nomination of a special subset of preconditions (sometimes referring to these as constraints) whose truth values can be computed directly from the assignments to the initial state, allowing for faster processing. For these planners, type constraints may actually help speed the plan search process by reducing the set of objects the planner must consider for bindings to parameters of operators and literals.

Types are also associated with operators, enabling the author to use a portion of the type hierarchy to encompass entities much like features in Myers' formulation. Every operator, parameter, and object instance has one or more associated types, and zero or more associated **measurements**. A **measurement** consists of a **dimension** and a **weight**. A **dimension** is a symbolic name selected from a global list of unique author-defined dimensions. A **weight** specifies a relative intensity of the dimension normalized on the interval [-1, 1]. The default weight '0' represents a neutral intensity, -1 is maximally negative and 1 is maximally positive.

Expressive Power of DEF

The **dimension** construct in DEF corresponds to the **measure** of Myers' metatheory. Both are symbolic values chosen from an author-defined list, e.g., *affordability, comfort,* or *magic*. A key difference is that where DEF uses numeric **weights** to gauge instances on each **dimension**, Myers uses **measure values** from a set of symbols that are defined for each feature category and ordered by the plan author for each **measure**.

For example, in DEF, object instance *Lear Jet* may have the **measurement** {*affordability*, -0.98}, where *affordability* is a **dimension**, and **weight** is near the minimal value of -1 on the scale [-1, 1].

With Myers' metatheory, object instance *Lear Jet* may be assigned the measure value *Expensive* for the measure *affordability*, from the ordered set of measure values {*Cheap*, *Moderate*, *Expensive}.*

At the operator level, DEF allows types and dimensions to describe operators in the same way they describe object instances. Myers uses **features** to describe operators at a higher level of abstraction than DEF. The strategy chosen with DEF is to use a reduced low-level vocabulary to elaborate the problem domain description and defer their aggregation into more complex abstractions like **features** and **feature categories** to higher level user interfaces. Hopefully, this will allow for abstractions of arbitrary complexity at the interface level, while preserving an underlying representation that facilitates efficient reasoning about the qualities of individual plans and the qualitative differences between plans.

To make qualitative judgments about plans, Myers' measures are converted to proportionally distributed values over the interval [0, 1]. DEF requires explicit normalization of weights over an interval [-1, 1] (chosen to facilitate a default neutral weight of 0). Clearly, this shifts some responsibility to the interface to ensure that human authors assign weights with this normalization in mind. An interface using DEF can provide abstractions such as symbolic ranges, {*Cheap*, *Moderate*, *Expensive}* and convert these values to proportional internal representations. However, an interface is not precluded from allowing more precise or non-proportional numerical representations when appropriate.

An expressive advantage of DEF is that types and measurements are applicable to every operator, operator parameter, and condition parameter. Authorial goals are often articulated in terms of the types of actions contained in a story. The knowledge to support this type of reasoning can be represented through measurements applied to operators. Suppose, as in the film *The Princess Bride*, a young boy would like to make sure that the story does not contain too much kissing. *Kissing* could be introduced as a dimension and every operator associated with the act of kissing could be assigned measurements on the order of (*kissing*, .95). Other operators could have neutral values of 0, or negative values. *Kissing* could be selected as an evaluation criterion and the plans whose evaluation functions return low values of *kissing* could be favored.

Object instances could also have attributes that are directly derived from authorial goals. Perhaps the author would like to favor stories that contain a lot of enchanted objects. A dimension of "*magical*" could be created and applied with high levels to magic rings, scrolls, and potions, and low levels to chewing gum wrappers and socks. A planning heuristic that takes these measurements as an input can offer a high-degree of fidelity to discrimination between candidate plans.

Higher-level narrative constructs will necessitate the use of more complex representations. Suppose the author wants a 'happy' story. Is give-money(giving-player, receiving-player) a 'happy' action? It might be happy for receiving-player but not for giving-player depending on the state of the world. Using parameter-level measurements in DEF, a default positive measurement of happiness could be given to the receiving-player and a default negative measurement to the giving-player. Still, what if the measurement value of one parameter may depend on the bindings to other parameters in the same operator? For example, the giving-player might be happy to give money to her child, but unhappy to give the money to a thief.

One approach would be to recognize that these two situations describe actions that differ from the perspective of drama (mothering vs. mugging), even if they may have the same add and delete list from the perspective of classical planning. Thus, the action can be split into a give-moneyto-child action where ischildof(player2, player1) is added as a precondition, and a give-money-to-thief action where isthief(player2) is added as a precondition. Then the happiness of player1 can be assigned different values in each action. It may be possible for a tool using DEF could create cloned actions like these when the user indicates that happiness is a function of the sub-types of player1 and player2 and use a more compact representation to solicit and display such preferences. Still, the role the operator plays within the larger context of the plan may also significantly effect the user's evaluation of the plan. DEF merely serves as a starting point for reasoning about interactive narrative.

