Reasonable Machines:
Analogical Reasoning in Autonomous Agent Design

David C. Wyland

Reasonable Machines
15213 Bowden Ct.
Morgan Hill, CA 95037 USA
dcwyland@ix.netcom.com

Abstract. This paper focuses on agents for open environments (AOE) and on the use of analogical reasoning (AR) in their design. The autonomous mobile robot is an example of an AOE. It operates in an open, dynamic environment, i.e. unpredictable but familiar. An AOE should be a reasonable machine: it should act as we would for the same environment, history and goals. A reasonable machine has responses that are mediated among its goals and actions in the current context. AR can be used to design reasonable machines that implement this mediation. An agent using AR reasons from known examples to define actions in the current context. This paper proposes a form of AR called Analog Logic and discusses its potential to achieve AOE design goals. AR provides philosophical grounding of the symbols used and a new approach for dealing with continuous symbol systems and agent communication.

1 Introduction

This paper focuses on agents for open environments (AOE) and on the use of analogical reasoning in their architecture and design. The autonomous mobile robot (AMR) is an example of an AOE. The AMR operates in an open, dynamic environment. Open, in this sense, means unpredictable but familiar. An AMR is never in exactly the same place or exactly the same state twice. Its position will not be the same, its goals will not be the same and it can encounter obstacles and perturbations at any time. However, it still must function to achieve its goals. Because they deal with open environments, AOE applications include exploration (planetary rovers), natural materials processing (food processing, agriculture), maintenance, security and human interaction. In the last case, simple human interaction itself is the open environment: unpredictable but familiar. This paper proposes that analogical reasoning as embodied in analog logic allows us to design what we want in such an autonomous agent.

1.1 Autonomous Agents: Some Definitions

An autonomous agent is a machine that must make decisions during its activities (run time) rather than before starting its activities (compile time). Autonomous agents can be classified in terms of their environment. An agent's environment can be classified in two ways: simple versus complex and closed versus open. Environmental complexity can be defined in terms of the agent's total number of internal, external and action states. External states are the number of unique states of all the external inputs. Internal states are the number of unique states of the agent's internal storage elements, and action states are the number of unique actions that the agent can take. The state spaces can be discrete or continuous. These states and their sequence determine the behavior of the agent. The size of the total state space defines the complexity of the agent's operating environment, both in space (external states), time (internal states) and activity (action states). Agents for simple environments, such as many software robots, have few states. See [1], [2] and [3] for discussions of agent design. See [4] and [5] for discussions of software agent design.

Simple agents can use simple planning, such as conventional programming and search techniques. As the state space of the agent increases, conventional deductive planning becomes exponentially more difficult. At this point, the designer begins to resort to heuristic methods and eventually heuristic planning in general. Planning based on analogy such as case based planning is an example of heuristic planning. Hammond in [6] describes case based planning as "planning by remembering." Bernard in [7] discusses a spacecraft application with pre-compiled contingency plans, effectively plans as "synthetic memories. "

Closed environments can be closed and passive or closed and active. In a closed, passive environment, all states are known and predictable in advance. The starting state and all state transitions are known in advance. Industrial robots such as painting and spot welding robots on automobile production lines are examples of machines that work in closed, passive environments. They execute pre programmed fixed action sequences with little or no feedback from their environment. These machines are not generally considered agents because they are not autonomous: they make no run time choices.

In a closed, active environment, all states are known but their sequence is not predictable in advance. The starting state is known in advance, but not all state transitions are known in advance. The agent must make choices at run time on actions to take depending on the context. Software robots, some industrial robots and most mobile robots fit this category. An industrial robot that picks parts from a moving belt must examine the orientation of the part and choose a grasp action plan for picking up the part. Mobile robots for closed environments know all the objects they may encounter, but they make choices about how to deal with the obstacles when they encounter them.

A spacecraft agent such as described in [7] is a good example of an agent in a closed, active environment. In this example, the spacecraft does pre-planned experiments using instruments and navigation elements that can fail. In addition to the nominal experiment plans, the spacecraft stores contingency plans for equipment failures. When a piece of equipment fails, the on-board planner invokes the appropriate contingency plan to compensate for the failure. Equipment failure in the spacecraft may not directly detectable. The on-board planner uses models of the equipment to detect failures. The models provide a generalized algorithmic method of detecting failure. They encode the failure mechanisms and the sensor readings they generate. The models provide a deductively generated heuristic for relating sensor readings to equipment failure detection.

