Directed Improvisation by Computer Characters

Barbara Hayes-Roth, Lee Brownston, and Erik Sincoff

Knowledge Systems Laboratory
Stanford University
701 Welch Road, Bldg. A
Palo Alto, California 94304, USA
Telephone: (415) 723-0505
Secretary: (415) 723-3445
Fax: (415) 725-5850
E-Mail: Hayes-Roth@CS.Stanford.edu
Brownston@HPP.Stanford.edu
Sincoff@CS.Stanford.edu

Technical Report: KSL-95-04

Abstract

We present a directed improvisation paradigm, in which computer characters improvise a joint course of behavior that follows users' directions, but also engages and entertains users with the novelty, life-like qualities, and performance properties of their improvisations. We present requirements for improvisational characters that differ from the usual requirements for conventional computer agents and present an architecture that is designed to meet the new requirements. Two implemented characters exploit some of these architectural features to meet simple versions of the requirements. Finally, we illustrate the utility of improvisational characters for a variety of applications related to the arts and entertainment, including a suite of interaction modes in our testbed environment, a Virtual Theater for Children.

1. Introduction

To improvise is to perform a new work in real time, without detailed preparation and by making use of the resources at hand. When improvisation is constrained by exogenously supplied directions, the new work realizes a prescribed structure in an inventive form. The most striking examples of directed improvisation occur in the theater [7, 24, 31] when actors invite the audience to supply a theme or some constraints and then deliver a delightfully entertaining rendition of the theme that cleverly meets the constraints. Directed improvisation also occurs in other domains: jazz music [10], writing [13, 39], scientific investigation [35], planning [17, 20, 36], reactive behavior [1], conversation [29], human-machine communication [36], children's planning, story telling, and playcrafting [2, 3, 33], and even life-course management [6]. Indeed, directed improvisation may be the predominant mode of human behavior and interaction.

We are developing a theory and approach to building computer characters that can perform directed improvisation [22]. Here, the new work is a course of behavior performed by the characters. Directors (who may be human users or other computer agents) give the characters abstract instructions. The characters work together to improvise an engaging course of behavior within the constraints of the directions. Section 2 illustrates directed improvisation, outlines requirements for improvisational characters, and contrasts these with requirements for conventional agents. Our approach combines a generic agent architecture with configurable components to support efficient and economical development of a variety of specific improvisational characters for different applications. It incorporates clean interfaces, so that a given character's "mind" can be embodied in different physical forms within and between media (e.g., animation, virtual reality, text). Section 3 presents our proposed agent architecture, illustrated by simple characters we have created, and descriptions of current work in progress.

We believe that improvisational characters will be useful elements of diverse applications in the arts and entertainment. Section 4 presents our current testbed application, a Virtual Theater for Children, and describes the several interaction modes it supports. Section 5 presents concluding remarks.

2. Requirements for Improvisational Characters

To illustrate the requirements for improvisational characters, Figure 1 presents a simple episode involving a large character (LC) and a small character (SC).[1] A

---

[1]Here, we embody characters as "woggles" in the animation system developed by Joe Bates of Carnegie-Mellon University, which also incorporates the RAL software of Production Systems Technologies.

child has directed LC to act curious and friendly; she has directed SC to act playful. Working within these abstract directions, the characters improvise their own behavior and interpret their partners' behavior as the episode unfolds. At first, LC is alone in the environment. Acting curious, it decides to "look around for something" (1). SC enters (2), observes LC standing still, but infers nothing about LC's behavior (3). Acting playful, it decides to "start something," namely to play alone for a while and then hide (4). Observing SC enter and begin to play alone, LC infers that SC is shy (5). Acting friendly toward a shy character, LC decides to interact with SC by approaching, greeting, and inviting it to play (6). Although they don't know it, LC and SC now have conflicting plans and one of them must change. Which one it is depends on which character acts first. If LC invites SC to play, SC must drop its own plan and agree to play. If SC hides first, LC might respond by "getting into the game." Or if SC notices LC approaching, it might drop its plan to hide and do something else. In any case, each character will readily change its plan to accommodate its partner.

