Tigrito : A Multi-Mode Interactive Improvisational Agent

Heidy Maldonado, Antoine Picard, Patrick Doyle and Barbara Hayes-Roth

Knowledge Systems Laboratory
Stanford University
Gates Information Sciences Building
Stanford, CA 94305-9020, USA
Telephone: (650) 723-0948
E-mail : {heidym/antoine/pdoyle}@cs.stanford.edu, bhr@hpp.stanford.edu

ABSTRACT
This paper presents the implementation of Tigrito, an affective computer character. We outline how Tigrito can be used to study chidren's sense of engagement and relationship with virtual toys in different modes of interaction.

KEYWORDS
Interactive, avatar, believability, autonomous, agent.

INTRODUCTION
•Seven-year-old Katie laughs as the stuffed tiger on the screen sings. Then she uses the cursor to push the "Applaud" button.

•Six year-old-Jimmy watches open-mouthed as the on-screen tigers playfully pounce on one another.

The program that so amuses Katie and Jimmy is not a typical computer game. The children are not trying to score the most points, finish first, or achieve any competitive goals to "win" a game, though they are playing with the computer. Both Katie and Jimmy have established a rapport with the on-screen characters, an affective relationship that matures over time and can both suffer and benefit from the child's actions. This affective link makes the experience emotionally rewarding, and extends beyond mere anthropomorphization; children not only believe that the characters exist, but deeply care about their virtual friends. In this paper we describe an architecture designed to compare the sense of engagement and suspension of disbelief across different modes of interaction, as well as the affective relationship that emerges from the interaction between the child and a virtual toy.

We have constructed an agent engine, affectionately called Tigrito, that can act either autonomously or under user direction. In our studies, there are three modes of user interaction. In the first, the child interacts directly with Tigrito, who behaves as an autonomous "pet." In the second, the child plays with Tigrito through an avatar, a second tiger; lastly, there is a "movie mode" in which the child gives high-level directions to the two autonomous tigers.

IUI 98 San Francisco CA USA

Both the autonomous and user-controlled characters have transient moods that are influenced by the child's actions, and each character has a unique personality, which it expresses through the actions it performs. Rather than digitally rendering our characters, we instead use animated video clips of a stuffed white tiger cub to represent Tigrito in action. This encourages a child's suspension of disbelief and promotes the illusion of reality by presenting a faithful rendering of a real, familiar object. Through this approach, children will feel that an everyday toy has come to life before their own eyes. As Chuck Jones says, "Believability. That is what we're striving for." [5]

OVERVIEW
Past research has explored many different forms of interaction with autonomous characters, without giving the user the power to choose between them. For example, the Oz Project [1, 2] featured highly detailed models of emotion used to create a framework within which the user interacted with the autonomous characters through an avatar. Unfortunately, the interface they chose could not convey the complexity of the emotional model. The characters in the Virtual Theater's Master/Servant [4], on the other hand, improvised within a broadly-defined storyline, but lacked interactivity once the plot was set. The ALIVE Project's [6] approach let the user interact directly with agents within immersive environments but required a complex and cumbersome system to recognize the user's actions. We wanted to build an agent capable of functioning in each of the aforementioned styles of interactivity, to compare the user's sense of engagement. The framework for that interaction also had to be kept simple enough for children to understand the characters' mood changes.

In each mode of interaction, Tigrito is represented as an animated character whose moods are displayed on sliders. A second set of sliders represents the mood of the character Tigrito is interacting with, either the child, the child's avatar, or an autonomous character. Both the child's moods and Tigrito's are influenced by the other's actions. When the child is participating, s/he is given a dynamic list of actions to choose from.

Our character possesses a distinct personality which he expresses through actions based on his mood and position. The characters' moods emulate those used in the Virtual Theater's animated puppets : "[The] moods vary along three continuous dimensions: an emotional one ranging from happy to sad, a physiological dimension ranging from peppy to tired and a social dimension ranging from friendly to shy."[4] In our implementation these independent moods are called happiness, friendliness and energy.

