AAAI Technical Report WS-11-18

Seven Design Challenges for Fully-Realized Experience Management

David L. Roberts
Department of Computer Science
North Carolina State University
robertsd@csc.ncsu.edu

Abstract

Drama Managers, a specific type of the more general Experience Manager, have become a common subject of study in the interactive narrative literature. With a range of representational and computational approaches, authors have repeatedly developed techniques that enable computers to generate, reason about, and adapt narratives in an interactive virtual setting. In order to fully realize an experience manager, seven representational and computational problems need to be solved, generally on a case-by-case basis. In other words, the choice to use an Experience Manager is the choice to model the design as, and implement solutions to, seven interdependent design problems. We explicitly articulate those design problems and provide a number of examples of methods that both motivate the design problems as well as illustrate a range of approaches to solving them.

1 Introduction

The challenges in authoring computer-based interactive environments has received notable attention in the research community. As hardware technology continues to mature, possibilities for increasingly complex computer-based entertainment, training, or educational experiences arise. The players in those experiences are increasingly expecting a higher degree of agency and meaningful interactivity (Wardrip-Fruin et al. 2009). These opportunities for complexity and increased expectations by players bring new challenges for authors and designers. To address these challenges, researchers have spent considerable effort designing representations and algorithms in an attempt to provide increased expressivity, eliminate the requirement on completely specifying the entire space of experiences, or improve the efficiency of designing experiences.

When you take into account authors' desires to produce a dramatic or pedagogical experience for players, this task of authoring for a highly dynamic environment, that needs to afford meaningful interaction for player agency while maintaining a degree of control for authors to produce their dramatic or pedagogical experiences, becomes a massive undertaking. Commercial computer games produced today can be the result of tens, if not hundreds, of person-years of a coordinated effort among artists, programmers, designers, and authors. An important goal in developing new technologies for authoring such interactive virtual experiences is to provide a paradigm that preserves expressive power for authors without increasing either their authorial effort or their need for advanced technical expertise. In other words, *we need authorial paradigms that make the process of creating interactive experiences more accessible to non-technical experts and easier to accomplish for smaller groups of people.*

One approach to realizing this goal is to implement an omniscient coordinator that tracks players' experiences and adapts the environment to bring about a targeted progression of events. Such a *drama* or *experience manager* is tasked with guiding players in an experience prescribed by authors (Laurel 1986; Riedl et al. 2008). In this paper, we will refer to drama managers and experience managers interchangeably. In general, such managers may have different types of goals, *e.g.*, pedagogical in nature for edutainment systems or training in nature for interactive training systems.

In the past two decades numerous approaches to experience management have been developed. See Mateas (1999) for a survey of early work and Arinbjarnar, Barber and Kudenko (2009) or Roberts and Isbell (2008) for surveys of more recent work. When implementing one of these systems, there are numerous design decisions required of authors. Aside from implementing the environment and authoring the story itself, the seven design questions that must be answered prior to the complete realization of an experience manager can be grouped into two categories: 1) ***representational*** and 2) ***computational***. Within each of these categories, there are a number of more specific, interrelated, design questions that must be answered. In this paper, we will illuminate these design decisions, using examples from the literature to highlight potential answers.

It is our hope that by explicitly describing these design problems, researchers and practitioners alike will have a common framework for discussing, comparing, and attacking problems of interactivity and experience management.

The choice to implement a dramatic, educational, or training experience using an experience manager is, in fact, a choice to model the design of the system as a set of technical problems that need to be solved—similar to a choice to use "Case-Based Reasoning" (Kolodner and Leake 1996) (CBR) to solve an AI problem. A CBR system is a reasoning system that uses a four step process to problem solving, each step in the process representing a distinct technical design question that a CBR system designer much solve in order to implement a complete solution. We believe that by providing a concise description of the design questions associated with the implementation of an experience manager we will enable future progress to occur in the development of experience managers by allowing different researchers to attack pieces of the problem with a better understanding of how their work relates to the work of others.

In the subsequent sections of this paper, we will describe each of the computational and representational problems, first briefly and then in more detail. Along the way, we will cite numerous examples to illustrate the range of potential answers to these design questions.

2 The Computational Questions

To improve the efficiency of authoring an interactive experience, we would like for authors to be able to specify some small set of inputs and for the system to extrapolate those inputs into a fully functioning experience manager, capable of reasoning about players' experiences in relationship to aesthetic criteria specified by authors and evolving the environment in order to realize those goals while strictly preserving the affordance for players to have, or at a minimum perceive, a sense of agency. Put more simply, we want for authors to tell the system naturally what they want and for the rest to "just happen." To realize this, experience managers must have certain computational abilities. We briefly describe those here, and in more detail below.