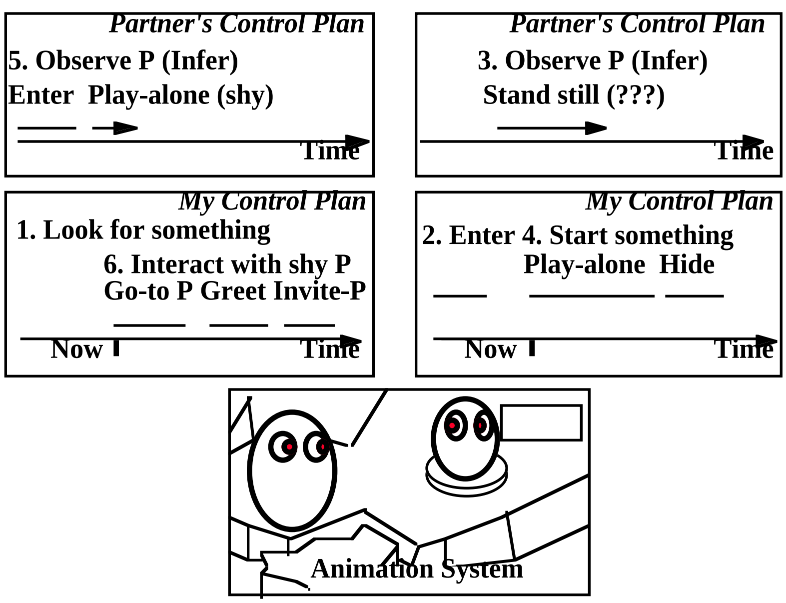

Figure 1. Illustration of collaborative improvisation.

Improvisational characters must have *functional capabilities for perception, reasoning, and action in complex, uncertain, real-time environments.* These capabilities are modeled to some degree in most conventional computer agents (e.g., physical robots, softbots, monitoring agents) [1, 11, 15-18, 20, 21, 25, 26, 28, 30, 40] However, improvisational characters must integrate a greater variety of these capabilities than do most existing agents. For example LC and SC must continually monitor and interpret one another's behavior, monitor the rest of their environment, and control their own behavior to follow current plans—while they mentally improvise internal directions that elaborate the child's directions and adapt to their partners' behavior.

Improvisational characters must *follow directions*. This requirement is similar in spirit to requirements for instructable agents [14, 23, 37], for object-oriented agent simulations [12, 32, 34], and for advanced programming languages in which programmers specify what to do, rather than how to do it [9, 38]. But directions for improvisation are much more abstract (e.g., be curious, friendly, playful), requiring more interpretation and initiative in determining an eventual course of action. In fact, where most direction-following behavior is evaluated in terms of the specificity with which an agent meets the user's implicit intentions, directed improvisation is evaluated for the creative surprises in a character's behavior.

Improvisational characters must exhibit *everyday intelligence*. Most current work focuses on one of four classes of agents: software agents that capture small amounts of knowledge and specialized reasoning skills; robots that emphasize perception and navigation skills, with limited cognitive skills; expert systems that emphasize narrow and deep domain-specific rules for use in static contexts; and monitoring agents that emphasize signal processing and expert reasoning skills in technical domains. By contrast, improvisational characters must function acceptably well in a broad range of situations, without committing egregious errors. They must exhibit common sense about the physical world (e.g., knowing how to hide), a naive psychological model (e.g., recognizing shy behavior) and obvious social conventions (e.g., responding to a friendly greeting). In contrast to conventional intelligent agents, computer characters require what Bates [4] calls "broad, but shallow competence."

Improvisational characters must exhibit *life-like qualities.* Some recent work (e.g., on softbots and interface agents) includes efforts to make agents appealing, or at least non-irritating, to users [11, 25, 26, 28]. By contrast, improvisational characters present much stronger requirements for life-like qualities, which we conceptualize to include: interesting variability in a character's interpretation of a given direction (e.g., act playful) on different occasions; random variability in the way a character performs a specific behavior on different occasions (e.g., greeting) ; idiosyncrasies in the behaviors of different characters (e.g., what they do and how they do it); plausible motivations for characters' behavior (e.g., hiding to signal a desire to play); and recognizable emotions associated with characters' behaviors and interactions (e.g., shyness). The individual qualities that agents bring to a production are powerful sources of texture and depth in their contributions to the joint performance. In contrast to the all-business mentality of traditional agents, effective improvisers bring believable characters to life [5].

Improvisational characters must *collaborate closely with one another and share control.* Most multi-agent paradigms assume that agents have qualified interests in working together and limited interactions during the actual performance of their collective activities [8]. In fact a good part of the work on multi-agent systems focuses deeply on issues of negotiation and planning of interactions to optimize individual and joint returns on that costly investment. By contrast, good improvisers do not negotiate or plan their interactions; like LC and SC, they simply cooperate, wholeheartedly, as best they can, at every opportunity [19].

