Directable Synthetic Characters

Mazin Assanie

University of Michigan Artificial Intelligence Lab
1101 Beal Avenue
Ann Arbor, Michigan 48103
mazina@umich.edu

Abstract

Assuming a centralized narrative control mechanism, we must build synthetic characters that can accept and enact direction in real-time from the narrative manager. These characters must adhere to both the character design as intended by the author, as well as the performance requirements of their autonomous behavior, such as military doctrine or social roles. To explore how we can build such artifacts, we first analyze the requirements that directable synthetic characters must meet. We formalize these in terms of both directability and believability. We then examine the various types of conflicts that can occur when our synthetic characters are faced with a direction. Finally, we propose an agent design that integrates deliberate goal pursuit with coherent improvisation to achieve the type of believable behavior such directable systems would require.

Introduction

Developers want to create deep, immersive, intricately detailed, freely explorable worlds. Furthermore, developers are required to make sure that an experience emerges in this hard-to-control, dynamic virtual world that meets some author goal, such as an interesting story, general enjoyability, fright, mystery or what not. To this end, narrative management systems will be created that will influence the unfolding narrative by manipulating the world. The key manipulation technique we are concerned with in this research is the directing of the synthetic characters.

In a nutshell, we want synthetic characters who behave as improvisational actors with an earpiece from a director who is sitting in a control room with cameras trained on the whole world. When faced with either the unexpected actions of the user or direction via their earpiece, they know how to believably improvise their actions so that they achieve the direction. This may be difficult because the direction can produce behavior that either conflicts with the author's vision for the character or is incongruent with the character's previous behavior. Building agents that can satisfy this particular challenge is what this work is focused on.

Design Goals of Directable Characters

One of the difficulties with this area is that it requires a variety of capabilities – as is evidenced by the eclectic nature of the systems developed – and it must meet a large set of poorly defined, often subjective requirements (Bates *et al* 1991). Though the focus of this research is the directability of the agents, we need to determine the set of design goal characteristics required for an agent's behavior to be considered believable. This will allow us to focus our examination on those characteristics most relevant to directability.

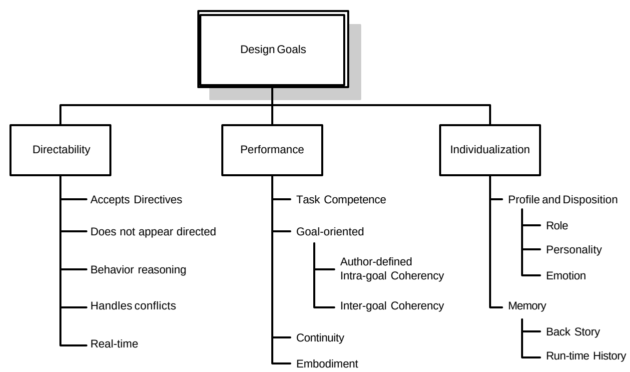

Figure 1: Design Goals of an Ideal Directable Synthetic Character

It is important to note that these goals are not strict requirements. In many cases, some of these design goals are intentionally violated for dramatic purpose. Nevertheless, these items still hold as general design goals. It is also important to note that this taxonomy is a description of the requirements of an ideal complete believable agent. Overall, these goals are very ambitious and, as a complete set, have not been achieved by any single system. We will strive for these goals only insofar as we expect them to interact with the issue of directability.

The first category describes the important elements of directability in our problem formulation. The agent should be capable of accepting the full range of directive commands as defined by the external narrative manager. To integrate the achievement of directives with its previous autonomous activity, the agent must find ways to cleverly fold the directive into its current behavior. To do this generally, it must be capable of reasoning over multiple goals in order that it may avoid conflict between them and find inconspicuous ways to achieve the direction. All of this direction integration must be done within the real-time constraints of the environment in which these agents typically operate (Laird et al 2002).