The three styles of interaction implemented are illustrated in Figures 2, 3 and 4. In each mode the autonomous animated tiger and his moods are visible, so that current situation is available at a glance. In the avatar mode (Figure 2), the child directs Hobbes (the rightmost tiger) at a high level by moving his mood sliders and at a lower level by choosing the actions he should perform, using the buttons in the lower right corner of the screen. The child can only change Tigrito's mood by the effect that Hobbes' actions have on him. The list of available actions is updated the moment that Hobbes' mood changes. Figure 2 also illustrates the optional text commentary, which shows the text-version of the interaction.

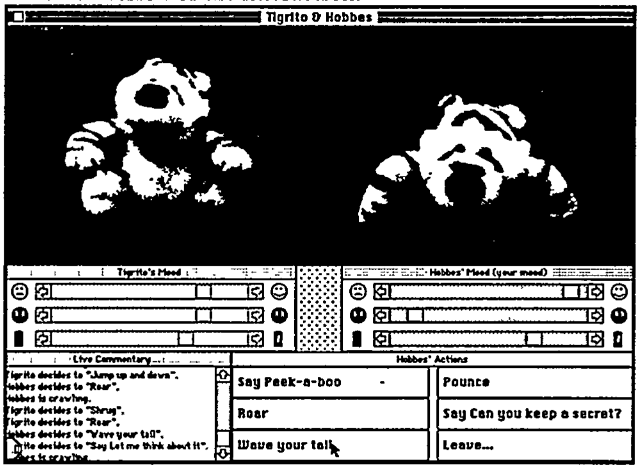

Fig. 2: Avatar Mode

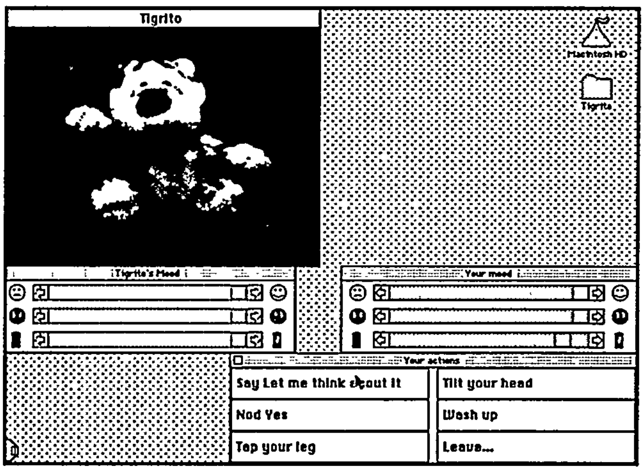

Fig. 3: First-person Mode

In the first-person mode (Figure 3), the child interacts directly with Tigrito, using sliders to set his/her moods and choosing from the list of actions. Although Tigrito's moods are displayed on the screen, the child can only change them by performing actions that Tigrito can perceive.

We refer to the third mode as "movie mode" (Figure 4). In this case, the child's directions are high-level in nature, since s/he can only control the tigers' moods by moving the mood sliders. The actions performed by the tigers are left entirely up to the agents for each character.