There are three computational design problems must be solved to fully implement a drama management system:

1. **Goal Selection**: The system must have a way of representing the state of the narrative and encoding (in terms of goals) the author's desired aesthetics for the experience. In addition, the system must have a way to reason about the player's behavior in the environment in order to select the appropriate short-term narrative goals that make progress towards the criteria set forth by the author.
2. **Action/Plan Selection/Generation**: The system must have actions that provide it a way to affect the environment. More importantly, the system must be able to reason about how the actions it takes will affect both the player's experience and its ability to achieve the criteria the author has specified for it, and in doing so either select or generate the best action to realize a short-term goal.
3. **Action/Plan Refinement**: The system must have a way to ensure consistency of the actions it takes given the current state of the environment.

To date, the bulk of work on drama management has been focused on the goal selection problem. Various approaches have been designed and to varying degrees implemented and tested in simulation or with actual game environments. Despite the significant representational and computational power provided by many of the approaches to experience management found in the literature, the systems have relied heavily on the author to implement solutions to the action/plan selection/generation and refinement problems.

3 The Representational Questions

Having briefly mentioned the computational issues at play in the implementation of an experience manager, we now discuss the data structures and inputs upon which those algorithms will operate. Indeed, these concepts are inextricably linked—the design choices an author makes for representing the narrative, the environment, *etc.* will have a profound effect on the computational properties of their algorithms. Similarly, the choice of algorithms to solve the three computational design questions will dictate, to a degree, what is required of the representations of the narrative, environment, *etc.* Thus, although we are discussing these design questions separately, a designer actually must take a more holistic view when developing their solutions.

Briefly, there are four representational choices that must be made in order to implement an experience manager:

1. **Representation of content**: Fundamentally, interactive experiences—games, edutainment, training simulations, *etc.*—are inseparable from their content. Separating the question of generation (*e.g.*, story generation vs. authoring), how content gets represented computationally can change significantly from one implementation of an experience manager to another.
2. **Representation of actions**: In order for an experience manager to effect change in the environment and, in doing so, to shape players' experiences, they must have access to actions that operate on the environment in some manner. Authors must provide enough information about actions to enable two things to occur: 1) actions must have a concrete effect on the environment, either completely specified by the author ahead of time or by some means of compilation into a concrete implementation; and 2) actions must afford a description of their effect on the world that will serve as a basis for the computational reasoning about player experiences.
3. **Representation of player behavior**: To fully reason about player experiences, managers must have access to information about how players will, or are likely to, act in the environment. Further, information about how actions taken to reconfigure the environment will affect players' behaviors is crucial to a successful implementation.
4. **Representation of author's criteria**: In order for an experience manager to reason about and guide players' experiences, it must have some representation of what it should guide players towards. This representation can be explicit or implicit, and is a significant input into the computational processes that underly an experience manager.

In the growing body of literature on experience management, there have been a large number of published solutions to these questions. The approaches have used a range of representational schemes from bags of beats (Mateas and Stern 2005) for representing content, to formal decision processes (Nelson et al. 2006b) with precedence matrices for content, generic action operators, and probabilistic models of player behavior. Each of these representational choices has certain benefits and certain drawbacks and is more or less suited to particular computational approaches.

4 Representation of Content

Interactive narratives bridge a divide between sandbox environments where players are free to explore and a more traditional storytelling medium, like a book or movie, where the author completely specifies the contents ahead of time.

In a traditional linear narrative, authors typically outline the content ahead of time. Often this is accomplished through the use of storyboarding, a process by which characteristic scenes that depict the setting and characters are organized into a sequence. This process can be very effective for creating linear story structures. It enables authors to get an understanding of the plot quickly and easily, while maintaining the ability to reorganize as necessary.

On the other end of the spectrum are sandbox virtual experiences. Unlike linear plot progressions, sandbox experiences commit heavily to interactivity, potentially at the expense of any notion of story or purpose. Like a child playing in a physical sandbox, participants are free to use whatever is made available to them to construct their own experience. In the virtual setting, rather than using plastic shovels and pails to create, participants use keyboards and mice to operate on the virtual world designers have created for them.

Considering linear narratives on one end of the spectrum and sandbox environments on the other, interactive narratives can be considered to be in the middle, or a little bit of both. Like a linear narrative, participants are supposed to experience a (somewhat) prescribed experience that is defined by the author; however, like a sandbox, participants are supposed to have a high degree of control over their interaction. Thus, the task of representing content for an interactive virtual experience is the task of representing a *set of experiences within a larger space of possible experiences.* The desirable set of experiences defined by authors bring the narrative-like quality to the interactive experience, and, the larger space of possible experiences bring the sandbox-like quality to the interactive experience.

