Asynchronous Hierarchical Architecture for Controlling and Animating Interface Agents

Kenji Mase, *Sidney Fels, †Armin Bruderlin,
Yasuyuki Sumi and Rieko Kadobayashi
ATR Media Integration & Communications Research Laboratories,
Kyoto, Japan
e-mail: mase@mic.atr.co.jp

Abstract

An interface agent with a life-like character on a personal and mobile computer based guidance system is a plausible interface design approach for supporting and mediating communication between the exhibitors and visitors of showcase places such as museums, laboratory open houses and trade shows. This paper discusses the design framework and applications of such interface agents based on an asynchronous hierarchical agent(AHA) architecture and empirical knowledge of character animation design. First, a locomotion design tool is introduced with multi-level abstraction of interface design, followed by the AHA architecture which is suitable for designing agents allowing user's control access at these different levels of abstraction. The architecture is tested with a board game called Malefitz in which the user can control player agents at various levels of abstraction. Lastly, a mobile guidance system with a cartoon-based life-like interface agent was developed and tested. Personalized agents can provide an intimate relationship with the users through context-aware behavior.

1 Introduction

We speculate that, in the future, an interface agent will accompany a user throughout the day, performing a role that a human butler or secretary would perform at home or the office. Such a personal interface agent should have a tangible personality as the foundation of its character but also have the flexibility to play different roles depending on the context or the requests for the sake of its agentee, i.e., the user of the agent. Furthermore, as the agent adapts to the agentee (and vice versa) we foresee that the relationship between them will become more intimate over time. An interface agent is defined in this paper as an autonomous agent which mediates between a human user and computer cyberspace. "Agent" refers to an autonomous computer program which performs some task or tasks on behalf of an entity, communicates asynchronously with other entities in the world, and responds asynchronously to events in the world. Entities may be people, other agents, other traditional programs or objects in the world.

We are designing a multi-agent architecture which is suitable to develop such interface agents for various application systems. The architecture is also unified to control animated agents characters[1]. Without any practical tasks, an interface agent would remain a toy to play with or a friend to chat to. We are also pursuing two applications; a virtualized museum[2] and a context-aware mobile assistant system named C-MAP[3]. Agents will appear in both systems and play active roles. These works are depending on and related to each other under the umbrella research project on "*computer mediated creative communications*". However, the topics are weakly coupled and so implemented at this moment. A multi-agent architecture is suitable for this type of project management because the program and system modules can be integrated easily.

Our efforts to develop interface agents are in line with related work in various fields, such as, character animation in computer graphics[4], anthropomorphic social agent[5][6], autonomous interactive characters[7][8], and software agents[9]. Our objective is to provide a personal interface agent which has the capability to adapt its character to its agentee. Thus, a functional interface can be provided. The adaptation may be manual in the initial stage and then autonomous later as the research progresses. The visual outfit is necessary to bring about the intimate relationships between the agent and agentee.

In the following sections, we first discuss agentality as a functional labeling of agents followed by discussions on the design framework and applications of such interface agents. In Section 3 we present the locomotive animation mechanism of visual agents, which uses a multi-level abstraction of interface design. In Section 4 we expand on how the asynchronous hierarchical agent architecture works with an example implementation of a Malefitz board game in which the user can control the

---

*Currently with University of British Columbia, BC, Canada

†Currently with Sony Pictures Imageworks, CA, U.S.A.

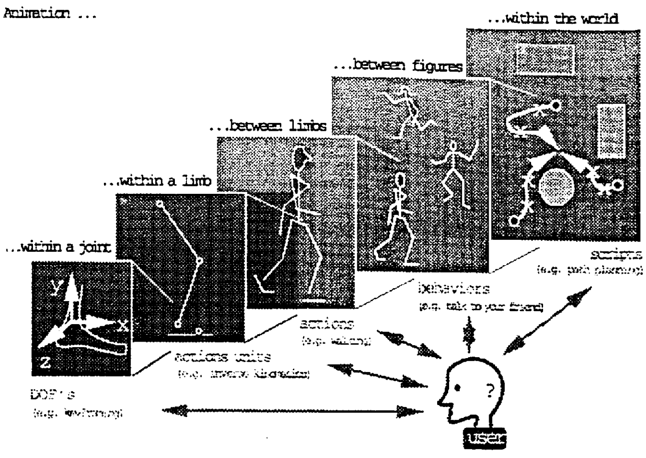


Figure 1: Hierarchical Animation Control Interface

agents at various levels of abstraction. In Section 5, application to museum exhibitions is discussed, in particular a virtualized museum which incorporates the C-MAP personal and mobile interface agent. This system is a cartoon-based life-like character that seamlessly guides the visitor of a museum through both physical and virtual exhibits.

