Agents, Robots & Artificial Intelligence

4.

We now turn to the literature on computer agent and human interfaces based on the idea of *agency*. In contrast to the previous two chapters, which dealt with the idea of agency in a general sense, this chapter presents it as it instantiates itself in what has been called *software agents* (c.f. quote in margin). A natural extension of the idea of sofware agents is the notion of a *social agent*, which are software agents, robots, or autonomous creatures that posses some social-interaction know-how—the kind of which we reviewed in the last two chapters—and can thus engage in social interaction with people on some level. First we will look closely at ways of embodying and presenting agents, and then review the most relevant work on agent architectures and artificial Intelligence.

"The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology. They had in view a system that, when given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck. An agent would be a 'soft robot' living and doing its business within the computer's world."

—Alan Kay (1984, p. 58)

4.1 The Agent Metaphor

Although the idea of software agents dates back to the sixties [Kay 1984], it is not until recently that the potential value of the agent metaphor for human-computer communication is becoming accepted [Hasegawa et al. 1995, Nagao & Takeuchi 1994, Rich et al. 1994, Maes 1994, Maulsby et al. 1993, Chin 1991, Laurel et al. 1990, Laurel 1990, Oren 1990, Crowston & Malone 1988].

Searching for an unambiguous definition of the term "agent" would be futile, but *a* definition is better than none. The Merriam Webster's Collegiate Dictionary has 4 different definitions for the term (side bar), none of which will suffice on its own. Kozierok and Maes [1993] define an agent as "a semi-intelligent, semi-autonomous system which assists a user in dealing with one or more computer applications." The definition used here is similar, but leaves out the reference to applications:

An interface agent is a metaphor for an agenda or a collection of task-level goals in the computer, imparted to it by the user, and the capability to carry out those, within reasonable expectations.

agent *n* (ME, fr. ML agent-, agens, fr. L, prp. of agere to drive, lead, act, do; akin to ON *aka* to travel in a vehicle, Gk *agein* to drive, lead) (15c) **1:** one that acts or exerts power **2a:** something that produces or is capable of producing an effect: an active or efficient cause **b:** a chemically, physically, or biologically active principle **3:** a means or instrument by which a guiding intelligence achieves a result **4:** one who is authorized to act for or in the place of another: as **a:** a representative, emissary, or official of a government (crown-) (federal -) **b:** one engaged in undercover activities (as espionage): spy (secret -) **c:** a business representative (as of an athlete or entertainer) (a theatrical -)

—Merriam Webster's Collegiate Dictionary, Tenth Edition

In addition to leaving out references to current computer applications, this definition slightly more specific than the one presented by Kozierok and Maes [1993]. It still contains direct reference to computers since computers are the only known synthetic entity to possess the necessary power to make anything close to what we intuitively might refer to as agents. The definition does not distinguish between kinds of intelligence or kinds of skills, and these will be addressed as we go on.

Not all agents in the real world are humanoid: dogs for example communicate via a subset of the human mutlimodal "command set". Since we are interested in the full range of multimodal interaction, the discussion will naturally focus on humanoid agents—those that bear a resemblance to humans in *appearance* and *skills*—as opposed to agents that resemble arachnids, insects or dogs. Such a distinction is necessary because so much of face-to-face communication is based on assumptions about skills (i.e. intelligence level, or competence) and appearance, both spatial and visual representation.

Agents represent thus the ability of the computer to accomplish something on behalf of the user [cf. Minsky & Riecken 1994]. To do this they posses high-level knowledge about a particular task domain or domains.[1] How the user conveys these wishes to the agent is an issue of human-computer interface design, and of course a central issue of this thesis. For example, Chin [1991] describes an agent that gives users advice about UNIX commands during interactive sessions. This system is a text-based natural language system using a keyboard as the input device and written English as the means of communication. Maes & Kozierok [1993] describe an agent that selects information from news sources depending on their relevance to what the user has found interesting in the past. These kinds of agents could be called terminal-based, because they rely on the traditional interaction methods of keyboard, mouse and monitor. For agents that can see and listen to the user, the issue is somewhat more involved.