This becomes challenging, from a representational point of view, for two main reasons. First is complexity. As we introduce more and more of an affordance for players to interact in a meaningful way, there is an exponential increase in the content that must be created to support their interactions (Bruckman 1990). The second challenge is a bit more subtle. It has to do with the ability for the representation to capture what is narratively important, while omitting extraneous alternatives available in the sandbox. For example, content defined for the purposes of the experience manager need not follow a typical wall clock progression through time. Events may happen at non-uniform intervals in accordance with the actions of the player. In other words, *the representation of content to the experience manager need not fully express the content the player will experience.* The content that is represented to the experience manager may only be a subset of what is actually authored if some of the details are irrelevant from the perspective of the experience manager. For example, a player's choice to explore a dungeon may lead them through a long series of rooms. But, from the experience manager's point of view, the information disseminated to the player in one of the rooms is all that is important. In that case, what gets represented to the experience manager would be the fact that the player entered the room of import, rather than the path taken to that room.

Thus, representing content requires two design decisions. The first is the format of the representation, and the second is what level of granularity to represent it at. Magerko has classified the format of representations into three categories: *planning languages*, *modular languages*, and *hybrid languages* (2007). Published experience management systems have represented content using a variety of techniques, ranging from relatively concrete to very abstract. Among the most concrete are the "bags of beats" that the Façade drama manager uses (Mateas and Stern 2000; 2003a; 2003b). A beat, being an atomic narrative unit, contains concrete instructions for its effects in the environment. Beats are abstracted into groups with some coherent theme, and the experience progresses as the beats (and the bags themselves) are sequenced. On the other end of the spectrum are the Declarative Optimization-based Drama Management (DODM) plot events (*cf.* (Nelson and Mateas 2005; Nelson et al. 2006a; Weyhrauch 1997)). DODM plot events, combined with an appropriate set of precedence constraints, implicitly create a space of sequences induced by all orderings that are topologically consistent with the precedence constraints. The notion of a DODM plot event is abstract, and may be extremely granular in comparison to the low-level environmental events the player experiences. One additional commonly used content representation is based on artificial intelligence planning techniques like STRIPS operators (Russell and Norvig 2009). Numerous systems have used this logical approach to representing content (*cf.* (Riedl, Saretto, and Young 2003; Young 2001)).

5 Representation of Actions

One of the benefits of using an experience manager is the capability it provides to adapt the narrative experience in response to players' actions. We term these adaptions "actions" because the manager will enact a change to the environment to bring about a change in players' experiences. Rowe *et al.* argue that there are three types of narrative adaptions: *plot adaptions*, *discourse adaptions*, and *user tailoring* (2010). While this classification may not be exhaustive, it does point out that actions serve different purposes.

Actions operate on the environment. The result of an action may, for example, prevent navigation to part of the environment, cause a synthetic interaction to occur, adjust the discourse technique, or any number of other things. For example, in the PaSSAGE system (Thue and Bulitko 2006; Thue et al. 2007), the actions the experience manager takes cause the player to experience plot events better suited to their individual play style. These actions directly modify the environment to change the content players experience.

Independent of how the actions are implemented in the environment, there are a number of representational choices designers must make. Ultimately, all of the actions are in service of effecting a change in players' experiences. Further, all of the effects of the actions must be on the environment directly, even if the action itself is designed to effect some change in the player—actions that operate on players can only do so indirectly through environment changes.

There are two dimensions, each with two values, that illustrate the range of action representation choices. The first dimension, the *effect* dimension, dictates whether an action is represented as having a deterministic outcome (as is the case with many artificial intelligence planning systems (Young 2001; Young, Moore, and Pollack 1994; Riedl and Stern 2006)), or has a probabilistic effect (as is the case with statistical methods (Mott and Lester 2006; Roberts et al. 2007; 2006)). If an action results in a concrete change to the environment, such as locking a door, a deterministic representation is appropriate; however, if an action to communicate information to the player results in a newspaper containing the information being placed in the environment, there is no guarantee the player will read it. While the placement of the newspaper may be deterministic, the conveyance of the information is probabilistic.

The newspaper example highlights the second dimension of action representation, the *operator* dimension. Actions can be represented as operating on the environment directly or operating on the player, mediated through the environment. Due to the nature of these virtual experiences, it is impossible to know for sure the outcome of an action designed to operate on the player. Therefore, those actions must be represented probabilistically. For example, the social psychological influence employed as actions that operate on the player in certain DODM implementations (Roberts 2010; Roberts et al. 2009) effect a change in the likelihood of players making certain decisions, but can never guarantee a particular outcome. On the other hand, actions that operate on the environment can have deterministic or probabilistic outcomes. Depending on the implementation, any change made to the environment can be deterministic or probabilistic.