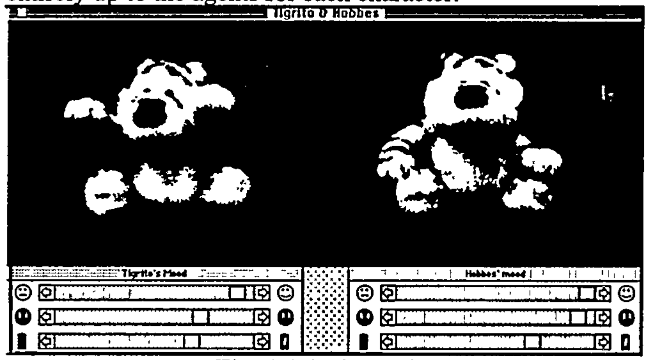
Fig. 4: Movie Mode

AGENT IMPLEMENTATION

The implementation of Tigrito is based around a decision loop in which the agent uses mood values to select actions. The significance of the actions are read at run-time from a modular character file. For example, when Hobbes performs the action "Roar," it sees from its character file that roaring causes the other character's friendliness and happiness to decrease. As a result, while Hobbes is roaring, Tigrito's moods decrease accordingly. The on-screen representation of the actions is a set of Quicktime movies filmed with a stuffed animal. In order to provide visual continuity to the movies, a set of base physical positions are also defined in the character file and each action-movie starts and ends in one of the predefined positions.

To generate a realistic interaction, our agent needs to decide what action or transition to perform in order to represent its mood accurately. As a first step of the decision loop, the agent evaluates how much the actions of the other character have affected its mood since the last iteration. If the current mood is too far from the ideal mood of the current position (as defined in the character file), it chooses to perform a transition to a different position; otherwise an action will be chosen.

In the case where an expressive action is to be performed, we use the least square method to evaluate the appropriateness of each action. However, instead of searching for the single most appropriate action, the best ones are recorded along with a measure of their appropriateness. The number of stored actions depends on the interface restrictions introduced by the multiple modes of interaction. Once all actions that can be performed in the current position have been reviewed, one of them is chosen by an algorithm that favors actions whose mood values are closest to the current mood. After an action is selected, the movie is played and the mood changes associated with that action are effected on the other character.

Transitions between positions follow a slightly different algorithm. To decide what the most appropriate position is, the agent again uses a least square method to compare the current mood to the ideal moods of all known positions. Once the best position is chosen, it searches through all defined transitions to find the shortest sequence that will lead it to its desired position. The first transition in this sequence is selected as the next movie. The rest of the sequence is not preserved since the character's mood may have changed again, while the chosen transition was being performed. Therefore, a whole new sequence is computed, if necessary, during the next iteration of the decision loop. This strategy, though inefficient, prevents the character from following potentially lengthy sequences of transitions that may be unnecessary. Even with a large character file with a large number of positions, the search space does not get large very rapidly because of the following restrictions: if many of transitions are defined between these many positions, the path from starting to finishing position should be very short and breadth-first search will rapidly find this short solution; if few transitions are defined, the appropriate path may be long and take us through numerous other positions first but the number of transitions would also cause the branching factor to be very low, allowing us to keep the search space in check.

All the detailed information about the significance of the actor's behavior is read at run-time from a character file, allowing for easy modification and addition of actions to the actor's repertoire. The character file stores the following information to manage the behavior of the character:

- A list of all the positions in which the character can find itself. Each position is given a unique name that is be used in the text display and a specific numerical representation of the moods for which that position is most appropriate.
- A list of the "transition" movies that allow the tiger to go from one position to another so that it knows, for example, how to go from sad to happy, angry to sad, etc.
- A list of the expressive actions that the tiger can perform (such "Wave Hi," "Shout," "Sing"). For each of these, the character file also specifies the position that the action is performed in, the appropriate mood for the character performing the action and the effect that the action should have on the other character's mood.

