Is it an Agent, or just a Program?:
A Taxonomy for Autonomous Agents

Stan Franklin and Art Graesser
Institute for Intelligent Systems
University of Memphis

Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

Abstract

The advent of software agents gave rise to much discussion of just what such an agent is, and of how they differ from programs in general. Here we propose a formal definition of an autonomous agent which clearly distinguishes a software agent from just any program. We also offer the beginnings of a natural kinds taxonomy of autonomous agents, and discuss possibilities for further classification. Finally, we discuss subagents and multiagent systems.

Introduction

On meeting a friend or colleague that we haven't seen for a while, or a new acquaintance, some version of the following conversation often ensues:

What are you working on these days?
Control structures for autonomous agents.
Autonomous agents? What do you mean by that?

A brief explanation is then followed by:

But agents sound just like computer programs. How are they different?

This elicits a more satisfying explanation that distinguishes between agent and program. The nature of this "more satisfying explanation" motivates this essay. After a review of some of the many ways the term "agent" has been used within the context of autonomous agents, we'll propose and defend a notion of agent that is clearly distinct from a program. This discussion will lead us to a discussion of possible classifications for autonomous agents.

What is an agent?

Workers involved in agent research have offered a variety of definitions, each hoping to explicate his or her use of the word "agent." These definitions range from the simple to the lengthy and demanding. We suspect that each of them grew directly out of the set of examples of agents that the definer had in mind. (This is certainly the case for the version we'll propose below.) Let's orient ourselves by examining and comparing some of these definitions.

The MuBot Agent [http://www.crystaliz.com/logicware/mubot.html] *"The term agent is used to represent two orthogonal concepts. The first is the agent's ability for autonomous execution. The second is the agent's ability to perform domain oriented reasoning."*P> This pointer at definitions come from an online white paper by Sankar Virdhagriswaran of Crystaliz, Inc.,

defining mobile agent technology. Autonomous execution is clearly central to agency.

The AIMA Agent [Russell and Norvig 1995, page 33] *"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors."*

AIMA is an acronym for "Artificial Intelligence: a Modern Approach," a remarkably successful new AI text that was used in 200 colleges and universities in 1995. The authors were interested in software agents embodying AI techniques. Clearly, the AIMA definition depends heavily on what we take as the environment, and on what sensing and acting mean. If we define the environment as whatever provides input and receives output, and take receiving input to be sensing and producing output to be acting, every program is an agent. Thus, if we want to arrive at a useful contrast between agent and program, we must restrict at least some of the notions of environment, sensing and acting.

The Maes Agent [Maes 1995, page 108] *"Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed."*

Pattie Maes, of MIT's Media Lab, is one of the pioneers of agent research. She adds a crucial element to her definition of an agent: agents must act autonomously so as to "realize a set of goals." Also environments are restricted to being complex and dynamic. It's not clear whether this rules out a payroll program without further restrictions.

The KidSim Agent [Smith, Cypher and Spohrer 1994] *"Let us define an agent as a persistent software entity dedicated to a specific purpose. 'Persistent' distinguishes agents from subroutines; agents have their own ideas about how to accomplish tasks, their own agendas. 'Special purpose' distinguishes them from entire multifunction applications; agents are typically much smaller."*

The authors are with Apple. The explicit requirement of persistence is a new and important addition here. Though many agents are "special purpose" we suspect this is not an essential feature of agency.

The Hayes-Roth Agent [Hayes-Roth 1995] *Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.*

Barbara Hayes-Roth of Stanford's Knowledge Systems Laboratory insists that agents reason during the process of action selection. If reasoning is interpreted broadly, her agent architecture does allow for reflex actions as well as planned actions.

The IBM Agent [http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm] *"Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires."*

This definition, from IBM's Intelligent Agent Strategy white paper, views an intelligent agent as acting for another, with authority granted by the other. A typical example might be an information gathering agent, though the white paper talks of eight possible applications. Would

you stretch "some degree of independence" to include a payroll program? What if it called itself on a certain day of the month?