Thus, the designing an action representation first requires a choice as to what the actions will operate on. If the actions will operate on the environment, then their outcome can be represented either probabilistically or deterministically.

6 Representation of Player Behavior

The algorithms that get implemented for an experience manager are designed to strike a balance between the author's criteria and players' expressions of agency. Therefore, it is vital that a model of how players will behave in the environment be included in some form. The information contained in the representation of player behavior provides a description of what the player is likely to do or experience both in the presence of experience manager actions and in their absence. This information plays a vital role in the determination of when and how an experience manager should act.

Smith *et al.* propose a taxonomy that characterizes the design of player models along four dimensions: *scope*, *purpose*, *domain*, and *source* (2011). While designs may vary along those four dimensions, specific instantiations of models may take on other characteristics. Various levels of granularity and various semantic interpretations can be ascribed to specific instances.

One such approach to representing player behavior is a relatively abstract model of player behavior that aggregates low-level player actions that occur between significant plot events. This type of model is common in DODM systems (*cf.* (Nelson and Mateas 2005; Weyhrauch 1997; Roberts et al. 2007)) as well as planning systems (*cf.* (Riedl, Saretto, and Young 2003; Young 2001)). Like the action representation, the representation of player behavior can be probabilistic or deterministic. Probabilistic representations of player behavior illustrate how the player is likely to respond to experience manager actions in various settings. They describe both a range of possibilities for player actions as well as their relative probabilities. On the other hand, deterministic models of player behavior define either a single outcome that players will cause, or a set of possible outcomes without any indication of their likelihood.

An alternative player model approach uses preferences. The model-driven player behavior representation of PaSSAGE (Thue and Bulitko 2006) and the case-based player preference model used in a variant of DODM (Sharma et al. 2007) are examples. In these representations, players are assumed to be "preference seeking," and therefore represented as likely to pursue content they will enjoy. Players' content preferences are either knowledge engineered (PaSSAGE) or learned through a comparison of players' decisions to those of previous players (CBR approach to DODM).

Orthogonally, the representation of player behavior can be implemented at a high or low level of granularity with respect to the representation of content. High granularity models will provide detailed information about how the player will interact with the environment, even if that information is not immediately pertinent to the content representation. These types of representations will be harder to author because of the increased level of specificity, but may yield a more accurate implementation. For example, the player behavior model used in EMPath (Sullivan, Chen, and Mateas 2008; 2009) takes into account the physical proximity of plot events when determining their likelihood. This is in contrast to models that assume all legal subsequent plot events are equally likely (*cf.* (Weyhrauch 1997; Nelson and Mateas 2005)). These models are examples of a high granularity representation of player behavior.

Thus, there are two design dimensions for representing player behavior: representing players' actions vs. representing players' preference; and representing behavior with a high level of granularity to include extraneous information about low-level environmental details or not.

7 Representation of Authors' Criteria

Because experience managers incorporate qualities of a linear narrative into open sandbox environments, managers need some representation of what the author's aesthetic criteria for the player to experience are. The choice of this representation will be a choice in concert with the computational choice for goal selection (described in Section 8).

The expressive power of the representation of authorial goals will directly effect the ability for the author to specify certain types of goals. For example, the author may desire for the player to experience one of a set of predefined experiences, or may favor experiences where the player has the most options. Perhaps the author wants the experience manager to do the minimal amount of manipulation, or prefers a more heavy-handed approach. Some authors may prefer a range of experiences for players that is random (in a controlled way), while others may desire for more deterministic control over the minutiae of the experience.

There are representational decisions that need to be made in order to effect the type of control that authors may desire. The manner in which the author's criteria are represented can either be abstract or concrete. For example, the Façade experience manager (Mateas and Stern 2000; 2003b), being a drama manager represents goals in terms of the narrative progression. Narratives progress according to a key value. In this case, that value is "dramatic tension." The author's goals are abstractly represented as a curve where the level of tension rises slowly throughout the experience until the resolution at the end. The content that gets sequenced for the player is annotated with information about how it effects changes in dramatic tension, so the experience manager can order content to match the desired tension arc as closely as possible. Note, these annotations are an abstract representation of criteria—the change in tension is something the player experiences in response to the selected content.

On the other hand, some systems will rely on a more concrete representation of the author's criteria that is defined in terms of the content representation. For example, the Mimesis system (Young 2001), along with many other artificial intelligence planning experience managers, defines the author's criteria in terms of a concrete sequence of events. This is in contrast to the abstraction present in the Façade drama manger criteria specification.

