

A Taxonomy of Agents

J. Alfredo Sánchez

Technical Report ICT-97-1

ICT
Interactive and Cooperative Technologies Lab
Universidad de las Américas-Puebla
Department of Computer Systems Engineering
A. P. 100
Cholula, Puebla 72820 México

January 1997

A Taxonomy of Agents

J. Alfredo Sánchez

*Interactive and Cooperative Technologies Lab
Universidad de las Américas-Puebla
Cholula, Puebla, 72820 México
alfredo@cca.pue.udlap.mx
<http://udlapvms.pue.udlap.mx/~alfredo>*

ABSTRACT

The notion of *agent* has become increasingly popular in computer science research and the software industry. However, reconciling different views of agency and providing a precise definition of agent has remained a challenging task for researchers. This paper proposes a taxonomy of agents that provides a high-level, user-oriented view of existing conceptions of agency and their inter-relationships. Within this taxonomy, agents that are perceived directly by the end user (*user agents*) are placed at the same level as agents used in designing complex software systems and network communications. The class of user agents is further refined by developing useful subclasses and characterizing techniques commonly used in their implementation.

1. Introduction

The notion of *agent* has become immensely popular in computer science research and the software industry. Futuristic depictions of agent-enabled computer environments, where humans routinely interact with diligent software assistants, have contributed to this popularity. The potential applications of agent technology lie in areas as diverse as user interfaces, telecommunications, network management, electronic commerce and information gathering. However, in this incipient stage of agency development, researchers, product developers and end users face the crucial problems of defining *agent* and establishing criteria to determine the differences among existing views of agency. For example, differing views of agency have led to definitions of agent that are apparently unrelated. White [1996] refers to agents as entities consisting of procedures and their states, which represent a computer when such entities are sent to other computers for communication purposes. Shoham [1993] conceives of an agent as an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices and commitments. Etzioni [1993] presents agents as software robots (“softbots”) endowed with sensors and effectors which are capable of changing their external state by issuing commands.

Agents designed to be perceived directly by end users and to perform tasks on their behalf are an innovative class of agents that can facilitate human-computer interaction. The potential of agent-based user interfaces has been discussed extensively [Kay 1984, 1990; Laurel 1990, 1991; Negroponte 1990, 1995]. However, there has been a tendency to consider “interface issues” as peripheral to the central theme of agency and to refer to *interface agents* as merely an application of the *agent* notion used in the construction of software systems (see, e.g., [White 1996; Wooldridge and Jennings 1995a]).

This paper proposes a taxonomy of agents that provides a high-level, user-oriented view of existing conceptions of agency. Within this taxonomy, user agents are elevated to stand on their own as an abstraction forming the basis for a human-computer interaction paradigm. User agent subclasses are suggested, and the major techniques employed in current research developments are surveyed. User agents respond to particular motivations and raise a set of specific research issues [Sánchez et al. 1995]. Addressing such issues requires the development and deployment of agents in the context of system architectures and testbeds designed to support user agency research.

The paper is organized as follows. Section 2 develops the first level of a taxonomy of agents. Section 3 focuses on the specific class of user agents, defines user agent subclasses and provides examples from published research work in the field. Finally, a framework for characterizing ongoing work in user agents is discussed in Section 4.

2. Towards a Taxonomy of Agents

This section provides a high-level perspective of agency based on three major uses of the notion of *agent*. Further disambiguation of agency conceptions is achieved by summarizing various interpretations of the common attribute of *autonomy*, and exposing the most common problems associated with the terminology appearing frequently in agent-related literature.

2.1 Classes of Agents

Three major views of agency can be distinguished in the literature. First, computer scientists and software engineers have used agents as an abstraction to conceptualize, design and implement complex systems. This class of agents will be referred to as *programmer agents*. Second, agents may be viewed as autonomously migrating entities that act on behalf of network nodes in a distributed environment. This class will be termed *network agents*. Finally, agents have been proposed as an abstraction for end users to interact with computer systems. This view defines the class referred to as *user agents*.

2.1.1 Programmer Agents

Computer scientists deal with complex hardware and software entities. The abstraction mechanisms used to describe the behavior of these entities vary widely. Flow charts, data flow diagrams and object models have been popular *mechanistic* aids for visualizing automata operation. As the complexity of software systems has increased, *animistic* abstractions have been found useful in explaining system behavior. Given the programmers' familiarity with real-world entities that exhibit *intentions*, ascribing autonomy and intent to system components provides a convenient way to describe, analyze and predict the operation of entire systems without further concern about their internal details. This is the perspective adopted by Wooldridge and Jennings [1995a] to define an agent as *a system consistently described by the intentional stance* (in reference to Dennett [1987]). The emphasis here is on modeling the *processes* performed by the computer for the benefit of the *programmer*. As this view matures, an approach to software development is likely to emerge under the banner of *agent-oriented software engineering*.

Agents as an abstraction for building complex systems are being investigated in diverse fields. Agents are popular in Artificial Intelligence (AI) research [Brooks 1991a, 1991b; Etzioni 1993; Maes 1994; Russell and Norvig 1995], Distributed AI [Gasser and Huhns 1989], and Decentralized

AI [Demazeau and Müller 1990]. Other application areas include distributed systems [Agha 1986], software engineering [Nigay and Coutaz 1992], teleconferencing systems [Lee et al. 1993], and resource management [Miller and Drexler 1988].

Research in programmer agents has produced a number of architectures, testbeds, methodologies and tools [Hanks et al 1993; Wooldridge and Jennings 1995a; Wooldridge et al. 1996]. For example, *agent-oriented programming* (AOP) [Shoham 1993] is a framework that specializes object-oriented programming by introducing the notion of *mental states* for objects, which thus become agents (mental states include beliefs, capabilities, obligations, etc.). Computation in AOP takes place as agents interact with one another via communicative acts as categorized by Speech Act Theory [Searle 1969]. AOP concepts have been implemented by the AGENT0 [Shoham 1993] and PLACA [Thomas 1995] interpreters. *Oasis* [Cheong 1992] is a compiled programming language for distributed applications that allows processes to be modeled as a collection of agents. In Oasis, an agent is defined as an extension of the concept of *class* in object-oriented programming. This extension consists of *goal clauses* that are executed upon class instantiation. *April* [McCabe and Clark 1995] is an object-based programming language intended to support agent construction in distributed environments. However, April does not directly offer high-level agent-related abstractions. KQML (Knowledge Query and Manipulation Language) [Finin et al. 1993] is the most salient project in the area of protocols for inter-agent communication. KQML borrows from Speech Act Theory to specify messages to be exchanged by agents as *performatives* (messages intended to perform some action by virtue of being sent).

