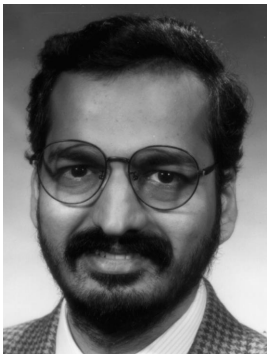

COGNITIVE AGENTS

Michael N. Huhns • *University of South Carolina* • *huhns@sc.edu*
Munindar P. Singh • *North Carolina State University* • *singh@ncsu.edu*

As agents get smarter and our expectations of their capabilities rise correspondingly, we will no doubt treat them more anthropomorphically. It is already common for both system designers and users to attribute beliefs and intentions to agents—much the way people attribute such cognitions to their pets. For example, we might say that our cat appears aloof because it believes it is superior to its owners.

We also use such terms for machines, especially when we are annoyed with them. For example, if an ATM gives us the wrong foreign exchange, we might wonder if it intends to cheat us—a possibility that would be replaced by more benign attributions if the ATM actually gave more money than we expected.

In the case of agents, however, the basis for ascribing beliefs and intentions to their actions is not just sentimental or frivolous. Even though computational agents are pieces of machinery, their designs must be specified and behaviors commanded by humans. And because humans think and speak using cognitive terms such as beliefs, knowledge, desires, and intentions, it is more natural to use the same cognitive concepts when constructing agents and assigning tasks to them.

Cognitive Reflections of Environmental States

An agent operates in some physical or computational environment. An agent is itself a physical system of some sort. Even a pure software agent is embodied on a computer that gives a home to the agent's internal structures (data structures, if you will) and enacts its program.

For an agent to act properly in a changing environment, some combination of its data structures and program must reflect the information it has about its environment. Because this information would reflect the state of the environment according to the agent, it can be termed its *knowledge* or a set of its *beliefs*. (The distinction between knowledge and beliefs is stronger in ordinary language than it is in the literature about agents, where knowledge is usually treated simply as true belief. Some researchers represent the relationship between the two using additional attributes, such as justifications, but we'll just accept the simpler definition.) *Desires* correspond to the state of the environment the agent prefers. *Intentions* correspond to the state of the environment the agent is trying to achieve, which should be a consistent subset of the agent's desires and directly connected to the agent's actions.

Notice that it is the human designer who determines the agent's beliefs, desires, and intentions in an environment. However, to make sense, these beliefs, desires, and intentions must be related to the agent's perceptions and actions. This relationship can be captured in an agent architecture such as the one shown in Figure 1.

Applying Cognitive Concepts

The relationship is mediated by the agent's reasoning subsystem. For simplicity and as is customary, let's assume that the agent's desires are given. The cognitive concepts can then be used in two ways:[1]

- *Means-ends reasoning*. The agent must decide what intentions to adopt or revise, and what actions to perform.
- *Plan recognition*. The agent must infer the beliefs, desires, and intentions of other agents in order to cooperate or compete with them.

For example, suppose agent Al from Figure 1 desires both ice cream and soup, but given that the weather is cold (and based on beliefs not mentioned here), Al intends to have only soup. Means-end reasoning causes Al to get soup from the pantry and heat it in the microwave oven. In a second example, Al sees Bo perform actions or hears Bo's statements indicating that Bo is opening the refrigerator door. From this action, Al uses plan recognition to infer that Bo is about to get ice cream. Knowing Bo to be "rational," Al figures that Bo does not believe it is cold outside. Since Al is a helpful agent, he tells Bo that it is cold outside.

Although the examples describe only two agents in a kitchen, the agents are reasoning about each other's actions, which could be actions to access information if the agents were in a Web environment.

A cognitive basis for an agent is especially applicable when the agent is to serve as a personal assistant in a user interface. Based on its own intentions and beliefs, the agent can infer what its user needs by understanding his or her intentions.[2] This facilitates "tasking," allowing users to tell agents

