Lecture #3B Developing Agent Systems, Cognitive Architectures, and Adaptive Programming

- Developing Agent Systems
- Paper discussion – cognitive architectures (Brachman, DARPA, 2002, Systems that know what they’re doing)
- Paper discussion – Adaptive software

Developing Agent Systems

- Agent Architecture
  - Software engineering models of agents
  - (Deliberative, Reactive, Hybrid)
- Agent Language
  - Software systems for programming and experimenting with agents. More in future lectures!

Multiagent Architectures

1. Direct communication
   - Contract-net approach
     - (requests for proposal, bids, contracts)
   - Specification approach
     - (supplying other agents with its capabilities and needs)
   - Advantages: no reliance on the existence, capabilities, or biases of any other programs
   - Disadvantages: too costly when too many agents used, implementation complexity

2. Assisted coordination
   - a federated system (facilitators)

FIPA’s Agent Platform

- Foundation for Intelligent Physical Agents
- Agent management system (supervisory control like creation, deletion, suspension, resumption, authentication, migration, etc.)
- Directory facilitator (agent existence, service, capabilities, protocols, etc... It’s like “yellow pages”)
- Agent name server (an index of all agents resident on a platform, mapping between globally unique agent name and local transport addresses)
- Agent communication channel (communication method connecting all agents in an AP and between AP’s, routing of messages)

Agent Languages

- Specification language
- Scripting (programming) language
  - AOP (agent-oriented programming), PLACA (planning communication agents), Java, Telescript/language, engine, protocol), Tcl/Tk
- Agent shell
  - Jess, DYNACLIPS, AGENT_CLIPS, KAPICLIPS
``Agentification” of Old Programs

- Transducer
- Wrapper
- Rewrite
- Proven useful already!

Cognitive Architectures

- Paper discussion - Brachman, DARPA, 2002
- Systems that know what they’re doing
- [http://www2.hawaii.edu/~nreed/ics606/papers/brachman02.pdf](http://www2.hawaii.edu/~nreed/ics606/papers/brachman02.pdf)

Systems That Know What They're Doing

- Paper discussion - Brachman, DARPA, 2002
- Systems that know what they’re doing
- [http://www2.hawaii.edu/~nreed/ics606/papers/brachman02.pdf](http://www2.hawaii.edu/~nreed/ics606/papers/brachman02.pdf)

Why Build Adaptive Software?

- Adaptive Software, Norvig & Cohn
- Software Engineering Problems
  - Time-consuming and costly to develop
  - Brittle when used (e.g. crashes)
- Software Engineering Solutions
  - 1970s – structured programming
  - 1980s – object-oriented programming
  - Today – adaptive programming

What is Adaptive Programming?

- Applications that can adapt to
  - Changing user needs
  - Changing environments
  - Explicitly represents actions and goals
- Anticipate change (it will happen)
- Common Lisp is ideal for adaptive programming

Summary

- Agent Development
- Cognitive Architectures for Agents
- Adaptive Software