2 Agentality: Function of an Agent

The need for multiple abstraction layers and multiple interaction models for agentees and agents is related to *agentality*. As we refer to a person differently by personal pronouns, such as *I*, *you* and *they*, in a segment of discourse, we can refer to an interface agent from different perspectives depending upon their relationship to us (and our task) during discourse. Three categories of agents we consider are avatars, assistants (or secretary agents) and computer actors. Unlike the personal pronouns, this categorization of an agent, which we call *agentality*, identifies an agent based upon its function. However, as the functional difference of agents makes the relationship between agents and agentees different, the agentality relationship closely corresponds to the analogy of personal pronouns. For example, we may say "Do you see the avatar standing there? It's *me*.", "Secretary Agent, will *you* check my e-mail?", "*He* is my personal actor agent." and so on.

The largest difference among the types of agents is the level of control accessible by user. The hierarchical structure of agent architecture fits to with functional labeling. As illustrated in the Section 4.2, different agentalities are important for agents playing in the relatively simple Malefitz game world. The agent users sometimes want to control the finest level of detail of the agent's behaviors while at other times only coarse, motivational control is needed.

This is particularly important when an agent is designed as a personal *interface agent* for guidance such as an agent for museum guidance. The user can make an agent be an avatar, assistant or actor by simply changing the level of control he/she exercises. In a virtual museum application, a user may want the agent to play three kinds of roles during the visit. First, an avatar in the virtual world is helpful to extend his/her reality and grasp the situation in the virtual world. Second, a virtual assistant (guide) agent may answer questions about the visit and bibliographical inquiries. Third, a virtual actor can entertain the visitor by showing a play or pro-actively conveying information. An avatar may become an actor when such a show is open for participation by virtual visitors. In this way, the agent provides the user with a consistent, interactive change of agentality.

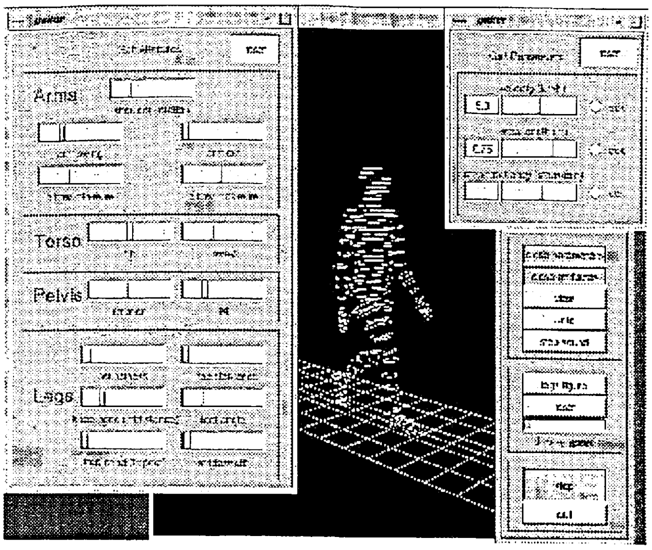


Figure 2: Gaïtor: animation design tool for locomotion

3 Animation Control of Visual Agents

Realistic and/or believable motion is important to give an agent an intelligent appearance. Unfortunately, it is very tedious and requires great skill to produce realistic and believable motion by traditional key-framing. As the actions of an agent become more complex, it is not practical to design all agent actions by the labor of a human animator nor by motion capturing. An algorithm-driven animation process is strongly desired.

By incorporating knowledge of how real humans move into control algorithms, certain motions can be animated more convincingly, autonomously and efficiently than is typical using standard control algorithms. Interestingly, the animation control interface can be layered hierarchically (figure 1). The user, i.e., animation designer, wants to have control in every level of layers during the production. For example, at first he/she has a basic script of a story or rough ideas of behaviors for each character. Then he/she wants to modify the actions and action units in detail to add reality and believability. Ideally, these modifications should not affect the scripts and behaviors. However, a change of velocity of some angle can easily break the whole balance of the figure or, for example, break down the synchronous walking of multi-