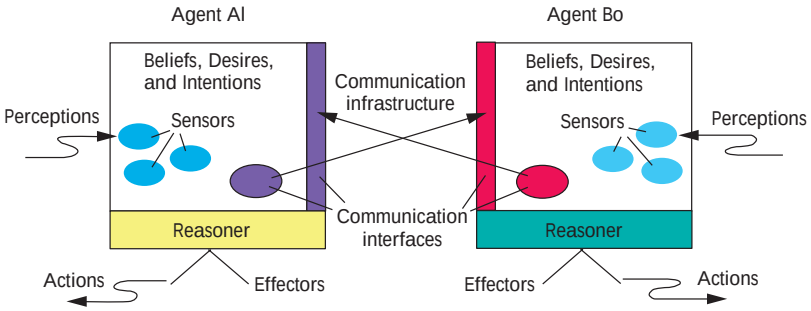

Figure 1. An architecture for an agent that captures the beliefs, desires, and intentions ascribed to the agent, and relates them to the agent's perceptions and actions in an environment.

what to do rather than *how* to do it. Such an agent might be able to solve its user's problem even if it happens to be formulated incorrectly, which can easily be the case when the user is asking for assistance.

Although intuitive, cognitive concepts often have a number of connotations in the vernacular. If we are going to build computing systems from cognitive concepts, we must be sure of our interpretations and how precisely they relate to an agent's construction.

One class of properties that cognitive theories seek to capture is the relationships among the concepts. For example,

- Beliefs are mutually consistent. (This can be a demanding property to realize in a practical system and usually requires an agent's beliefs to be restricted in some way.)
- An agent will intend an action only while it believes the action is possible.
- An agent need not intend something that would happen anyway.

These kinds of properties have long been under development. Much progress has been made, but not every important aspect has been worked out. However, developers are proceeding with practical systems based on these concepts, providing valuable input in refining the theories.

(Note: Readers interested in details on cognitive concepts can enter the vast literature on the subject through selections from our book, *Readings in Agents*.[3])

Putting Cognitive Concepts to Work

Three broad approaches have been defined for implementing cognitive concepts.[4] The approaches are reflected in different agent architectures.

In the first approach, designers use cognitive concepts to model an agent's reasoning. The agent represents its beliefs, intentions, and desires in modular data structures and performs explicit manipulations on those structures to carry out means-ends reasoning or plan recognition. When the cognitive concepts are defined formally, the explicit manipulations can be accomplished through the application of a suitable theorem prover. Among the best of the systems using this approach is Artimis,[5] an intentional system designed for human interaction and applied in a spoken-dialog interface for information access. This system is based on a logic of beliefs and intentions, which it uses to carry out effective dialogues with users. Although its application domain is somewhat narrow, Artimis' use of a theorem prover for cognitive concepts makes it one of the purest systems of its kind.

In the second approach, designers can still use explicit representations of cognitive concepts, but the concepts are processed procedurally, rather than via theorem proving. The procedural approaches have better performance than theorem proving, but performance comes at the expense of implementation simplicity and the superior semantic basis of theorem provers. Most practical agent architectures based on cognitive concepts fall into the procedural category. A notable example is the belief, desire, and intention (BDI) architecture[4] instantiated in the procedural reasoning system (**PRS**) and the distributed multiagent reasoning system (**dMARS**).

The third approach uses cognitive concepts only for design and analysis. A designer can think of an agent's behavior in cognitive terms, but the agent itself would not have any explicit representations of the cognitive concepts. The agent might be just a simple finite-state machine operating in a restricted environment. There is continuing interest in such a "situated automata" approach, and it is especially promising in settings where higher performance is desired, but the agent's construction does not have to be highly complex.

In each of these approaches, the designer ascribes cognitive concepts to the agent. The designer considers not only the agent's data structures, but also how these structures are linked to its sensors and effectors (see Figure 1), and how the sensors and effectors in turn are linked to the real environment. This is difficult, however, and the agent's resultant behavior might not fully reflect the designer's intentions. An agent might still have false beliefs, inappropriate desires, or impossible intentions. Such eventualities can occur, for example, if the agent is operating outside its normal design range. In such a situation, the intentional stance (see the sidebar on this page) can help designers and users understand and analyze why the agent is behaving in an apparently inappropriate manner.

Communicating Cognitively

Several researchers have proposed using cognitive concepts as a semantic basis for agent communications.[6] One of the leading candidates for such a semantics is based on Arcol, the communication language used within Artimis.[5] Interestingly, this application (not only of Arcol, but also in general) appears extremely misguided. The intentional concepts are well-suited to designing agents, but are not suited to giving a basis to a public, standardizable view of communication.[6]

A challenge for using the cognitive concepts is that although they are natural in several respects and can guide implementations, full-blown implementations that try to be faithful to every aspect of the model can end up being computationally demanding. As the cognitive concepts are put to use in real applications, the principles for simplifying the implementations will emerge. In any case, because of their naturalness to humans, the cognitive concepts are here to stay, and we will do well to consider them in the design of our agents. ■

Michael Huhns is professor of electrical and computer engineering at the University of South Carolina and Munindar Singh is assistant professor of computer science at North Carolina State University. Both are members of the editorial board for *IEEE Internet Computing*. Huhns and Singh have collaborated on various aspects of agents for almost a decade. Their coedited *Reading in Agents* appeared in 1998.