The Wooldridge&shyp;Jennings Agent [Wooldridge and Jennings 1995, page 2] *"... a hardware or (more usually) software-based computer system that enjoys the following properties:*

- *autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;*
- *social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;*
- *reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;*
- *pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavour by taking the initiative."*

The Wooldridge and Jennings definition, in addition to spelling out autonomy, sensing and acting, allows for a broad, but finite, range of environments. They further add a communications requirement. What would be the status of a payroll program with a graphical interface and a decidedly primitive communication language?

The SodaBot Agent [Michael Coen http://www.ai.mit.edu/people/sodabot/slideshow/total/P001.html] *"Software agents are programs that engage in dialogs [and] negotiate and coordinate transfer of information."*

SodaBot is a development environment for software agent being constructed at the MIT AI Lab by Michael Coen. Note the apparently almost empty intersection between this definition and the preceding seven. we say "apparently" since negotiating, for example, requires both sensing and acting. And dialoging requires communication. Still the feeling of this definition is vastly different from the first few, and would seem to rule out almost all standard programs.

The Foner Agent [Lenny Foner - Download from ftp://media.mit.edu/pub/Foner/Papers/Julia/Agents--Julia.ps or online at http://foner.www.media.mit.edu/people/foner/Julia/ (click on "What's an agent? Crucial notions")]

Foner requires much more of an agent. His agents collaborate with their users to improve the accomplishment of the users' tasks. This requires, in addition to autonomy, that the agent dialog with the user, be trustworthy, and degrade gracefully in the face of a "communications mismatch." However, this quick paraphrase doesn't do justice to Foner's analysis.

The Brustoloni Agent [Brustoloni 1991, Franklin 1995, p. 265] *"Autonomous agents are systems capable of autonomous, purposeful action in the real world."*

The Brustoloni agent, unlike the prior agents, must live and act "in the real world." This definition excludes software agents and programs in general. Brustoloni also insists that his agents be "reactive &shyp; that is, be able to respond to external, asynchronous stimuli in a timely fashion."

As these definitions make clear, there's no general agreement as to what constitutes an agent, or as to how agents differ from programs. The Software Agents Mailing List on the Internet provides a FAQ (frequently asked questions) that says,

The FAQ Agent [http://www.ee.mcgill.ca:80/~belmarc/agent_faq.html] *"This FAQ will not attempt to provide an authoritative definition ..."*

It does provide a list of attributes often found in agents: Autonomous, goal-oriented, collaborative, flexible, self-starting, temporal continuity, character, communicative, adaptive, mobile, [Etzioni and Weld]. Several of these would seem to rule out our payroll program.

The Essence of Agency

We normally avoid prescriptive arguments about how a word should be used. Russell and Norvig put it this way: "The notion of an agent is meant to be a tool for analyzing systems, not an absolute characterization that divides the world into agents and non-agents." [1995, page 33] The only concepts that yield sharp edge categories are mathematical concepts, and they succeed only because they are content free. Agents "live" in the real world (or some world), and real world concepts yield fuzzy categories.

Nevertheless, we will propose a mathematical style definition of an autonomous agent, knowing full well that it must fail around the edges. Our definition attempts to capture the essence of being an agent, and to define the broadest class of agents. Further restrictions can then be added to define more particular classes of agents. Ideally, such an endeavor would produce a nomenclature of agents that could be used relatively unambiguously by researchers in the field, resulting in clearer communications.

The definitions of the previous section seem to derive from one or both of two common uses of the word agent: 1) one who acts, or who can act, and 2) one who acts in place of another with permission. Since "one who acts in place of " acts, the second usage requires the first. Hence, let's go for a definition of the first notion.

What are examples of agents in this first sense upon which we can build our mathematical style definition? Well, humans act, as do most other animals. (I say most since some animals act during a portion of their lives and not during others, for example the sea squirt [Dethie 1986].) Also, some autonomous mobile robots act, for example Brooks' Herbert [Brooks 1990, p. 8; Franklin 1995, p263-5]. All of these are real world agents. Software agents "live" in computer operating systems, databases, networks, MUDs, etc. Almost all the definitions in the previous section refer to software agents. Finally, artificial life agents "live" in artificial environments on a computer screen or in its memory [Langton 1989, Franklin 1995, pp. 185-208]. What do these agents share that constitutes the essence of being an agent?