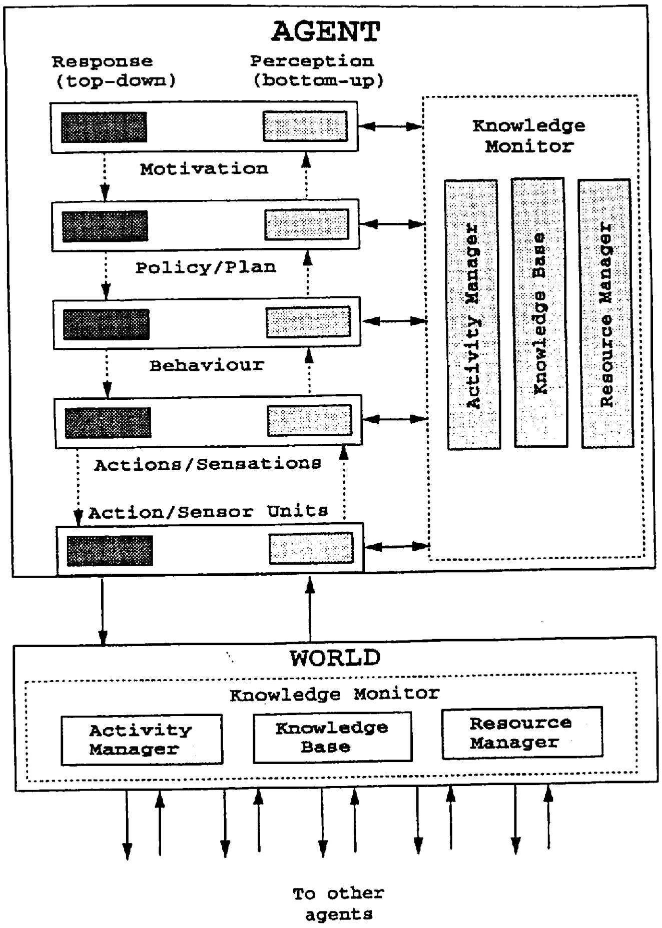


Figure 3: AHA: Asynchronous-Hierarchical Agent Architecture

ple figures. We need the system that allows users fine tuning on one layer while maintaining settings of other layers and adjusting boundary conditions.

We have developed an interactive, real-time animation system of human locomotion (walking and running) which has two-level access control corresponding to a behavior layer and an action layer[4]. Figure 2 is an interface display of the system. In the locomotive agent, walking speed, step length, and other characteristics are the behavior layer parameters, and other attribute parameters, such as, torso bend and pelvic rotations, are the action layer parameters.

With this strategy, it is easy for an autonomous agent to control behavior of a locomotive agent by incorporating the empirical control mechanism of the interactive system. The agents need only use parameters for behaviors and actions in order to obtain the desired behaviors. These parameters may be defined in advance or interactively at the time of execution.

4 AHA Architecture

Two fundamental components are in the Asynchronous Hierarchical Agent (AHA) architecture as shown in Figure 3. First, the multilevel hierarchical structure of the system is comprised of motivation/needs, policy/planning, behaviors, action/sensation and action/sensor units. Each layer contains responsive and perceptual functions, which interpret input signals from the adjacent layer. Our approach resembles the multilayered approach found in the InteRRaP system [10]. Our system has a structure called the knowledge monitor that contains a knowledge base structured so that all control layers have access to the information it contains; providing access to knowledge that applies to all levels of control abstraction. Knowledge specific to a particular layer though is contained within that layer. The multi-layered computation approach is also seen in [11] [12] [10].

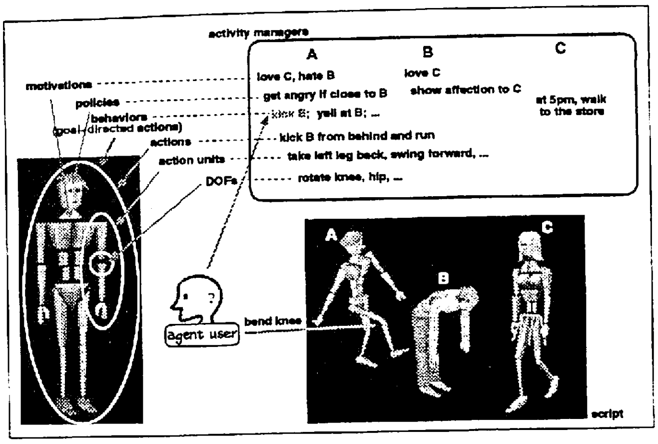


Figure 4: Multi-level Interaction via Scripts and Direct Manipulation

Our agent architecture allows the user to control the agent at different levels of abstraction. The various levels of abstraction correspond to different agentalities, allowing uniform design of avatars, assistants and actors. Agents interact with each other through the world, where the shared knowledge monitor manages common activities and a common knowledge base. The knowledge monitor of the world acts like the blackboard of our multi-agent systems[13].