To summarize, DEF associates a set of one or more types and zero or more measurements with every operator, operator parameter, condition parameter, and object instance. DEF is a domain-independent representation intended to by leveraged by a user interface for use with any planner that can work with a STRIPS-style domain description.

Qualitative Reasoning With DEF

One motivation for DEF is to provide a general framework for elaborations of the plan author's preferences for the objects and actions in the domain over a variety of criteria. It is left to the planner and whatever interfaces are put between the human and the planner to make use of these preferences to influence plan reasoning. An evaluation function can be easily constructed from the **measurements** in DEF to apply the qualitative reasoning power of Myers' work to resulting plans, simply by transforming the **weights** from their interval [-1, 1] to the interval [0,1] employed by Myers' algorithms. This function can be used to iteratively refine and navigate through the plan space, or it can influence can be in the heuristics that are applied by the planner to direct search, perhaps in conjunction with other DEF constructs.

Another mechanism for reasoning about types recently introduced by Myers (2005) is also easily applicable in the DEF context. Myers defines a function MinSuperType(V) which finds the most specific super-type common to a set of elements V . This allows the author to characterize the differences between plans or parts of plans, through the five distinct set relationships that correspond to particular subsets of their typed elements. Set arithmetic functions are described that help pinpoint key strategic differences between plans and show areas where plans are not as different as they might seem.

Incorporating DEF in a Planning System

While DEF supplies the raw materials for qualitative reasoning about plan structures, it requires an interface to allow non-technical authors to apply it to a planning system. This interface should represent the problem solving state at multiple levels of abstractions, similar to the four-layer model employed by Allen and Ferguson (2002). Their model allowed the user to move from high-level hierarchical objectives, through task structures that summarized classes of concrete solutions, to more primitive descriptions of particular plan fragments and world states.

Implementation of such an interface has begun with a program called Bowman, which is currently part of the Zócalo suite of planning tools available at NCSU at http://zocalo.csc.ncsu.edu. Bowman provides a GUI that allows authors to describe types, objects, operators, conditions and the initial and goal state of a planning problem. Bowman seamlessly passes an XML representation of the planning problem to a planning web-service to generate plans. The planner interface supports requests for the next N plans, planning for N seconds, or simply until a complete plan is found.

Bowman shows not only individual plans but also the entire plan space through scalable vector graphics (SVGs) that can be navigated through mouse clicks. Bowman can depict the plan space as a tree of nodes, where each node is a partial plan with zero or more plan flaws to be resolved. A plan flaw is an open precondition, a threatened causal link, or a flawed decomposition. Plan nodes with zero flaws are shown in green and plans with one or more flaws are shown in progressively more pale shades of yellow.

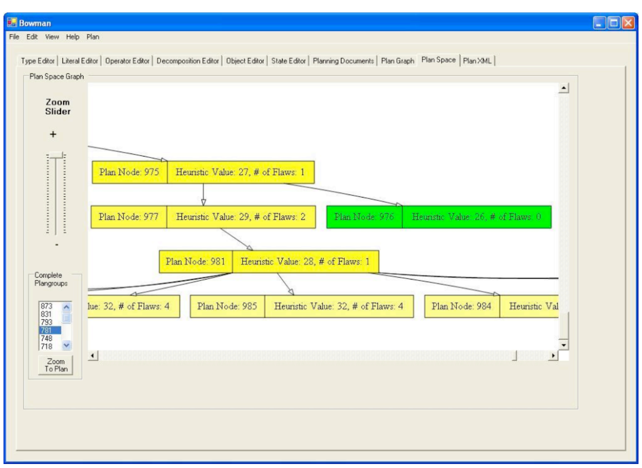

Figure 1 Bowman Plan Space View

The Bowman user can navigate the plan space and individual plans either through direct manipulation with the mouse or through dialog-based search.

Application to Interactive Narrative

Bowman can work with any planner that supports its straightforward XML representation of the planning and conforms to a simple web service interface. However, to provide support for interactive narrative domains, Bowman assumes a particular type of planner is being used. In its initial deployment, Bowman assumes that this is the Zócalo planner, based on Longbow, a decompositional (HTN) partial order causal link planner described by Young, Pollack and Moore (1994). Riedl, Saretto, and Young (2003) extended Longbow to support what they called *narrative mediation* to manage and respond to user actions in interactive narrative. Mediation policies are invoked in response to unplanned user actions that would threaten conditions in the world required for a planned future action. When such an action occurs, a mediation policy may nominate alternative actions called *failure modes* that may be substituted at run-time for the threatening user action.