REFERENCES

1. M.E. Pollack, "The Uses of Plans," *Artificial Intelligence*, Vol. 57, No. 1, 1992, pp. 43–68.
2. M.N. Huhns and M.P. Singh, "Personal Assistants," *IEEE Internet Computing*, Vol. 2, No. 5, Sep./Oct. 1998, pp. 90–92.
3. M.N. Huhns and M.P. Singh, *Readings in Agents*, Morgan Kaufmann, San Francisco, Calif., 1998.
4. M.P. Singh, A.S. Rao, and M.P. Georgeff, "Formal Methods in DAI: Logic-Based Representation and Reasoning," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Gerhard Weiss, ed., MIT Press, Cambridge, Mass., 1998, pp. 331–376. Forthcoming. Preprint available from www.csc.ncsu.edu/faculty/mpsingh/papers/mas/formal-DAI.ps.
5. P. Breiter and M.D. Sadek, "A Rational Agent as a Kernel of a Cooperative Dialogue System: Implementing a Logical Theory of Interaction," in *Proc. of the ECAI Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, Berlin, 1996, pp. 261–276.
6. M.N. Huhns and M.P. Singh, "Conversational Agents," *IEEE Internet Computing*, Vol. 1, No. 2, Mar./Apr. 1997, pp. 73–75.

URLs for this column

dMars • www.aaii.oz.au/proj/dmars_tech_overview/dMARS-overview.html

LALO • www.CRIM.CA/sbc/english/lalo/

PRS • www.ai.sri.com/~prs/

UMPRS and Jam! • members.home.net:80/marcush/IRS/

THE INTENTIONAL STANCE

The intentional stance is the philosophical view underlying the use of cognitive concepts for agents. It was first promulgated by the computer scientist John McCarthy[1] and developed further by the philosopher Daniel Dennett.[2] The intentional stance simply states that cognitive concepts can be ascribed to any physical system and that it is beneficial to do so for complex systems. Indeed, if a system is sufficiently complex—such that complete physical details can never be known—the intentional stance might be the only one that enables us to understand how it acts.

People often use cognitive concepts to understand how others behave. For example, we try to anticipate the actions of other drivers on the road by inferences about their beliefs and intentions. We could never function as car drivers if we had to reason about the neural states of the other drivers just to figure out if they were about to change lanes. In fact, even though neuroscience hasn't yet developed to the stage where human brains can be mathematically modeled, people and even animals have always been able to figure out each other's beliefs and intentions to more or less correctly predict each other's actions. The idea, then, is to use cognitive concepts to talk about the states of computational agents without needing to know how those agents are implemented.

References

1. J. McCarthy, "Ascribing Mental Qualities to Machines." In *Philosophical Perspectives in Artificial Intelligence*, M. Ringle, ed., Harvester Press, Brighton, UK, 1979.
2. D.C. Dennett, *The Intentional Stance*, MIT Press, Cambridge, Mass., 1987.

SYSTEMS OF THE BIMONTH

A number of systems now exist that owe a large part of their functionality to cognitive concepts. Two systems with practical applications are dMARS and Artimis as alluded to above. dMARS is not publicly available, but two related systems are, namely, University of Michigan PRS (**UMPRS**) **and Jam!**.

Langage d'Agents Logiciel Objet (**LALO**) is a programming language based on agent-oriented programming, in broad terms a variant of the cognitive concepts discussed here.[1] The STEAM system uses a model based on beliefs and intentions to carry out effective teamwork among agents.[2]

Check them out!

References

1. Y. Shoham, "Agent-Oriented Programming," in *Readings in Agents*, Morgan Kaufmann, San Francisco, Calif., 1998, pp. 329–349. (Reprinted from *Artificial Intelligence*, 1993.)
2. M. Tambe, "Agent Architectures for Flexible, Practical Teamwork," in *Proc. Nat'l Conf. on Artificial Intelligence*, AAAI Press, Menlo Park, CA, 1997, pp. 22–28.