Another way to characterize this distinction is as the difference between a *feature-based* representation or an *example-based* representation. In early work on the DODM drama manager (Nelson and Mateas 2005; Nelson et al. 2006b; Weyhrauch 1997), the author-defined aesthetic goals were represented in terms of a mathematical formula, called an "evaluation function." Using specific *features* of the sequence of narrative events, a numerical score could be assigned with higher scores indicating a higher desirability in the eyes of the author. These functions were defined as a linear combination of features about the plot progression. In later work on the DODM formalism, authorial goals were encoded using "exemplars" (Roberts et al. 2007; Roberts 2010). In that representation, example plot progressions formed the center of a neighborhood of desirable stories. The "closer" to a neighborhood center a specific plot progression is, the more desirable it is and the "farther" from a neighborhood center, the less desirable.

Thus, in designing a representation for the author's criteria, designers must choose whether to represent them deterministically or probabilistically, and further whether to represent them as a *function* of the content experienced by players or as the content itself. To our knowledge, there has been no research to date that examines which of these paradigms are best suited for authors at any level of experience. That is a significant area for future research.

8 Goal Selection

Having described the representational design choices that authors must make while implementing an experience management system, we now turn our discussion to computational questions. Chief among those questions is that of *goal selection*. This problem has received a disproportionate amount of attention in the literature on experience management. Concisely, the problem of goal selection for experience management is: "Given the content the player has experienced thus far, what content should come next to provide the best experience for the player (according to the author's criteria)?" More specifically, the experience manager must determine what action to take at the current time that will result in the best outcome. To accomplish this task, the experience manager must have knowledge of the environment, the history of content experienced by the player, their likely reactions to any changes to the environment, the set of things that can be done to effect change in the environment, and how those changes relate to realizing authors' criteria. With the addition of history information, the four representations described above are required for computing goals.

There is an important distinction to make between the criteria that authors define for the experience and the computation of goals. The former is a description that summarizes the overall quality of the entire experience from a particular perspective. The latter is a short term concept that is a building block to realize the former. These short term goals are, in effect, a decomposition of the author's criteria into smaller, more immediately attainable steps. Another way to think about the goal selection problem is as the "temporal credit assignment problem" (Sutton 1984). The temporal credit assignment problem arises in repeated decision making situations. Perhaps the most intuitive example of this is the game chess. In chess, players alternate making moves on a board until the end game. While a game of chess may result from forty or more moves by each player, the end result of a win, loss, or draw is only determined by the last move; however, it may be that the twentieth move out of forty contributed largely to the ultimate outcome. The process of assigning "credit" to the actions in a sequence that contribute to the outcome requires performing temporal credit assignment. Similarly to chess, the ultimate experience a player has is what gets measured against authors' goals, so the process of computing shorter-term goals that will contribute to satisfying authors' criteria requires temporal credit assignment. In fact, it has been argued that chess is actually an interactive narrative (Roberts, Riedl, and Isbell 2009).

In the published literature on experience management, a variety of computational methods have been applied to the problem of goal selection—even if the term hasn't been used before. These methods have ranged from efficient linear algebra computations (Thue and Bulitko 2006; Thue et al. 2007) to statistical machine learning methods (Mott and Lester 2006). For example, consider the DODM incarnation that uses past player preferences and case-based reasoning (CBR) to make decisions about how to adapt the story (Sharma et al. 2007). Using case-based reasoning, decisions are made by comparing the current situation to previous situations, finding those that are closest, and determining what goal to pursue based on the goals that were pursued in the past. On the other end of the spectrum, are Targeted Trajectory Distribution Markov Decision Process (TTD-MDPs). TTD-MDPs are a statistical machine learning formalism that leverage probabilistic reasoning to create a "policy" specifying the relative probability of short term goals in relation to the authors overall criteria. They have been applied to the goal selection problem in DODM (Roberts et al. 2006; Bhat et al. 2007; Roberts et al. 2007; Roberts 2010).

Continuing with these two examples, CBR and TTD-MDPs, there are vastly differing requirements for the representational decisions that must be made. In the case of CBR, the representation of player behavior is based on a corpus of data collected about previous players. These data indicate what the appropriate action for a player that has exhibited similar playing characteristics in the past is. Thus, the representation of behavior is a deterministic one. Further, the representation of authors' criteria for a CBR-based system is made in terms of a single assumption: "Do what players are most likely to enjoy." This makes it deterministic and concrete. One the other hand, the representation of player behavior in TTD-MDPs is probabilistic. It is defined as a probabilistic transition function between content events. So it is concrete. The representation of authorial goals is defined as a probability distribution as well. Depending on the particular implementation, this could be a distribution directly over the contents or over some features of the contents, making it either concrete or abstract depending on the implementation (Roberts et al. 2007). When applying TTD-MDPs to compute goals for experience management, the computation is to determine a probability distribution over action that will result in the desired relative probability of subsequent plot events.