Each is situated in, and is a part on some environment. Each senses its environment and act autonomously upon it. No other entity is required to feed it input, or to interpret and use its output. Each acts in pursuit of it's own agenda, whether satisfying evolved drives as in humans and animals, or pursuing goals designed in by some other agent, as in software agents. (Artificial life agents may be of either variety.) Each acts so that its current actions may effect its later sensing, that is its actions effect its environment. Finally, each acts continually over some period of time. A software agent, once invoked, typically runs until it decides not to. An artificial life agent often runs until it's eaten or otherwise dies. Of course, some human can pull the plug, but not always. Mobile agents on the Internet may be beyond calling back by the user.

To us, these requirements constitute the essence of being an agent. Let's formalize them into a definition.

*An **autonomous agent** is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*

One way of clarifying the boundaries of this definition is by looking at extreme cases. Humans and some animals are at the high end of being an agent, with multiple, conflicting drives, multiples senses, multiple possible actions, and complex sophisticated control structures (minds [Franklin 1995]) . At the low end, with one or two senses, a single action, and an absurdly simple control structure (mind?) we find a thermostat. A thermostat? Yes, a thermostat satisfies all the requirements of the definition, as does a bacterium. Strange things sometimes happen at the extremes. Espousing a definition entails these risks.

Our definition yields a large and varied class of agents as was to be expected of one requiring only the essence. No doubt it's too large to be useful as is. Adding additional requirements for different purposes will produce useful subclasses of agents. We'll discuss some of these in the next section. But first, there are a couple of basic points to clarify.

Autonomous agents are situated in some environment. Change the environment and we may no longer have an agent. A robot with only visual sensors in an environment without light is not an agent. Systems are agents or not with respect to some environment. The AIMA agent discussed above requires that an agent "can be viewed" as sensing and acting in an environment, that is, there must exist an environment in which it is an agent.

What about ordinary programs? A payroll program in a real world environment could be said to sense the world via it's input and act on it via its output, but is not an agent because its output would not normally effect what it senses later. A payroll program also fails the "over time" test of temporal continuity. It runs once and then goes into a coma, waiting to be called again. Most ordinary programs are ruled out by one or both of these conditions, regardless of how we stretch to define a suitable environment. All software agents are programs, but not all programs are agents.

Nor are software agents defined by their tasks. A spell checker adjunct to a wordprocessor is typically not an agent for the reasons given in the preceding paragraph. However, a spell checker that watched as I typed and corrected on the fly might well be an agent. Tasks can be specified so as to require agents to fulfill them.

Subroutines of agents need not be agents for the same reasons that programs need not be. However agents can have subagents. Herbert, the robot mentioned above, is built using a subsumption architecture [Brooks 1990], a layered architecture in which each layer senses and acts in order to perform its task. Each layer satisfies all the requirements of an autonomous agent. Thus the layers constitute a multiagent system that controls Herbert. Sumpy [Song, Franklin and Negatu, 1996] is a software agent living in a unix file system. Sumpy, also built using subsumption architecture, consists of subagents that wander, that compress files, that backup files, and that put Sumpy to sleep when the system is busy. Thus, Sumpy is both an agent and a multiagent system.

Our definition of an autonomous agents has succeeded in distinguishing between agents and programs. An agent need not be a program at all; it may be a robot or a school teacher. Software agents are, by definition, programs, but a program must measure up to several marks to be an agent. But our definition of autonomous agents yield a class of agents so large as not to promise great utility. Let's look at subclasses of agents with more promise.

Agent Classifications

The various definitions discussed above involve a host of properties of an agent. Having settled on a much less restrictive definition of an autonomous agent, these properties may help us further classify agents in useful ways. The table that follows lists several of the properties mentioned above.

Property	Other Names	Meaning
reactive	(sensing and acting)	responds in a timely fashion to changes in the environment
autonomous		exercises control over its own actions
goal-oriented	pro-active purposeful	does not simply act in response to the environment
temporally continuous		is a continuously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	adaptive	changes its behavior based on its previous experience
mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable "personality" and emotional state.