In an open environment, all states are not known and not predictable in advance. The states themselves are only partially known in advance, and the state transitions are only partly predictable in advance. Open environments are inherently active: if the states are partially unknown, the results of the agent actions – and therefore the new state – cannot be predicted in advance. Outdoor mobile robots such as planetary rovers fit in this category. The terrain and obstacles are only generally known in advance, never exactly known.

Mobile robots in open environments are never in exactly the same place or exactly the same state twice. You can't return to exactly the same place because of natural limitations on measurement and positioning capability of the robot. "Near enough" is just that: you return to a similar place, not exactly the same place. This means that you have to deal with familiar places, not the same places. The same goes for the robot's state, which is the combination of all sense inputs and history. Your history may have changed because of learning, if nothing else. Also if you are not in exactly the same place, your sensors will not generate the same values, even if they are perfect, have no noise and there were no other changes in the environment. The rover also will never encounter the same object twice in the sense that, even if it re-encounters "the same" object, it will encounter it from a slightly different position the second time. Situations and objects are similar to but not the same as previous situations and objects.

We will consider agents for complex open environments (AOE). An AOE needs both autonomy and a method for dealing with incomplete knowledge at run time. Open environments need heuristic methods for dealing with incomplete knowledge. Complex environments also lead to heuristic methods as a tactic to deal with the combinational complexity of many states. The problem of combinational complexity is similar to incomplete information: the information may be there, but you cannot derive it due to the finite nature of compute resources.

1.2 What We Want: Reasonable Machines

What would we like in an AOE, beyond functionality? We would like it to act reasonably, to act as we would for the same environment, history and goals. A reasonable machine would have responses that are mediated among its goals and actions in the current context. For example, if we told an agent to get a cup of coffee, we do not want it to drop the tray of glasses it is carrying and immediately go get the coffee. This implies multiple, context dependent goals and priorities. It also implies autonomy: the agent must decide how and when to carry out its intended actions in order to mediate among its context dependent goals and priorities.

We also want a machine that can carry out commands in open environments. An open environment is unpredictable but familiar. How can an environment (or anything) be unpredictable but familiar? Faces are an example. When you meet someone new, he or she has an unpredictable but familiar face. It is unpredictable because you have never seen this particular face before; it is familiar because it has two eyes, a nose, a mouth, and so forth. like any other human face. The new face is familiar: it "looks like" many other faces you have seen. An open environment may have unpredictable but familiar features. You have never seen this exact rock before, this exact animal or person before, this box or obstacle before, but it is similar to other rocks, animals, persons, boxes and obstacles you have seen before. If an environment is unpredictable but familiar, you cannot pre-plan how you will respond to events and actions, but you can respond to the environment based on similar prior examples.

We want social agents that carry on conversations with us. If it is going to be a "gofer" (as in go get something for me) agent, we need to talk with it a lot. Not only to give it commands, but to give it the background of the commands. We also want to check on how it is doing and to instruct it in new behaviors. These conversations should be similar in form and content to conversations between human agents, for example a conversation between a human and a store clerk about an intended purchase. This does not mean discussing Eastern Philosophy or life in France; it means discussing the agent's goals, status, priorities, experiences, knowledge and capabilities. The reason for the conversations is to insure that the agent understands what we mean when we say something.

What can an agent talk about? Only what it has directly experienced, what it can experience or information about activities that it can relate to what it has or can experience. An agent can talk about its sensor values and abstractions (patterns) made from its sensor inputs. For example, it can tell whether it has something in front of it. It can also sense whether the object has a quality, such as being red. Further, it can have abstractions about something it senses infrequently, such as a specific person that it can uniquely identify. It can also have indirect abstractions about its actuators, as in what happens at sensor inputs when an actuator moves.

What would you want an agent to talk about? Examples include: 1) its current goals, its status in achieving them and new goals we would like it to pursue, 2) goal priorities and changes to them, 3) its current beliefs about the world, 4) its experiences and the relations between them, and 5) new behaviors to be directly learned and behaviors to be learned by example through stories. Although the list is extensive, each conversation is potentially simple and grounded in the experience of the robot. If the conversation is limited to topics that the robot can directly or vicariously experience, we can design a robot and conversational apparatus to do this.