Finally, improvisational characters must exhibit *improvisational expertise.* Most prior work on intelligent agents focuses on their competence at performing some work-related job and especially their success in achieving well-defined goals. By contrast, directed improvisation is explicitly process-oriented—the joint course of behavior enacted by the characters <u>is</u> their product. Other than meeting the constraints of users' directions, there is no "correct" or "incorrect" performance. There can be many alternative, equally "successful" performances of a given script—that is, performances that follow the directions in a manner that is novel, surprising, amusing, dramatic, or otherwise engaging to human observers. Simply following the directions is easy. The art lies in the improvisation. Thus, if we ask LC and SC to improvise again under the same abstract directions, we would be disappointed if they did not do something different from what they did the first time. The wonder of improvisation is that each performance is unique.

3. Agent Architecture

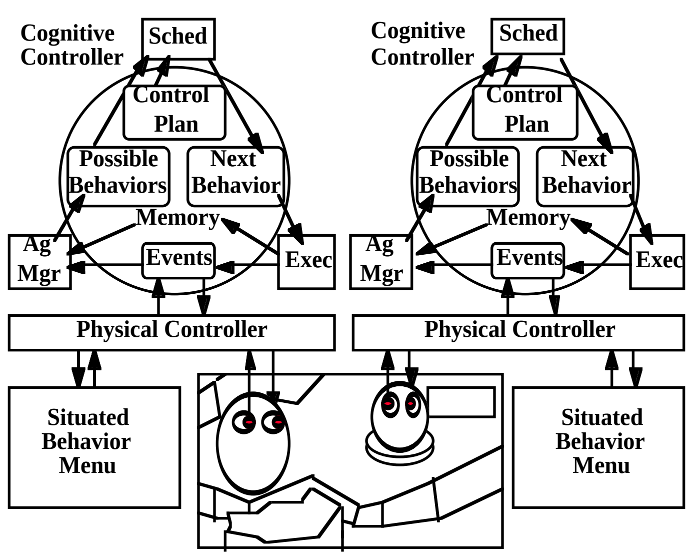

Figure 2. Framework for a Two-Character Production.

Each improvisational character is an intelligent agent, whose architecture (Figure 2) comprises coordinated, but asynchronous controllers for cognitive and physical behaviors [16, 17, 21]. Each controller implements a dynamic control model (Figure 3) [15, 18], allowing the agent to: (a) continually notice possible situated behaviors; (b) describe its intended behavior in abstract control plans; and (c) generate and modify its control plans at run time. At each point in time, the agent selects and performs situated behaviors that match its current control plans.

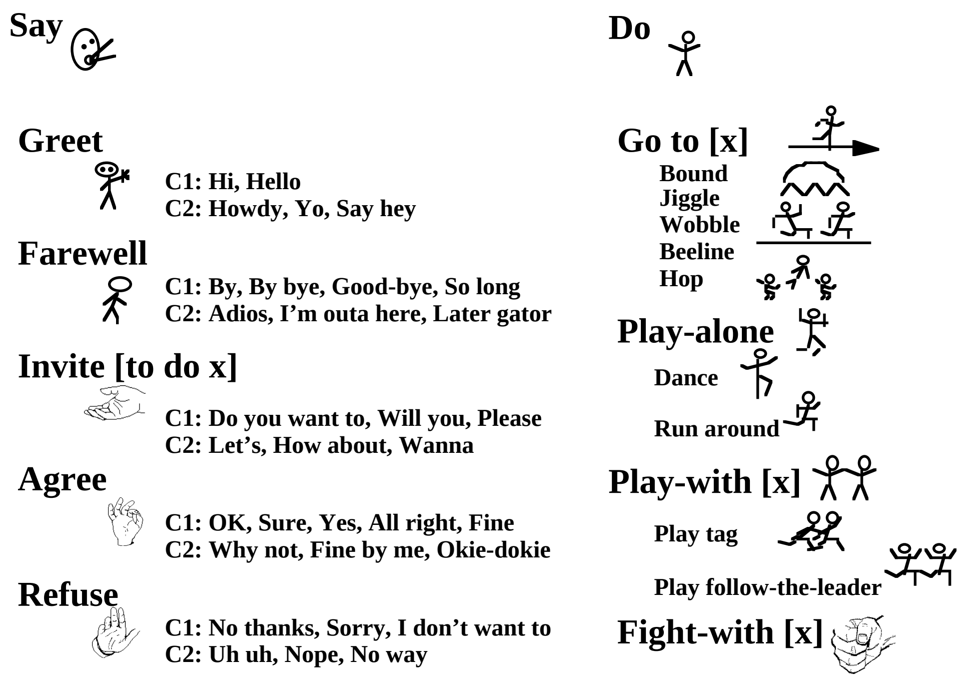

Figure 3. Excerpts from the behavioral repertoires of C1 and C2.