2.1.2 Network Agents

Communication in distributed computing environments often relies on *remote procedure call* (RPC) protocols. In RPC, a client process invokes a remote action by sending a message (including data arguments) to a server process and blocks until a reply is received with the results from the requested operations [RPC 1988]. At that point, call results are extracted and the caller's execution resumes. Network traffic may increase dramatically as the number of clients and the number and complexity of tasks increase.

Network agents offer an alternative to RPC. Communicating computers can supply the code of the procedures to be executed by remote machines, allowing the computation to migrate to the site where data resides. Messages sent across the network include instructions to be executed and data reflecting the procedure's current state. Each executing procedure is considered an autonomous agent that may decide to migrate to a different host in the network according to its task

requirements. An agent may start executing on one machine and, when other resources are needed, jump to the machine having those resources, continue to perform its computation there and, if needed, migrate again. While performing its work in a receiving host, the migrating process is regarded as an agent representing the interests of the sending machine. Network agents have also been referred to as *mobile agents* [White 1996], *itinerant agents* [Chess et al. 1995] and *transportable agents* [Gray et al. 1996]. The mobile agent-based model for distributed computing has also been called *remote programming* [White 1996].

Implementing mobile agents requires that participating computers agree on the set of instructions that can be sent over the network and executed by receiving hosts. These instructions constitute a *mobile agent programming language*. An engine capable of interpreting such a language must reside on each participating host. A natural approach to providing a secure environment is to implement agencies as interpreters able to monitor instructions being executed by agents and providing facilities for agent migration and reception. If mobile agents are to communicate with each other, agencies must provide mechanisms for agents to meet and exchange data as well. Implementations of the mobile agents model are just starting to appear. Projects notable in the development of mobile agent facilities include *Agent Tcl* [Gray 1995], *Ara* [Peine 1995], *TACOMA* [Johansen et al. 1995] and *Telescript* [White 1996]. In all cases, engines that can execute statements in an interpreted language have been implemented and statements that allow programmers to transparently specify agent migration and agent meetings are central constructs.

2.1.3 User Agents

End users have consistently attributed autonomy and other aspects of agency to computer systems, even when this view has not been explicitly promoted by user interfaces. For example, in an experiment conducted by Friedman [1995], eighty three percent of the participating users attributed autonomous decision making capabilities or intentions to computer programs. A natural next step in the development of human-computer interfaces is to make the agent abstraction explicitly available to the end users. Just as software developers use programmer and network agents as abstractions to cope with the complexity of system design, end users can better deal with system complexity by viewing programs as animistic entities.

Since user agents directly affect the way human-computer interfaces are perceived, the term *interface agents* has frequently been associated with this view of agency (see, for example, [Laurel 1990], [Kozierok and Maes 1993], [Wooldridge and Jennings 1995a] and [Sánchez 1993]). *User interface agents* would be a more precise way to designate agents in this view; after all,

“interfaces,” also exist between software modules and communicating computers (or between any independent systems). In this paper, however, the term *user agents* is preferred to contrast the user’s view with that of the programmer’s or the network’s in the other two classes of agents. Further user agent subclasses and references to salient research projects involving user agents are presented in Section 3.

2.2 Concept of Autonomy

The motivations for each of the above approaches to agency are different, and constructing agents in each category requires different tools and involves different sets of issues. Although it is possible to have distributed applications based on network agents which also consist of programmer agents, or user agents that are mobile, no one category of agents is necessary for the others to exist. What is common to all three categories is the *perception* of system components as having some degree of *autonomy* – a perception that has been fostered and has resulted in an innovative, useful, and compelling paradigm for computation and communication. The notion of autonomy is also shared by two common definitions for agent: a *teleological* definition in which agents are perceived as having goals that they autonomously try to achieve and a *delegational* definition in which agents are considered to act on behalf of another with some degree of independence.

2.2.1 Autonomy and Perception

Autonomous behavior has to be *perceived* by someone in order for a process to be considered an agent. In the end, autonomy (and hence agency) is subjective and exists only in the mind of the beholder: a process may be regarded as a teleological agent by one person, while someone else may prefer to see a deterministic automaton. In order to realize the potential of agents, it is important to identify those properties of a system that promote its perception as an autonomous entity.

2.2.2 Autonomy as Goal-Directed Behavior

Autonomous entities are often thought of as goal-directed. Covrigaru and Lindsay [1991] distinguish between *achievable goals*, those goals that can be considered “terminated” once a well-defined result or system state is reached, and *homeostatic goals*, those goals that cannot be considered “terminated” once a desired state is reached. In the case of homeostatic goals, when changes in the environment take place, system activity is reinitiated in order to provoke a return to

some equilibrium state. Systems perceived to be autonomous will generally have both homeostatic and achievable goals.

2.2.3 Autonomy as a Social Relation

Autonomy has also been associated with *independence*. In fact, *autonomous* and *independent* are listed as synonyms meaning “not subject to the rule or control of another” [Webster 1989]. However, *total* autonomy in this sense would imply agents existing in isolation, their autonomy becoming irrelevant and uninteresting [Castelfranchi 1995]. Though autonomous, a system must influence or adopt other systems’ goals; autonomy is thus necessarily limited and social. Castelfranchi [1995] presents this argument in the context of programmer agents, but it can be generalized to other categories: network agents should be able to meet with other agents and undergo state changes as a result of exchanged data (other systems’ influences); user agents would be useless if they were not capable of being influenced by and adopting the goals of the end user.

2.3 Terminology

One of the problems in understanding existing approaches to agency is the profusion of terms used to refer to agents. In AI, for example, the term *software agent* has been used by Etzioni [1993] to contrast software-oriented with hardware-oriented approaches to studying AI issues. The term continues to designate an approach to agent-based system construction (programmer agents) that provides assistance to end users; but the term de-emphasizes user interface issues [Etzioni and Weld 1994]. Indeed, a problem with this term is that it lacks specificity, as any computational agents not involving hardware (be them programmer, network or user agents) should be considered software agents.

A different problem occurs with the term *intelligent agent*, which is used sometimes in an all-encompassing way to refer to the study of agents and their applications [ACM 1994; Wooldridge and Jennings 1995b; Wooldridge et al. 1996]. Intelligence (another elusive concept) may be a desirable characteristic for agents. However, autonomous agents and intelligent agents can be thought of as equivalent terms only if autonomy (rather than problem solving ability) is considered *the* key property that defines intelligence.