To leverage narrative mediation, Bowman must be able to represent the particular mediation policies in effect for particular actions. Since failure modes are simply lists of operators, it would be possible to provide guidance on the subset of operators that are good candidates for failure modes through DEF constructs. In addition, DEF may be used to inform Bowman of abstract specifications of the characteristics of failure modes required to resolve plan bottlenecks. A first step is to use DEF to differentiate between types of agents.

Agent Types

For Bowman to be useful in addressing narrative mediation, it must contain in its representation of operators the types of agents that fill various roles. Thus a first step toward supporting narrative mediation is to distinguish between user-controlled agents and system-controlled agents (often called NPCs, or Non Player-controlled Characters). This distinction can be accomplished through a convention applied to the population of the type hierarchy of DEF. A subtree of the hierarchy can be fixed to contain “agent” and “inanimate”. “Agent” can be subdivided into “NPC” and “User”. A generalized mechanism can be realized in Bowman to allow the author to designate the subtree of the global type tree that is associated with the user (the default convention being “User” above) and that which is associated with NPC agents.

For interactive planning domains, the definition of operator in DEF is then extended to *require* specification of the type of agent that is capable of invoking the action (“User”, “NPC”, or the un-committed “Agent” as a default). Also, individual object instances are extended to have a required agent type specification.

Mediation Strategies

As described in by Riedl, Saretto, and Young (2003) the planner is responsible for detecting user actions that could threaten the story plan. For each of these *exceptional* actions, the system must determine if changing part of the unexecuted portion of the plan can *accommodate* the action or if an *intervention* is required. An *intervention* requires that the requested action does not execute. Instead an instance of a non-threatening action, called a *failure mode* is substituted for the requested action in real-time. For example, if the user tries to shoot a character that is required to achieve a narrative goal later in the plan, a *failure mode* of "shoot-and-miss" or "jamming-shoot" might be substituted for the threatening "shoot(?gun, ?target, ?victim)" action. For this substitution to occur in real-time while as the user invokes the action, the intervention policies must be communicated to the story world system as soon as the initial story plan is created and whenever that story plan is altered.

How does the system know which actions can be invoked by the user? With DEF and Bowman, the author must explicitly specify the type of agent capable of invoking any particular action. Bowman provides this information to the planner, and also uses it to advise the human author of the set of actions for which failure modes are appropriate. For example, if the author at some point in the construction of the narrative has specified that the "shoot(?gun, ?target, ?victim)" action is one that can be invoked by an agent that can be a human user, Bowman can later take the initiative to advise the human author this action is contained in the set of actions for which failure modes may be useful. This indication could be provided prior to the invocation of the planner, or following invocation of the planner, after which the set may have been reduced by the set of actions for which the accommodation strategy proved sufficient. Bowman can further aid the human author in creating failure modes by reminding the author of the preconditions and effects of the underlying operators, highlighted with the planner's understanding of particular threatening effects if the planner has provided such guidance. Finally, the plan space depiction can be enhanced with explicit representation of the impact of mediation strategies.

Narrative "Macro" Libraries

Bowman allows for libraries of planning "macros" to be made available for plan authors. These sets of related literals and operators can assist plan authors to achieve typical planning goals with less total work. As a motivating example, consider the issue of inter-agent relationships and their effect on character believability. Narrative systems have been built with rich emotional models of agents (Gratch and Marsella 2004). But in most commercial games the attitudes of NPCs are modeled with a single bit of memory – "friend" vs. "foe". Bowman proposes a modest extension of this model to enhance the believability of characters through three-valued attitudes defined by the narrative author. These attitudinal values can shift based on actions chosen by the agents in the world to enhance the component of character believability that is inferred by characters changing their behaviors based on social history. Examples of attitudes could be "ally/foe", "trust", "niceness", "sincerity", or "goal-directedness". Bowman predefines a set of literals that handle initialization of attitudes and transitions from one attitude to another. These attitude "macros" can be invoked through simpler constructs that the author manipulates. This may seem an overly shallow modeling of social relationships, but it does match up quite well with the emotional "bank account" metaphor pioneered by psychologists like George Bach and later popularized by Stephen Covey (1989).