9 Action/Plan Selection/Generation

The second of the three computational design choices that authors must make is the *action/plan selection/generation* problem. It is the first step, of two, in formulating and implementing a plan to realize the goal that is identified by the goal selection algorithm. We term it the action/plan selection/generation problem for two reasons: the things that experience managers can do in the environment may be atomic (without temporal extent) in the case of actions, or may require a series of atomic actions in the case of plans. Further, these actions or plans can be pre-specified by authors, making the task a retrieval and selection problem, or they can be created by the experience manager on the fly, making the task a generation problem. Let's consider this in more detail.

The problem of selecting or generating an action or plan for an experience manager boils down to this: given a goal that has been identified, what is the best strategy to realize it in the environment. *Given the range of types of modifications the experience manager can make to the environment, what is the right one to use in the given situation?* This is exactly the selection/generation problem. Note that this *strategic* choice is different from how the actions will be implemented, which is a problem of *tactics* and the focus of the refinement problem described in the next section.

We believe this paper is the first time this design choice has been explicitly articulated as a part of the experience management design process. Therefore, for the most part, the approaches to this problem in the literature have been to hand-author a solution and little has been written on any computational techniques. The notable exception to this is work where social psychological influence has been modeled to inform the generation of natural language that will persuade players to make certain decisions (Roberts 2010). In that work, a library of "schema" that contained natural language templates authored specifically to effect influence was used. The templates in the schema were abstract in the sense that they didn't apply to anything in particular in the environment. They simply represented the strategy for the application of influence. Schema were created for different influence techniques that could be applied in different settings. For example, one schema for applying the "scarcity" principle (Cialdini 1998) to objects was created and another for applying it to actions. There were also schema for applying the "reciprocity" principle (Cialdini 1998) that used the notion of reciprocal concessions. Because there were a variety of schema, each applicable in different settings, the task of plan selection was to identify which schema were applicable, examine their effectiveness, and determine which one would match the selected goal most closely.

Despite being the only published technique for selection or generation, there are a large number of methods that could be used to solve this design hurdle. The influence work was based on theories of marketing communication, which lend themselves to natural language; however, other theories more appropriate for physical manipulation of a graphical environment could be used. One possibility is to use a model of visual attention (Knudsen 2007) to draw players' attention, and therefore actions to specific parts of the environment. There are also plenty of options for this design decision that don't require a knowledge of psychology. For example, in a dungeon setting, navigational assistance can be provided (or a more heavy-handed navigational requirement can be imposed) simply by locking and unlocking doors between rooms as appropriate. To make this example more concrete, supposed the goal selection algorithm determines it is best to encourage the player to fight an opponent that resides four rooms away from the player's current location. The experience manager then can generate a plan to force the player into that room by running a path planning algorithm to determine a good path to the target room, and then locking all the doors that don't follow the path. Here, we use "locking" to mean all things that could be performed in the environment to block the path (*e.g.*, actually locking doors, removing doors from the environment, having them lead to empty rooms, camouflaging them, *etc.*).

Thus, to select or generate actions or plans for the environment that move toward realizing a selected goal, authors must provide a framework for reasoning about how change can be effected. Running the gamut from completely pre-specifying rules for every possible environment change, to leveraging existing theories that must be adapted for the current situation, the selection/generation design question can be solved in any number of ways.

10 Action/Plan Refinement

The third of the three computational steps required to implement an experience manager is the *action/plan refinement* step. This process takes the strategy determined by the generation/selection process and implements tactics to make the strategy consistent and usable in the environment. During refinement, the details of the current state of the environment are considered and the specifics of the action are created so as to ensure the action achieves the desired effect in the environment without breaking the narrative. In early work on the DODM approach to experience management, Nelson *et al.* describe how abstract actions must be "refined" to make them concrete, and therefore useful (2006); however, they stop short of suggesting this process can (or should) be automated.

The choice of refinement algorithm will be very closely tied to the algorithm used for selection/generation—the output of the selection/generation process is the main input into the refinement process. Separating the refinement process from the selection/generation process enables modularity and extensibility, but both can be accomplished by the same algorithm if designers choose to do so. The modularity that is achieved by separating the processes enables the representational and other two computational efforts to be reused across domains. That is, the strategy of how to effect change can apply in many settings, but the details of what gets changed to implement that strategy will be different depending on the environment. Thus, by separating the processes, solutions to the selection/generation problem that are applicable in many different settings can be designed, and paired with an environment-specific refinement algorithm.

