Logic-based Agents and Softbots

Jane Hsu

Logic-Based Architectures

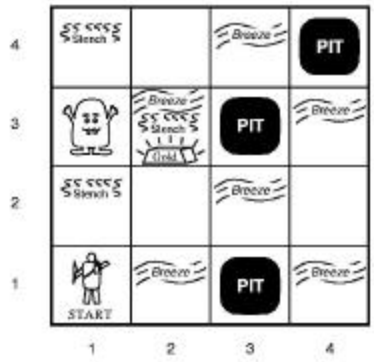
Decision making is realized through logical deduction.

It is the *traditional AI* (e.g. symbolic AI) approach ¡ intelligent behavior can be created in a system that manipulates symbols.

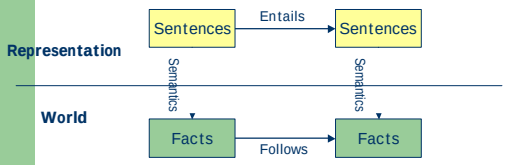
- ¡ Physical Symbol System Hypothesis

Agents as theorem provers.

The Wumpus World



Knowledge Representation



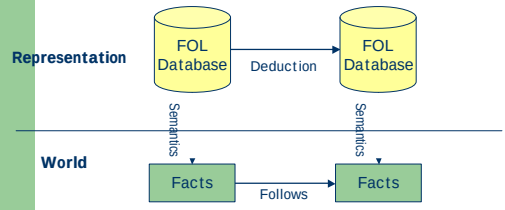
Syntax ¡ describes the possible configurations that can constitute sentences.

Semantics ¡ determines the facts in the world to which the sentences refer.

KR: Languages

Pictures, drawings
Natural language sentences
Logic: propositional, first-order
Rule-based: production systems, frames
Graphical: semantic nets, conceptual graphs
Probabilistic: Bayesian networks, fuzzy sets
Non-symbolic: neural networks
Genetic: chromosomes

Logic-based Representation



Deliberate Agents

Internal state: a database of FOL formulae

- Open(valve221)
- Temperature(reactor4726, 321)
- Pressure(tank776, 28)

Decision making is modeled as deduction rules.

see: $S \rightarrow P$

next: $D \times P \rightarrow D$

action: $D \rightarrow A$

Reactive Architectures

Decision making is implemented in some form of direct mapping from situation to action.

- Rejection of symbolic representations
- Intelligent behavior is NOT disembodied ¡ it has to be a product of the interaction the agent maintains with its environment.
- Intelligent behavior emerges from the interaction of various simpler behaviors.

BDI Architectures

Decision making depends upon the manipulation of data structures representing the beliefs, desires, and intentions of the agent.

Practical reasoning

- Deliberation: what goals we want to achieve
- Means-ends analysis: how to achieve those goals
- E.g. What are you going to do after college?

Intentions

Intentions drive *means-ends reasoning*.

Intentions constrain future deliberation.

Intentions persist.

Intentions influence beliefs upon which future practical reasoning is based.

The BDI Model

Belief revision function (brf)

Beliefs

Generate options

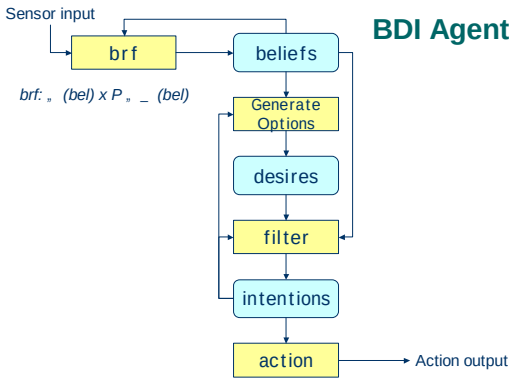
Desires

Filters

Intentions

Actions

BDI Agent



Layered Architectures

Decision making is realized via various software layers, each of which is more-or-less explicit reasoning about the environment at different levels of abstraction.