In terms of performance, the agent's behavior needs to achieve a competent level of performance on par with that of the author's envisioned character. To do this, the agent should pursue goals and find methods to achieve them. *To the degree defined by the author*, the agent's goal should be evident, and its action should appear to contribute to those goals (Sengers 1998b). Moreover, to do this in a believable fashion, this behavior should be based on a model of limited character perception with realistically embodied effectors (Freed et al 2000). Of particular importance to believably integrating direction, the agent should not schizophrenically switch between goals. Rather, all of its behavior should be fairly continuous and fit together in a coherent fashion (Sengers 1998a). In the next section, we will discuss different forms of coherency in greater detail.

The individualization criteria establish the agent's dramatic character. The agent should possess a distinguishable personality that manifests itself in the agent's actions. In some cases, social roles can be used to prime the user's expectations and more clearly define who the agent is (Hayes-Roth and Rousseau 1997; Reilly 1994; Reilly 1996; Rousseau and Hayes-Roth 1996, 1997). Furthermore, the agent should have either an emotional model or a method to approximate emotionally consistent behavior (Bates et al 1992a, b; Hayes-Roth and Doyle 1998; Reilly and Bates 1992; Reilly 1996). Finally, our ideal believable agent should have memory of both the events that occur in the simulated environment and the purported back-story of the character.

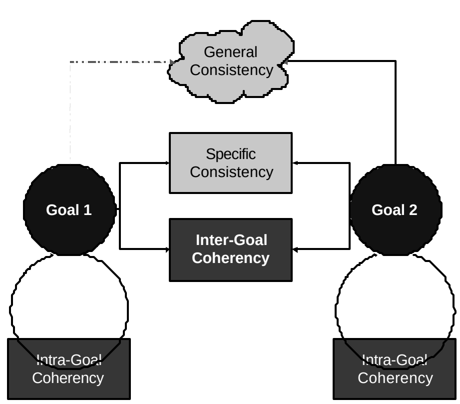

Figure 2: Directed Goal Conflicts

Directed Goal Conflicts

Assuming that direction comes in the form of a known agent goal, we can use our design criteria to generate a set of possible conflicts that a directed goal can introduce. We will define two categories of conflicts:

Goal Consistency: compatibility between goals or among components of goals.

Goal Coherency: the quality of having a clear and logical structure between goals and being understandable in and of themselves.

We can split our definition of believable goal consistency into two cases: specific consistency and general consistency. We define *specific consistency* as complete compatibility between the conditions of two goals or the components of two goals. Specific inconsistency can occur because either (a) the goal conditions themselves conflict – both goals cannot coexist at the same time – or (b) there are potential threats in the achievement of the two goals -- the precondition of some step of one goal is potentially clobbered by a step of the other goal. Obviously, this is completely analogous to conflicts and threats in planning literature.

We say that behavior is *generally inconsistent* when it is incongruent with the user's general knowledge about the agent's purported personality, role and emotional disposition. In other words, generally inconsistent behavior is somehow "out of character."

We can also delineate two forms of believable goal coherency. Intra-goal coherence deals with whether one particular goal is itself understandable. To make sense of an agent's action, the user must have *some* knowledge of both the goals of the agent and how the actions contribute to those goals. This is an obvious, but often overlooked requirement for all believable agents (Sengers 1998b). As a caveat, one should note that in a real system, this constraint is far from strict. An agent that goes around announcing every one of its intentions would seem quite silly; nevertheless, making goals evident is an important and specific step that should be explicitly considered. When, how often and in what manner intentional information is communicated to the user is an issue that, in most cases, would be determined by the character author.

Inter-goal coherence is concerned whether the switch from goal to goal has structure that makes sense. Both the agent's goal switches and lack of goal switches must seem rational. There should be some evident reason to necessitate the changing to a new goal. For example, an earlier goal may have been completed, or the new goal is more important. Furthermore, the agent may want to finish old goals that are near completion, and/or resume suspended old goals. Inter-goal coherence is also concerned with natural goal transitions – this is what is meant by *clear and logical structure* in the definition. Ideally, one goal should seamlessly transition into the next.

Research Direction

To examine how we can build agents that satisfy these requirements, we will explore extensions to the hand-encoded expert rule-based agents – which we will abbreviate at HERBs – that we have built for other virtual environments, for e.g the QuakeBot and Tac-Air Soar pilots (Laird and Jones 1998; Laird et al 1999). Our goal is to explore one facet of directability, namely inter-goal coherence. To do this, we will modify the decision-making mechanism of our agents so that it can improvise new behavior that satisfies the direction while maintaining coherent goal transitions.