An example of designing each agentality is as follows. The avatar requires frequent control at the action layer by the user. It does not have an explicit objective or sense of intelligence but just acts like a marionette. The agent, however, adapts the action unit so that dynamic actions look natural. The assistant agent expects explicit inputs of goals and plans from the user. It watches the behaviors of users as action examples or uses sensed situations about the user and the environment. The basic behaviors and layers beneath are pre-coded. The actor agent will follow the pre-coded script most of the time looking at the human user and other agents and responding properly within the range of freedom allowed within the script.

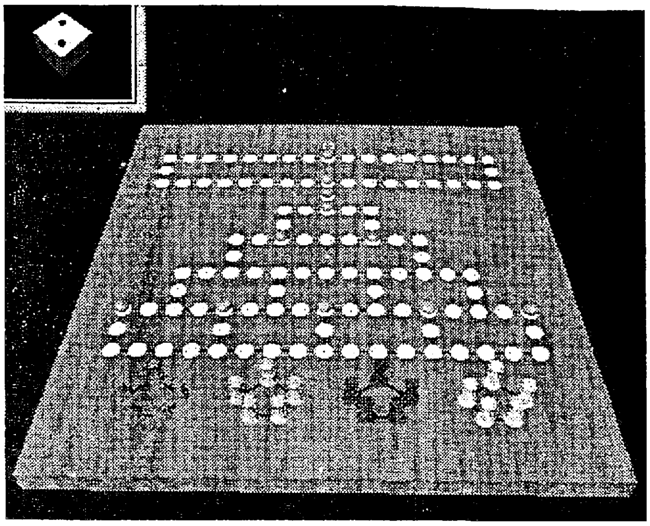


Figure 5: Malefitz Board Game Layout

4.1 Example Agent Interaction

Figure 4 shows an example of an agent user interacting with his user agent A via the multi-layered interface. Agent A has chosen the indicated motivations, policies, behaviors, etc. and its agent user has decided to make A "kick B" by interactively setting A's behavior. Likewise, an agent user might choose low-level, direct manipulation control by bending the knee of his agent a bit more.

Currently, our hierarchical interface agent project is still at a developing stage, and we do not yet have a general system to control life-like agents as illustrated in Figure 4. However, we have implemented the basic AHA framework, and are now testing it in the simplified scenario of a board game called the Malefitz game. Also, we have implemented a mobile environment where the interface agent inhabits and guides users. Section 5 illustrates the current status of the environment and shows a few examples of life-like interface agents. They are 2-D cartoon characters and controlled by hard programmed behaviors. However, it partially uses the AHA architecture and demonstrates interesting interactions with agentees.

4.2 Implementation of Malefitz Game

We are testing AHA on a simple game called Malefitz. Malefitz is a board game for four or six players. The goal of the game is to move one of player's pieces to the goal position at the top of the board (see figure 5 for layout). The distance of piece movement is determined by a die roll. Each user has the choices of which piece to move, direction at the intersections and position of obstacle stone that can block other players. The game involves complex strategy planning such as piece movements, stone movements, dynamic change of enemies and friends, etc.