Why not allow the agent to talk about things it cannot directly experience? Because we will wind up with a "parrot" agent that can respond convincingly to conversational cues, but has no understanding of the content of the conversation. A dictionary agent is an example. You could have an agent that would respond with dictionary text given a word as input. However, the agent would not "understand" the text it emitted. This is the symbol grounding problem, as discussed in [8] and [9]. The word is a symbol, but it is not related to the agent's ability to experience the thing the word refers to. This corresponds to cocktail party conversation by someone who has lengthy opinions on a book he or she has not read. The computer program Eliza does this. It uses a simple program to deceive the user into thinking that it understands its conversation with the user. Even though it is very convincing, it stands as an archetypical example of something that appears to the user to understand conversation but definitely does not.

What does "understand" mean, in agent terms? An agent understands a sentence and the words in it if it can relate them to its past experiences and actions, and make decisions based on this relationship. In story terms, this is equivalent to being able to paraphrase a story, to tell a new story that is equivalent to an existing story using different words, phrases and examples

2 Designing Reasonable Machines

Designing an AOE means designing a machine for an unknown and unpredictable environment. This is somewhat a contradiction in terms. Design is based on deductive reasoning. You select components for their known characteristics. These are the "axioms" of your "proof," i.e. your design. You combine these components to achieve the desired result in the same manner as constructing a proof. If you know how the components work, you can exactly predict how the design will work. If you do not have the component you need, you must create one. If you do not have the environment you need, you modify it until it meets your needs, or you model it as a new component and restart your proof. Sometimes the characteristics of the components or the environment are only partly known. In this case, you do testing during the design to complete your knowledge. You make the necessary adjustments before the design is done.

For an AOE, the characteristics of the components and the environment are never completely known, they change with time and we cannot modify the environment to suit the design. Therefore, we cannot design, in the sense outlined above. So how can we design one? The answer is that we design a machine that can work effectively with incomplete knowledge. It must do the best it can, since it cannot know how to do the best absolutely. Such a machine will never have (or at least never know that it has) the kind of complete knowledge we require for a design, even if it learns. If it ever does, it will mean that the environment is no longer open: it has become completely predictable. At this point it is no longer an AOE, as we have described it.

The AOE has to deal with incomplete information in an unpredictable environment, yet take effective action according to its goals. This is an important planning problem, as discussed in [10] who notes that plans fail more often from incomplete beliefs than false beliefs. How can an agent take effective action in a new situation? Answer: it will have to make a guess. And to be effective, it will have to make the best guess it can. It can make a good guess by comparing the current situation to similar situations it has previously encountered, then taking the action that worked best in those situations. Once the action has been taken and the results of that action have developed, the agent can record this as a new situation-action experience pair for use as a candidate in the next new situation.

This solves our small dilemma. Such a machine can be designed conventionally yet act reasonably. A simple analogy is a payroll program. The payroll program is rigorously designed, yet nothing in the design predicts what payroll checks will be printed next week.

3 Analogical Reasoning

An agent has to deal with an unpredictable environment and yet act in a reasonable manner. One way to do this is to have its current activities guided by past examples of activity given a similar environment. Since no combination of sensed environment, goals and history can be exactly the same as any previous combination, the agent will have to look for similarities between the current situation and prior situations. It then will mediate its actions between the actions corresponding to the previous situations.

This style of reasoning called analogical reasoning, which is reasoning by analogy to previously similar situations. Analogical reasoning assumes that what worked in the past will work in the future. Reasoning by analogy involves measuring the similarity between the current situation and past situations, and then taking the same action you took in past situations based on similarity. The action taken may be a blend of past actions weighted by similarity, or it may be a choice of the action associated with the most similar case.

Analogical reasoning depends on the measurement of similarity of the current state to remembered examples and on the blending (or choice of the best) of remembered actions based on similarity. The challenges in analogical reasoning are how similarity is measured, how the examples are remembered and how the actions associated with the examples are remembered and combined to create new action in the current situation.

To see how analogical reasoning works, let us compare it to deductive reasoning. Deductive reasoning is based on equality, while analogical reasoning is based on similarity. Both reasoning systems define and manipulate their symbols according to rules. The symbols in both systems “point” to things in some context, some part of the world. In deductive reasoning, symbols have a one of two possible values: true or not true. This value indicates whether the thing indicated by the symbol is equal or not equal to an ideal thing associated with the symbol. For example when counting stones, each stone is equal to all other stones and to an ideal stone for the purposes of counting. Otherwise, we could not count things.