The main difficulty that HERBs face when dealing with direction is that they cannot find a way to smoothly transition into the directed goal unless the author envisions the scenario and encodes a transition. Therefore, the critical new ability that the synthetic character needs is a general mechanism to effect coherent transitions. Our approach is to find or, in some cases, create new goals that allow the agent to smoothly transition between the old goal and the directed goal. To achieve this, the agent composes *a set of goals* that would be reasonable to pursue. It can then evaluate whether the transition from each of these goals to the directed goal is more natural than simply pursuing the directed goal. If so, it can pursue the most natural transition as a mediating goal and then repeat the process.

To implement this extension, we need to develop two main augmentations, which we will discuss in the upcoming sections.

- First, we must create *coherency heuristics* that evaluate how smooth a transition between two goals may be. The agent needs these to decide whether one transition is more coherent than another.
- Second, we need to extend the standard HERB goal selection mechanism so that it can effectively use these heuristics to improvise the new goal into its behavior.

For the time being, we will leave out the issue of creating new transitioning goals. This topic has been addressed extensively in Phoebe Sengers' thesis work (Sengers 1998a). Though we make no significant additions, her techniques would work quite well within our proposed system.

Goal Selection

To improvise direction, we cannot select goals in the standard HERBs fashion. We must change two facets of the goal reasoning process, namely that of goal commitment and that of goal selection. Specifically, we must delay goal commitment and use our new heuristics for the final goal selection.

A character's operator hierarchy outlines the reasoning that the agent could perform to achieve its goals. The following is an excerpt from an operator hierarchy used in one of the Soar QuakeBots, an action-oriented agent that navigates a virtual battlefield and battles other agents:

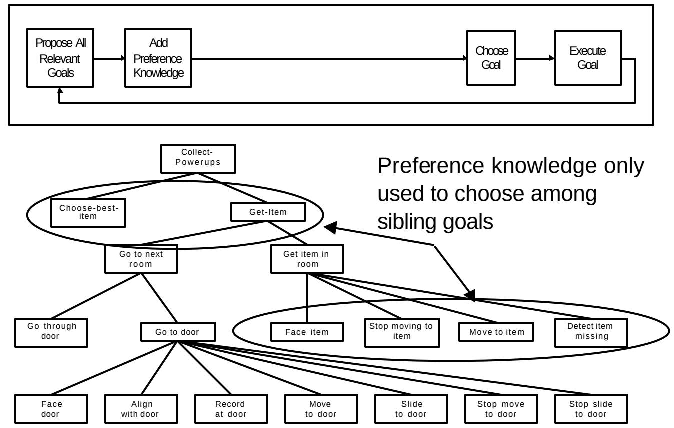

Figure 3: Quakebot Operator Hierarchy and Goal Selection Mechanism

As illustrated, the agent decomposes high-level goals into all possible subgoals that can contribute to its achievement. Eventually, it is able to decompose these goals into actions that describe either how the agent changes its internal state or how it performs some external action. On any level, multiple goals may be proposed. *Preference knowledge* is used to select and *commit*[1] to one of the subgoals of a given goal. The reasoning goes down to the next level, in which the agent may once again choose another goal to commit to.

We originally stated that to improvise, we want our agent to consider multiple contextually possible goals and use the coherency-based preference knowledge generated by our heuristics to choose between them. However, the way a standard HERB system works is that preference knowledge is *only used to select among sibling goals*. In other words, it only chooses among the contextually possible goals *given the higher-level goals it has committed to*. We will modify this so that coherency preference knowledge can potentially be used to select from any two goals in the operator hierarchy.

We do this as follows. We use our standard hand-coded preference knowledge to choose the best goal for a given context. However, rather than committing to that goal and executing the next phase in the reasoning, we add it to an

[1] Please note that this commitment is not firm in the intentional sense; rather, it is a commitment to consider only one particular level of the hierarchy when choosing what to do next. This commitment can easily be disregarded if the context of the situation changes. It is only relevant to this discussion because it delineates the groupings of operators upon which preferences can be applied.