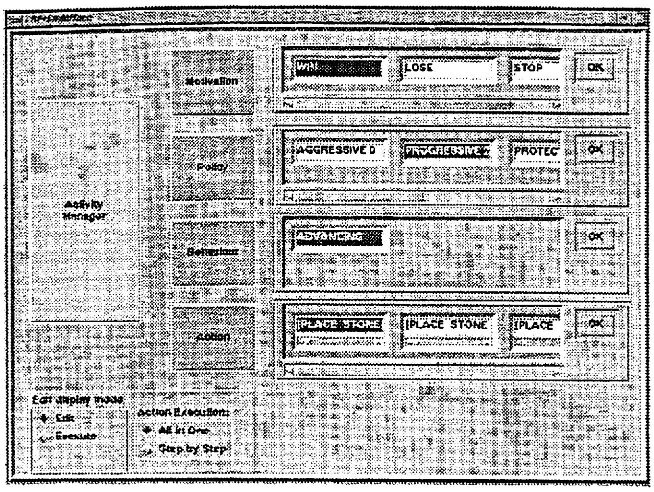


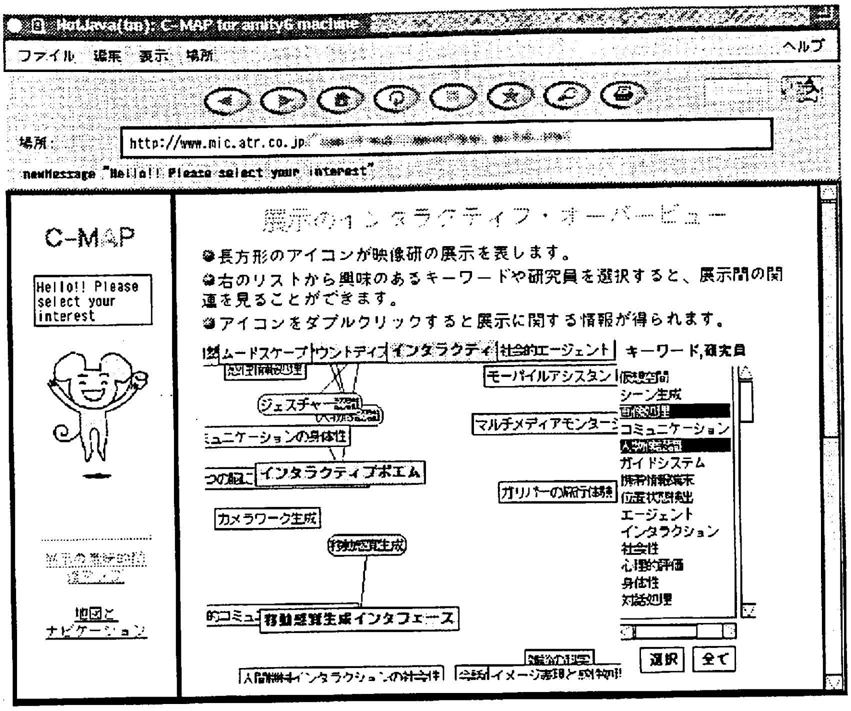
Figure 6: Agentee's Interface for Manipulating Agent in Malefitz Game: Agentee can control and design motivation/policy/behavior/action of agent at different levels of layered abstraction.

To illustrate how we constructed a Malefitz player using the AHA architecture, consider the top four layers again in Figure 3. In our initial experiments, we mainly focussed on the response part of the various layers in the AHA agent to play the game. We are currently working on creating the perceptual part of the hierarchy.

We partitioned the *motivation* layer into motivations and personality traits. The possible motivations are to WIN, to LOSE, to HURRY, and to STOP playing. The personalities are CONSERVATIVE, FRIENDLY and AGGRESSIVE. These motivations can be activated which causes lower level functions to be implemented in the policy layer (see below). Likewise, functions at this layer initiate response activities. For example, to roll the die after the previous player (agent) has finished his turn requires the perception of this (external) event to propagate up to the motivation layer. Now, the motivation layer must decide what to do. The reason an agent rolls the die comes from its motivations. Thus, the motivations to WIN, LOSE or HURRY the game cause the necessary activities in the layers below to roll the die (first action). The outcome of the die roll is perceived again, travels up the hierarchy, and initiates a move of one of the pieces (second action) based on the current motivation and the state of the levels below. However, if an AHA is motivated to STOP playing the game, it likely would not roll the die at all.

Agents have five main *policies* which are instantiated depending upon the state of the game and the current motivations of the agent. The policies are: AGGRESSIVE, PROGRESSIVE, PROTECTIVE, DISTURBING, and CAUTIOUS. As an example, a protective policy encourages movements that keep the player's pieces grouped together. It also influences stone placement so that they are placed to protect the player's pieces.

In response to the policies which become active, various *behaviors* activate. Possible available behaviors are: KICK-OUT (move to send an opponent's piece back to



newMessage "Hello!! Please select your interest"

Hello!! Please select your interest

●長方形のアイコンが映像研の展示を表します。
●右のリストから興味のあるキーワードや研究員を選択すると、展示間の関連を見ることができます。
●アイコンをダブルクリックすると展示に関する情報が得られます。

Figure 7: Screen shot of Mobile Assistant Computer: life-like character provides expressive messages

its home base), CHASING, ADVANCING, BLOCKING, BARRICADING, SUPPORTING and FLOCKING. For illustration, a chasing behavior might be instantiated when a disturbing policy is active. This behavior causes actions to be generated to move a piece to chase an opponent's piece.

Agents have four actions available to cause changes in the environment: ROLL-DIE, SEND-HOME, MOVE-PIECE and PLACE-STONE. These are the minimum number of actions needed for the agent to play the game. We are currently adding additional actions for the game world such as SAY (to talk to other agents) and GESTURE (to indicate suggestions non-verbally). As support for more complex agent behaviors increases, there will be considerably more actions required. The actions are the only communication path available between the agents and the world.

Several activities in each layer may be active at the same time, which may cause multiple or conflicting responses in the lower layers. For example, in response to several policies which are active, as well as the current die roll and board configuration, there may be several behaviors which are appropriate. In this case, the behavior which has the most support from the policy layer is executed by the agent. In the agent interface, the behaviors are prioritized using this scheme and displayed so that the agentee can see what her agent considered and will execute. Figure 6 shows the interface with which the agentee can override her agent's choices. Note that the interface provides access for the agentee at any of the four levels.