In analogical reasoning, each symbol has one of many possible values indicating degree of similarity between the thing pointed to by the symbol and an ideal associated with the symbol. For example when comparing a dining room table to a small coffee table, the dining room table is more similar to an ideal example of a table than the coffee table is, and it would have a higher value for similarity to the ideal table as a result. Otherwise, we could not compare things.

To compare these two systems, consider Socrates and his mortality. In deductive reasoning, we state the following: 1) Socrates is a man, 2) all men are mortal and 3) therefore, Socrates is mortal. To put it more precisely: 1) Socrates meets the criteria of being a man, therefore Socrates is a man; 2) If x meets the criteria of being a man, x is mortal by assertion, and 3) Socrates meets the criteria of being a man, therefore Socrates is mortal. In step 2) we rely implicitly on the fact that x is either true or not true, and therefore the mortality of x is also asserted to be true if x is true.

An analogical reasoning version of Socrates mortality does not work well because man and mortal are true or false: there are no intermediate values. However, let's try Socrates' table, as follows: 1) Socrates' table is tall. 2) Tall tables have long legs. 3) Therefore, Socrates' table has long legs. To put it more precisely: 1) Socrates' table has a measure of tallness. 2) To the degree that table x is tall, x has long legs. 3) Therefore, to the degree that Socrates' table is tall, Socrates' table has long legs.

Note that analogical reasoning, as described above, defines a continuous symbol system.

Analogical reasoning is becoming interesting as a tool for agent design. Pollock in [10] provides some justification for using analogical reasoning in planning. Hofstadter in [11] argues that pattern finding and corresponding analogy making is at the core of intelligence. In [12], analogical representation is recommended as an internal representation for reasoning.

Analogical reasoning has had successes in Case Based Reasoning (CBR) and fuzzy logic applications. CBR is reasoning by analogy, as discussed in [13] Each case in CBR is a remembered example of a situation-action plan that achieved (or failed to achieve) a goal. Cases are relatively large, complete plans. As a result, CBR is considered large grain (large, complex example) reasoning.

Fuzzy logic as used in almost all fuzzy logic control systems implements analogical reasoning as its control method. This is not generally recognized. However, the fuzzy logic control examples in [14] and [15] are most easily understood as examples of analogical reasoning.

Fuzzy logic has had successes in agent design, specifically autonomous mobile robots. The first and second place winners of the AAAI 93 autonomous mobile robot competition used fuzzy logic as the reactive layer that connects the sensors to the actuators and (digital) symbolic planners as supervisors. See [16] and [17].

Analogical reasoning can provide some significant benefits. Since analogical reasoning is heuristic and based on learned or taught examples, it avoids the frame problem. The frame problem, as reviewed in [18], is the problem of deductively predicting the future, an impossible task. The specific problem is to determine what changes and does not change one instant from now. Changes can occur because of actions an agent takes, or they may occur spontaneously as a result of outside forces. The problem is to predict the future so that a deductive reasoning system can plan for optimal action to achieve its goals. Analogical reasoning avoids the problem by making a basic assumption: what worked last time will work this time. This may not always be true, but it is the best we can do in an unpredictable world of chronically incomplete knowledge.

Analogical reasoning provides a new and potentially more effective way of dealing with words, language and conversation. Analogical reasoning can provide the basis for a continuous symbol system, and words represented in this system have degrees of meaning depending on context. This is compatible with current research on the mental concepts we use and the words that refer to these concepts, as discussed in [19] and [20]. By providing a more accurate understanding and representation of words and by grounding them in examples, analogical reasoning should be able to significantly improve and simplify communication between agents and humans.

3.1 Analog Logic

Analog logic implements analogical reasoning. It uses analogical inference to provide a formal bridge between similarity and implied action. Analog inference rules remember the relation between examples and their associated actions, and it combines the actions according to the relative value of their associated similarity values.

Analogical reasoning depends the measurement of the similarity of the current state to remembered examples and on the blending (or choice of the best) of remembered actions based on similarity. The challenges in analogical reasoning are how similarity is measured, how the examples are remembered and how the actions associated with the examples are remembered and combined to create new action in the current situation.

Analog logic uses the concept of measurement of qualities of things – such as temperature – to define similarity, and it uses the comparison of these measurements to relate similarities. The qualities of remembered examples are contained in measurement operators. These operators compare the current quality of a thing against an implied reference and return a relative similarity value between 0 and 100%.