To make this more concrete, let's consider the use of social psychological influence as actions again (Roberts et al. 2009; Roberts 2010). The domain in which that was applied and tested was a web-based choose-your-own-adventure-style story. When the experience manager selected a goal, an appropriate schema was selected which resulted in a sequence of natural language templates. Refining this action requires the templates be unified and adapted to fit into the environment by formatting the resulting utterance for the web browser to insert it into the story. This is the refinement process. In other environments, refinement might include applying formatting for graphical virtual worlds, creating non-player character dialogue, or perhaps creating an object with the text written on it and having the player find it, *e.g.*, a newspaper.

The task of designing a refinement algorithm for an experience manager is the task of creating a set of production rules for formatting and presenting content to players. By separating the formatting from the selection/generation process, adapting an experience manager for different environments becomes a much easier task. Further, reusing refinement strategies across stories becomes possible as well.

11 Conclusion

In this paper, we have explicitly articulated seven distinct, but inter-dependent, design problems that must be solved in order to implement an experience manager. The distinction between representational design choices and computational design choices is slightly muddied by the their dependence on one another. The choice of representation affords certain computational abilities (or does not afford them) and, the choice of computational approach necessitates certain representational choices. While all four of the representational design tasks and three computational design tasks are distinct problems that require a solution prior to an experience manager becoming usable, they must be considered together in order to create the ultimate solution.

While many of these problems have published solutions in the literature, to our knowledge nobody has articulated them thus far. In highlighting these challenges and their relationship, we hope to have 1) created a common vocabulary for speaking about parts of experience management implementations; 2) pointed out where additional efforts by researchers may benefit authors; and 3) spurred a better understanding about how research efforts can relate to each other, and hopefully created an increased potential for interoperability among experience manager systems.

References

Arinbjarnar, M.; Barber, H.; and Kudenko, D. 2009. A critical review of interactive drama systems. In *AISB 2009 Convention: Adaptive & Emergent Behavior & Complex System, Symposium on AI & Games*.

Bhat, S.; Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; and Mateas, M. 2007. A Globally Optimal Algorithm for TTD-MDPs. In *Proc. of the Sixth Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS07)*.

Bruckman, A. 1990. The combinatorics of storytelling: Mystery train interactive. Interactive Cinema Group internal paper, MIT Media Lab.

Cialdini, R. B. 1998. *Influence: The Psychology of Persuasion*. Collins.

Knudsen, E. I. 2007. Fundamental components of attention. *Annual Review of Neuroscience* 30(1):57–78.

Kolodner, J. L., and Leake, D. B. 1996. A tutorial intro. to case-based reasoning. In Leake, D. B., ed., *Case-Based Reasoning: Experiences, Lessons, and Future Directions*.

Laurel, B. 1986. *Toward the Design of a Computer-Based Interactive Fantasy System*. Ph.D. Dissertation, Drama department, Ohio State University.

Magerko, B. 2007. A comparative analysis of story representations for interactive narrative systems. In *Proceedings of the 3rd Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE 07)*.

Mateas, M., and Stern, A. 2000. Towards Integrating Plot and Character for Interactive Drama. In *Working Notes of the Social Intelligent Agents: The Human in the Loop Symposium, AAAI Fall Symposium Series*.

Mateas, M., and Stern, A. 2003a. Integrating Plot, Character, and Natural Language Processing in the Interactive Drama Façade. In *Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE03)*.

Mateas, M., and Stern, A. 2003b. Façade: An Experiment in Building a Fully-Realized Interactive Drama. In *Game Developers Conference: Game Design Track*.

Mateas, M., and Stern, A. 2005. Structuring Content in the Façade Interactive Drama Architecture. In *Proc. of the 1st AI and Interactive Digital Entertainment (AIIDE05)*.

Mateas, M. 1999. An Oz-Centric Review of Interactive Drama and Believable Agents. In Woodridge, M., and Veloso, M. M., eds., *AI Today: Recent Trends and Developments*, volume 1600 of *Lecture Notes in AI*. Springer.

Mott, B. W., and Lester, J. C. 2006. U-Director: A Decision-Theoretic Narrative Planning Architecture for Storytelling Environments. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, 977–984. ACM Press.

Nelson, M. J., and Mateas, M. 2005. Search-Based Drama Management in the Interactive Fiction Anchorhead. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05)*.

Nelson, M. J.; Ashmore, C.; and Mateas, M. 2006. Authoring an interactive narrative with declarative optimization-based drama management. In *Proceedings of the 2nd Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2006)*.

