Introduction to Autonomous Agents and Multi-Agent Systems

Michael Luck
Dept of Electronics and Computer Science
University of Southampton, UK.
mml@ecs.soton.ac.uk
http://www.ecs.soton.ac.uk/~mml

Tutorial Outline

- Agents
  - What are they?
  - Why are they a good idea?
- Agent Architectures
  - Deliberative (especially BDI models)
  - Hybrid
  - Reactive
- Agent Interactions
- Agent Resources

Remote Agent Experiment (RAX)

- Deep Space One mission to validate technologies
- AI software in primary command of a spacecraft

RAX

- Comprises
  - planner/scheduler to generate plans for general mission goals
  - smart executive to execute plans
  - Mode identification and recovery to detect failures
- Goals not pre-planned so more flexible
- Tests include simulated failures
- Tests in May 1999

Agents

- Relatively new field (10 years?)
- Dramatic growth
- Popularity
- Increasing numbers of applications
- Multi-disciplinary
- Problems:
  - Agent backlash?
  - Sound conceptual foundation?

Agent Definitions

- Smith et al: “persistent software entity dedicated to a specific purpose”
- Selker: “computer programs that simulate a human relationship by doing something that another person could do for you “
- Riecken: “integrated reasoning processes”

CACM, July 1994
... and more
- anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors
  - Russell and Norvig
- An autonomous agent … senses that environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future.”
  - Franklin and Graesser

Why agents?
- Increasingly difficult to deal with large-scale information systems using traditional software:
  - distributed and open, lacking central control and standardised communication;
  - heterogeneous: compatibility and interfacing problems;
  - rapid change: new subsystems appear, existing ones disappear
  - rapid growth: huge amount of unstructured information;
  - human involvement: sophisticated interaction and cooperation.

Agent Types
- Software agents
- Interface agents
- Personal assistant agents
- Believable agents
- Electronic mail agents
- Information agents
- Teaching agents

Application Areas
- Agent monitoring of web sites
- Agent filtering of email and newsgroups
- Personal information management
- Electronic marketplaces
  - “an agent is a credit card with an attitude” - Richard Sharpe
- Negotiation between and within organisations

Agent Dimensions
- Reactivity
- Pro-activeness
- Autonomy
- Rationality
- Benevolence
- Veracity
- Temporal continuity
- Adaptability
- Mobility
- Social ability

Lack of Agreement
- Does it matter?
- Richness aids acceptance
- Broad range of applicability
- Cross-fertilising subfields
- Lack of precision
- Abuse of terminology
Weak Notion of Agents

- Four key qualities:
  - Autonomous: function without intervention
  - Proactive: goal-directed behaviour
  - Reactive: perceive and respond to changing environment
  - Social ability: interaction with others

- Wooldridge and Jennings, 1994/1995

Strong notion of agents

- In addition to the weak notion, also uses mental components such as
  - belief
  - desire
  - intention
  - knowledge
  - etc

Objects

Agents

Autonomous Agents

Agent

"encapsulated computer system, situated in some environment, and capable of "flexible autonomous action in that environment in order to meet its design objectives" (Wooldridge)

- control over internal state and over own behaviour
- experiences environment through sensors and acts through effectors
- reactive: respond in timely fashion to environmental change
- proactive: act in anticipation of future goals
Multiple Agents
In most cases, single agent is insufficient
- no such thing as a single agent system (!?)
- multiple agents are the norm, to represent:
  - natural decentralisation
  - multiple loci of control
  - multiple perspectives
  - competing interests

Agent Interactions
- Interaction between agents is inevitable
  - to achieve individual objectives, to manage inter-depencies
- Conceptualised as taking place at knowledge-level
  - which goals, at what time, by whom, what for
- Flexible run-time initiation and responses
  - cf. design-time, hard-wired nature of extant approaches

Organisations
Agents act/interact to achieve objectives:
- on behalf of individuals/companies
- part of a wider problem solving initiative

underlying organisational relationship between the agents

Organisations
This organisational context:
- influences agents' behaviour
  - relationships need to be made explicit
    - peers
    - teams, coalitions
    - authority relationships
- is subject to ongoing change
  - provide computational apparatus for creating, maintaining and disbanding structures

A Canonical View

Decomposition: Agents
- In terms of entities that have:
  - own persistent thread of control (active: “say go”)
  - control over their own destiny (autonomous: “say no”)
- Makes engineering of complex systems easier:
  - natural representation of multiple loci of control
    - “real systems have no top”
  - allows competing objectives to be represented and reconciled in context sensitive fashion
Decomposition: Interactions

- Agents make decisions about nature & scope of interactions at run time
- Makes engineering of complex systems easier:
  - unexpected interaction is expected
  - not all interactions need be set at design time
  - simplified management of control relationships between components
  - coordination occurs on as-needed basis between continuously active entities