Analog logic consists of symbols and inference operators. Symbols point to things in the world and represent abstract qualities of these things. Each symbol has a value between zero and one. This value indicates the degree to which a quality is present in the thing, from not at all present to fully present, for example the relative redness of a ball, from not at all red to maximally red. The value associated with the symbol is called a signal. In analog logic, this signal has a value between zero and one. In deductive, digital logic, this signal would have a value of either zero or one.

Signal values are defined as constants, generated by sensors or generated by inference. A sensor measures something in the outside world and returns a value that is a measurement of a quality of that thing, such as degree of redness of a ball. If the sensor's output does not directly indicate the desired quality, inference operators can be used to generate the desired quality value from the output of the sensor(s).

Inference maps one or more input quality values to an output quality value. Inference in analog logic is by assertion: for a given input value(s), the inference function asserts the output value. There are two types of inference operators: direct and indirect. Direct inference maps one or more inputs to a single output.

Indirect inference uses its inputs to map other inputs to a single output. In indirect inference, each inference input consists of two values: the control value and the recommended output value. For example in the choice operator, the control input with the maximum value selects the single recommended output value associated with that input. In the blend operator, the control inputs are used to perform a weighted average combination of the recommended values, weighted by their associated quality values.

Direct inference operators are called measurement operators (MO's). An MO is typically defined by identifying example points in the input space of the MO. These example points, or exemplars, define the output values for specific input values. Other values are derived by interpolation between the exemplars. For example, an MO indicating hot outside temperature might have exemplars at 70 degrees and 90 degrees. The resulting MO has a value of 0.0 at 70 degrees and below, rising to 1.0 at 90 degrees and above. This MO could convert a outside thermometer sensor output into the quality hot temperature of outside.

The examples that define an MO define the context of the measurement it performs. To define "red ball," we could use a red ball and a non-red ball as examples. Since the measurement operator is defined in reference to examples, its value is a measure of the similarity of the thing measured to those examples. It indicates the degree of similarity between the thing in the environment the symbol refers to and a perfect example of the quality. Zero means no similarity, and one means complete similarity.

MO's are often defined in sets by example points in the input space common to the set of MO's. Each example point, or exemplar, defines a separate MO. Each MO has a value of 1.0 at its exemplar point, declining to a value of 0.0 at adjacent exemplars and a value of 0.0 everywhere else. The result is a set of triangular MO's, with the peak of each triangle corresponding to its defining exemplar. For example, a set of MO's spanning outside temperature might have defining exemplars at (-20, 40, 65, 90, 120) degrees F corresponding to MO's of (maximum cold, cold, normal, hot, maximum hot). Such a set of MO's is useful in blend and choice inferences. As the temperature moves from –20 to + 120 degrees F, pairs of MO's become successively active, with one MO output value increasing to 1.0 as the other decreases to 0.0. When used with a blend operator, the blend operator output value moves smoothly from one recommended output value to the next. The combination of MO set definition by exemplars and its use with the blend operator results in analogical reasoning. The output from the blend operator is a weighted average of the recommend output values associated with the exemplar values that define the MO's.

The indirect inference operators, blend and choice, provide analogical inference as we have defined it. Analogical reasoning depends on the measurement of similarity of the current state to remembered examples and on the blending (or choice of the best) of remembered actions based on similarity. The blend operator performs a weighted average of recommended output values (the remembered actions), weighted by the relative values of the quality (similarity) input values. The choice operator selects the recommended output values (the remembered action), associated with the highest quality (similarity) input value.

3.2 Analog Logic and Language

Humans can define membership operators by subjective testing. In MO definition by subjective testing, you give a candidate group of people a set of input values and ask them to specify the corresponding output values. The resulting values are statistically averaged to produce the values for the MO. For example, consider an MO that generates the quality of Hot Outside Temperature from a thermometer sensor reading. The candidate group might define Hot to have a value of 1.0 for 90 degrees F or above and a value of 0.0 for 70 degrees F or below, with values between 70 and 90 degrees by interpolation.

The subjective approach works. If the candidate group understands the quality being inferred, the group members return statistically similar values for the inferred values. [21] documents this somewhat surprising result. Perhaps this is not so surprising in retrospect. The qualities of analog logic correspond to words in ordinary language. To make decisions acceptable to a group of people based on whether a thing is hot or not, there must be a closely correlated agreement by each member of the group of what hot means in terms of temperature. This can allow a designer to directly define an MO by using knowledge held in common with the group of people concerned with the output of the MO. If all members agree closely on the MO inference values, anyone (including the designer) can define the MO. Group testing can then be used to verify the values, if necessary.