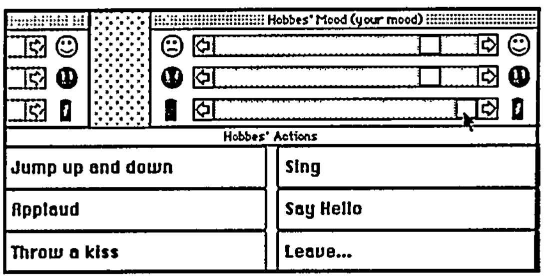

Fig. 5: High Energy

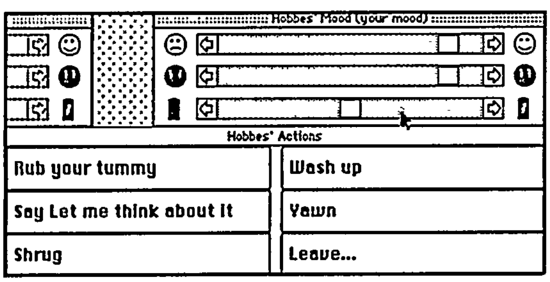

Fig. 6: Medium Energy

SAMPLE BEHAVIOR

In figures 5 and 6, even though the character's position is "Standing Happily" in both situations, the change of a mood value (in this case energy) causes the set of available actions to vary. Whereas in Figure 5, high energy actions like "Jump up and down" and "Sing" were deemed appropriate, in Figure 6, the re-evaluation, they are substituted in the appropriate action list by "Yawn," "Shrug" and other low energy actions.

In contrast, Fig. 7 shows how a more drastic change of mood (once again energy) causes the tiger to realize that his position no longer aptly portrays his mood, and he changes position, going from happy (Figure 6) to pensive (Figure 7) and now offers a selection of the actions available from the pensive position rather than the happy position.

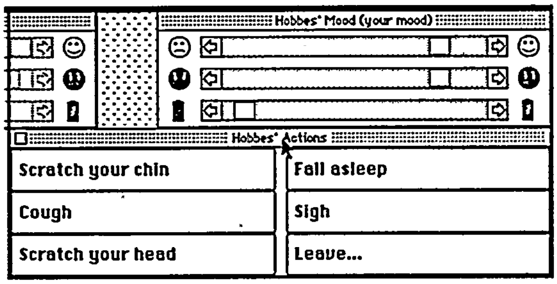

Fig. 7: Low Energy

REALISTIC BEHAVIOR

Our design incorporates several mechanisms to maintain the child's sense of engagement with repetitive behavior, mood invariability, and unresponsiveness to his/her actions.

While repetitive behavior may be considered a realistic personality trait, constant repetition is not a very engaging form of interaction. If a character's mood is not being modified dramatically by the other character's actions, it would be likely to favor an action that has already been favored in the previous iteration of the decision loop. As a result, there is a high probability that this behavior would be repeated. To avoid this pitfall, the agent keeps track of the number of times a certain action has been performed within the last few turns and modifies the action weights to disfavor repeated actions. As a result, the actor is still more likely to perform an action close to its mood than one more distant, but the most recently performed action has a lesser chance of being chosen. Currently, the weights for repetition is a constant but a future improvement will be to allow different actions to have their weighing affected differently by repetition. This would cause behaviors that can logically be repeated, such as an angry character kicking another, to have a greater tendency to repeat than others, such as greeting.

Another problem with mood-reinforcing behaviors is that they can lead to deadlock or extreme reinforcement. For example, both actors could be so happy, friendly and energetic that their actions might only have positive effects on one another, causing their moods to remain unchanged at their maximum positive values. To avoid such situations, the system infuses instability in the moods. Our original solution to this problem was to have each mood decay quadratically. This algorithm had the desired effect on the energy mood value, since the character would become tired after performing several high energy actions, and would gain energy after performing low energy actions (resting). However, this pattern of change was not realistic for happiness and friendliness. A different approach to the variation of these moods is to have them jump to a random value when they have not changed for a long time. However, this tactic is also unsatisfactory, since it causes the character to have dramatic mood swings that give it a rather schizophrenic appearance that, though interesting, is undesirable in our actors. Instead, our implementation forces happiness and friendliness to change by small random amounts in random directions, while happiness decays quadratically. This ensures that the actors does not get locked into certain moods while keeping the variations realistic within the framework of their interaction.