Information Herbivores

Massive memory and network resources required

Amortized over millions of queries per day

Minimal cycles devoted to each individual

No memory of previous requests

Least common denominator service

No Time for Intelligence

Softbots: The Problem

Problem: information explosion

Approach: deploy softbots on the web

Softbots: an intelligent program that uses software tools on a person¡s behalf.

- ¡ Software: not physical, not simulated
- ¡ Active: unlike the Unix Consultant
- ¡ Integrated: unlike interface agents, Microsoft wizards

Softbots are information carnivores!

Domain

The software robot uses a Unix shell and the World Wide Web to interact with a wide range of Internet resources.

Effectors: ftp, telnet, mail etc.

Sensors: archie, gopher, finger, netfind, etc.

Softbots as Interface Technology

Goal-oriented: person says *what*, softbot responsible for *how* and *where*.

Integrated: a uniform interface, leveraging existing services.

Expressive: additional expressive power beyond underlying tools.

Tolerant: softbot attempts to decipher incorrect or incomplete requests.

Example: find Bill¡s e-mail address.

Sample Task Request

Send the budget memos to Mitchell at CMU.

```
(forall (?d :in files)
     (if (and (file.type ?d memo.document)
              (subject.of.doc ?d ¡budget¡)
              (not (string.in.file ¡draft¡ ?d))
          (delivered.to ?d ?obj341)))
```

Disambiguation

Human requests are usually incompletely specified, potentially ambiguous, or even impossible to satisfy.

- Which Mitchell was intended?
- What if there is no Mitchell at CMU?
- Does the softbot know all the Mitchells at CMU?
- Which documents should be sent?
- Where are they located?
- How should softbot transmit the memos?
  e.g. email, fax, remote printing, etc.
- What if the memos are confidential?
- What if Mitchell is out of town?

Potential Solutions

Consult its knowledge base

Search for individuals or objects on the Internet matching a given description

- Access a single resource that provides such information
- Form a plan to seek out matching individuals

Infer based on the documents being sent

Infer based on the context of the request

Ask the human to further constrain the specification

Service Explosion

Airline reservations, stock prices, map servers, White Pages, package tracking, electronic commerce, job listing etc.

What is available at each site?

How to integrate multiple services?

How to choose between competitors?

- Quality, scope, reliability etc

How to discover new services?

How to insulate users from details?

Rodney: The Internet Softbot

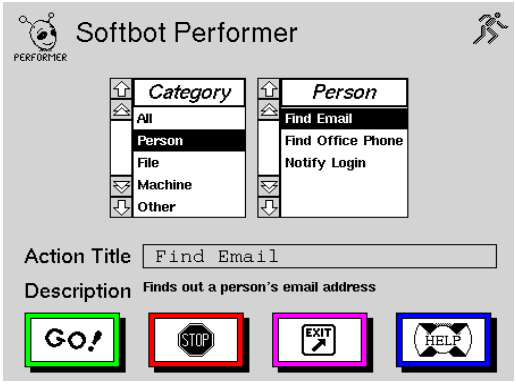
Declarative representation of software tools

Human requests translated into planner goals

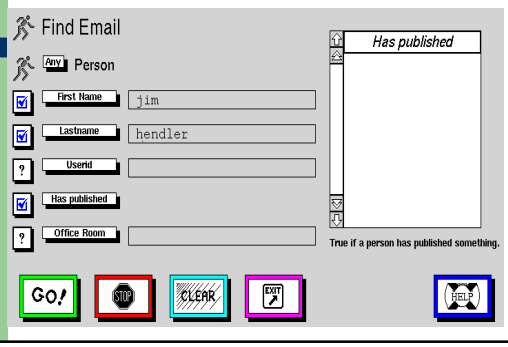
XII Planner generates and executes a sequence of software ¡actions¡

Sound and efficient closed-world reasoning.