1. The main reason for creating agents, and not simply making a suite of "tools" that one can select between, is that in addition to making the "tools" very sophisticated—i.e. moving toward their automation—we also want to automate the selection between these "tools". What inevitably merges out of such a creation is somethingone is hard-pressed to call anything but an "agent".

FIGURE 4-1. Cartoon illustrating the issue of embodiment and multimodal interaction.

When its owner addresses it, Tobor the vacuum cleaner turns in the direction of the speech and starts to decode the audio emanating from the human. The owner tells it to vacuum in a particular location, as indicated with a manual gesture. Miraculously, Tobor recognizes this as a deictic gesture and looks in the right direction even as the owner continues to speak. It then looks back when the utterance is finished. When asked if it understood, it nods enthusiastically.

A robot with such sophisticated communications skills still doesn't exist, but when it does I sure will buy one.

4.1.1 Agent Embodiment

"There is no place ... for a disembodied 'system' as a source of agency, communication, or collaboration: indeed, such disembodiment forces its mirror image on the participant and precludes the possibility of holistic response."

— Brenda Laurel (1992, p. 69)

The case for computer embodiment is most obviously seen in robotics, where the distinction between an agent and its environment is by default: a robot has a body that separates it from the rest of the world, and can thus be addressed and treated as an individual entity (see e.g. Brooks [1990] and Bares et al. [1989]). The issue of embodiment is important for face-to-face communication since the face/body system serves several functions, as we saw in the last chapter (page 37). The body parts that play the largest role in non-verbal communication are obviously the face, head, hands and, to some extent, the trunk. These have been treated thoroughly in Chapter 4. Here we will discuss two orthogonal issues of embodiment: *Visual representation* and *spatial representation*. Both play an important part in social communication. Visual representation includes the appearance of the agent—its physical form. Spatial representation is the kind of embodiment the agent has in the world and the way it can change its position in space.

4.1.2 Visual Representation

"With cartoon faces... becoming data measures, we would appear to have reached the limit of graphical economy of presentation, imagination, and, let it be admitted, eccentricity."

—Edward R. Tufte (1990, p. 142)

It may be argued that the most obvious and important form of embodiment for social interaction is a face. One of the earliest uses of facial expression to display machine status were the "Chernoff Faces" [Chernoff 1973]. Various features in a graphically generated face, like distance between the eyes, width of mouth, size of head, etc. were linked to variables in the status of a nuclear reactor: heat, pressure, etc. Since physical variables have nothing to do with human communication, this use of a face as a display method stands in strong contrast to the use of a face for social interaction, where its purpose is to facilitate dynamic, continuous exchange between the computer and its user.

While computer graphics work concerned with faces has to date focused extensively on their visual appearance, interactivity and effectiveness for information transmission has not been of primary concern. Convincing facial animation has proven to be a difficult task. A common limitation of physically-modeled faces [Essa 1995, Essa et al. 1994, Pelachaud et al. 1991, Waters 1990, Takeuchi & Nagao 1993, Waite 1989] is that the meaning of their expressions is often vague and a computer-controlled human face looks abnormal, even repulsive. An ideal solution to this would be to exaggerate the facial expressions, but within a physical modeling framework this may look unconvincing or awkward. An alternative is what might be called a "caricature" approach [Thórisson 1993a, 1993b, Librande 1992, Britton 1991, Laurel 1990] where details in the face are minimized and the important features exaggerated (see Hamm [1967] for an excellent discussion on cartooning the head and face). Brennan [1985] created a system that could automatically generate caricature line-drawings of real people from examples that had been entered by hand. Librande [1992] describes a system called *Xspace* that can generate hundreds of artistically acceptable two-dimensional drawings from a small example base. Simplified faces seem a very attractive alternative to physical modeling for animating interface agents, both in terms of computational cost and expressive power.

Another important issue in visual representation are the hands. Hands, as discussed above, can carry a lot of meaning and are also crucial in process control—directing the flow of the dialogue [McNeill 1992]. Again, details in the hands' representation below the gross anatomy level are not important for this purpose since crucial communicative information is generally not carried in their photo-realistic aspects.