The Java Language Environment [Sun 1995] and Tcl/Tk [Ousterhout 1994; Rose and Borenstein 1994] are sometimes considered network agency tools (e.g. see [Wayner 1996]). Although both

enable some form of code migration, they currently lack mechanisms for transferring state or for autonomous code migration.

Programs are being developed that recursively “traverse” World Wide Web (WWW) [Berners-Lee et al. 1992] documents, creating indices and facilitating keyword-based document search and retrieval. These programs have been referred to as *web wanderers*, *web robots*, *spiders*, and *web crawlers*. Some names may suggest that program code actually travels and visits nodes in the network; however, this is not generally the case. Web traversal is most commonly implemented as recursive document requests using conventional client/server mechanisms. Koster [1996] has compiled a catalog containing over sixty different programs serving as WWW robots, none of which is implemented as a network agent in the sense defined in this paper.

3. User Agents

The classes of agents defined in Section 2.1 can be used as the basis for more detailed taxonomies. From an applications perspective, user agents can be further refined into three subclasses, which will be termed *information agents*, *task agents*, and *synthetic agents*.

3.1 Information Agents

Information agents help users deal with information spaces that are typically unorganized and highly dynamic. The WWW has presented an attractive environment for testing various implementations of information agents, for example, WebWatcher [Armstrong et al. 1995], Letizia [Lieberman 1995] and LIRA [Balabanovic and Shoham 1995]. WebWatcher provides the user with *potentially useful* links given an information seeking goal. Letizia conducts a breadth-first traversal looking for pages that are related to the current document or to the user’s inferred interests; interesting pages are briefly displayed in suggestion windows. LIRA learns user interests through explicit user feedback and presents a selection of WWW pages matching those interests. Lashkari et al. [1994] have developed a framework in which multiple agents collaborate to assist users in prioritizing and handling e-mail messages. Other agents that assist users in browsing large databases or extracting personalized information include HyperActive agents [Sánchez et al. 1994], Video Navigator [Burke and Hammond 1995], and The Movie Critic [Songline 1995].

3.2 Task Agents

Task agents help users perform computer-supported tasks. These agents run concurrently with user applications, watch user activity and offer to automate certain actions. Two kinds of task agents can be distinguished: *personal agents*, which assist individual users; and *group agents*, which participate in computer-mediated collaborative tasks.

Examples of personal task agents include intelligent tutors such as COACH [Selker 1994] and a note-taking apprentice developed by [Schlimmer and Hermens 1993]. COACH provides personalized advice to students using a computer to learn about specific domains, such as the Lisp programming language or the Unix operating system. Schlimmer and Hermens' note-taking apprentice continuously predicts likely completions for notes taken on a pen-based computer; the user may select the agent's predictions in order to produce faster and more accurate notes.

In contrast, *Dave* [Lakin 1994] is an agent that provides assistance to groups of users working cooperatively. Dave has access to all objects in a large, shared display and becomes one more actor during a performance by intervening from time to time when certain patterns in a special-purpose visual language are recognized. Kautz et al. [1995] have designed an environment in which user agents communicate with each other to locate users who are experts on a given subject, a common problem in collaborative work settings.

3.3 Synthetic Agents

Synthetic agents create engaging environments for users by introducing lifelike characters into the computer interface. They promote a *suspension of disbelief* from the user and provide the illusion of autonomous, animistic entities. Although synthetic agents have been used mainly in the entertainment arena, their impact on general user agent issues deserves careful consideration. Developments in this area are varied. *Julia* [Mauldin 1994] is an enticing agent that participates as an independent player in TINYMUD, a *multi-user dimension* (MUD) [Curtis 1992]. Julia can converse with other players, explore the MUD's "rooms" and objects, assist users in navigating the environment, and answer questions about *herself*, other players, rooms and objects. Bates [1994] has described a small simulated world containing real-time, interactive, self-animating creatures called *Woggles*. Woggles are designed to have individual personalities, display emotions, engage in social behaviors and react to their environment, which includes a fourth Woggle controlled by a human *interactor*. ALIVE [Maes 1995] is an environment in which users can immerse themselves and interact with animated three-dimensional creatures with different behaviors modeled according

to notions from ethology. Other developments and ideas related to synthetic agents may be found in LCC [1995].

3.4 The Context of User Agents

Figure 1 summarizes the taxonomy of agents developed thus far. As noted previously, implementations are possible that rely on all three major classes of agents simultaneously. This taxonomy, however, stresses the importance of the entity that *perceives* and benefits from the notion of agent (i.e. the software developer, the computer network, or the end user).