Humans can – and typically do – define MO's and other inferences so their output values correspond to commonly understood words, for example, hot temperature of outdoor air. This allows the human designer to understand and have proper expectations about the output value, and it allows philosophically grounded communication between the human and an agent. When the agent or the human uses the term "hot temperature of outdoor air," each knows what it means because each knows what it refers to.

3.3 Analog Logic and Fuzzy Logic

Analog logic is based on fuzzy logic as used in fuzzy logic control system design. (See [22], [23] and [14] for a discussion of fuzzy logic.) However since fuzzy logic is a general formalism for modeling degrees of truth including uncertainty and the ambiguity of language, it provides no specific guidance or conceptual grounding for the operations required in these control systems. If you study fuzzy logic for control applications, you find no formal method to justify the choice of inference (defuzzification) functions other than "these seem to work well and are popular." However, if you study fuzzy control application in [14] and [15], you will find fuzzy logic is used as a mathematical formalism for analogical reasoning, without being recognized as such.

In many of these fuzzy control designs, example points in the system input space define fuzzy membership functions, the fuzzifier operators. Each membership function has an output value indicating the degree to which the input is near its defining exemplar point relative to adjacent points. These membership function values are used for inference. Each membership function output is applied to an inference operator, the defuzzyfier operator. Associated with each membership function is a recommended output value that is supplied to the inference operator. The commonly used “center of moment” defuzzifier operator corresponds to the analog logic blend operator. It generates a weighted average of the recommended output values, weighted by the input values. The result of this operation is that the output of the defuzzifier is the blending of remembered actions (recommended output values) based on similarity (relative nearness to example points). Stripped of fuzzy logic terminology, this is equivalent to analogical inference.

Analog logic is a variant or subset of fuzzy logic that targets the domain of analogical reasoning, and analogical reasoning provides the conceptual grounding for its operations. Analog logic attributes its symbol values to measures of quality, rather than degrees of vagueness or ambiguity. The variables of fuzzy logic can measure many different things, including vagueness and ambiguity; however, this capability is not used in analog logic. Analog logic has no inherent vagueness. The membership functions of fuzzy logic become the measurement operators of analog logic. Measurement and other inference operators return specific values (crisp values in fuzzy terms) that indicate the degree to which a quality is present and that expresses the degree of similarity to remembered examples. Inference in analog logic is an asserted mapping from one quality measurement to a new quality measurement.

4 Summary

Autonomous agents such as planetary rovers work in open environments. Open environments are unpredictable but familiar. The challenge is to create agents that work effectively in these environments. We want agents that act as reasonable machines, to act as we would for the same environment, history and goals. This implies mediated responses, multiple context dependent goals and priority, and autonomy to carry out its goals. We also want the agent to be social, to carry on conversations about its experiences, its goals and its progress toward achieving its goals.

To design an agent for an open environment (AOE), we have to design a machine that works with limited knowledge in an unpredictable environment. Such a machine must make a guess about what to do next. By reasoning from experience, it tries to make the best guess it can. If it can learn the new experience, it may be able to make a better guess next time.

Reasoning from experience is analogical reasoning. Analogical reasoning makes one basic assumption: what worked before will work again. Analogical reasoning involves measuring the similarity between the current situation and past situations, and then taking the same action you took in past situations based on similarity.

Analog logic implements analogical reasoning. It does this by creating a continuous symbol system with elements and operators that implement the analogical reasoning.

Analogical reasoning avoids the frame problem by assuming what worked before will work again. This may not always be true, but it is the best we can do in an unpredictable world of chronically incomplete knowledge. Also, by providing a more accurate understanding and representation of words and by grounding them in examples, analogical reasoning and analog logic should be able to significantly improve and simplify communication between agents and humans.