The situated behaviors an agent "notices" are instances from its repertoire of cognitive and physical classes (Figure 3), the "building blocks" for improvisation. Different instances of a behavior class express life-like variability and moods (e.g., a character may go to a destination by beeline or hop, depending on whether it feels determined or playful). Idiosyncratic instances (e.g., in repertoires or styles of behavior) express life-like individual differences. As mentioned above and illustrated in the examples, we have developed two simple characters by inserting new "minds" into animated "woggle" bodies. Unlike the CMU woggles, our characters have multiple gaits, speak lines, follow directions, and improvise. Each one has about 10 classes of physical behaviors and 20 classes of verbal behaviors, each with 1-5 instances. Their verbal behaviors and personalities were conceived and recorded by Aaron and Nora Hayes-Roth, who were 13 and 10 years old at the time.[2] Our architecture has a clean interface so that a given mind can be

---

[2]These differences produce qualitatively different interactive experiences. CMU's charming "Edge of Intention" system allows users to enter the world of three well-defined characters (Wolf, Bear, and Shrimp), in the form of a mouse-controlled user-woggle. Our Virtual Theater (discussed below) allows users to engage in creative self-expression as collaborative creators, directors, and performers of animated stories and plays.

embedded in different bodies in different media (e.g., virtual reality, text, animation). We currently are developing a new animation system that will permit children to create new bodies (and sets and props) by configuring components (e.g., body parts, facial features, hair, clothes) from a library.

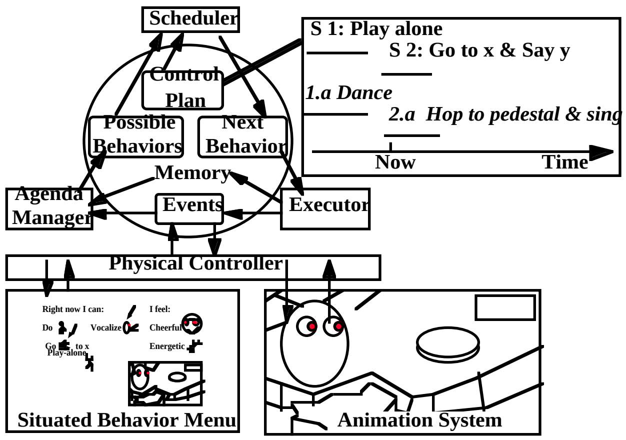

Figure 4. Control plans: user directions (1, 2) and improvisational plans (1a, 2a).

The performing arts literature [7, 10, 24, 31] suggests that the essential ingredients of improvisational expertise are: (a) an openness to situated possibilities (*Just let the words flow. Do not fear mistakes. Turn off the censor. Look for relationships. Do not plan too far ahead.*); (b) a discerning pursuit of promising possibilities (*Relate present actions to past actions. Keep the action on stage. Take it to the extreme*); and (c) an extreme and unqualified commitment to cooperate with partners (*Accept [don't block] offers. Give your partner what she/he needs. Share control.*) Our architecture is a natural framework for implementing these heuristics. The agenda manager generates alternative situated behaviors—the raw materials of improvisation. Dynamic control plans represent children's directions, along with characters' own improvisational plans.

The control plan is a key architectural feature. Because it can be abstract, the character may be able to choose among several logically acceptable behaviors. There can be multiple simultaneous control plans, some of which represent exogenously supplied directions and some of which represent the character's own improvisational decisions (Figure 4). Each component plan can have its own temporal, sequential, or hierarchical structure. Finally, since the control plan is a data structure, executed actions can modify it, thereby modifying the criteria used to schedule actions in the future. Thus, characters can incorporate and adapt their behavior to directions given asynchronously during a performance, as well as their own internally generated improvisational decisions. As illustrated in the two-character improvisation discussed above, we are extending the architecture to allow characters to use similar mechanisms to interpret and predict the plans and behavior of their improvisation partners. We are implementing the several different abstract forms of improvisational expertise in Table 1.

Table 1. Forms of Directed Improvisation.