Agents may be usefully classified according to the subset of these properties that they enjoy. Every agent, by our definition, satisfies the first four properties. Adding other properties produces potentially useful classes of agents, for example, mobile, learning agents. Thus a hierarchical classification based on set inclusion occurs naturally. Mobile, learning agents are then a subclass of mobile agents.

There are, of course, other possible classifying schemes. For example, we might classify software agents according to the tasks they perform, for example, information gathering agents or email filtering agents. Or, we might classify them according to their control architecture. Sumpy, then, would be a fuzzy subsumption agent, while Etzioni and Weld's Softbot would be a planning agent [1994]. Agents may also be classified by the range and sensitivity of their senses, or by the range and effectiveness of their actions, or by how much internal state they possess.

Brustoloni's taxonomy of software agents [1991] begins with a three-way classification into regulation agents, planning agents, or adaptive agents. A regulation agent, probably named with regulation of temperature by a thermostat or similar regulation of bodily homeostasis, reacts to each sensory input as it comes in, and always knows what to do. It neither plans nor learns. Planning agents plan, either in the usual AI sense (problem solving agent), or using the case-based paradigm (case-based agents), or using operations research based methods (OR agents), or using various randomizing algorithms (randomizing agent). Brustoloni's adaptive agents not only plan, but learn. Thus there are adaptive problem solving agents, and so on, yielding a two layer taxonomy.

Yet another possible classification scheme might involve the environment in which the agent finds itself, for example software agents as opposed to artificial life agents. And, there must be many, many more such possibilities. Which one, or ones, shall we choose?

A Natural Kinds Taxonomy of Agents

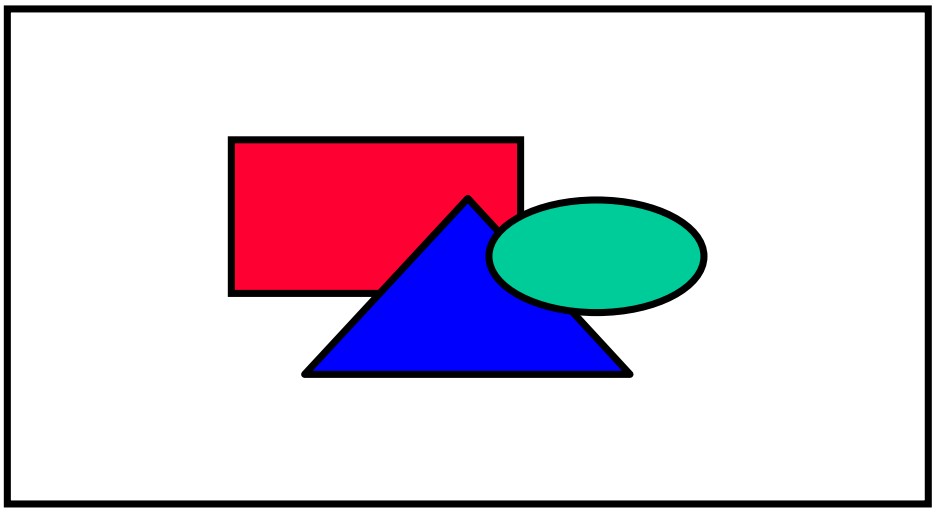
In thinking about a taxonomy of agents two possible models come to mind, the biological model

and the mathematical model. The biological taxonomy takes the form of a tree with "living creatures" at the root and individual species at the leaves. For example, we humans are classified as

- kingdom - animal
- phylum - chordata
- class - mammalia
- order - primate
- family - pongidae
- subfamily - hominidae
- genus - homo
- species - sapiens

where each line represents a branching point of the tree. Might it be possible to create such a taxonomy of autonomous agents? Let's start and see where we get.

At the kingdom level let's classify our agents as either biological, robotic, or computational, as these seem to be natural kinds [Keil, (1989)]. Every culture and even very young children readily distinguish between animate organisms, artifacts and abstract concepts. At the phylum level we can reasonably subclassify computational into software agents and artificial life agents. At the class level we might subclassify software agents into task-specific agents (like Sumpy), entertainment agents (like Julia), and computer viruses. At this point we've succeeded in categorizing our major classes of autonomous agents, that is the known families of examples.