<table>
<thead>
<tr>
<th>Complex System</th>
<th>Agent-Based System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-systems</td>
<td>Agent organisations</td>
</tr>
<tr>
<td>Sub-system components</td>
<td></td>
</tr>
<tr>
<td>Interactions between sub-systems and sub-system components</td>
<td></td>
</tr>
<tr>
<td>Relationships between sub-systems and sub-system components</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complex System</th>
<th>Agent-Based System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-systems</td>
<td>Agent organisations</td>
</tr>
<tr>
<td>Sub-system components</td>
<td></td>
</tr>
<tr>
<td>Interactions between sub-systems and sub-system components</td>
<td></td>
</tr>
<tr>
<td>Relationships between sub-systems and sub-system components</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complex System</th>
<th>Agent-Based System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-systems</td>
<td>Agent organisations</td>
</tr>
<tr>
<td>Sub-system components</td>
<td></td>
</tr>
<tr>
<td>Interactions between sub-systems and sub-system components</td>
<td></td>
</tr>
<tr>
<td>Relationships between sub-systems and sub-system components</td>
<td></td>
</tr>
</tbody>
</table>

Agents Consistent with Trends in Software Engineering

- Conceptual basis rooted in problem domain
- Increasing localisation and encapsulation
- apply to control, as well as state and behaviour

Exploit mechanisms for representing & managing organisational relationships
Structures for modeling collectives
Agents Consistent with Trends in Software Engineering

- Conceptual basis rooted in problem domain
- Increasing localisation and encapsulation
- Greater support for re-use of designs and programs
  - whole sub-system components (cf. components, patterns)
  - flexible interactions (cf. patterns, architectures)
- e.g. agent architectures, system structures
- e.g. contract net protocol, auction protocols

Agents Support System Development by Synthesis

An agent is a stable intermediate form
- able to operate to achieve its objectives and interact with others in flexible ways
- construct “system” by bringing agents together and watching overall functionality emerge from their interplay
- well suited to developments in:
  - open systems (e.g. Internet)
  - e-commerce

Single Agent Architectures

- Deliberative Agent Systems
  - Symbolic representation and manipulation
  - IRMA, GRATE, PRS/dMARS
- Reactive
  - Stimulus-Response Agent Systems
  - Subsumption Architecture
  - Agent Network Architecture
- Hybrid Agent Systems
  - Act both deliberatively and reactively
  - TouringMachine
  - InterRRaP

Towards BDI Architectures

- BDI aims to model rational or intentional agency
- The symbols representing the world correspond to mental attitudes
- Three categories:
  - informative (knowledge, belief, assumptions)
  - motivational (desires, motivations, goals)
  - deliberative (intentions, plans)

BDI Systems

- BDI = Belief, Desires and Intentions
- Many agent architectures are BDI based
- Original system was PRS
- More recent versions include dMARS.
- Other related systems include AgentSpeak(L) and Agentis
Folk Psychology

- I believed the tutorial today was at 8:30am so I intended to arrive yesterday from London.
- I believed the planes were not delayed and desired not to be late so I intended to arrive by 6pm.
- Compelling because
  - familiar: what it wants, knows and intends - easier to understand and predict behaviour.
  - Other agents can understand and predict behaviour
  - Relationship between these three categories may give us a handle on intelligent action in general.

BDI Architectures

- Beliefs - modelling world state.
- Desires - choice between possible states.
- Intentions - commitment to achieving particular state.

PRS/dMARS

- Beliefs: information about the world
- Goals: tasks to achieve
- Plan library: procedural knowledge
- Intentions: partially instantiated selected plans

Procedural Reasoning System (PRS)

PRS Architecture

- In general, an agent cannot achieve all its desires.
- Must therefore fix upon a subset.
- Commit resources to achieving them.
- Chosen desires are intentions.
- Agents continue to try to achieve intentions until either
  - believe intention is satisfied, or
  - believe intention is no longer achievable.

PRS Plans

- BDI model is operationalised in PRS/dMARS agents by plans.
- Plans are recipes for courses of action.
- Each plan contains:
  - invocation condition: circumstances for plan consideration;
  - context, circumstances for successful plan execution;
  - maintenance condition: must be true while plan is executing, in order for it to succeed; and
  - body: course of action, consisting of both goals and actions.
PRS Plan Structure

1. Plan
   - Invocation
   - Context
   - Body
   - Maintenance
   - Success
   - Failure

2. PRS Operation 1
   - Observe world and agent state, and update event queue to reflect observed events.
   - Generate new possible goals (tasks), by finding plans whose trigger matches event queue.
   - Select matching plan for execution (an intended means).

3. PRS Operation 2
   - Push the intended means onto the appropriate intention stack in the current set.
   - Select an intention stack and execute next step of its topmost plan (intended means):
     - if the step is an action, perform it;
     - if it is a subgoal, post it on the event queue.

4. Applications
   - Air-traffic control
   - spacecraft systems
   - telecommunications management
   - air-combat modelling

5. Theoretical BDI models
   - Theories to understand the relationship between the attitudes (plus time, plus control)
   - Modal Logics are used with abstract semantics
   - No concrete link between logic and system.
   - How can you tell whether a system is an embodiment of axioms of BDI?
   - Most BDI specifications are high level and are not easy to implement directly.
   - Relationship between theory and system intuitive only.