Solo Improvisation	*Collaborative Improvisation*
One-Step Improvisation	
Choose among logically equivalent behaviors Direction: Go to pedestal Improvise: Hop to pedestal-3	Respond to a partner's behaviors Partner: Greeting Improvise: Return greeting
Sequential Improvisation	
Construct a coherent path to a dramatic moment Direction: Play alone, Rest Improvise: Play alone, Get tired, Rest	Recognize and coordinate with a partner's behavior sequence Partner: Going toward pedestal Improvise: Go towardpedestal, Meet at pedestal
Patterned Improvisation	
Instantiate an improvisational schema Direction: Dance: Improvise: Iterate(Hop, twirl)	Recognize and participate in a partner's schema Partner: Play hide and seek? Improvise: I count to 10, etc.

5. Testbed Application: A Virtual Theater

We are experimenting with improvisational characters in the context of a Virtual Theater for Children. Actually, this is a suite of applications, permitting several different interaction modes within the theater metaphor.

In *animated-puppets mode*, children direct characters interactively by making choices from *situated behavior menus*. For example, in Figure 5, two children interactively direct two animated puppets, LC and SC. LC feels cheerful and energetic. It considers going somewhere, playing alone, or inviting its partner to play. SC feels OK, but tired. It considers going to the rest area, playing alone, or speaking. One child directs LC to invite SC to play. It improvises accordingly (e.g., approach SC, greet it, say "Do you want to play follow the leader?") LC's behaviors change the shared situation and, as a result, the behaviors both characters subsequently consider. In particular, SC will now consider alternative replies to LC (all positive, if SC is a good improviser!).The children work side by side, directing their characters' moods and behavior to create a shared story, just as they would with physical puppets. But here the characters collaborate on the playcrafting, improvising within the constraints of the children's directions.

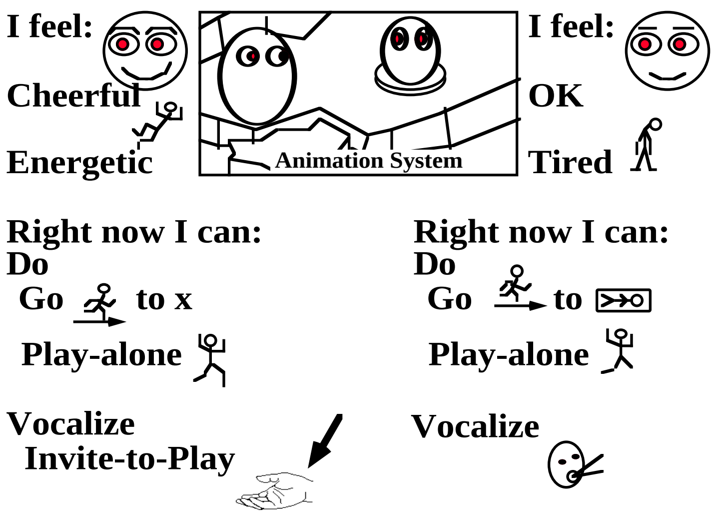

Figure 5. Snapshot from situated behavior menus for animated puppets.

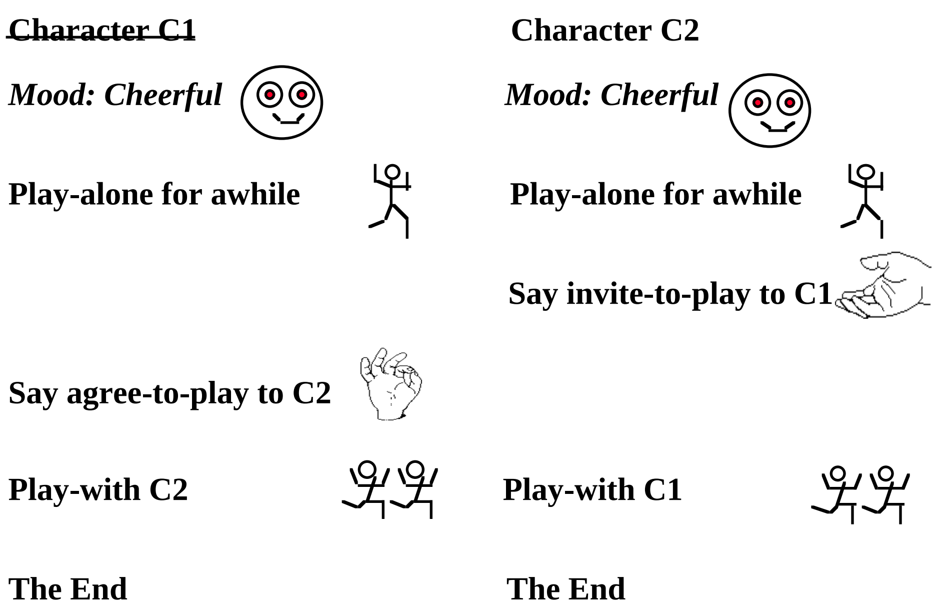

Figure 6. Abstract script for animated actors.