The agent interface also has an activity manager display so that agentees can see the current state of their agent or the world. This display is useful for the agentee to monitor the progress of his agents so that he can decide whether to override some aspect of the agent's behavior.

Our implementation of an AHA agent player (user agent) for Malefitz is by no means complete; however, our agents are able to play the game by making reasonable moves depending upon the die roll. At any time, agentees (real players playing through their user agents) can influence the "decisions" of their agents at several levels. We also provide "avatar" agentality, that is, users can choose to participate without agent support, thus playing directly against other virtual users and user agents.

Our next step of development will include the addition of perceptual parts to the agents, as well as the improvement of the graphical modeling of the agents and inter-agent communication support using the AHA framework.

5 Application: Guide Agent in Museum

In this section, we introduce a computer supported communication environment in which autonomous agents inhabit and interact with human users.

We have proposed the Meta-Museum concept [2] as a computer-supported communication environment of the future. The Meta-Museum is an interdisciplinary museum incorporating augmented reality, mobile computing, computer vision and interface agent technologies to facilitate the communication between visitors and the

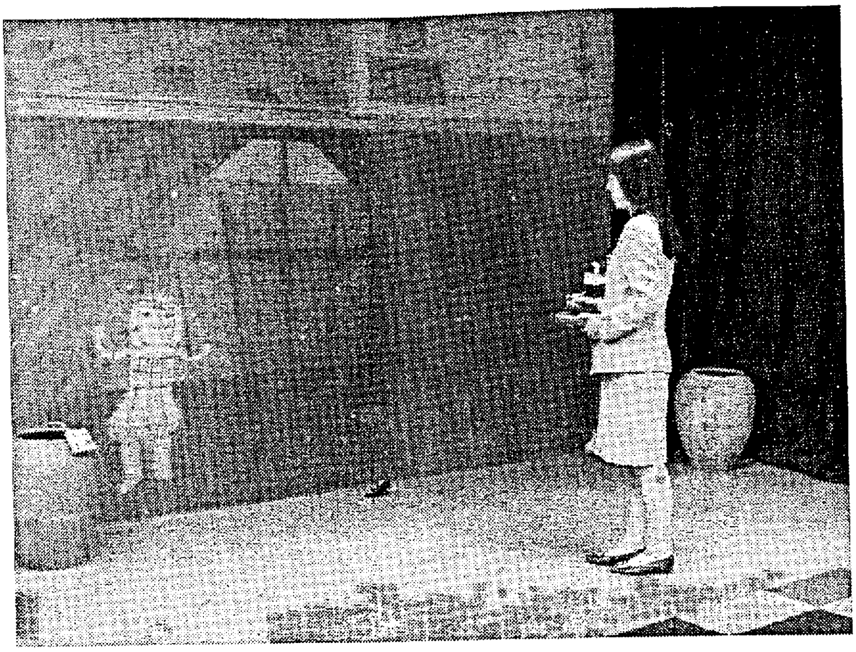


Figure 8: Mobile and Personal Guide Agent in VisTA-walk: personal agent transposes itself from mobile computer to screen of virtual exhibit having a full-body gesture interface

people behind the exhibits. It will transcend the conventional museums and even the recent virtual museums by not only combining the physical and the virtual exhibits seamlessly but also incorporating contexts of visitors and exhibitions.

The objective of Meta-Museum is to provide an interactive, exciting, entertaining and educational experience to visitors. The people in the environment will be able to share knowledge, ideas, ways of thinking, and visions in order to know each other, to create new ideas, to form communities, etc. One of the research goals of Meta-Museum is to offer rich and effective interaction with a museum's archives and the people behind them through communication mediated by interface agents.

The exhibit guidance is a key to support the personal and comprehensive experience in the Meta-Museum in addition to the exhibit presentation and the database management. We have been developing the VisTA-walk system and the C-MAP mobile guide system. VisTA-walk[14][1] is the first prototype system of Meta-Museum and a virtual walk-through system with a vision-based interface[15] to provide full-body gesture interaction. We have modeled an ancient Japanese village where visitors can visually experience archaeological reconstruction in VisTA-walk. C-MAP [3] is a context-aware mobile assistant system that provides a mobile guide agent for assisting visitors during exhibitions. We have recently combined these two systems.

5.1 C-MAP Guide Agent

The C-MAP guide agent usually inhabits a mobile computer assisting the physical tour of exhibitions[2]. However, if the user stands in front of the VisTA-walk system, the interface agent transports itself to the VisTA screen from the mobile computer. Then it participates in guiding the agentee through a virtual tour of the ancient village. The participation of the agent is based upon the visitor's (agentee's) preferences which are automatically estimated based on the discerned behavior of the visitor during previous physical and virtual tours of the exhibition.