Of primary concern in the visual representation of interface agents is the dynamic appearance of the agent: how it moves and reacts over time. This is even more important than static appearance, as we know from the qualitatively different experience of looking at people on photographs and interacting with them in real-time. Most of the work in this arena has been in animation [Sabiston 1991, Lasseter 1987, Thomas &

Johnston 1981], and adopted to a limited extent in interactive agent design [Bates et al. 1992]. This is an area that requires much more research and is closely linked with research on animal motor capabilities. In this thesis, a choice was made to use a cartoon-style representation of the agent (see "Character Animation" on page 203).

4.1.3 Spatial Representation

Giving a listening computer a spatial location makes it possible for its user to rely on conventions about "address" and point of view in his interactions[1]—something that is impossible if the computer listener is omnipresent.[2] And by making a computer agent situated in the real-world along with the user and the task at hand, a person can move between the agent and the task by virtue of social convention.

Common space between a user and a computer agent can be accomplished in two prototypical ways: The user can be brought into the computer's space, as is done in immersive virtual environments (Figure 4-3) where the user wears head-mounted goggles with stereoscopic graphics [Held & Durlach 1992, Sheridan 1992], or the agent can be brought into the user's world, as seen most clearly in robotics. This can be done by

FIGURE 4-2. Although HAL-9000's omnipresent fish-eye lens (on left) in *2001: A Space Oddyssey* [1968] proved highly effective for dramatic effect, ergonomists are quick to point out its inanimate embodiment and lack of visual feedback as troublemakers in a conversational interface.

1. This is true whether the computer's location is within the user's interaction space, such as in face-to-face conversation, or external, such as in a phone call.
2. From the human's perspective, of course, in other words, the user can make no assumptions can be made about the computer's visual or auditory "point of view."

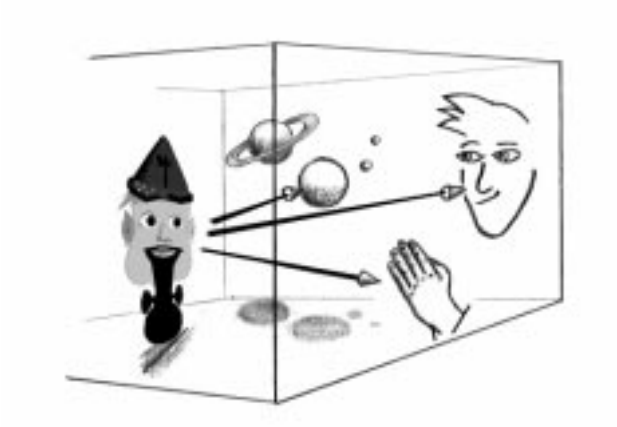


FIGURE 4-3. To achieve common space, the user can be brought into the agent's world

either making a robotic head (and body) or by allowing agent and objects to be displayed on separate monitors, placed at an angle to each other (Figure 4-4), or by giving the agent a physical body. These two methods are really two extremes on a continuum of ways to achieve integration between virtual and real worlds. Both extremes have their problems and virtues. Immersion allows both the user and agent to reference things and each other within the graphical world, eliminating the complexities involved in sensing and referencing real-world objects. It probably would be the method of choice for adventure games where the goal is total immersion and all the objects of interest are within the computer's world. A disadvantage of this approach is that the user has to "dress up" to have a common space with the agent. This precludes the agent from perceiving anything outside its own virtual world. The second option places an agent in real-space, which allows it to reference objects in the computer's world (although the reference space is now a 2-D projection) and still keeps open the option of referencing real-world objects, depending on the agent's perceptual prowess. One problem with this approach is the need to represent two distinct spaces: one within the workspace world and one in the real world. Another is sensing the surroundings and the user. However, if this can be done in a non-intrusive way, bringing the agent into the user's world offers more seamless integration of user-agent interaction with the user's work. This is the approach taken here.