Nelson, M. J.; Mateas, M.; Roberts, D. L.; and Isbell, C. L. 2006a. Declarative Optimization-Based Drama Management in the Interactive Fiction Anchorhead. *IEEE Computer Graphics and Applications (Special Issue on Interactive Narrative)* 26(3):30–39.

Nelson, M. J.; Roberts, D. L.; Isbell, C. L.; and Mateas, M. 2006b. Reinforcement Learning for Declarative Optimization-Based Drama Management. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*.

Riedl, M. O., and Stern, A. 2006. Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training. In *In Proc. of the 15th Conf. on Behavior Representation in Modeling and Simulation (BRIMS06)*.

Riedl, M. O.; Stern, A.; Dini, D.; and Alderman, J. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4(2).

Riedl, M. O.; Saretto, C. J.; and Young, R. M. 2003. Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03)*.

Roberts, D. L., and Isbell, C. L. 2008. A Survey and Qualitative Analysis of Recent Advances in Drama Management. *Int. Transactions on Systems Science and Applications, Issue on Agent Based Systems for Human Learning* 3(1):61–75.

Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; Mateas, M.; and Littman, M. L. 2006. Targeting Specific Distributions of Trajectories in MDPs. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI06)*.

Roberts, D. L.; Bhat, S.; Clair, K. S.; and Isbell, C. L. 2007. Authorial Idioms for Target Distributions in TTD-MDPs. In *Proc. of the 22nd Conf. on Artificial Intelligence (AAAI07)*.

Roberts, D. L.; Furst, M. L.; Isbell, C. L.; and Dorn, B. 2009. Using Influence and Persuasion to Shape Player Experiences. In *2009 Sandbox: ACM SIGGRAPH Video Game Proceedings (SIGGRAPH Sandbox)*.

Roberts, D. L.; Riedl, M. O.; and Isbell, C. L. 2009. Beyond adversarial: The case of game ai as storytelling. In *Breaking New Ground: Innovation in Games, Play, Practice and Theory, Proceedings of the Digital Games Research Association 2009 Conference (DIGRA 2009)*.

Roberts, D. L. 2010. *Computational Approaches for Reasoning About and Shaping Player Experiences in Interactive Narratives*. Ph.D. Dissertation, Georgia Tech.

Rowe, J.; Shores, L.; Mott, B.; and Lester, J. 2010. A framework for narrative adaptation in interactive story-based learning environments. In *In Proceedings of the FDG'10 Workshop on Intelligent Narrative Technologies III (INT3)*.

Russell, S., and Norvig, P. 2009. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3nd edition.

Sharma, M.; Ontañón, S.; Strong, C.; Mehta, M.; and Ram, A. 2007. Towards Player Preference Modeling for Drama Management in Interactive Stories. In *Proceedings of the Twentieth International FLAIRS Conference (FLAIRS07)*.

Smith, A. M.; Lewis, C.; Hullett, K.; Smith, G.; and Sullivan, A. 2011. An inclusive view of player modeling. In *Proceedings of the 6th International Conference on Foundations of Digital Games (FDG 2011)*.

Sullivan, A.; Chen, S.; and Mateas, M. 2008. Integrating Drama Management into an Adventure Game. In *Proc. of the Fourth Conf. on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 08)*. AAAI Press.

Sullivan, A.; Chen, S.; and Mateas, M. 2009. From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience. In *Proc. of the 2009 AAAI Spring Symposium on Intelligent Narrative Technologies II*.

Sutton, R. 1984. *Temporal Credit Assignment in Reinforcement Learning*. Ph.D. Dissertation.

Thue, D., and Bulitko, V. 2006. Modelling Goal-Directed Players in Digital Games. In *Proc. of the 2nd AI and Interactive Digital Entertainment Conference (AIIDE06)*.

Thue, D.; Bulitko, V.; Spetch, M.; and Wasylishen, E. 2007. Interactive storytelling: A player modeling approach. In *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE07)*.

Wardrip-Fruin, N.; Mateas, M.; Dow, S.; and Sali, S. 2009. Agency reconsidered. In *Breaking New Ground: Innovation in Games, Play, Practice and Theory, Proc. of the Digital Games Research Ass. 2009 Conf. (DIGRA 09)*.

Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, School of Computer Science.

Young, R. M.; Moore, J. D.; and Pollack, M. E. 1994. Towards a Principled Representation of Discourse Plans. In *Proc. of the 16th Conf. of the Cognitive Science Society*.

Young, R. M. 2001. An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment. In *Working Notes of the AAAI Spring Symp. on AI and Interactive Entertainment*.