Figure 7 shows a screen shot of the C-MAP mobile assistant computer. In the upper-left corner is the animated character along with a message box that the animated character uses to send messages to the visitor. Also shown in the figure is the semantic map of the exhibition. It shows semantic relationships using Euclidean distances of keywords and objects of each exhibit. It also can show the physical map on the screen.

The objectives for the C-MAP guide agent are modest at this time because we want to test the infrastructure first. We plan to later proceed to more complex models of agent behaviors. The main objectives for the guide agent implemented at this point are: (i) track agentee's physical position, (ii) track agentee's interest on semantic map, (iii) suggest alternative exhibit to visit depending on agentee's interest and (iv) provide an animated character that represents the guide agent's internal state.

---

[1] VisTA: Visualization Tool for Archaeological data

[2] All systems have been implemented in our laboratory buildings. Consequently, the exhibits are research-related posters.

For each agent, the planner is implemented on the agent server machine using [incr Tcl], while other components are implemented on the portable machine in Java. We partially used an AHA architecture to implement our guide agents. By using the AHA approach, the guide agent responds to events asynchronously as they happen. Our current implementation of the planner is very simple and operates at the behavior layer.

We created 12 different life-like animated characters as the visual representation of personal agents which the visitor can choose from. Each character has short animations of four different behaviors that can be controlled remotely. The four different behaviors includes thinking, suggesting, pushing and idle. Additionally, the animated character can be set to exhibit a particular behavior for a given duration. The behaviors are implemented as a FIFO queue. When the queue is empty the idle behavior is displayed. The animated character applet is responsible for sending the agent server the name of the animated character the visitor chooses when they registers. Above the animated character is a small message window which can also be controlled remotely.

5.2 CMAP Agent in VisTA-walk

Figure 8 shows the visitor agentee guided by the guide agent as it appeared on the VisTA-walk screen. Notice that in the visitor's hands is a mobile hand-held computer. An infra-red sensor detects the visitor's existence from a badge she is wearing and her identification code which is communicated to the C-MAP agent server. The agent server sends the visitor's personal agent character to the graphics workstation used for the VisTA-walk system.

There are three windows displayed on a large projection screen for VisTA-walk: a walk-through view, a bird's-eye view and a document browser (i.e. WWW browser). The WWW browser is linked to the objects (e.g. houses, warehouses, pots) in the walk-through view to provide information prepared as requested by the visitor's selection using gesture pointing or mouse-clicking. When the visitor moves in a relaxed pose (i.e. no gestures) they control the walk-though.

The agent guides the visitor in three ways: fully automatic guided tour, semi-automatic guided tour and manual guided tour. In the fully automatic mode, the visitor doesn't have any control of walking through but enjoys passive guided tour based on the agent's planning. In the semi-automatic mode, the visitor has control of walking-through but the agent suggests points to visit. The suggestion is displayed by the *follow-me* behavior, to which the visitor can choose to follow or not. Instead, if the visitor goes to a different point, the agent follows her and tries to give another suggestion after the visit[3]. In the manual mode, the agent only follows the visitor to provide company. In future, we plan to have the agent provide instructive guide presentations at points of interest depending on the visitor's interests.

Selection of the guiding mode is done automatically but may be overridden manually. The agent planner chooses a mode based on the visitor's context, in particular, context is based on the activity level recorded through the previous visits measured by interaction logs. In the current implementation, the selection algorithm is still simple such that it uses the fully automatic mode if the amount of interaction with the mobile computer is small.

The choice of character for the personal agent is open to the user when the mobile-computer is loaned to them during a registration process. Some of the characters are: robot, dad, ghost, rocket, goat, etc. Our experience found that user's responds positively when the personally chosen agent appears on the VisTA-walk screen[4]. The agent's character heightens his or her enjoyment of and interest in the exhibition.

6 Summary and Future Directions

In this paper the project on interface agent research at ATR-MIC has been presented. We are currently implementing more complex behaviors of the animated agent using the AHA architecture for the applications outlined. We are gradually getting results on each sub-topic as described above. It is our belief that the vertically layered AHA architecture and the agentality aspect of our agents will provide a convenient and clear mechanism for users and designers to interact with agents.

The display and control of changing agentality has yet to be developed. Interface agents should support visitors as personal agents or exhibit agents. As the personal agent has three agentalities, it changes the major portion of its interaction to the visitor as she/he proceeds through the exhibition. For example, an assistant agent will appear to guide the walking course through the museum and provide information of interest to the visitor. An actor agent will show how the artifacts were used, visually traveling back to the original era and location. An avatar agent is also implemented in the VisTA-walk system, but only to provide the visitor's position in the virtual village at this moment.