For example, suppose the author would like to model the ally/foe attitude. Assume that we name this quality "*ally*". The following literals express possible initial states for the "*ally*" attitude between *?agent-a* and *?agent-b*:

- *attitude-negative*(*ally*(*?agent-a, ?agent-b*)) means that *?agent-a* is a foe of *?agent-b*
- *attitude-neutral*(*ally(?agent-a, ?agent-b*)) means that *?agent-a* is neither an ally nor a foe of *?agent-b*
- *attitude-positive*(*ally(?agent-a, ?agent-b*)) means that *?agent-a* is an ally of *?agent-b*

The pre-defined library of helper operators that support attitude maintenance are:

attitude-up-to-neutral(?attitude-name, ?a, ?b)

Preconditions:

attitude-negative(?attitude-name(?a, ?b))

incr(?attitude-name, ?a, ?b)

Effects:

attitude-neutral(?attitude-name(?a, ?b))

^incr(?attitude-name, ?a, ?b)

attitude-up-to-positive(?attitude-name, ?a, ?b)

Preconditions:

attitude- neutral (?attitude-name(?a, ?b))

incr(?attitude-name, ?a, ?b)

Effects:

attitude- positive (?attitude-name(?a, ?b))

^incr(?attitude-name, ?a, ?b)

A reciprocal set of "*attitude-down*" actions that reference the "*decr*" literal are also automatically added to the domain description. Then, the only thing the plan author needs to do is establish initial attitude values for relationships that need to be modeled and insert "*incr*" or "*decr*" attitude literals as the effects of actions that make sense in the world. The planner will find the appropriate "*attitude-up*" or "*attitude-down*" actions to insert in the plan to meet attitudinal goals specified later in the plan (if any).

For example, if the author intends for character *?agent-a* to become an ally of *?agent-b*, the author could add condition *attitude-neutral(?ally, ?agent-a, ?agent-b)* to the initial state, *attitude-positive(?ally, ?agent-a, ?agent-b*) to the goal state and create an operator like *join-up(?ally, ?agent-a, ?agent-b)* that has *incr(?ally, ?agent-a, ?agent-b)* as effect. A planner could add the *attitude-up-to-positive* operator to establish the *attitude-positive* goal condition. Then an operator like *join-up* would be selected to establish the *incr* condition.

Creating an inter-agent relationship library of macros is just one example of what might be done with a sufficiently general interface like Bowman. Bowman can serve as a test bed for assessing the usefulness of other macro-like libraries of related operators and literals.

Research Ambitions

DEF and Bowman can aid in narrative construction and qualitative reasoning about plans. At the point of this writing, Bowman works with the Zócalo planner to construct planning domains and planning problems and navigate through plan spaces and individual plans. However, only the "*type"* component of the DEF framework is realized in the current version of Bowman. To achieve the ambitions outlined in this paper, much work remains.

First, the full DEF framework must be implemented in Bowman, so that agent types, mediation strategies and narrative "macros" can be specified. Second, authors must be given an ability to articulate their narrative preferences through DEF constructs. Third, because authors are engaged in a creative process, it will be important for Bowman to provide a rich interaction history as in COLLAGEN (Rich and Sidner 1998) to allow exploration of alternate approaches. Plan summarization techniques will be needed to annotate the plan space with DEF constructs and increase the effectiveness of the "navigation" paradigm. The qualitative metrics proposed by Myers should overlay all these capabilities to provide a clear understanding of the differences between plans.

In such a system, the author could use "world manipulation sliders" like a sound engineer uses a mixing board to control the relative levels of different components in a production. In the case of interactive narrative, these components may be primitive constructs like DEF dimensions, or more complex features like the number of different execution paths or the amount of "conflict" or "happiness" in the story. A tight coupling with the planning system could allow the Bowman to opportunistically highlight areas of the plan that would best benefit from the attention of the human author. To conserve that attention, Bowman would benefit from an internal model of the author's intentions and apply them automatically to situations that are deemed similar. The result would be a system whose behavior is interesting enough to hold the interest of people who create interesting characters for a living.

Conclusion

This paper introduced a general planning domain metatheory called DEF and a general plan-authoring interface called Bowman, currently under development at North Carolina State University. These tools are being used to support author-preference realization in interactive narrative. As these tools grown into high-level interfaces accessible to non-technical authors, new avenues of planning research may become accessible as well.

Acknowledgements

The authors wish to thank the members of the Liquid Narrative Group at NC State University for their assistance in developing Zocalo. This is material is based upon work supported under a National Science Foundation Graduate Research Fellowship and National Science CAREER award 0092586.

References

Allen, J. F., and Ferguson, G. 2002. Human-Machine collaborative planning. *In Proceedings of the Third International Workshop on Planning and Scheduling for Space*.