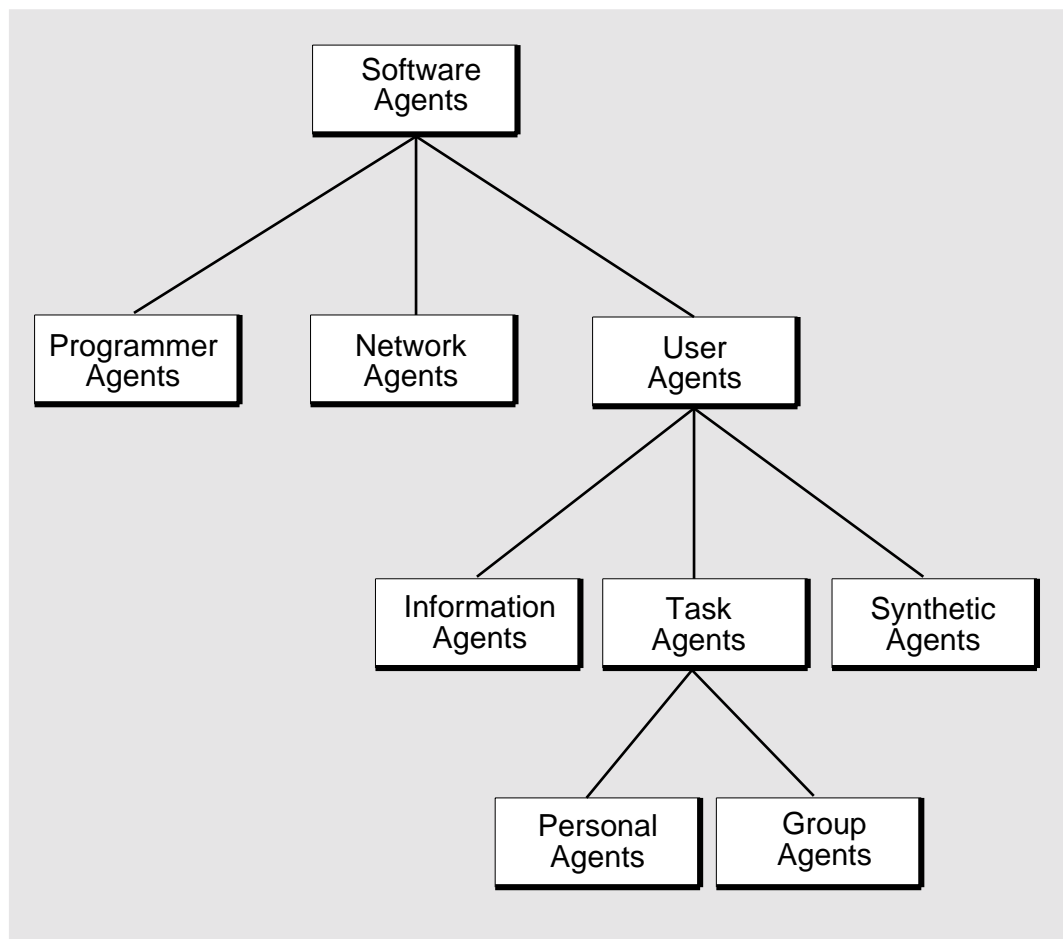


Figure 1. A taxonomy of agents.

3.5 Related Work

There have been other efforts to define and systematize the wide spectrum of research on agents. Unfortunately, definitions and taxonomies have tended to exclude existing views of agents, and

surveys have grouped efforts that are not necessarily related. For example, Shoham [1993, 1995] provides an instructive discussion on agent characteristics, but does not include network agents and regards user agents merely as an application of agent-based computing. Foner [1993] defines “true agents” and provides criteria that programs must pass to be considered such. However, the discussion is limited to synthetic agents. The taxonomy of multi-agent systems proposed by Bird [1993] centers on programmer agents, while the framework presented in [Sánchez et al. 1994] focuses solely on user agents. The special issue of Communications of the ACM devoted to agents [ACM 1994] presents an excellent sample of ongoing research in the area, but no attempt is made to classify or otherwise structure the various projects discussed.

Wooldridge and Jennings [1995a] and Wooldridge et al. [1996] offer what are perhaps the most exhaustive surveys of agency work, and organize existing efforts into three convenient, closely inter-related lines of research: theories, architectures, and languages. This structure provides insight into relevant research issues and has been helpful in constructing the classification proposed here. Their survey, however, concentrates on programmer agents, subscribing to Maes’ definition of user agents (interface agents) as nothing more than “programs that employ artificial intelligence techniques in order to provide assistance to a user dealing with a particular application” [Maes 1994]. They do not discuss the motivations for network agents and include languages for network agent construction (such as Telescript) under the same category as languages such as AGENT0 and PLACA, intended to model programmer agents with mental states and Speech Act Theory-based communication.

4. Characterizing User Agent Developments

A framework for characterization of user agents was proposed in Sánchez [1993] and presented with slight modifications in Sánchez et al. [1994]. In such a framework, which for brevity will subsequently be referred to as *Frame93*, user agent implementations were described according to five major criteria: method of activation, scope of action, location of behavior, type of knowledge, and source of knowledge, as shown in Table I. In what follows, the criteria are briefly reviewed and further refined to allow for a perspective view of the differences and commonalities exhibited by ongoing developments in user agency.

4.1 Method of Activation

User agents may be explicitly activated by humans and other agents, or triggered by the occurrence of certain events in the environment. Frame93 included the possible invocation of user agents by any arbitrary application. Considering the taxonomy of agents just introduced, however, from the end user's point of view it is more natural to limit the study of this design aspect to whether or not a user agent may be activated by other user agents. Activation by arbitrary applications either lies in the realm of programmer agents or can simply be considered event-triggering.

Table I. The *Frame93* characterization framework (adapted from [Sánchez. et al. 1994]).

	Activated by			Scope		Location of behavior			Knowledge Type		Knowledge Source		
	user	apps.	events	local	global	mono-lithic	embe-dded	indep-endent	lexical	seman-tic	progra-mmmer	user	learned
Information Management													
Object Lens [Lai and Malone 1988]	♦	♦	♦	♦				♦	♦			♦	
Khronika [Lövsstrand 1991]			♦		♦	♦			♦			♦	
MAFIA [Lutz et al. 1990]	♦		♦	♦	♦	♦				♦	♦	♦	
HyperNews [Andersen et al. 1989]	♦				♦	♦			♦		♦		
News Readers [Sheth and Maes 1993]	♦			♦				♦	♦			♦	♦
RightPages [Story et al. 1992]			♦	♦		♦			♦			♦	
Work Support													
COKES [Kaye and Karam 1987]		♦	♦	♦	♦	♦				♦	♦	♦	
PAGES [Mäntylä et al. 1990]			♦	♦	♦		♦			♦	♦	♦	
ProMInanD [Karbe and Ramsperger 1990]			♦	♦	♦		♦			♦	♦		
D-POLYMER [Bhandaru and Croft 1990]	♦		♦		♦	♦				♦	♦		
MCS [Woitass 1990]	♦		♦	♦	♦			♦		♦	♦	♦	
Learning Agents [Kozierok and Maes 1993]			♦	♦	♦			♦		♦		♦	♦
Hypertext Apprentice [Bernstein 1990]	♦			♦		♦			♦		♦		
HyperBase Agents [Littleford 1991]			♦		♦		♦		♦		♦		
Eager [Cypher 1991]	♦		♦	♦		♦				♦	♦		♦
Learning													
Guides [Oren et al. 1990]	♦		♦	♦				♦	♦		♦		
Evocative Agents [Adelson 1992]	♦			♦				♦	♦			♦	

4.2 Scope of Actions

A user agent may act locally if it considers only the individual user needs and preferences, or globally, if the agent can utilize knowledge about the organization, other users, and other agents in performing its mission. In order to provide personalized services, agents typically rely on individual information such as preference profiles or keywords. Agents such as *Dave* or the *Expertise Locators* mentioned previously benefit from global knowledge in the environment and from other participating agents for their operation.

4.3 Architecture (Location of Behavior)

Frame93 suggested *location of behavior* as a criterion to identify the architectural relationship between the behavioral components of user agents and a data repository. Agents could thus be a monolithic process acting as a mediator between users and the data, a collection of user-selectable processes, or behavioral elements distributed among objects in a database. In the light of the proposed taxonomy, this criterion may be appropriate for characterizing information agents, but presents problems for describing agent operation that is not necessarily dependent on large, existing data repositories (task and synthetic agents). There has been a trend to develop agents as an enhancement to existing environments and interfaces, as illustrated by the various information and task agents mentioned in the previous section. These agents typically operate side-by-side with the existing system and represent an option for the users to augment their potential to utilize conventional environments. However, for special-purpose agents such as COACH, or for the exploration of new concepts (as in the case of synthetic agents), integrated, independent architectures and environments have been constructed in which agents operate. It is useful then to consider a criterion that will be termed *Architecture* and that characterizes agents according to the approach followed in their construction. Architectures in the first group will be referred to as *augmenting architectures*, while those in the second group will be termed *integrated architectures*.

