COSC 4401/5326 Fall 2004

Introduction to Intelligent/Autonomous Agents and Multiagent Systems

Yves Lespérance

Readings: Wooldridge, *Introduction to Multiagent Systems*, ch. 1, ch. 2 sec. 1–5.

What's an Agent?

"Upon logging in to your computer, you are presented with a list of email messages, sorted into order of importance by your personal digital assistant (PDA). You are then presented with a similar list of news articles; the assistant draws your attention to one particular article, which describes hitherto unknown work that is very close to your own. After an electronic discussion with a number of other PDAs, your PDA has already obtained a relevant technical report for you from an FTP site, in the anticipation that it will be of interest." [Wooldridge & Jennings 95]

Core Attributes of Agents

Autonomy: act independently without (constant) human intervention.

Reactivity: perceive their environment and respond in timely manner to relevant changes.

Pro-activeness/goal-directedness: do not just react, but take the initiative and act to achieve their goals.

Social-ability: can interact/collaborate with other agents and users; often includes use of an agent communication language based on speech acts (inform, request, promise, etc.), e.g. KQML (Knowledge Query & Manipulation Language) or FIPA (Foundation for Intelligent Physical Agents) ACL.

System viewed as a *society* of interacting agents.

Several Notions of Agent

On a continuum from light to heavy-duty:

- Mobile agents: processes that can move from host to host as they execute (Aglets, Agent tcl, Telescript).
- Autonomous agents: long-lived processes that assist the user without necessarily using AI techniques; often required to have some of the core agent attributes.
- Processes in a complex system that are usefully described as having mental states such as beliefs, goals, intentions, etc.; agency as an abstraction mechanism.

- Processes that can communicate in an agent communication language based on speech acts (e.g. KQML, FIPA ACL); related to the above, since speech act communication requires agents to communicate as if they had mental states.

- Intelligent agents: processes that maintain symbolic representations of their beliefs, goals, etc., and chose how to act by performing reasoning.

In this course, focus on intelligent agents.

Why the Interest in Agents-Based Approaches?

Agents-Based Approaches appear to offer solutions to the many new challenges faced by information infrastructure.

With growth in distributed computing, Internet, WWW, e-commerce, etc., more and more applications must operate in *open systems*, where the components change over time, and there is a lot of information available from multiple sources, much of it unstructured.

Agent communication languages and protocols, matchmaker and facilitator agents, could be the glue that allows these applications configure themselves to work together dynamically.

Ontology broker and interpreter agents could allow applications (databases) to understand each other even though they use different terminology.

Agents technology can support the automation of web services, the “Semantic Web”.

Types of Work on Individual Agents

Work on the internal (micro) aspects of agents can be divided in 3 areas:

Agent theories: formal specifications of agent properties — what kind of mental states they have and how they are related to each other and to action; should support reasoning about agents.

Agent architectures : how to design and implement systems that have these properties — planning, decision making, belief update, etc., and their integration.

Agent Programming Languages: languages for programming agents and their implementation.

Types of Work on Multiagent Systems

Work on the external (macro) aspects of agents can be divided in 4 areas:

Protocol Design & Negotiation: mechanisms for reaching agreements between competitive agents; often involves auctions; based on game and economic theory; analysis of protocol properties.

Task Allocation and Coordination: mechanisms for allocating tasks and sharing results, multiagent planning, ensuring that team members stay coordinated.

Agent Communication Languages: speech acts languages, content languages, ontologies.

Types of Work on Multiagent Systems (cont.)

Multiagent Platforms/Infrastructures: languages and tools for programming multiagent systems (e.g. SRI's Open Agent Architecture (OAA) platform, Telecom Italia's JADE platform which is FIPA compliant); basic services for MAS (e.g. networking, yellow pages, etc.).

Applications of Agents

Can be divided into:

- distributed systems,
- personal assistants.

Distributed sytems applications:

- Workflow and business process management; e.g. ADEPT [Jennings et al 96] for telecom network quote generation, uses PRS for plan execution, CORBA & for ARCHON for communication, not a lot of AI.
- Industrial systems management; e.g. ARCHON [Jennings et al 95] for power management, etc., has agent wrappers around legacy software that provide planning, representation, & communication.

Applications of Agents (cont.)