Further Classification

Suppose we wished to classify software agents further. How might we go about it? The major subclassification schemes that come to mind are via control structures, via environments (database, file system, network, Internet), via language (in which written) or via applications. Each might be useful. Let's try the first.

Let's list some of the possible initial classification schemes for software agents via their control structures. Brustoloni offers regulation, planning and adaptive. Another strategy would be to classify by type of control mechanism, algorithmic, rule-based, planner, fuzzy, neural net, machine learning, etc. Or we might distinguish agents with a central executive from those enjoying distributed control. Other binary classifications might be planning vs. non-planning, learning vs. non-learning, mobile vs. non-mobile, communicative vs. non- communicative, etc.

Suppose we used the binary classification above, including central vs. distributed, in the order

mentioned, to create a binary classification tree. The first branching would be according to the first pair. On each of these branches we then branch according to the second pair, and on each of these four we branch again via the third pair, and so on. We've essentially listed a pool of features and classified according to subsets of these features.

Viewing our taxonomic tree from this perspective calls to mind a mathematical taxonomy which also employs collections of properties. A mathematician might define a topological space (please don't bother yourself about the meanings of this mathematical term or others). This essential definition defines the class of spaces to be studied. Then the notion of a Hausdorff space might be defined by an explicit property of some spaces. Thus the subclass of Hausdorff spaces is specified. Next the notion of a compact space may be defined, yielding the subclass of compact spaces. The intersection of these two is the subclass of compact Hausdorff spaces, about which theorems are often proved. The topological classification continues in this way with defining properties giving rise to subclasses of spaces which are then studied.

This type of classification scheme is known as a matrix organization among psychologists. Each feature defines a dimension. With n features an n-dimensional matrix is created, so that each cell of the matrix corresponds to a collection of features, and provides one possible category for the classification.

Having given the essential definition of an autonomous agent above, the class of agent is specified. We may then speak of planning agents, or of mobile agents, or even of mobile, communicative, planning agents, each specifying a subclass of agents. Of course, we must have given definitions of these three properties. Having the basic definition of an autonomous agent to build on, and using features for further classification, we may rephrase some of the definitions given earlier in a more convenient manner:

- A *KidSim Agent* is dedicated to a specific purpose, i.e., is a task-specific agent.
- A *Hayes-Roth Agent* reasons to interpret perceptions, solve problems, draw inferences, and determine actions, i.e., is a reasoning agent.
- An *IBM Agent* carries out some set of operations on behalf of a user or another program, i.e., is a task-specific agent.
- A *Wooldridge&shyp;Jennings* Agent interacts with other agents (and possibly humans) via some kind of agent-communication language, i.e., is a communicative agent.
- A *SodaBot Agent* engages in dialog , and negotiates and coordinates transfer of information, i.e., is a negotiating, information agent.

•Subagents and Societies of Agents

•Sumpy, the file system maintenance agent mentioned above, can be thought of as a single agent, or as a multiagent system consisting of Wanderer, Compressor, Back-Up and Sleepy. Each of these have independent access to sensors (certain unix commands such as ls) and to actions (other unix commands such as cd), and each has its own simple agenda. Also, each runs continuously, and acts so as to effect its next sensing. Thus, each may be considered an agent in it's own right, and hence a subagent. Sumpy is thus a multiagent system.

Some agents with a layered architecture are not multiagent systems. Müller, Pischel, and Thiel (1995) classify such architectures into vertically and horizontally layered. In horizontally layered systems each layer has access to sensing and acting, making a decomposition into subagents likely. In vertically layered system, only the lowest layer senses, and only the highest acts, making a multiagent decomposition unlikely.

As a multiagent system, Sumpy is particularly simple in that there is almost no communication

between the subagents. Each is, of course, privy to sensing initiated by the others, and Sleepy's action effects the others. Also, each subagent sometimes suppresses the actions of the lower layers. One might ask if Wanderer is truly autonomous if Compressor can suppress its actions. A person in jail, or in an elevator, has lost some freedom of movement, but is still autonomous. Environment may be expected to imposes limits on an agent's actions.