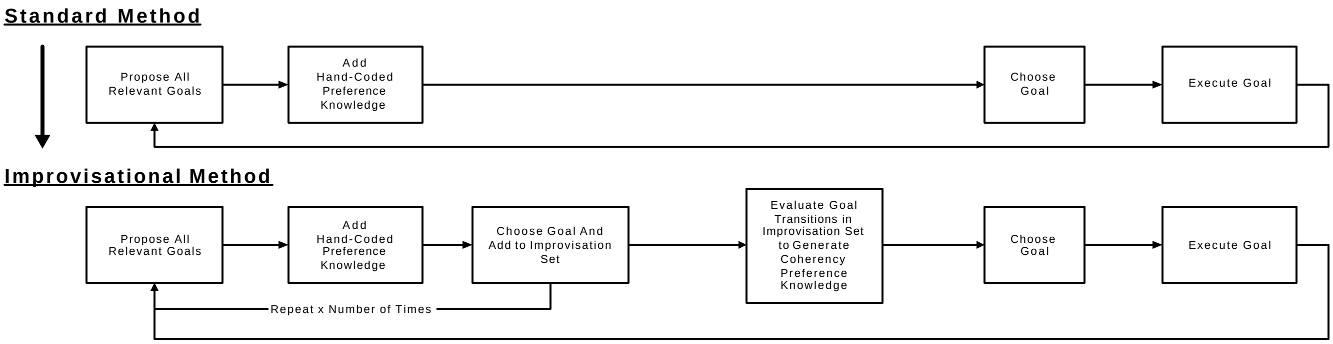

Figure 4: Modified Goal Selection Mechanism

"improvisation set" of goals. We then remove that goal from consideration and use that same hand-coded preference knowledge to find another goal to add, possibly from another part of the operator hierarchy. This can be repeated until we get an improvisation set of the desired size.

To summarize, the standard HERB systems goal selection mechanism is modified as follows:

Approach Overview

The first phase consists of a hand-coded expert system that uses a hierarchical decision making mechanism found in similar agents we have created for other dynamic environments (Laird and Duchi 2000; Laird and Jones 1998; Laird et al 1999). What is different is that, rather than just propose and apply an operator, the agent creates a declarative representation of a goal. We call these representations PIGs or partially-instantiated goals. PIGs are context-dependent declarative data structures that specify a goal and possibly some additional data such as summary information or the results of any problem-solving that occurred when the PIG was added. This information is important because it will facilitate the upcoming coherency evaluations. The agent can then generate a set of these PIGs, which we call the "improvisation set."

The second phase consists of an inter-goal coherence advocate, which operates between the abstract tasks of goal reasoning and action reasoning. This component helps the agent have coherent transitions between the last goal and the next. It does this in one of two ways. In most cases, it does this by using four different heuristics to add coherency-based preference knowledge. This knowledge allows the agent to choose the PIG that, if pursued, would result in the most coherent behavior given the previous goals. In cases when choosing among the eligible PIGs is insufficient to bring about a coherent transition, the coherence advocate can add a special-purpose mediating goal that will make for a smoother transition.

The third phase contains the improvisational execution system. The first thing that must happen during this phase is the selection of a PIG to pursue. Then, using the context of the situation and a set of appropriate execution knowledge, the agent performs primitive actions to effect the goal. Note that this context includes both the characteristics of the environment and the general attributes of the agent such as the personality and emotional disposition. Using these attributes, the algorithm chooses actions that both meet the goals and are generally consistent with the agent's character.

There are three main advantages to using this "advocate" style approach. The first is that it provides anytime improvement of the agent's behavior that can be used as time permits. Without any use of coherence advocates and an improvisation set size of one, the system is equivalent to the standard hand-coded agents, albeit with some overhead. Adding the advocate can incrementally improve the believability of our standard agent's behavior. In a real-time environment, this is an obvious benefit. The second advantage of using an advocate approach is that we have several different strategies at our disposal. Some strategies may be better suited to certain types of transitions. Finally, it allows us promote coherency, in most cases, through the homogenous and general mechanism of providing preference knowledge over the improvisation goal set.