In *animated-actors mode*, children direct characters prospectively by giving them *abstract behavior scripts*. For example, in Figure 6, the characters begin by playing independently (e.g., dancing, hopping on the pedestal). C2 determines when "awhile" has passed and suggests a game (e.g., hide and seek, follow the leader). C1 agrees and the two characters play the game together. The characters also may improvise script-independent behaviors (e.g., exchange greetings in a chance encounter while playing alone).

In *improv troupe mode,* children prepare abstract scripts for animated actors, but use *fully instantiated behavior menus* (Figure 7) to direct all of the specific behavior of a subset of them as animated puppets. Thus the children function as the minds of actors in an animated improv troupe.

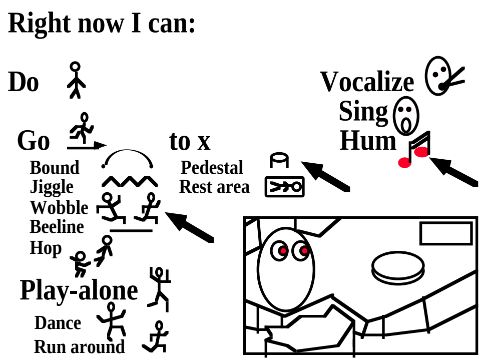

Figure 7. Snapshot from a fully instantiated situated behavior menu.

Character C1	Character C2
Turn away from C2 Start counting to 10	
	Beeline to pedestal Go behind pedestal
After counting to 10, wobble around set	
	When C1 approaches, jump out, say "Boo!"
Jump and say "Oh!"	

Time ↓

Figure 8. Abstracting a script for hide and seek by generalizing selected variables in the improvisation.

In *improv director mode*, children prepare abstract scripts for animated actors, but continue to direct all of them occasionally during the performance through abstract behavior menus, as illustrated in Figure 5 for animated puppets.

In *collaborative playcrafting mode*, children use animated puppets to improvise stories. When they see something they like, they abstract and store the corresponding script segment in the "repertory" for future use (Figure 8).

In *interactive script refinement mode*, children prepare abstract scripts for animated actors. When the actors improvise something appealing, the children create corresponding *fully instantiated behavior scripts* for future performances (Figure 9).

<u>Character C1</u>	<u>Character C2</u>
Start: Lower platform	Start: Upper platform
Wobble to near pedestal-1	Bound down stair
Hop onto pedestal-1	Bound down stair
Hop to pedestal-3 &	Bound down stair
say Yippee-1	Bound down stair
Hop to pedestal-5 &	Bound down stair
say Yippee-2	Wobble to near pedestal-4
Hop to pedestal-4 &	
say Yippee-3	
	Nod to C1
	Say "Do you want to
Nod to C2	play follow-the-leader?"
Say "Sure, let's go."	etc.
etc.	

Figure 9. A fully instantiated behavior script.

In *interactive story mode*, children direct the specific behavior of the protagonist in a story (Alice in "Alice in Wonderland"), while the other characters' improvisations are constrained by the story's abstract narrative plot and performance-time directions given by an internal "story master."

6. Concluding Remarks

Besides demonstrating the appeal of directed improvisation by computer characters, our Virtual Theater has other useful properties. It provides an enjoyable *experience-based learning environment* [27] for writing, computer programming, artistic skills, and social skills. It has social and commercial potential as a *new kind of computer game* that will appeal to a large population of children (primarily girls) who do not enjoy the kinds of games that currently dominate the market and, as a consequence, miss an early entry point to computing technology. It has artistic and commercial potential as a *new artistic "form"* for interactive story experiences. Finally, it invites collaborative and distributed activities among children in different locations (Figure 10).