4.4 Type of Knowledge

Agent implementations have utilized techniques relying on *lexical* manipulation of information and on the use of *semantic* representations to model their world. It is interesting to note that most information agents have strongly relied on lexical techniques from the field of Information Retrieval. Agents in the other subclasses, however, have made some use of semantical representations. For example, Video Navigator uses semantic networks to model the relationships

between film directors and their influences, and Julia semantically encodes the domain-specific information used in conversations.

4.5 Source of Knowledge

Finally, user agents have relied to different extents on knowledge directly encoded by programmers, knowledge supplied by the user in the form of feedback or initial profiles, and knowledge inferred from observing the user's actions. A summary of the updated framework for characterization of user agents, including each of the projects referenced in this section, is presented in Table II. A more detailed review of each of the research projects included in the table is offered in Sánchez [1996].

Table II. Characterizing work on user agents.

	Activated by			Scope		Architecture		Knowledge Type		Knowledge Source		
	user	agents	events	local	global	integrated	augmenting	lexical	semantic	programmer	user	learned
Information Agents												
WebWatcher [Armstrong et al. 1995]	♦		♦	♦			♦	♦		♦	♦	♦
Letizia [Lieberman 1995]	♦		♦	♦			♦	♦				♦
LIRA [Balabanovic and Shoham 1995]	♦			♦			♦	♦			♦	
Collab. Mail Agents [Lashkari et al. 1994]	♦	♦	♦	♦		♦	♦	♦	♦	♦	♦	♦
Video Navigator [Burke and Hammond 1995]	♦		♦	♦	♦	♦		♦	♦	♦	♦	
Task Agents												
Note Taker [Schlimmer and Hermens 1993]	♦		♦	♦			♦	♦		♦		♦
COACH [Selker 1994]	♦		♦	♦		♦			♦	♦		♦
Dave [Lakin 1994]	♦		♦		♦		♦		♦	♦		
Expertise Locators [Kautz et al. 1995]	♦	♦		♦	♦		♦	♦		♦		
Synthetic Agents												
Julia [Mauldin 1994]	♦	♦	♦		♦		♦	♦	♦	♦	♦	
Woggles [Bates 1994]	♦	♦	♦		♦	♦			♦	♦		
ALIVE Agents [Maes 1995]	♦	♦	♦		♦	♦			♦	♦		

References

- ACM. 1994. Special issue on intelligent agents. *Commun. ACM* 37, 7 (July).
- Adelson, B. 1992. Evocative agents and multi-media interface design. In *Proceedings of the CHI '92 Conference on Human Factors in Computing Systems*, P. Bauersfeld, J. Bennett, and G. Lych, Eds. ACM Press, New York, N.Y., 351-356.
- Agha, G. 1986. *Actors: A Model of Concurrent Computation in Distributed Systems*. The MIT Press, Cambridge, Mass.
- Andersen, M., Nielsen, J., and Rasmussen, H. 1989. A similarity-based hypertext browser for reading the Unix network news. *Hypermedia* 1, 3, 255-265.
- Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T. 1995. WebWatcher: A learning apprentice for the World Wide Web. In *Information Gathering from Heterogeneous, Distributed Environments: Papers from the 1995 AAAI Spring Symposium* (Menlo Park, Calif., March), C. Knoblock and A. Levy, Eds. AAAI Press, Menlo Park, Calif., 6-12.
- Azevedo, F., and Leggett, J. 1995. AcTool: An agent building environment. Tech. Rep. TAMU-HRL 95-003, Hypermedia Research Lab, Department of Computer Science, Texas A&M University, College Station, Tex.
- Balabanovic, M., and Shoham, Y. 1995. Learning information retrieval agents: Experiments with automated web browsing. In *Information Gathering from Heterogeneous, Distributed Environments: Papers from the 1995 AAAI Spring Symposium* (Menlo Park, Calif., March), C. Knoblock and A. Levy, Eds. AAAI Press, Menlo Park, Calif., 13-18. (Also available from <http://www.isi.edu/sims/knoblock/sss95/proceedings.html>.)
- Bates, J. 1994. The role of emotion in believable agents. *Commun. ACM* 37, 7 (July), 122-125.
- Bell, G., Parisi, A., and Pesce, M. 1995. The Virtual Reality Modeling Language version 1.0 specification. Available from the authors: gavin@sgi.com or mpesce@hyperreal.com. (Also available from <http://webpace.sgi.com/Archive/Spec1.0/index.html>.)
- Berners-Lee, T., Cailliau, R., Gorff, J., and Pollerman B. 1992. World-Wide Web: The information universe. *Electronic Networking: Research Applications and Policy* 2, 1 (Spring), 52-58.
- Berners-Lee, T., Masinter, L., and McCahill, M. 1994. Uniform Resource Locators (URL). Internet Request for Comments No. RFC 1738. Available from InterNIC Directory and Database Services, AT&T, 5000 Hadley Road, Room 1B13 South Plainfield, N.J. (Also available from [gopher://ds0.internic.net:70](http://ds0.internic.net:70).)
- Birmingham, W., Drabenstott, K., Frost, C., Warner, A., and Willis, K. 1994. The University of Michigan digital library: This is not your father's library. In *Proceedings of Digital Libraries '94* (College Station, Tex., June). Hypermedia Research Laboratory, Texas A&M University, College Station, Tex., 53-60.
- Bernstein, M. 1990. An apprentice that discovers hypertext links. In *Hypertext: Concepts, Systems, and Applications. Proceedings of the European Conference on Hypertext* (Versailles, France, Nov.), 212-223.