Coherency Heuristics

The basic assumption behind our heuristics is that goals that are similar on some dimension produce transitions that are more coherent. We have four basic techniques for evaluating similarity.

1. **Operator proposal order**: This heuristic uses the default preferences in the operator hierarchy, i.e. the hand-encoded preference knowledge, as a measure of what the agent would most likely do in a particular situation. If the agent would have done that goal anyway, it should be a more natural transition. This heuristic is very general, in the sense that it can encompass any meta-level preference mechanism engineered in the original hierarchy.

2. **Goal hierarchy distance**: This heuristic leverages the hierarchical structure of the goal operator hierarchy to get a rough estimate of how similar two goals may be by measuring how "near" one goal is to another. The assumption is that goals close to each other on the tree are likely to be more similar than ones further away. Similarly, goals that subsume each other are very closely related
3. **Effector information**: This heuristic generates an evaluation based on summary information of the primitive actions that the given goals may produce. The premise behind this heuristic is that using two goals with similar actions will result in clusters of similar actions. This may make any transition harder to detect. Such activity may even appear synergistic.
4. **Engineered knowledge**: We can annotate the goals with some knowledge about the type of goal being pursued. This could range from task classes to the various attributes of general consistency. Using this heuristic, we can measure transition coherency on that specific dimension. The following are the main types of knowledge that we will annotate the proposed goals with:
   - ***Relation to emotion, personality or social model*** Though this research does not make any commitment to what model of personality, emotion or social role is used, the algorithm does need to know how goals relate to these items. At a minimum, those models must provide some venue to measure how similar one goal is to another on these dimensions.
   - ***A classification of the type of activity*** This could be useful if it provides another way to group goals that is not captured in the operator hierarchy.
   - ***A measure of the character's value of a goal*** If the author can provide this, the algorithm could more easily detect when it is switching to a less important goal, i.e. a switch that may appear to the user to be less rational. This is in addition to the relative importance already measured by the proposal order Note that this does not have to be static; the agent could functionally determine this when the PIG is constructed.
   - ***Methods that measure how close to completion a goal is*** The algorithm could use this knowledge to determine whether abandoning or resuming a goal would make the agent appear less rational.

One obvious difficulty is deriving the information necessary to calculate these heuristics and combining the metrics to evaluate relative transitions.

Interestingly, this approach is analogous to the strategies employed in human dramatic improvisation. Hayes-Roth describes some of the real-world heuristics used by improvisational actors as "*accept all offers…do the natural thing … and reincorporate previously generated elements*" (Hayes-Roth and Gent 1996). Our approach does this in some sense. By creating a small menu of different goals for a given context, even though it normally knows what is "best" for it to do, the agent is, in some sense, "accepting all offers." By having the heuristics we've described and a mechanism to prefer those goals that are most coherent with previous goals, the agent will intentionally prefer courses of actions that are more "natural" and "reincorporate previously generated elements."

Restructured Knowledge

To use the techniques we have described, it is critical that we give our agent as many options as possible. One way to do this is to structure the agent's knowledge so that it can consider its goals possible in more situations (even if some of those options are suboptimal.)

Typically, an HERB agent's knowledge is engineered so that it knows exactly what to do in a given situation. As a result, goal proposal knowledge often conflates *whether an agent can pursue a goal* with *when an agent should pursue the goal*. If we separate the latter out into preference knowledge, we can give the agent more options.

For example, consider an agent that is pursuing an "attack" goal. Perhaps, this agent has three possible ways