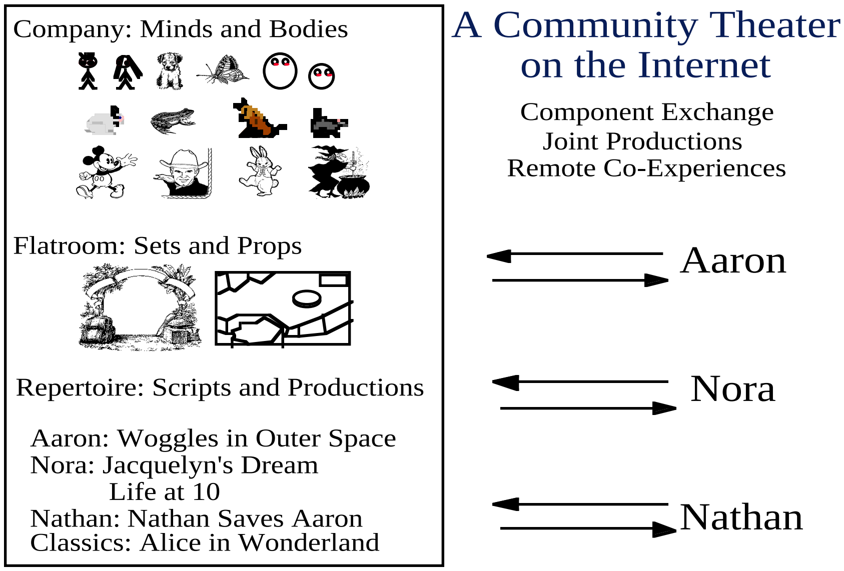

Figure 10. A collaborative and distributed community theater.

There are many other sorts of applications for our Virtual Theater metaphor, for example: (a) Interactive Historical Reenactment Exhibits for museums or theme parks would allow people to enter the everyday world of important times and places; and (b) Virtual On-the-Job Training Environments would allow people to practice and learn how to interact effectively with diverse customers, clients, staff, or co-workers in various work-related situations.

Directed improvisation by computer characters also has promise as a general paradigm for human-computer interaction. First, it provides a framework in which intelligent computer characters can cooperate with users and one another to achieve goals. Minimally, it offers an efficient communication mode, in which users need not explicitly instruct characters in all of the details incidental to performing a task. Ideally, it offers characters who bring expertise to the task at hand and enhance achievement of the objectives. Second, it explicitly allows flexibility in the manner in which goals can be achieved. Many important jobs carry intrinsic uncertainty in critical run-time conditions, such as resource availability, operating constraints, and performance requirements [16]. Directed improvisation allows users to give abstract directions, within which computer characters can improvise behavior that is compatible with run-time conditions. Third, it provides a natural and familiar style of interaction that mimics the improvisational quality of most human behavior and interaction. In contrast to the stereotypic "feel" of most human-computer interaction, directed improvisation has the variable, idiosyncratic, slightly unpredictable, "give-and-take" of human interaction. Finally, directed improvisation introduces an amusing, engaging, delightful quality to interactive experience in the form of computer characters who combine task-relevant obedience with task-compatible improvisation.

Acknowledgments. The original development of the proposed agent architecture was supported by ARPA.

References

1. Agre, P.E., and Chapman, D. What are plans for? Robotics and Autonomous Systems, 6, 17-34, 1990.

2. Applebee, A.N. The Child's Concept of Story. Chicago: University of Chicago Press, 1978.

3. Baker-Sennett, J. Matusov, E., and Rogoff, B. Sociocultural processes of creative planning in children's playcrafting. In P. Light and G. Butterworth (eds.), Context and Cognition: Ways of Learning and Knowing. NY: Harvester Wheatsheaf, 1992.

4. Bates, J. The role of emotion in believable agents. Communications of the ACM, 1994, 37, 122-125.

5. Bates, J., Hayes-Roth, B., Laurel, B., and Nilsson, N. Working notes of the AAAI Symposium on believable agents, Menlo Park, CA: AAAI, 1994.

6. Bateson, M.C. Composing a Life. NY: The Atlantic Monthly Press, 1989.

7. Belt, L., and Stockley, R. Improvisation through Theater Sports. Seattle: Thespis Productions, 1991.

8. Charib-draa, B., Mandiau, R., and Millot, P. Distributed artificial intelligence: An annotated bibliography. SigArt Bulletin, 3,1992, 20-37.