Another important factor in preserving the child's sense of engagement is avoiding a feeling of disconnection between the two characters. For example, an autonomous character who is playing alone cannot continue behaving in the same manner when the other actor responds to every action. To account for this, our system introduces a delay between actions that gradually increases if the other character remains inactive for long periods of time. During periods of low interaction this feature slows Tigrito to a more realistic pace while preserving his ability to perform occasional actions, encouraging the child's participation.

FUTURE DIRECTIONS

Our design in this project aimed to produce a high-level abstraction of the agent where the features of each character were specified in the character file for modularity. Although the representation of actions by their effect on the mood of the other character seemed sufficient to in the two character case, the need for a more precise understanding of the significance of the actions of the other character is evident. To preserve our abstraction, the more precise specification of the actions will have to come in the form of a generalized encoding, that can be used in each character file independently of the others. This would ensure that each character is a distinct entity, which can interact with any other character in this framework, without making assumptions about the behaviors known by others.

An alternative approach to the above problem would be to have each character attach a particular meaning not only to its own actions but also to the actions of others. This makes sense within the framework of human interactions where, while two individuals may recognize the same action, their interpretations may vary. For example, some Eastern European cultures use a vertical nod to say no, and shake their head side to side to indicate yes, contrary to the Western significance of these motions. Allowing characters to put their own interpretation on each action will give us a chance to create such widely differing personalities and study their interactions.

CONCLUSION

By nature, people react emotionally in all their interactions, including interactions with machines. Research has shown that human beings become emotionally involved with computers whether they are acting as machines [6, 7] or as simulations of real beings [3, 8]. In systematic user testing, we intend to determine whether those tendencies are equally strong across all our modes of interaction. Even with the informal testing done to date, we are pleased to find that our interactive characters kept their behaviors coherent, believable and engaging.

The research community has suggested several uses for interactive creatures and characters, both autonomous and user-controlled, by embedding them in traditional computer application genres, such as education and entertainment. Although interactive improvisational characters can make great adaptive opponents, interesting supplementary characters, companions through interactive stories, guides in virtual environments, or pets and tutors, among others, we believe that they can provide an emotional, and entertaining experience on their own, and hope that our system inspires other designers of virtual toys for children.

ACKNOWLEDGMENTS

Our work has been supported by ARPA Contract N66001-95-D-8642 Subcontract #137-1 through Teknowledge, Inc., a gift from Intel, a Seed Grant from Stanford's Center for the Study of Language and Information and Office of Technology Licensing, and Fundayacucho's Galileo Program. We thank Angel Puerta for his help in developing this paper, and Stanford University's Meyer Media Center for the production of the accompanying video.

REFERENCES

1. Bates, J. The role of emotion in believable agents. *Communications of the ACM*, 1994.
2. Bates, J., Loyall, A.B., Reilly, W.S. An architecture for Action, Emotion and Social Behavior. *Proceedings of the Fourth European Workshop on Modeling Autonomous Agents in Multi-Agent World*, 1992.
3. Foner, L. Entertaining agents: a sociological case study. In *Proceedings of the First International Conference on Autonomous* Agents, ACM Press, 1997.
4. Hayes-Roth, B., van Gent, R. Improvisational Puppets, Actors and Avatars. *Proceedings of the Computer Game Developers' Conference*, 1996.
5. Jones, Chuck. Chuck Amuck: The life and times of an animated cartoonist. Avon Books, New York, November 1990. 304 pages.
6. Maes, P., Darrell, T., Blumberg, B., Pentland, S., Foner, L. Interacting with Animated Autonomous Agents. *Communications of the ACM*, Vol. 37, No. 7, July 1994.
7. Nass, C., Moon, Y., Fogg, B., Reeves, B. and Dryer, D. Can computer personalities be human personalities? *International Journal of Human-Computer Studies,* 1995.
8. Nass, C., Steven, J. and Tauber, E. Computer are social actors. *In Proceedings of the CHI Conference.* 1994.
9. Weizenbaum, J. Eliza - A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, Vol. 9, No. 1, January1966.