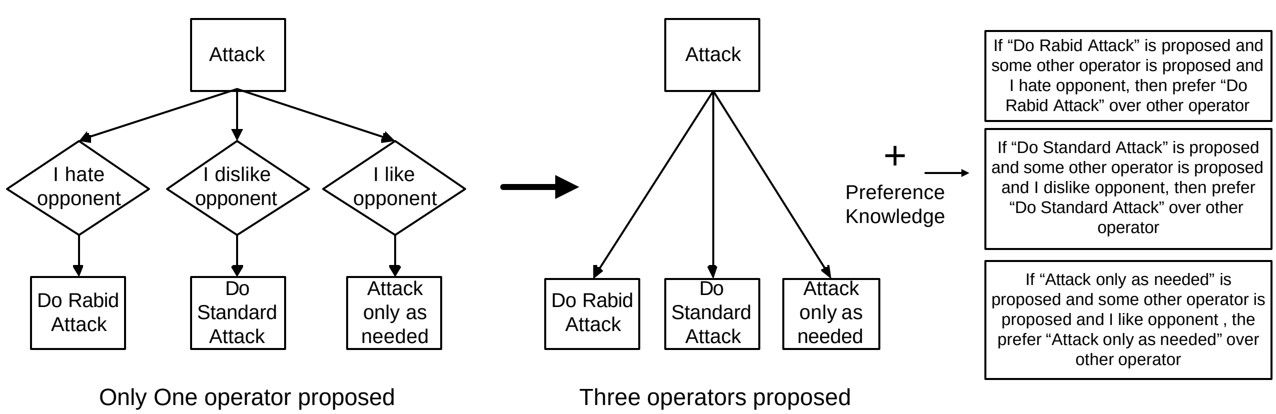

Figure 5: Restructured Attack Operator

to attack based on whether it hates, dislikes or is ambivalent to the opponent. One way to encode this would be to have every one of those affection conditions within the different goal proposals. Another way would be to propose all of the different attack operators, with fewer preconditions, and then use preference knowledge, based on those removed preconditions, to choose which one to do in a given context. By moving some of the proposal knowledge into preference knowledge, we give the agent more options to improvise over. Naturally, the proposals must still be meaningful. In this example, using a different attack operator still makes sense.

Evaluation

One of the problems with a research area that has so many fuzzy and subjective goals is that it is difficult to determine the success of a system. This is true on many levels. If the goals are poorly defined, one cannot concretely determine if the system satisfies them. Furthermore, even when desired results are achieved, it is not clear whether the solution is general or simply the fortuitous result of the hard-coded knowledge base. Often, these systems point at some emergent behavior that is interesting and simply assume that such results can be reproduced. Furthermore, it is very difficult to determine what components of the system contributed to any results that may be achieved.

To make progress in this research area, we need to pay more attention to this issue of credit assignment. Furthermore, we must develop concrete evaluation criteria for evaluating coherency.

Experimental Variations

To test the impact of various components, we plan to create several different variations of the algorithm and see how the quality of the behavior differs.

1. We will try different sized improvisation sets. Our goal is to examine the relationship between increased goal options and quality of improvised behavior, as well as how much computational load increases with a larger PIG limit.
2. We will also try different ways of increasing the limit and see how that affects the answers to the above questions. For example, we can iteratively change proposal limit when direction is received, until a sufficiently good goal is found.
3. We will experiment with both the number of intermediate PIGs permitted.
4. Since we have multiple types of coherence advocacy strategies, we must isolate and test each one of them individually.
5. Furthermore, since only a portion of the heuristics can be used in a given time period, we must also consider the ordering of their use. Moreover, we can try the algorithm with the artificial luxury of infinite time available to see how well it would do in an ideal situation.
6. When the inter-goal coherency cannot find a sufficiently smooth transition among the candidate PIGs, it resorts to adding a mediating goal. This threshold value, which is closely related to the open research question of how to combine the results of multiple heuristics, is a value we will explore.
7. The algorithm also places time limits on each of the phases. We need to examine the manner in which changes to those time limits affects the resulting behavior.
8. We also plan to compare a standard hand-coded agent to a neutered version of the improvisational agent, i.e. PIG limit of 1 and no advocates, to investigate two issues. The goal is to examine both the overhead introduced by our approach and whether the behavior will differ with the knowledge-restructured agent?

Evaluation Criteria

One way to evaluate the effectiveness of this approach id to develop metrics that may indirectly correlate to how coherent the behavior generated is. For example, we may examine the number of imposed goal switches an agent takes to achieve a direction. One could argue that fewer switches indicate more consistent and coherent behavior. Another metric we are considering is the amount of time spent in a portion of operator hierarchy. A large number of abrupt switches from one portion of the tree to another may be indicative of less coherent behavior.

