

Elicitation and Application of Narrative Constraints Through Mixed-Initiative Planning

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 can encode the narrative goals and preferences of the human author of planned interactive narrative. The authors describe a graphical user interface that exploits this metatheory to elicit 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 runtime. 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 met the “Holodeck” standard (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. 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 iterative experimentations and explorations of plan possibilities.

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. 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 a 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 *isallocation(loc1)* could be added to the operator's list of preconditions, and when object instances of type location are created, corresponding *isallocation(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 be 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. Integrated, scalable vector graphic rendering of both the plan space, and individual plans allow authors to navigate from plan to plan to explore structural differences.

Application to Interactive Narrative

The Zócalo planner used by Bowman is based on Longbow, a decompositional (HTN) partial order causal link planner described by Young, Pollack and Moore (1994). Later, Riedl, Saretto, and Young (2003) describe a technique called *narrative mediation* for managing and responding 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.

In such planners, explicit representation of the particular mediation policies in effect for particular actions is another aspect that can be profitably leveraged within DEF. 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, or abstract specification of the characteristics of failure modes required to resolve plan bottlenecks.

Agent Types

To accommodate the specification of mediation strategies by the human author of interactive narrative two applications of DEF are immediately apparent. First, a mechanism is needed to distinguish between user-controlled agents and system-controlled agents. These are 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. The second level of the hierarchy, just below “anyThing”, can be fixed to contain “agent” and “inanimate”. “Agent” can be further 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. The definition of operator is extended to include specification of the type of agent that is capable of invoking the action (“User”, “NPC”, or the non-committal “Agent”).

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-but-gun-jams” might be substituted for the threatening “shoot(?gun)” action.

Bowman can expand the depiction of complete plans to include the application of all available mediation strategies. The plan author can use Bowman to compare these expanded complete plans to see how resilient each is to user action. Some authors may be interested in ensuring that the alternative narrative paths dictated by alternative user actions are different or similar based on various qualitative criteria. Bowman can summarize these differences through the application of custom heuristics. The author may then choose to change the number or variety of mediation strategies, to achieve their narrative purposes.

Custom Heuristics

Heuristics are rules used to measure the desirability of one plan compared to another. The planner uses heuristics to rank the order in which nodes of the search space will be expanded. Typical uses of heuristics are to favor plans that have fewer steps, plan that have fewer flaws, plans that have more abstract (hierarchical) as opposed to primitive steps, or plans where particularly problematic causal links are established early. With Bowman, the plan author can construct heuristics based on any of the attributes described in DEF: types, dimensions, or measurements to apply relative weights on different flaws or features of a plan. Thus, the author can encode arbitrary narrative preferences and use iterative refinement of the plan space to ensure that optimal levels of “kissing” are in each story, ensure that the possible execution paths have the desired level of conformity or diversity, or simply understand the shortest and longest success paths through a narrative.

Narrative Goal Conflicts

As authors build more narrative goals into their planning problems, it may become more difficult to find complete plans, if these preferences are treated as hard constraints. In the narrative domain, it is likely that authors would prefer a sub-optimal plan to having no solution at all. However, for the planner to recognize all conflicting advice is an intractable problem, as each unsatisfiable piece of advice will drive a search through the entire solution space. Myers (2000b) researched two techniques for partial satisfaction of user-specified advice. The first, minimize introduced local advice violations (MILAV) guarantees that some plan will be produced but may ignore some advice that could be satisfiable. During search, MILAV performs a one-step look-ahead for each task node ranking the impact of all possible choices (i.e. operator selection, variable instantiations) according to the number of new advice violations that each will introduce. Search control then uses this ranking to select options that

minimize introduced advice violations. Unfortunately, locally optimal decisions made via this heuristic may not be globally optimal as there is only a one-step lookahead. The second technique for partial advice satisfaction is Local Maxima Search (LMS). This algorithm selects a seed solution that satisfies a selected subset of user advice and employs a hill-climbing algorithm to iteratively improve on the current best solution by adding a single new piece of advice at a time. This continues until no more advice can be added that will lead to a complete plan. At that point a new seed node is chosen and the process repeats. Thus, LMS embodies an anytime approach that will yield plans of increasing advice satisfaction quality as more time is allotted. Several optimizations for seed selection have been tested

Although both of these techniques seek to maximize the quantity of author-specified advice that shapes the planning process, neither provides a mechanism for the author to differentiate the importance of one piece of advice from another. Bowman allows the author to assign an enforcement priority to each precondition and constraint in the planning problem through a graphical “slider” which is translated into a range of values from 0 to 1. Bowman can provide a master slider for the overall “narrative rigidity threshold” that functions over the same range. If this slider is set on 0.5, for example, all constraints that are labeled with rigidity values between 0 and 0.5 are excluded from the plan problem sent to the planner. This allows the author to exploit the mixed-initiative interface through repeatedly requesting plans at different levels or the slider to see how varying degrees of their authorial preferences affect the diversity of the plan space. The idea of different treatment for conditions derived from different sources is not new, as it was featured prominently in the description of the NONLIN algorithm (Tate 1976). Bowman simply facilitates the author’s control over these nominations.

Nevertheless, these mechanisms do not provide Bowman with sufficient power to capture conditional preferences over variable instantiations within operators. For example, although it allows the author specify a preference that the *?transport-vehicle* for any *move-by-vehicle* actions in the plan be constrained to be of type *gasoline-powered-vehicle*, the specified preference level will be the same for all instances of that action in the plan, regardless of other variable instantiations. A more powerful mechanism would allow authors to differentiate preferences for *gasoline-powered-vehicles* based on whether or not the *?environment* parameter of the action is set to *urban* or *rural*, or based on the *relative-cost-of-gasoline* as defined in the initial state. Myers addressed these types of advice through the role-fill mechanism of her domain Metatheory. For Bowman to handle such requests, one alternative is to formulate these as **ceteris paribus (cp)** preference statements.

CP preference statements may be elicited from the author through logical connectives, similar to file search dialogs found in some operating systems. For example, the italicized entries in the following description may correspond to drop-down menu settings: “Use *gasoline-powered-vehicles* WHEN *?environment* is of type *urban* OR *relative-cost-of-gasoline* = *high* AND ...” Logical connectives may be added by the author to construct

arbitrarily complex statements. Prestwich (et. al., 2005) describe methods by which the optimal solutions for such preferences can be determined. These procedures could be applied to derive from all of the conditional narrative constraints an optimal subset that can be encoded through traditional operator preconditions prior to the invocation of the planner. Bowman can apply special treatment to this derived set of constraints to allow the author to elevate or demote its importance relative to all the other constraints over authorial preferences and rules of the world.

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 Zócalo. 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*.
- Amant, R. S., Healey, C. G., Riedl, M., Kocherlakota, S., Pegram, D. A., and Torhola, M. 2001. Intelligent visualization in a planning simulation. In *IUI '01: Proceedings Of The 6th International Conference On Intelligent User Interfaces* (New York, NY, USA), ACM Press, pp. 153–159.
- Aylett, R.S. 2000. Emergent Narrative, Social Immersion And “Storification” *Proceedings, Narrative Interaction for Learning Environments*, Edinburgh.
- 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.

- Ferguson, G., and Allen, J. F. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *AAAI/LAAI* pp. 567–572.
- Fikes, R., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 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.
- 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.
- 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.
- Prestwich, S., Rossi, F., Venable, K. B., and Walsh, T. Constraint-based Preferential Optimization* In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*. Morgan Kaufmann, 2005.
- 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.
- 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.