- Air traffic control; e.g. OASIS [Steeb et al 88,Kinny et al 96], has agents for each aircraft and some global scheduling agents, uses PRS.
- Distributed sensing; e.g. DVMT [Lesser & Erman 80] for tracking and guiding (military) vehicles using multiple sensors.
- Cooperative information system & digital libraries; e.g. Carnot [Huhns et al 92], InfoSleuth health info access system [Bayardo et al 97], [Durfee 97], [Zuno 97].
- Multi robot control; e.g. RoboCop soccer [RoboCup 01].
- Social simulation; e.g. EOS project [Doran & Palmer 95].

Applications of Agents (cont.)

Personal assistant applications:

- Personal information agents; e.g. MAXIMS [Maes 94] for email handling & filtering, learns email handling rules by watching user; e.g. NewT for news filtering [Maes 94] and [Etzioni & Weld 94]'s Internet softbot.
- Comparison shopping agents; e.g. Jango [Doorembos et al 97], has vendor description learner and comparison shopping component.
- Auction bots; e.g. Kasbah [Chavez & Maes 96].
- Interface agents, often supporting speech or natural language input, multimedia, mobile computing, cooperative work; e.g. e.g. Mailtalk [Cohen et al 94], COLLAGEN [Rich & Sidner 97].

Applications of Agents (cont.)

- Believable agents for games and virtual realty; e.g. OZ project [Bates et al 92], [Hayes-Roth et al 95], [Cassel et al 94].
- Robot control; e.g. Rhino museum guide robot [Burgard et al 98]
- Spacecraft control; e.g. NASA Deep Space 1 [Muscettola et al 98].

See Wooldridge's Intro to MAS ch. 11, Parunak's chapter in Weiss's MAS book, Jennings & Wooldridge's Agent Technology, Huhns & Singh's Readings in Agents.

Cognitive Robotics

Theory and implementation of agents that reason, act and perceive in a changing, incompletely known, unpredictable environment.
Reasoning about:

- Goals
- Actions
- When to perceive and what to look for
- The cognitive states of other agents.

Cognitive robotics is concerned with integrating reasoning, perception and action within a uniform theoretical and implementation framework.

Toronto Cognitive Robotics Framework

Includes:

- logical model of agency that integrates reasoning, action, & perception, based on situation calculus;
- Golog, a high-level agent programming language;
- ConGolog, version of Golog with concurrent processes, etc.;
- IndiGolog, version of ConGolog with incremental execution for implementing embedded agents;
- prototype implementations in Prolog.

Joint work with Levesque and Reiter at U. of Toronto, De Giacomo at U. of Rome, & others.

Applications Developed

- Robot control: ConGolog and IndiGolog interfaced with navigation software running on various robots and used to program high-level controllers; tested in mail delivery applications; currently working on planning and operation in dynamic environments, multi-robot coordination, intelligent control of vision systems.
- Tools for implementing MAS with intelligent agents: igOAAlib library for including IndiGolog agents in OAA-based MAS, igJADE library for including IndiGolog agents in JADE-based MAS.
- Formalisms and tools for agent-oriented software engineering: modeling of business processes (e.g. e-commerce) and telecommunication applications with multiple agents and mental states; use in requirements engineering; tools for simulation and verification.

Applications Developed (cont.)

- Agent-based web service composition/customization/ delivery: air travel assistance demo; uses IndiGolog + JADE + PKS planner.

- Intelligent software agents:
  - meeting scheduling tool
  - home banking tool
  - CAD assistant

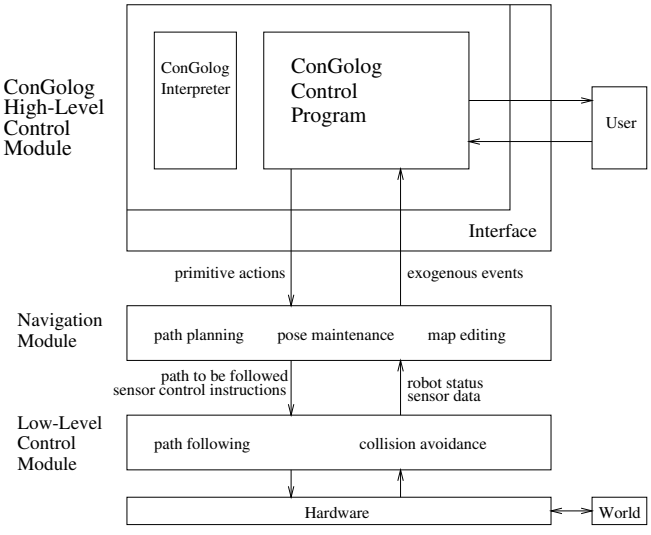
Example: High-Level Controller for Mail Delivery Robot [Lespérance et al. 99]