Another way to evaluate coherency is through subjective means. Wilkins and desJardins argue that in realistic domains, evaluating plans is frequently difficult since "humans often have evaluation criteria that cannot be captured precisely and commonsense knowledge that allows them to determine appropriate actions in unusual situations that were unforeseen when the domain was modeled" (Wilkins and desJardins 2001). Similarly, there are many aesthetic or more abstract facets of coherency that are hard to capture numerically or symbolically.

Users who participate in the interactive environment will be able to give feedback on what they found either successful or problematic. One way to do this is to place the user in the environments and ask them to hit a button when he or she thinks an agent is being directed. We can then compare this data to when and how the goals were selected in the agent.

Another way we can generate subjective data is through post run-time surveys. A good example of such an approach can be found in (El-Nasr et al 1999). In this work, El-Nasr attempts to evaluate a fuzzy-logic based emotional model. A survey asks users to evaluate several dimensions of the agent's behavior on a ten-point scale. This is done for three versions of the system: the standard model, one with learning and one that has completely randomized behavior. They then compared the systems by examining how well each did compared to the average value with a confidence interval of 95%. Though based on subjective feedback, we could also use similar methods to isolate statistically significant trends that can point to both successful and problematic areas of our algorithm.

References

Bates J, Loyall A B, Reilly W S, 1992a "An Architecture for Action, Emotion and Social Behavior"

Bates J, Loyall A B, Reilly W S, 1992b "Integrating Reactivity, Goals, and Emotion in a Broad Agent"

Bates J, Loyall B, Reilly W S, 1991 "Broad Agents", in *AAAI Spring Symposium on Integrated Intelligent Architectures*

El-Nasr M S, Ioerger T R, Yen J, 1999 "PETEEI: A PET with Evolving Emotional Intelligence", in *Autonomous Agents*

Freed M, Bear T, Goldman H, Hyatt G, Reber P, Tauber J, 2000 "Towards more human-like computer opponents", in *1999 AAAI Spring Symposium on AI and Interactive Entertainment*

Hayes-Roth B, Doyle P, 1998 "Animate Characters", in *Autonomous Agents and Multi-Agent Systems* (Netherlands: Netherlands) pp 195-230

Hayes-Roth B, Gent R v, 1996 "Improvisational Puppets, Actors and Avatars",

Hayes-Roth B, Rousseau D, 1997 "Improvisational Synthetic Actors with Flexible Personalities",

Laird J, Assanie M, Bachelor B, Benninghoff N, Enam S, Jones B, Kerfoot A, Lauver C, Magerko B, Sheiman J, Stokes D, Wallace S, 2002 "A Testbed for Developing Intelligent Synthetic Characters", in *AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment* (Stanford University: Stanford University)

Laird J, Duchi J C, 2000 "Creating Human-like Synthetic Characters with Multiple Skill Levels: A Case Study using the Soar Quakebot", in *AAAI 2000 Fall Symposium Series on Simulating Human Agents*

Laird J, Jones R M, 1998 "Building Advanced Autonomous AI Systems for Large-Scale Real-Time Simulations", in *Computer Game Developers Conference*

Laird J E, Coulter K J, Jones R M, Kenny P G, Koss F, Nielson P E, 1999 "Automated Intelligent Pilots for Combat Flight Simulation", in *AI Magazine* pp 27-42

Reilly W S, 1994 "The Art of Creating Emotional and Robust Interactive Characters",

Reilly W S, Bates J, 1992 "Building Emotional Agents"

Reilly W S N, 1996 "Believable Social and Emotional Agents"

Rousseau D, Hayes-Roth B, 1996 "Personality in Synthetic Agents",

Rousseau D, Hayes-Roth B, 1997 "Interacting with Personality-Rich Characters",

Sengers P, 1998a *Anti-Boxology: Agent Design in Cultural Context*, Carnegie Mellon University, Pittsburgh

Sengers P, 1998b "Do the Thing Right: An Architecture for Action-Expression", in *Autonomous Agents 1998*

Wilkins D E, desJardins M, 2001 "A Call for Knowledge-Based Planning" *Artificial Intelligence* **22** 99-115