Going back to our topological analogy, we might call a system with no communication between its subagents a discrete multiagent system. A multiagent system in which each agent communicates with every other might be called fully connected,. Thus multiagents systems can be classified according to the possible communications paths through the system. We might also classify such systems by their communications bandwidth.

In addition to multiagent systems that can reasonably be viewed as constituting a single agent, other multiagent system are better classified as societies of agents. For example, when a collection of scheduling agents gather to schedule a meeting between their users, they pursue a common goal and intelligent group behavior emerges (see Kautz, Selman, and Coen 1994 for a similar situation.) Yet, as a group, our definition of agent is not met in that persistence is missing. When scheduling is complete, our agents disperse, perhaps never to gather again in this same grouping. One could argue that the collection of all such scheduling agents at a given site constitute a single agent. To do so, the notions of sensing, acting, and having its own agenda would have to be considerably stretched. As Russell and Norvig have reminded us, the issue here is not truth or falsity, but what's useful in communicating about agents.

The notion of a society of agents leads to a caution. The term "agent" as used by Minsky (1985) does not necessarily refer to an autonomous agent as the term is used here. In the context of trying to explain intelligence, Minsky speaks of "mental agents," saying "Each mental agent by itself can only do some simple thing that needs no mind or thought at all." I suspect that some, if not many, of his agents don't meet all our criteria for autonomous agents.

•Conclusions

•An attempt has been made to capture the essence of agency in a formal definition, which allows a clear distinction between a software agent and an arbitrary program. The beginnings of a natural kinds taxonomy for autonomous agents is proposed, as is further classification via collections of features.

References

Brooks, Rodney A. (1990), "Elephants Don't Play Chess," In Pattie Maes, ed., **Designing Autonomous Agents**, Cambridge, MA: MIT Press

Brustoloni, Jose C. (1991), "Autonomous Agents: Characterization and Requirements," Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University

Dethie, Vincent G. (1986), "The Magic of Metamorphosis: Nature's Own Sleight of Hand," *Smithsonian*, v. 17, p. 122ff

Etzioni, Oren, and Daniel Weld (1994), A Softbot-Based Interface to the Internet. *Communications of the ACM*, 37, 7, 72&shyp;79.

Franklin, Stan (1995), **Artificial Minds**, Cambridge, MA: MIT Press

Hayes-Roth, B. (1995). "An Architecture for Adaptive Intelligent Systems," *Artificial*

Intelligence: Special Issue on Agents and Interactivity, 72, 329-365, .

Kautz, H., B. Selman, and M. Coen (1994), "Bottom-up Design of Software Agents." *Communications of the ACM*, 37, 7, 143-146

Keil, F. C. (1989). **Concepts, Kinds, and Cognitive Development**. Cambridge, MA: MIT Press.

Langton, Christopher, ed. (1989), **Artificial Life**, Redwood City, CA: Addison-Wesley

Maes, Pattie (1990) ed., **Designing Autonomous Agents**, Cambridge, MA: MIT Press

Maes, Pattie (1995), "Artificial Life Meets Entertainment: Life like Autonomous Agents," *Communications of the ACM*, 38, 11, 108-114

Minsky, Marvin (1985), **The Society of Mind**, New York: Simon and Schuster

Müller, J. P., M. Pischel, and M. Thiel (1995), "Modeling Reactive Behaviour in Vertically Layered Agent Architectures," in Wooldridge and Jennings Eds., **Intelligent Agents**, Berlin: Springer-Verlag, 261-276

Russell, Stuart J. and Peter Norvig (1995), **Artificial Intelligence: A Modern Approach**, Englewood Cliffs, NJ: Prentice Hall

Smith, D. C., A. Cypher and J. Spohrer (1994), "KidSim: Programming Agents Without a Programming Language," *Communications of the ACM*, 37, 7, 55-67

Song, Hongjun, Stan Franklin and Aregahegn Negatu (1996), "A Fuzzy Subsumption Softbot," Proceedings of the ISCA Int Conf on Intelligent Systems, Reno Nevada

Wooldridge, Michael and Nicholas R. Jennings (1995), "Agent Theories, Architectures, and Languages: a Survey," in Wooldridge and Jennings Eds., **Intelligent Agents**, Berlin: Springer-Verlag, 1-22