The terminal-based interface agents to date have been represented visually by simple icons [Maes 1994, Maes & Kozierok 1993, Seth & Maes 1993], pre-recorded video clips [Laurel et al. 1990, Laurel 1990, Oren 1990, etc.] and presented inside windows on regular desk-top computers (or they have simply been hidden from the user's view [Mitchell et al. 1994, Sparrell & Koons 1994, Chin 1991]). Because of this, their spa-

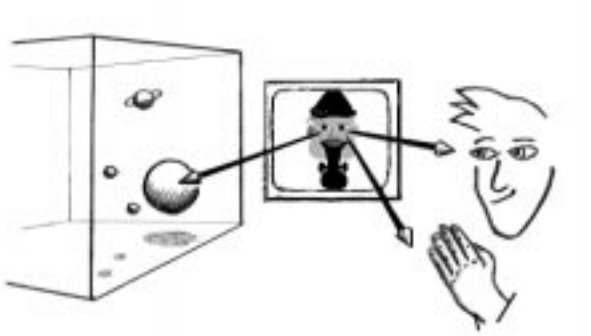


FIGURE 4-4. The agent can be brought into the user's world by giving it a physical embodiment such as a screen.

tial position has generally had no function at all. This is in part because of a lack on the machine's side to localize the user (and itself) in real-space. Another inherent complication is that representing both agent and work space in the same 2-D plane makes agent actions that rely on three-directional cues—such as deictic gestures of gaze and hands—difficult save for the simplest cases.

An exception to a history of a single 2-D plane representation was a system by Schmandt et al. [1985] employing speech recognition, called the Conversational Desktop. Their system employed space sensing technology to demonstrate how directionality—one of the cues for inferring "address"—plays a role in communication: If you are turned toward someone when speaking an utterance, chances are the utterance is meant for him or her. The system would only listen to the user's speech if s/he was turned to the computer screen. By giving computers information about spatial layout of users and objects—including themselves—the agents' glances and deictic gestures, as well as "point of view," can begin to have meaning in the context of the interaction.

4.2 Agent Architectures

Agent design in AI has mainly been in the area of robotics, where a physical entity—often mobile—is used as a testbed for the development of control strategies. Approaches taken to date can be classified into two categories, "classical AI" and "behavior-based AI" (c.f. Maes [1990b]). As Brooks [1990] has pointed out, even though a happy marriage of the two has yet to come about, the approaches are somewhat complementary and as I will argue later, both have features to offer for

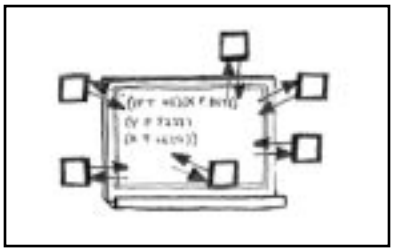


FIGURE 4-5. A blackboard serves as the common storage of intermediate and final results produced by a collection of independent processing modules (small squares) or 'Knowledge Sources.'

social agent design. Here we will look briefly at 7 system architectures, three from the classical AI pool, two from the behavior-based one, and two hybrid.

4.2.1 Classical A.I.

Blackboard systems were designed to handle unpredictable information like that encountered in speech recognition or planning [Hayes-Roth et al. 1988, Nii 1989]. This architecture is relevant here because it provides potential solutions to some of the problems multi-modal interfaces present, namely those of multiple levels of detail, multiple data types and high variability. The blackboard architecture attacks the problem of unpredictability by the use of a common data storage area, or blackboard, where results of intermediate processes, or knowledge sources (KS), are posted and can be inspected by other processes working on the same problem (Figure 4-5). Indeed, the problem of social behavior control includes some of the same problems as automatic speech recognition, where information on many levels—phonemic, lexical, syntactic, semantic, discoursal, pragmatic—can come to bear on the recognition process [Allen 1987]. HEARSAY [Reddy et al. 1973] was the first system to apply this architecture to a real-world problem. It consisted of multiple knowledge sources, each designed for recognizing and classifying a specific feature of natural speech. The system, and its modified version, HEARSAY-II, were designed to exhibit the following properties absent in prior systems [Nii 1989, p. 21]:

1. *The contribution of each source of knowledge (syntax, semantics, context, and so on) to the recognition of speech had to be measurable.*
2. *The absence of one or more knowledge sources should not have a crippling effect on the overall performance.*
3. *The system must permit graceful error recovery.*
4. *Change in performance requirements such as increased vocabulary size or modifications to the syntax or semantics should not require major modifications to the model.*

The interesting points to notice here are 2, 3 and 4. These do not only apply to requirements for speech recognition systems: they apply to any system that is to function semi-autonomously in a dynamic environment. Variations on the original version of the blackboard architecture have been successfully applied to areas such as vision and distributed computing [Nii 1989]. Jagannathan [1989] discusses approaches to applying blackboard systems to real-time applications. Modifications to the original versions for this purpose include mechanisms to allow interleaved execution of subsystems, as well as communication between them [Fehling et al. 1989], resource management, speed/effectiveness trade-off and reactive systems behavior [Dodhiawala 1989].