9. Cypher, A. Watch what I do: Programming by demonstration. Cambriedge, MA: MIT Press, 1993.

10. Dean, R. Creative Improvisation: Jazz, Contemporary Music and Beyond. Philadelphia: Open University Press, 1989.

11. Etzioni, O., and Weld, D. A softbot-based interface to the internet. Comm. of the ACM, 1994, 37, 72-79.

12. Fenton, J., and Beck, K. Playground: An object-oriented simulation system with agent rules for children of all ages. In Proc. of OOPSLA, NY: ACM, 1989.

13. Flower, L., and Hayes, J.R. The dynamics of composing: Making plans and juggling constraints. In L.W. Gregg and E.R. Steinberg (eds), Cognitive Processes in Writing. Hillsdale, NJ: Erlbaum, 1980.

14. Gans, A., Confrey, T., Harvey, J., and Hayes-Roth, B. How WATCH works. Stanford University, Knowledge Systems Laboratory, Report KSL 90-03, 1990.

15. Hayes-Roth, B. A blackboard architecture for control. Artificial Intelligence, 1985, 26, 251-321.

16. Hayes-Roth, B. An architecture for adaptive intelligent systems. Artificial Intelligence, in press, 1995.

17. Hayes-Roth, B. Architectural foundations for real-time performance in intelligent agents. Real-Time Systems, 2, 99-125, 1990.

18. Hayes-Roth, B. Opportunistic control of action. IEEE Trans. on Systems, Man, and Cybernetics, 12, 1575-1587 1993.

19. Hayes-Roth, B., and Brownston, L. Multiagent collaboration in directed improvisation. Stanford University Report KSL-94-69, 1994.

20. Hayes-Roth, B., and Hayes-Roth, F. A cognitive model of planning. Cognitive Science, 1979, 3, 275-310.

21. Hayes-Roth, B., Pfleger, K., Lalanda, P., Morignot, P., and Balabanovic, M. A domain-specific software architecture for adaptive intelligent systems. IEEE Transactions on Software Engineering, in press, 1994.

22. Hayes-Roth, B., Sincoff, E., Brownston, L., Huard, R., and Lent, B. Directed improvisation. Stanford University Report KSL-94-61, 1994.

23. Huffman, S. Instructable autonomous agents. U. of Mich., PhD Thesis, 1994.

24. Johnstone, K. IMPRO: Improvisation in the Theatre. NY: Penguin Books, 1987.

25. Kautz, H.A., Selman, B., and Coen, M. Bottom-up design of software agents. Comm. of the ACM, 1994, 37, 143-147.

26. Maes, P. Agents that reduce work and information overload. Comm. of the ACM, 1994, 37, 30-40.

27. Malone, T.W., and Lepper, M.R. Making learning fun: A taxonomy of intrinsic motivations for learning. In R.E. Snow and M.J. Farr (eds), Aptitude, learning and instruction. Hilssdale, N.J.: Erlbaum, 1987.

28. Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. Experience with a learning personal assistant. Comm. of ACM, 1994, 37, 80-91.

29. Moore, J. Presentation at the AAAI National Conference, Seattle, 1994.

30. Newell, A. Unified Theories of Cognition. Cambridge, MA: Harvard University Press, 1990.

31. Nachmanovitch, S. Free Play: Improvisation in life and art. Los Angeles: Jeremy P. Tarcher, Inc. 1990.

32. Papert, S. Mindstorms: Children, computers, and powerful ideas. New York: Basic Books. 1980.

33. Rogoff, B., Gauvain, M., and Gardner, W. The development of children's skills in adjusting plans to circumstances. In S. Friedman, E. Scholnick, and R. Cocking (eds.), Blueprints for Thinking: The role of planning in psychological development. New York: Cambridge University Press, 1987, 303-20.

34. Smith, D.C., Cypher, A., and Spohrer, J. KidSim: Programming agents without a programming language. Comm. of the ACM, 1994, 37, 54-67.

35. Stefik, M. Planning and meta-planning (MOLGEN: part 2) Artificial Intelligence, 16, 1981, 141-169.

36. Suchman, L. A. Plans and Situated Actions: The Problem of Machine/Human Communication. Cambridge, England: Cambridge University Press, 1987.

37. Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L., and Moore, M. Instructions, intentions, and expectations. U. of Pennsylvania: Report MS-CIS-93-61, 1993.

38. Winograd, T. Beyond programming languages. In D. Partridge (ed.), Artificial Intelligence and Software Engineering. Norwood, N.J.: Ablex, 1991.

39. Winokur, J. Writers on Writing. Philadelphia: Running Press, 1990.

40. Vere, S., and Bickmore, T. A basic agent. Computational Intelligence, 6, 1990, 41-60.