- Bhandaru, N., and Croft, B. 1990. An architecture for supporting goal-based cooperative work. In *Multi-User Interfaces and Applications*. S. Gibbs and A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, The Netherlands, 337-354.
- Bird, S. 1993. Towards a taxonomy of multi-agent systems. *International Journal of Man-Machine Studies* 39, 689-704.
- Brooks, R. 1991a. Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence* (San Mateo, Calif.). Morgan Kaufmann, San Francisco, Calif., 569-595.
- Brooks, R. 1991b. Intelligence without representation. *Artificial Intelligence* 47, 1-3 (Jan.), 139-159.
- Burke, R., and Hammond, K. 1995. Combining databases and knowledge bases for assisted browsing. In *Information Gathering from Heterogeneous, Distributed Environments: Papers from the 1995 AAAI Spring Symposium* (Menlo Park, Calif., March), C. Knoblock and A. Levy, Eds. AAAI Press, Menlo Park, Calif., 25-29.
- Cardelli, L. 1995. A language with distributed scope. *Computing Systems* 8, 1 (Winter), 27-59. (Also available from <http://www.research.digital.com/SRC/Obliq/Obliq.ps>.)
- Castelfranchi, C. 1995. Guarantees for autonomy in cognitive agent architecture. In *Intelligent Agents*, M. Wooldridge and N. Jennings, Eds. Springer-Verlag, New York, N.Y., 56-70.
- Cheong, F.-C. 1996. *Internet Agents—Spiders, Wanderers, Brokers and Bots*. New Riders, New York, N.Y.
- Cheong, F.-C. 1992. Oasis: An agent-oriented programming language for heterogeneous distributed environment. Ph.D. dissertation. Department of Computer Science and Engineering, University of Michigan, Ann Arbor, Mich. (Also available from <file://ftp.eecs.umich.edu/software/oasis/THESIS.tar.Z>.)
- Chess, D., Grosz, B., Harrison, C., Levine, D., Parris, C., and Tsudik, G. 1995. Itinerant agents for mobile computing. IBM Research Report No. RC20010, IBM T. J. Watson Research Center, Yorktown Heights, N. Y. (Also available from <http://www.research.ibm.com/xw-D953/rc20010.ps>.)
- Coen, M. 1994. SodaBot: A software agent environment and construction system. Tech. Rep. 1493, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Mass.
- Cohen, P., Cheyer, A., Wang, M., and Baeg, S. 1994. An open agent architecture. In *Software Agents: Papers from the 1994 Spring Symposium* (Menlo Park, Calif., March). AAAI Press, Menlo Park, Calif., 1-7.
- Covrigaru, A., and Lindsay, R. 1991. Deterministic autonomous systems. *AI Magazine* 12 (Fall), 110-117.
- Curtis, P. 1992. Mudding: Social phenomena in text-based virtual realities. In *Proceedings of DIAC92: Directions and Implications of Advanced Computing* (Berkeley, Calif., May). Computer Professionals for Social Responsibility, Palo Alto, Calif., 1-21. (Also available from <ftp://parcftp.xerox.com/pub/MOO/papers/DIAC92.ps>.)

- Crane, G. 1996. Building a digital library: The Perseus Project as a case study in the humanities. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 3-10.
- Cypher, A. 1991. Eager: Programming repetitive tasks by example. In *Proceedings of the CHI '91 Conference on Human Factors in Computing Systems* (New Orleans, La., April), 33-39.
- Demazeau, Y., and Müller, J.-P. (Eds.). 1990. *Decentralized A.I.* Elsevier Science Publishing Company, New York, N.Y.
- Dennett, D. 1987. *The Intentional Stance*. MIT Press, Cambridge, Mass.
- Etzioni, O. 1993. Intelligence without robots (a reply to Brooks). *AI Magazine* 14, 4 (Winter), 7-13.
- Etzioni, O. and Weld, D. 1994. A softbot-based interface to the Internet. *Commun. ACM* 37, 7 (July), 72-79.
- Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McGuire, J., Shapiro, S., and Beck, C. 1993. Specification of the KQML agent communication language. Tech. Rep. EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, Calif.
- Foner, L. 1993. What's an agent, anyway? A sociological case study. Agents Memo 93-01, MIT Media Lab, E15-305 20 Ames St., Cambridge, Mass.
- Fox, E., Akscyn, R., Furuta, R., and Leggett, J. 1995. Digital libraries. *Commun. ACM* 38, 4 (April), 23-28.
- Fox, E., Hix, D., Nowell, L., Brueni, D., Wake, W., and Heath, L. 1993. Users, user interfaces, and objects: Envision, a digital library. *Journal of the American Society for Information Science* 44, 8 (Sept.), 480-491.
- Furuta, R., Marshall, C., Shipman, F., and Leggett, J. 1996. Physical objects in the digital library. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 109-115.
- Gasser, L., and Huhns, M., (Eds.). 1989. *Distributed Artificial Intelligence: Volume II*. Morgan Kaufmann Publishers, Inc., San Mateo, Calif.
- Genesereth, M., and Ketchpel, S. 1994. Software agents. *Commun. ACM* 37, 7 (July), 47-53.
- Gray, R. 1995. Agent Tcl: Alpha Release 1.1. Draft available from Department of Computer Science, Dartmouth College, Hanover, N.H. (Also available from <ftp://bald.cs.dartmouth.edu/pub/agents/doc.1.1.ps.gz>.)
- Gray, R., Rus, D., and Kotz, D. 1996. Transportable information agents. Tech. Rep. PCS-TR95-26, Department of Computer Science, Dartmouth College, Hanover, N.H. (Also available from <http://www.cs.dartmouth.edu/pub/CS-techreports/TR96-278.ps.Z>.)
- Hanks, S., Pollack, M., and Cohen, P. 1993. Benchmarks, testbeds, controlled experimentation, and the design of agent architectures. *AI Magazine* 14, 4 (Winter), 17-42.