Another system using traditional AI methods is Chin's [1991] UCEgo. This system is an addition to a natural language UNIX consultant system (UC) that gives advice to users about commands and command options. The system's task can generally be described as that of goal detection and maintenance, using traditional planning techniques. The main difference between this system and the others discussed here is that it is specifically designed to interact with humans. The interaction is of the step-lock, unimodal kind, via a teletype. An example of interaction between a user and the system is shown in Figure 4-6.

> What does who -b do?

who does not have a -b option

>What does ruptime -t do?

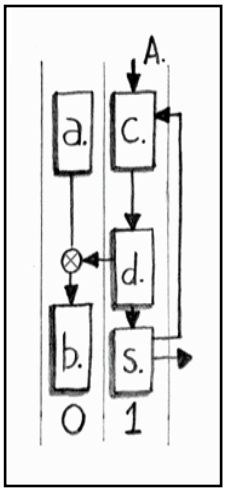
I'm sorry, I do not know that.

FIGURE 4-6. Example of a user-agent interaction in the UCEgo system [from Chin 1991]. The user's input starts with a >.

A third architecture in the classical AI category is Schema Theory[1] [Arbib 1992], which is historically an outgrowth form blackboard systems. Schema theory is an attempt to deal with the complexity of large systems that interact with the real world. A schema is both a storage of knowledge and the description of a process for applying that knowledge, and in this respect bears both resemblance to blackboard architectures' knowledge sources and Maes' competence modules (see below). This system postulates a set of basic perceptual and motor schemas that provide simple, prototypical perceptual capability and movement patterns. The schemas are combined to form assemblages of coordinated control programs which interweave their activations in accordance with the current task and sensory environment. Schema activations are largely task-driven, reflecting the goals of the organism and the physical and functional requirements of the task. As Arbib [1994, 1992] has noted, the generality of schema theory puts most distributed and layered systems, such as Minsky's Society of Mind [1989] or Brooks' *Subsumption* architecture [1990], under its umbrella.

4.2.2 Behavior-Based A.I.

As an example of the situated action or the behavior-based approach (cf. [Brooks 1991, 1990, 1986], [Meyer & Wilson 1991] and [Maes 1990a]), Brooks [1990] proposed what he calls a subsumption architecture where low-level behaviors of a robotic agent can be subsumed by higher-level, later-designed behaviors (Figure 4-7). This allows for incremental development of robot skills and a robustness that is difficult to achieve with traditional methods. Another example is Maes' [1989] architecture that is based on *competence modules*—software modules that contain enough information to execute a particular behavior from beginning to end (Figure 4-8). The modules are connected together by activation links that control their sequence of execution. The input to the modules can come both from internal goals and the environment. This architec-



GURE 4-7. In this example, a bsumption architecture has been ilt for a tour-guide robot. Level 0 ntains the behavior modules *local-nder* (a) and *move* (b). Modules in vel 1 include *up-counter* (c), *ndmark-list* (d) and *speak* (s), which tputs spoken information. The ndmark-list module suppresses the bot's wander behavior (valve arked x) so that it ends up ccessively at each landmark, and ggers the speech for each one as propriate, while the up-counter eps track of which landmarks have en visited. (Adopted from Lyons & endriks [1992].)

1. Arbib's Schema Theory should not be confused with Shank's scripts [Schank 1990, Schank & Abelson 1977], sometimes also referred to as schemas, which is a construct invented for modelling human memory and production mechanisms for stories.