High-level control module must react to two kinds of exogenous events:

- new shipment orders $orderShipment(sender, recipient, priority)$; when a new order comes in, it must be acknowledged immediately; if it has higher priority than the order the robot is currently serving, service of the current order must be aborted;

- signals from the navigation module, i.e. the $reachDest$ event announcing that the destination has been reached and the $getStuck$ event announcing that the robot has failed to reach its destination; when the robot successfully arrives at a customer's mailbox, it performs the pickups and deliveries; when it fails to get to the customer's mailbox, it decrements the customer's "credit" and if credit gets down to 0, the customer is notified he will no longer be served.

A Hierarchical Robot Control Architecture



ConGolog Mail Delivery Controller

The main procedure implementing the high-level controller concurrently executes 4 interrupts at different priorities:

proc $mainControl$
$\langle\, n : orderState(n) = JustIn$
$\rightarrow handleNewOrder(n)\,\rangle$
$\gg$
$\langle\, n : (orderState(n) = ToPickUp$
$\wedge\, Suspended(sender(n)))$
$\rightarrow cancelOrder(n)\,\rangle$
$\|$
$\langle\, n : (orderState(n) = ToPickUp$
$\vee\, orderState(n) = OnBoard$
$\vee\, robotPlace \neq CentralOffice)$
$\rightarrow robotMotionControl\,\rangle$
$\gg$
$\langle\, robotState = Moving \rightarrow$ **noOp** $\,\rangle$
endProc

ConGolog Mail Delivery Controller (cont.)

The third interrupt in $mainControl$ handles the robot's navigation, pick ups, and deliveries. It calls:

proc $robotMotionControl$
  if $\exists c\, CustToServe(c)$ **then** $tryServeCustomer$
  else $tryToWrapUp$;
  endIf
endProc

ConGolog Mail Delivery Controller (cont.)

To try to serve a customer, we execute:

proc $tryServeCustomer$
  $\pi\, c\, [BestCustToServe(c)?;$
    $startGoTo(mailbox(c));$
    $(robotState \neq Moving)?;$
    if $robotState = Reached$ **then**
      $freezeRobot;$
      $dropOffShipmentsTo(c);$
      $pickUpShipmentsFrom(c);$
      $resetRobot$
    else if $robotState = Stuck$ **then**
        $resetRobot;$ % abandon attempt
        $handleServiceFailure(c)$
    % else when service aborted nothing more to do
    endIf]
endProc

References

Wooldridge, M., *Introduction to Multiagent Systems,* Ch. 1, Ch. 2 Sec. 1 to 5, and Ch. 11.

Wooldridge, M. and Jennings, N.R., Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, **10** (2), 115–152, 1995.

Huhns, M.N. and Singh, M.P. *Readings in Agents,* Chapters 1 and 2.

Jennings, N.R. and Wooldridge, M., Applications of Intelligent Agents, in Jennings, N.R. and Wooldridge, M. (Eds.), *Agent Technology: Foundations, Applications, and Markets.*

Communications of the ACM, Vol. 37, no. 7, July 1994, special issue on Intelligent Agents.

Communications of the ACM, Vol. 42, no. 3, March 1999, special issue on Multiagent Systems on the Net and Agents in E-commerce.

IEEE Intelligent Systems, Vol. 16, no. 2, March-April, 2001, special issue on Agents and the Sematic Web.

Lespérance, Y., Tam, K. and Jenkin, M. Reactivity in a Logic-Based Robot Programming Framework. In Jennings, N.R. and Lespérance, Y., editors, *Intelligent Agents Volume VI - Agent Theories, Architectures, and Languages, 6th International Workshop, ATAL'99, Orlando, FL, USA, July 15-17, 1999, Proceedings,* 173-187, LNAI 1757, Springer-Verlag, Berlin, 2000.