

Agents

(TOO BRIEF)

5.1 What is an Agent?

Russell and Norvig define AI as in part “the study and construction of rational agents”; to them an *agent* is “anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors”.

An important idea for agents is *autonomy*: the agent does its own thing. But there are other properties that distinguish agents from other processes: social ability (interact with other agents), reactivity (perceive and respond to the environment), and pro-activeness (goal-directed behavior), etc. Some even go further to provide human concepts: attitudes, desires, beliefs, commitments. And want agents to be benevolent, adaptive or truthful.

Some stress the *principle of rational choice*: “do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent had”. However, this ignores the concept of risk.

One might classify agents as *programmer* agents, *network* agents, and *user* agents. The latter do interface stuff, retrieve information, and mediate between the user and the system. Network agents are mobile. Programmer agents are large objects.

Example applications: legacy systems (as wrappers), process control, information retrieval, electronic marketplaces, games and simulation. Simple idea is a typical business application. A bookseller has a collection of books. This is largely a database problem, but with the agents to negotiate or interact with suppliers. A similar idea is for a factory with several plants. When supplies are needed, agents can automatically scan the databases at each facility, make a decision about procurement, and then negotiate or interact with the suppliers.

5.2 Agent Design

Agents’ internal structure can grow. Some would be purely reflex. A common character is a state-based idea. If we add some “intelligence” to this, we can have goal-directed agents and utility-based agents which have degrees of success.

For building agents, most would use a *deliberative architecture*. Think all the software-engineering process for OOP and add to it. The add-on includes *BDI*: belief, desire, intention framework. In contrast, the *reactive architecture* emphasizes

emergent behavior. In this approach, the programmer provides methods that do primitive things, and by layering or learning, the agents gain the ability to perform more complex tasks.

Agent communication can be in an agent-oriented communication language somewhere between a network protocol and a database query language. FIPA, for example, identified 20 communicative acts, such as Propose, Refuse, Confirm, NotUnderstood. (Remember that language is not just informative but task oriented.) Some have proposed *stigmergy*: the act of one agent alters the environment so that other agents know. Implementations include JADE, JACK (both built on Java) and OAA.

Modeling includes UML with extensions. Use-cases and requirements modeling. One thing that plays a much larger role in agents in interactions. So for example, much work on bidding problems and protocol diagrams, which lead to a state-based agent. But maybe standard OOP is not right.

5.3 Auctions

One of the tasks proposed for agents is to conduct price negotiations on behalf of the user. This is a very rich and interesting subject. We will just scratch the surface with the discussion of sealed-bid auctions, which have been studied in economics.

The most natural *sealed-bid auction* is for the participants to state what they will bid (simultaneously and in secret) and the highest bid gets the item. But there is a winner's curse: winning suggests you value the item higher than anyone else—and maybe they're right. Furthermore, there is incentive to under-bid: if you get it, it's cheaper, and if you don't, maybe it wasn't worth what you said.

Consider in contrast a normal open escalating auction. The auctioneer ratchets up the price until only one person is left. The price they pay is the highest the second-place person was willing to pay.

So there is an idea for a *second-price sealed-bid auction*: everybody bids simultaneously and in secret as before; the highest bidder gets the item, but they only pay what the second-highest bid was. Perhaps surprising is that it can be shown that sincere bidding is a *Nash equilibrium* for this version: no player can benefit from unilaterally deviating from the strategy of accurate pricing (provided private data).

Of course, both these approaches lend themselves easily to collusion among the bidders.