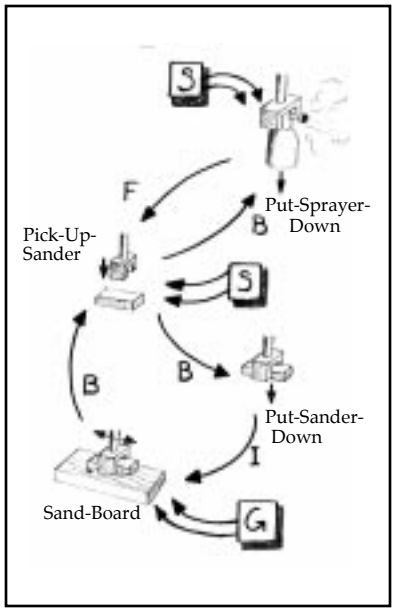


FIGURE 4-8. Toy example of the interaction between goals, states, inhibition links and spreading activation links in Maes' system [1990c]. The initial situation is {sprayer-in-hand, sander-on-table} a top of figure, and the initial goal is {board-sanded}. (B = backward spreading activation, F = forward, I = inhibition, G = goals, S = state. Adopted from Maes [1990c].)

ture, and other reactive approaches [Wilson 1991, Steels 1990, Agre & Chapman 1987] are very good for effective action selection, and they allow their robots to learn over time. However, they lack methods to deal with external and internal time-constraints and are limited in the planning they can handle.

4.2.3 Hybrid Systems

The traditional approach to the sensory-motor problem is a three-part system with a pipelined architecture. The sensory input feeds into the sensory module, which is followed by the cognition module, which is followed by the motor module. This pipelined approach has recently been questioned by the behavior-based AI research. However, hybrid systems trying to combine the best of both approaches have been few. One approach is worth mentioning, though. It is the NASREM architecture [Albus et al. 1987]. This model tries to incorporate knowledge gleaned from animal research on sensory-motor capabilities and integrate this into a comprehensive scheme for autonomous and tele-robot control. The system contains multiple levels of processing, each level containing the three components of sensory processing, world modeling and task decomposition. A global data storage is accessible from any level, but sensory modules also receive information from the level below, and task modules receive input from the level above. There are five levels all together: *Mission* (information relating to a full mission), *service* (information related to parts of a mission), *task* (sub-components of a service), *elemental move* (transition from symbolic commands of movements to spatially-defined commands), *primitive move* (generates smooth trajectories) and *servo* (simple hardware control). Although the layered approach of this model sounds promising for achieving the best of both worlds—fast responses to time-constrained events and slower responses to less time-constrained events—it has been criticized for trying too hard to encompass all possible systems, and thus losing its descriptive power [Thorpe 1992].

Of particular interest here is Cassell et al.'s [1994a, 1994b] system for automatic speech and gesture generation. The system employs two computer drawn human-looking characters that interact with each other (in non real-time) using speech, gaze, intonation, head and manual gesture. The system employs what the authors call PaT-Nets (Parallel Transition Networks; Figure 4-9) in which synchronization between gestures and speech is accomplished as simultaneously executing finite state machines (FSMs). (See page 84 for a discussion of the limitations of an FSM-based approach.) While the system is focused only on the generation of multimodal acts and is not concerned with the complications resulting from temporal constraints in perception, action planning and execution, it provides an insight into the complexities of synchronizing various levels of multimodal action generation, from the phoneme level up to the phrase and full utterance.

FIGURE 4-9. A PaT-Net for generating gaze movements. Nodes specify actions; transitions between nodes are both conditional and probabilistic. All leaf nodes branch back to the root node unconditionally (adopted from Cassell et al. [1994]).

4.3 Summary

We have now covered background material in two areas: multimodal research, in the previous chapter, and AI and agent-based systems in this chapter. Embodiment has been dealt with somewhat in the robotics literature, perhaps because disembodied agents are more common in this area than in psychology. Whereas the psychological literature tends to be descriptive, the computational approaches focus on both descriptive and prescriptive models. As of yet, computer implementations are mostly concerned with getting something to work, as opposed to modelling human face-to-face interaction correctly, and at all levels, but this may simply be because the field is relatively young. Robotics and cognitive science research has made several contributions relevant to the task of full-duplex feedback, among them the blackboard architecture and the behavior-based approach to planning. However, this work needs to be adapted to the task of generating face-to-face computer systems. The next step is then to characterize the specifics of multimodal interaction to make this possible.