A softbot is worth a thousand scripts.



Find Email GUI



The Monitor Window

```
ANSWER   1                    INSP
RN   CCCC: 1043-0989/93/$03.00.
CP   USA.
CC   C6170K.  C6110P.  C6140D.  C1230.
DE   case-based-reasoning.  frame-based-representation.  parallel-languages.
     parallel-programming.  planning-artificial-intelligence.
ID   case-based planning.  CaPER system.  case-based planner.  plan re-use.
     serial retrieval procedures.  indexed memory.  massively parallel
     frame-based AI language.  PARKA.  fast retrieval.  indexing.  very large
     casebases.  case retrieval.  memory retrieval.
AB   In case-based planning (CBP), previously generated plans are stored in
     memory and can be reused to solve similar planning problems in the
     future.  CaPER system is a case-based planner that is being developed to
     take advantage of the efficiencies of plan re-use while addressing some
     of the problems and limitations of case-based planners that use serial
     retrieval procedures on an indexed memory.  CaPER uses a massively
     parallel frame-based AI language (PARKA) and can do extremely fast

PRESS ENTER FOR NEXT PAGE OR ENTER A COMMAND -->


Search  Expand  QuickPrint  FileDirect  PreviousPage
NextOccurrence  PreviousOccurrence  ShortDisplay  Display  LimiT
SORT  Review  SaVequery  Change  Help  Quit-INSP                    EDISP_
```

The Planner Window

```
Session Status
   Plans Searched:      167    (on this problem: 80)
   Commands Executed:    16    (on this problem:  2)
   Facts in Model:      100
   Queries Answered:    564

Planner Status
   Plan Length: 7
   Number of Flaws: 7
   Executing Step: <NETFIND-PERSON>
   Working on Find Email.
   Finding the possible people.
```

Monitor netfind

```
        Office:     3267                        Office phone: 405-2696
        Home phone: 871-2210                    Associate Prof.
        Directory:  /fs/dextrose/hendler        Shell: /bin/csh
        Last login Thu Sep 22 14:41 on ttypa from dormouse.cs.umd.
        No plan.
        Mail is forwarded to hendler@dormouse.cs.umd.edu:
        [dormouse.cs.umd.edu]
        Login name: hendler                     In real life: Jim Hendler
        Office:     3135                        Office phone: 405-2696
        Home phone: 871-2210                    Associate Prof.
        Directory:  /dormouse2/hendler          Shell: /usr/imports/bin/tcsh
        Last read mail at Jan 31 17:43:37 1995
        On console since Jan 25 17:46:17: idle 6 days, 2 hours
        On ttyp0 since Jan 25 17:48:32 from unix:0.0: idle 2 hours, 54 minutes
        On ttyp1 since Jan 25 17:48:32 from unix:0.0: idle 6 days, 2 hours
        On ttyp2 since Jan 25 17:48:33 from unix:0.0: idle 6 days, 2 hours
        On ttyp3 since Jan 25 17:48:33 from unix:0.0: idle 2 hours, 5 minutes
        Last login Tue Jan 31 16:04 on ttyp4 from treacle.cs.umd.e
        Plan: No one ever lost an election by underestimating the stupidity of t
he
        American people.
         -- Anon. 1994
```

Query Results

The Email address of Jim Hendler is hendler@ringding.cs.umd.edu

Dismiss

- Search the INSPEC bibliographic database
- Find a publication by the person
- Use Netfind for where they are located
- Use XII Planner to decide the actions
- Get the search result

Closed-World Reasoning

Problem: reasoning with incomplete information

Blocks world solution:

- the infamous closed-world assumption

Recent work: open-world assumption

Softbot solution: sound, polynomial time closed-world inference and update.

Example: find all the non-stop United flights from Detroit to Portland (Etzioni, Golden, Weld, AIJ)

Softbot Family Tree