- IUG. 1995. *Illustra User's Guide*. Release 3.2. Illustra Information Technologies, Inc., Oakland, Calif.
- Johansen, D., van Renesse, R., and Schneider, F. 1995. An introduction to the TACOMA distributed system. Tech. Rep. 95-23, Institute of Mathematical and Physical Sciences, Department of Computer Science, University of Tromsø, Norway. (Also available from <http://www.cs.uit.no/Lokalt/Rapporter/Reports/9523.html>.)
- Kahn, R., and Wilensky, R. 1995. A framework for distributed digital object services. Tech. Rep. TN95-01, Corporation for National Research Initiatives, Reston, Va.
- Karbe, B., and Ramsperger, N. 1990. Influence of exception handling on the support of cooperative work. In *Multi-User Interfaces and Applications*. S. Gibbs and A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, The Netherlands, 355-370.
- Kautz, H., Milewski, A., and Selman, B. 1995. Agent amplified communication. In *Information Gathering from Heterogeneous, Distributed Environments: Papers from the 1995 AAAI Spring Symposium* (Menlo Park, Calif., March), C. Knoblock and A. Levy, Eds. AAAI Press, Menlo Park, Calif., 78-84.
- Kay, A. 1990. User interface: A personal view. In *The Art of Human-Computer Interface Design*, Brenda Laurel, Ed. Addison-Wesley, Reading, Mass., 191-207.
- Kay, A. 1984. Computer software. *Scientific American* 251, 3 (Sept.), 52-59.
- Kaye, A., and Karam, G. 1987. Cooperating knowledge-based assistants for the office. *ACM Trans. Off. Inf. Syst.* 5, 4 (Oct.), 332-353.
- Koster, M. (Ed.). 1996. List of robots. Available from editor: m.koster@webcrawler.com. (Also available from <http://info.webcrawler.com/mak/projects/robots/active.html>.)
- Kozierok, R., and Maes, P. 1993. A learning interface agent for scheduling meetings. In *Proceedings of the ACM-SIGCHI International Workshop on Intelligent User Interfaces* (Orlando, Fla., Jan.). ACM Press, New York, N.Y., 81-88.
- Lai, K., and Malone T. 1988. Object Lens: A "spreadsheet" for cooperative work. *ACM Trans. Off. Inf. Syst.* 6, 4 (Oct.), 332-353.
- Lakin, F. 1994. A visual agent for performance graphics. In *Software Agents: Papers from the 1994 AAAI Spring Symposium* (Menlo Park, Calif., March). AAAI Press, Menlo Park, Calif., 103-106.
- Lashkari, Y., Metral, M., and Maes, P. 1994. Collaborative interface agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI '94)* (Seattle, Wash., Aug.). AAAI Press, Menlo Park, Calif., 444-450.
- Laurel, B. 1991. *Computers as Theatre*. Addison-Wesley, Reading, Mass.
- Laurel, B. 1990. Interface agents: Metaphors with character. In *The Art of Human-Computer Interface Design*, Brenda Laurel, Ed. Addison-Wesley, Reading, Mass., 355-365.

- LCC. 1995. *Lifelike Computer Characters '95 Workshop Proceedings* (Snowbird, Utah, Sept.). Microsoft Research, Redmond, Wash. (Also available from <http://www.research.microsoft.com/lcc.htm>.)
- Lee, K., Mansfield, W., and Sheth, A. 1993. A framework for controlling cooperative agents. *IEEE Comput.* 26, 7 (July), 8-16.
- Leggett, J., Nürnberg, P., and Schneider, E. 1995. Principles of human-centered information systems. In *Proceedings of the International Conference on Artificial Reality and Tele-Existence/Conference on Virtual Reality Software and Technology (ICAT/VRST) '95* (Makuhari Messe, Chiba, Japan, Nov.), 173-176.
- Leggett, J., and Schnase J. 1994. Viewing Dexter with open eyes. *Commun. ACM* 37, 2 (Feb.), 76-86.
- Lieberman, H. 1996. Personal communication. Media Lab, Massachusetts Institute of Technology, Cambridge, Mass. (March).
- Littleford, A. 1991. Artificial intelligence and hypermedia. In *Hypertext/Hypermedia Handbook*, E. Berk and J. Devlin, Eds. McGraw-Hill, New York, N.Y., 357-378.
- Lövstrand, L. 1991. Being selectively aware with the Khronika system. In *Proceedings of the Second European Conference on Computer-Supported Cooperative Work* (Amsterdam, The Netherlands, Sept.), 265-277.
- Lutz, E., Kleist-Retzow, H., and Hoernig, K. 1990. MAFIA – An active mail-filter-agent for an intelligent document processing support. In *Multi-User Interfaces and Applications*. S. Gibbs and A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, The Netherlands, 235-252.
- Maes, P. 1995. Artificial Life meets entertainment: Lifelike autonomous agents. *Commun. ACM* 38, 11 (Nov.), 108-114.
- Maes, P. 1994. Modeling adaptive autonomous agents. *Artificial Life* 1, 1/2 (Fall/Winter), 135-162.
- Mäntylä, R., Alasuvanto, J., and Hämmäinen, H. 1990. PAGES: A testbed for groupware applications. In *Multi-User Interfaces and Applications*. S. Gibbs and A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, The Netherlands, 37-47.
- Mauldin, M. 1994. Chatterbots, tinymuds, and the Turing test: Entering the Loebner Prize competition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI '94)* (Seattle, Wash., August). AAAI Press, Menlo Park, Calif., 16-21.
- McCabe, F., and Clark, K. 1995. April - Agent PRocess Interaction Language. In *Intelligent Agents*, M. Wooldridge and N. Jennings, Eds. Springer-Verlag, New York, N.Y., 355-370.
- Miller, M., and Drexler, K. 1988. Markets and computation: Agoric open systems. In *The Ecology of Computation*, B. A. Huberman, Ed. Elsevier Science Publishers, Amsterdam, The Netherlands, 133-176.

- Negroponte, N. 1995. *Being Digital*. Alfred A. Knopf, New York, N.Y.
- Negroponte, N. 1990. Hospital Corners. In *The Art of Human-Computer Interface Design*, Brenda Laurel, Ed. Addison-Wesley, Reading, Mass., 347-353.
- Nigay, L., and Coutaz, J. 1992. PAC-Expert: Towards an automatic generation of dialogue controllers. Tech. Rep. RT 83, Laboratoire de Génie Informatique—IMAG, University of Grenoble, France.
- Norman, D. 1994. How might people interact with agents. *Commun. ACM* 37, 7 (July), 68-71.
- North, C., Shneiderman, B., and Plaisant, C. 1996. User controlled overviews of an image library: A case study of the visible human. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 74-82.
- Nürnberg, P. 1994. Implications of an open, extensible, and distributed hypermedia information system architecture for inter-process communication subsystem design. M.S. thesis, Department of Computer Science, Texas A&M University, College Station, Tex., December.
- Nürnberg, P., Furuta, R., Leggett, J., Marshall, C., and Shipman, F. 1995. Digital libraries: Issues and architectures. In *Proceedings of Digital Libraries '95* (Austin, Tex., June). Center for the Study of Digital Libraries, Texas A&M University, College Station, Tex., 147-153.
- Nürnberg, P., and Leggett, J. 1996. The HOSS inter-process communication facilities. Submitted to *Journal of Network and Computer Applications* (Available from the authors: pnuern@cSDL.tamu.edu.)
- Oren, T., Salomon, G., Kreitman, K., and Don, A. 1990. Guides: Characterizing the interface. In *The Art of Human-Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, Reading, Mass., 367-381.
- Orendorf, J., and Kacmar, C. 1996. A spatial approach to organizing and locating digital libraries and their content. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 83-89.
- Ousterhout, J. 1994. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, Mass.
- Peine, H. 1995. Lara reference manual). Draft available from University of Kaiserslautern, P.O. Box 3049, D-67653, Kaiserslautern, Germany. (Also available from <http://www.uni-kl.de/AG-Nehmer/Ara>.)
- Phelps, T., and Wilensky, R. 1996. Toward active, extensible, networked documents: Multivalent architecture and applications. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 100-108.
- Rao, R., Pedersen, J., Hearst, M., Mackinlay, J., Card, S., Masinter, L., Halvorsen, P.-K., and Robertson, G. 1995. Rich interaction in the digital library. *Commun. ACM* 38, 4 (Apr.), 29-39.
- Rose, M., and Borenstein, N. 1994. A model for enabled mail (EM). Draft available from the authors: nsb@nsb.fv.com or mrose@dbc.mtview.ca.us. (Also available from <http://oneworld.wa.com:80/laurence/enabled.html>.)