Aylett, R.S. 2000. Emergent Narrative, Social Immersion And "Storification" *Proceedings, Narrative Interaction for Learning Environments*, Edinburgh.

Benedikt, M. ed. 1990. The Lessons Of Lucasfilm's Habitat. *In Cyberspace: First Steps*. MIT Press, Cambridge, Massachusetts, p. 288.

Blythe, J., Kim, J., Ramachandran, S., and Gil, Y. 2001. An Integrated Environment For Knowledge Acquisition. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces* (New York, NY, USA, 2001), ACM Press, pp. 13–20.

Cavazza, M., 2000. Interactive Storytelling In Virtual Environments: Building The Holodeck. presented at the 6

International Conference on Virtual Systems and Multimedia (VSMM), 2000, October 4-6, Gifu, Japan.

Covey, S. 1989. *The Seven Habits of Highly Effective People*. Simon & Schuster, New York, NY, USA, pp. 188-199.

Ferguson, G., and Allen, J. F. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *AAAI/IAAI* pp. 567–572.

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

Gratch, J. and Marsella, A. 2004. A Domain-independent framework for modeling emotion, *Journal of Cognitive Systems Research,* Volume 5, Issue 4, pp. 269-306

Grosz, B. J., and Kraus, S. 1996. Collaborative Plans For Complex Group Action. *Artificial Intelligence* 86, 2, 269–357.

Lenat, D. B. 1995. CYC: A Large-Scale Investment In Knowledge Infrastructure. *Communications of the ACM* 38, 11, 33–38.

Louchart, S. and Aylett, R. 2003. Solving The Narrative Paradox In VEs - Lessons From Rpgs In *Intelligent Virtual Agents, 4th International Workshop IVA2003* eds T. Rist, R.Aylett, D.Ballin,

McCarthy, J. 1959. Programs With Common Sense. *In Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, Her Majesty's Stationary Office, pp. 75–91.

Murray, J. 1997. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace,* MIT Press, Cambridge, Massachusettes.

Myers, K. Strategic Advice For Hierarchical Planners. 1996a. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR96)*, L. C. Aiello, J. Doyle, and S. C. Shapiro, Eds., Morgan Kaufmann Publishers, pp. 112–123.

Myers, K. 2000a. Domain Metatheories: Enabling User-Centric Planning. In *Proceedings of the AAAI Workshop on Representational Issues for Real-World Planning Systems*, K. Myers and Y. Gil, Eds.

Myers, 2000b. K. Planning With Conflicting Advice. In *Proceedings of the Fifth International Conference on AI Planning Systems* (AIPS 2000).

Myers, K., and Lee, T. J. 1999. Generating Qualitatively Different Plans Through Metatheoretic Biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence,* AAAI Press.

Myers, K. L. 1996b. Advisable Planning Systems. In *Advanced Planning Technology*, A. Tate, Ed. AAAI Press, Menlo Park, CA.

Myers, K. L. 2005. Metatheoretic Plan Summarization And Comparison. In *Proceedings of the ICAPS-05 Workshop on Mixed-initiative Planning and Scheduling* Monterey, CA.

Nau, D. S. 2005. May All Your Plans Succeed! *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, July 2005. Invited talk.

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. M. 2003. Managing Interaction Between Users And Agents In A Multiagent Storytelling Environment*. In Proceedings of the Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-03).*

Riedl, M., and Young, R. M. 2004. An Intent-Driven Planner For Multi-Agent Story Generation*. In Proceedings of the Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*.

Riedl, M. O., and Amant, R. S. 2002. Toward Automated Exploration Of Interactive Systems. *In IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces* (New York, NY, USA, 2002), ACM Press, pp. 135–142.

Schiesel, Seth. 2006. "Kill the Big, Bad Dragon (Teamwork Required)." *New York Times*, 28 January 2006.

Tate, A., Dalton, J., and Levine, J. 1998. Generation Of Multiple Qualitatively Different Plan Options. *In AIPS*, pp. 27–35.

Tate, A. 1977. Generating Project Networks. In *Proceedings of IJCAI-77*, pp. 888-893.

Wolverton, M. 2004. Prioritizing Planning Decisions In Real-World Plan Authoring. In *Proceedings of the ICAPS04 Workshop on Connecting Planning Theory with Practice.*

Young, R.M., Pollack, M.E., and Moore, J.D. (1994). Decomposition and causality in partial-order planning. In *Proceedings of the Second International Conference on AI and Planning Systems*, 188-193, Chicago, IL, 1994.