References

1. M. Wooldridge and N. R. Jennings, “Agent Theories, Architectures, and Languages: A Survey," Proc. {ECAI}-Workshop on Agent Theories, Architectures and Languages 1994, Springer-verlag 1995, ISBN 3-540-58855-8. url = "citeseer.nj.nec.com/wooldridge94agent.html"
2. Foundations of Rational Agency, ed. Rao and Wooldridge, Kluwer Academic Publishers 1999; ISBN: 0792356012
3. M. Wooldridge, Reasoning About Rational Agents, MIT Press 2000, ISBN 0-262-23213-8.
4. Richard Murch & Tony Johnson, Intelligent Software Agents, Prentice Hall 1999 ISBN 0-13-011021-3
5. Joseph Williams, Bots and Other Internet Beasties, Sams Publishing 1996 ISBN 1-57521-016-9
6. K. J. Hammond", "Case-Based Planning: A Framework for Planning from Experience", Cognitive Science, volume 14, pages 385--443, 1990", url = "citeseer.nj.nec.com/article/hammond90casebased.html"
7. D. Bernard and G. Dorais & C. Fry & E. Jr & B. Kanefsky & J. Kurien & W. Millar and N. Muscettola & P. Nayak & B. Pell and K. Rajan & N. Rouquette & B. Smith & B. Williams, " Design of the Remote Agent Experiment for Spacecraft Autonomy," Proceedings of the IEEE Aerospace Conference, Snowmass, CO: IEEE 1998 url = "citeseer.nj.nec.com/bernard98design.html"
8. Tom Ziemke, “Rethinking Grounding,” Does Representation Need Reality? - Proceedings of the International Conference ‘New Trends in Cognitive Science’ (NTCS97) Perspectives from Cognitive Science, Neuroscience, Epistemology and Artificial Life, Austrian Society for Cognitive Science AsoCS technical Repore 97001, Vienna, Austria May 978.
9. Stevan Harnad, “Grounding Symbolic Capacity in Robotic Capacity,” Building Situated Embodied Agents, ed. L. Steels & R. Brooks, Lawrence Erlbaum Assoc 1990.
10. John Pollock, “Planning Agents,” Foundations of Rational Agency, ed. Rao and Wooldridge, Kluwer Academic Publishers 1999; ISBN: 0792356012
11. Douglas Hofstadter, Fluid Concepts and Creative Analogies, Basic Books 1995 ISBN 0-465-02475-0
12. Luc Steels, “Exploiting Analogical Representations,” Designing Autonomous Agents, ed. Pattie Maes, MIT Press 1990, ISBN 0-262-63135-0.

13. Janet Kolodner, Case Based Reasoning, Morgan Kaufman Publishers 1993, ISBN 0-7923-9075-X.
14. Constantin von Altrock, Fuzzy Logic & Neurofuzzy Applications Explained, Prentice-Hall 1995, ISBN 0-13-368465-2
15. Earl Cox, The Fuzzy Systems Handbook, Acedemic Press, ISBN 0-12-194455-7.
16. S. Goodridge, "A Fuzzy Behavior-Based Nervous System for an Autonomous Mobile Robot," Master's Thesis, North Carolina State University, 1994.
17. Simon Parsons, Ola Pettersson, Alessandro Saffiotti, Michael Wooldridge, "Robots with the Best of Intentions", Artificial Intelligence Today, pages 329-338, Springer Verlag 1987; ISBN: 3540664289 url = "citeseer.nj.nec.com/196598.html"
18. The Robot's Dilemma Revisited: The Frame Problem in Artificial Intelligence, ed. Kenneth M. Ford & Zenon W. Pylyshyn, Ablex Publishing Corporation 1996 ISBN 1-56750-143-5
19. Ulrike Hann & Nick Chater, "Concepts and Similarity," Knowledge, Concepts and Categories, ed. K. Lamberts and D. Shanks, MIT Press 1997, ISBN 0-262-62118-5.
20. Koen Lamberts, "Process Models of Categorization," Knowledge, Concepts and Categories, ed. K. Lamberts and D. Shanks, MIT Press 1997, ISBN 0-262-62118-5.
21. Elenor Rosch, On the Internal Structure of Perceptual and Semantic Categories,Cognitive Development and the Acquisition of Language, ed. Timothy Moore, Academic Press New York 1973.
22. [Zadeh 96] Lofti Zadeh, Fuzzy Sets, Fuzzy Logic and Fuzzy Systems: Selected Papers by Lofti A. Zadeh, ed. George J. Klir & Bo Yuan. World Scientific 1996, ISBN 9810224222.
23. H. J. Zimmermann, Fuzzy Set Theory and its Applications, Kluwer Academic Press 1991, ISBN 0-7923-9075-X.