Such a computer actor agent will play its role together with other computer and/or human actors based on a given script. In order to make the computer actor intelligent-looking, the timing of interaction is important. Timing is especially difficult to control when playing with a human actor because the agent's sensory devices do not know how long a human action takes during role-playing since the activity manager does not know *a priori* how long an actor takes to play a part. Scripting method such as [16] will be suitable for such a computer actor environments.

---

[3] In the current implementation, the agent waits for the visitor coming back to the suggested point.

[4] The integrated VisTA-walk and CMAP demonstration was performed during ATR laboratory open house day in November 1997.

The immersive presentation shown in Figure 8 is only one way of presenting the Meta-Museum exhibit. We plan to introduce a see-through HMD (head mounted display) device for personalized presentation of personal interface agents. We expect that the structure of creative thought activity and changing perspectives in agent mediated communication systems can be described together with agentality.

Acknowledgments

We thank Yasuyoshi Sakai and Ryohei Nakatsu, for their support and encouragement in this research. We also would like to thank Silvio Esser, Tameyuki Etani, and Kazushi Nishimoto for their contributions and comments to the work mentioned here.

References

[1] Bruderlin, A., Fels, S., Esser, S. and Mase, K.: Hierarchical Agent Interface for Animation, in *IJCAI-97 Workshop on Animated Interface Agents*, pp. 27–31 (1997).

[2] Mase, K., Kadobayashi, R. and Nakatsu, R.: Meta-Museum: A Supportive Augmented Reality Environment for Knowledge Sharing, in *Int'l Conf on Virtual Systems and Multimedia '96*, pp. 107–110, Gifu, Japan (1996).

[3] Fels, S., Sumi, Y., Etani, T., Simonet, N., Kobayashi, K. and Mase, K.: Progress of C-MAP: A Context-Aware Mobile Assistant, in *AAAI Spring Symposium*, CA (1998) (to appear).

[4] Bruderlin, A.: The Creative Process of Animation Human Movement, *Knowledge-Based Systems*, **9** No. 6, pp. 359–367 (1996).

[5] Nagao, K. and Takeuchi, A.: Social Interaction: Multimodal Conversation with Social Agents, *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, **1** pp. 22–28 (1994).

[6] Hasegawa, O., Itou, K., Kurita, T., Hayamizu, S., Tanaka, K., Sakaue, K. and Otsu, N.: Active Agent Oriented Multimodal Interface System, in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 82–87 (1995).

[7] Blumberg, B. M. and Galyean, T. A.: Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments, in *SIGGRAPH 95*, pp. 47–54 (1995).

[8] Justine Cassell et al., : Animated Conversation: Rule-based Generation of Facial Expression, Gesture & Spoken Intonation for Multiple Conversational Agents, in *SIGGRAPH94 Proceedings*, pp. 413–420, Orlando, FL (1994).

[9] Maes, P.: Agents that Reduce Work and Information Overload, *Communications of the ACM*, **37** No. 7, pp. 30–40 (1994).

[10] Müller, J. P., Pischel, M. and Thiel, M.: Modeling Reactive Behavior in Vertically Layered Agent Architectures, in Woolridge, M. and Jennings, N. R. eds., *Intelligent Agents*, pp. 261–276, Springer-Verlag (1994).

[11] Brooks, R. A.: A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, **2** No. 1, pp. 14–23 (1986).

[12] Bonasso, R. P., Kortenkamp, D., Miller, D. P. and Slack, M.: Experiences with an Architecture for Intelligent, Reactive Agents, in Woolridge, M., Müller, J. P. and Tambe, M. eds., *Intelligent Agents II*, pp. 187–202, Springer (1995).

[13] Hayes-Roth, B.: A Blackboard Architecture for Control, *Artificial Intelligence*, **26** No. 3, pp. 251–321 (1985).

[14] Kadobayashi, R., Nishimoto, K. and Mase, K.: Design and Evaluation of Gesture Interface of an Immersive Walk-through Application for Exploring Cyberspace, in *The Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, Nara, Japan (1998), (to appear).

[15] Wren, C. R., Azarbayejani, A., Darrell, T. and Pentland, A.: Pfinder: Real-Time Tracking of the Human Body, *IEEE trans. PAMI*, **19** No. 7, pp. 780–785 (1997).

[16] Pinhanez, C. S., Mase, K. and Bobick, A.: Interval Scripts: a Design Paradigm for Story-Based Interactive Systems, in *CHI97 Conference Proc.*, pp. 287–294, Atlanta, GA (1997).