RPC. 1988. Remote Procedure Call protocol specification, version 2. Internet Request for Comments RFC 1057. Available from InterNIC Directory and Database Services, AT&T, 5000 Hadley Road, Room 1B13 South Plainfield, N.J. (Also available from <http://ds.internic.net/rfc/rfc1057.txt>.)

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, N.J.

Sánchez, J. A. 1996. Agent services. Ph.D. Dissertation. Department of Computer Science, Texas A&M University, College Station, Tex., August.

Sánchez, J. A. 1995. The Object Manager 4.2 interface specification. Draft available from Center for the Study of Digital Libraries, Texas A&M University, College Station, Tex. (Also available from <http://www.csdl.tamu.edu/~joseash/om4.2.html>.)

Sánchez, J. A. 1994. User agents in the interface to digital libraries. In *Proceedings of Digital Libraries '94* (College Station, Tex., June). Hypermedia Research Laboratory, Texas A&M University, College Station, Tex., 217-218.

Sánchez, J. A. 1993. HyperActive: Extending an open hypermedia architecture to support agency. M.S. thesis. Department of Computer Science, Texas A&M University, College Station, Tex., December.

Sánchez, J. A., Azevedo, F. S., and Leggett, J. J. 1995. PARAgente: Exploring the issues in agent-based user interfaces. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (San Francisco, Calif., June), 320-327.

Sánchez, J. A., Leggett, J. J., and Schnase, J. L. 1994. Hyperactive: Extending an open hypermedia architecture to support agency. *ACM Transactions on Computer-Human Interaction* 1, 4 (Dec.), 357-382.

Schlimmer, J., and Hermens, L. 1993. Software agents: Completing patterns and constructing user interfaces. *Journal of Artificial Intelligence Research* 1 (Nov.), 61-89.

Schatz, B., Johnson, E., Cochrane, P., and Chen H. 1996. Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 126-133.

Searle, J. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, England.

Selker, T. 1994. Coach: A teaching agent that learns. *Commun. ACM* 37, 7 (July), 92-99.

Sheth, B., and Maes, P. 1993. Evolving agents for personalized information retrieval. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence and Applications* (Orlando, Fla., March), 1-11.

Shneiderman, B. 1993. Beyond intelligent machines: Just do it! *IEEE Software* 10, 1 (Jan.), 100-103.

- Shoham, Y. 1995. What we talk about when we talk about agents. Invited talk at the *First International Conference on Multi-agent Systems (ICMAS-95)* (San Francisco, Calif., June). Abstract available from <http://centaurus.cs.umass.edu:80/icmas/invited-speakers.html>.
- Shoham, Y. 1993. Agent-oriented programming. *Artificial Intelligence* 60, 1 (March), 51-92.
- Songline. 1995. The Movie Critic. Software available from Songline Studios, Sebastopol, Calif. (Also available from <http://www.moviecritic.com>.)
- Story, G., O’Gorman, L., Fox, D., Schaper, L., and Jagadish, H. 1992. The RightPages image-based electronic library for alerting and browsing. *IEEE Comput.* 25, 9 (Sept.), 17-26.
- Strzalkowski, T., Perez-Carballo, J., and Marinescu, M. 1996. Natural language information retrieval in digital libraries. In *Proceedings of the 1st ACM International Conference on Digital Libraries* (Bethesda, Md., March). ACM Press, New York, N.Y., 117-125.
- Sun. 1995. *The Java Language Environment*. White paper available from Sun Microsystems Computer Co., Mountain View, Calif. (Also available from <http://java.sun.com/whitePaper/java-whitepaper-1.html>.)
- Thomas, S. 1995. The PLACA Agent Programming Language. In *Intelligent Agents*, M. Wooldridge and N. Jennings, Eds. Springer-Verlag, New York, N.Y., 355-370.
- Wayner, P. 1996. Net programming for the masses. *Byte* 22, 2 (Feb.) 101-104.
- Webster. 1989. *Webster’s Ninth New Collegiate Dictionary*. Merriam-Webster Inc. Publishers, Springfield, Mass.
- White, J. 1996. Mobile agents. In *Software Agents*, J. Bradshaw, Ed. AAAI Press/The MIT Press, Menlo Park, Calif.
- Woitass, M. 1990. Coordination of intelligent office agents applied to meeting scheduling. In *Multi-User Interfaces and Applications*. S. Gibbs and A. Verrijn-Stuart, Eds. North- Holland, Amsterdam, The Netherlands, 371-384
- Wooldridge, M., and Jennings, N. 1995a. Agent theories, architectures and languages: A survey. In *Intelligent Agents*, M. Wooldridge and N. Jennings, Eds. Springer-Verlag, New York, N.Y., 1-39.
- Wooldridge, M., and Jennings, N. (Eds.). 1995b. *Intelligent Agents*. Springer-Verlag, New York, N.Y.
- Wooldridge, M., Müller, J., and Tambe, M. (Eds.). 1996. *Intelligent Agents II*. Springer-Verlag, New York, N.Y.

Table of Contents

1. Introduction	2
2. Towards a Taxonomy of Agents	3
2.1 Classes of Agents	3
2.2 Concept of Autonomy	6
2.3 Terminology	7
3. User Agents	8
3.1 Information Agents	8
3.2 Task Agents	9
3.3 Synthetic Agents	9
3.4 The Context of User Agents	10
3.5 Related Work	10
4. Characterizing User Agent Developments	11
4.1 Method of Activation	11
4.2 Scope of Actions	13
4.3 Architecture (Location of Behavior)	13
4.4 Type of Knowledge	13
4.5 Source of Knowledge	14
References	15