6. Reactive Architectures
   - Subsumption
   - Agent Network Architecture
The Subsumption Architecture

- Task-achieving behaviours
- More specific tasks at higher-levels
- Build each level separately until it works
- Higher levels intermittently cut in - lower levels unaware that higher levels influence the behaviour.
- Agent functions at early stage of development
- Influenced design of many architectures.
- No explicit reasoning!

Agent Network Architecture

- Collection of competence modules
- Each competes to control behaviour according to internal and external factors
- External: module activation, perception, goals
- Internal: by links:
  - activated modules increase activation along successor links
  - non-activated modules increase activation along predecessor links
  - all modules decrease activation of their conflictors

Hybrid Architectures

TouringMachines
InteRRaP

TouringMachines 1

- Designed for autonomous agents in dynamic worlds
- Three layers:
  - reactive layer - responds quickly to events not explicitly programmed in other layers
  - planning layer - generate, modify, execute (e.g. planning a route)
  - modelling layer - maintains models of environment, other agents and itself
Touring Machines 2
- Each layer directly connected to perception and action
- Any two layers can communicate
- Conflict between layers arises because each has incomplete view - architecture uses context-activated control rules.

Touring Machines

InterRRaP
- Layered hybrid architectures support:
  - Modelling environment at different abstraction levels
  - Different levels of responsiveness
  - Different levels of knowledge and reasoning required
- In a vertically-layered architecture only adjacent layers can communicate:
  - Behaviour-based layer (domain specific)
  - Plan-based layer (non-social goal-directed behaviour)
  - Cooperation-based layer (social behaviour - e.g. joint plans)

Agent oriented programming
- Demonstration of Shoham’s notion of agent oriented programming
- Programming paradigm based on societal view of computation
- Program agents in terms of intentional notions of e.g. belief, commitment, intention, …
- Intentional stance is useful for representing complex systems

AOP
- Three components:
  - Logic for specifying agents and mental state
  - Interpreted programming language for programming agents (AGENT-0, PLACA)
  - Process of agentification for representing other applications as agents.
Agents in AOP

- Agents in AGENT-0 have
  - capabilities
  - initial beliefs
  - initial commitments
  - commitment rules
- Rules are matched against messages received and current beliefs before taking action

Other Issues

- Interaction (multi-agent systems)
  - cooperation
  - communication (KQML, FIPA)
  - negotiation
  - protocols
- Standards
  - FIPA
  - OMG
- Mobility

Further Reading

The Contract Net Protocol
- The most common protocol between agents in both real applications and detailed simulations – Parunak
- Several efforts at extending CNP
- Several formalisations
- Used to demonstrate applicability of new theories and systems

Contract Net
- Agents dynamically create relationships in response to current processing requirements embodied in a contract.
- A node with a task to be achieved forms a contract with others who proceed to accomplish the task
- A contract is an agreement between a manager and contractor, resulting from the contractor successfully bidding for the contract.

Protocol Steps
- Task announcement from manager
- Nodes evaluate their suitability for task
- Bidding from potential contractors
- Manager ranks bids and awards contract to one or more contractors
- Manager monitors contractors, requests reports, integrates partial results

Stages of the CNP
- Sending and receiving announcements
- Bidding for contracts
- Making an award
Task Announcement

Distributed Sensing Example

- Task Abstraction Slot specifies identity and position of manager, enabling potential contractors to reply.
- Eligibility Specification specifies location and capabilities required by any bidders.
- Bid Specification indicates that a bidder must specify its position and sensing capabilities.

Task Evaluation and Award

- Nodes evaluate interest using task evaluation procedures specific to the problem at hand.
- Interested nodes submit bids.
- Manager selects nodes using bid evaluation procedures based on information in bid.
- Sends award messages to successful bidders.
- Contractors may subcontract parts of their task, to become managers.
- Contractors issue reports to the manager: interim, final.
- Manager terminates contract with message.

Further Reading

Cooperation

- Underpins multi-agent systems
- More than just coordinated simultaneous action
- Requires group intention
  - Cannot be same as individual intention, since beliefs are divergent
  - Problems if one member drops intention: group must also drop intention

Bratman’s Requirements for Cooperative Activity

- Mutual responsiveness: participants respond to others’ actions
- Commitment to joint activity (or cooperative intention)
- Commitment to mutual support
- Intentions should not be coerced
- Cooperative intentions should be common knowledge

Joint Intentions

- Belief that intention is no longer appropriate may lead to dropping goal.
- Need belief to be made known to group.
- Cohen and Levesque suggest use of a weak goal.
- Notion of joint persistent goal.

Joint Intentions

Cohen and Levesque

Joint intention is joint persistent goal to have knowingly performed an action or to have knowingly performed a sequence of events after which a goal is achieved.

Joint persistent goal is one held and mutually believed to be held by agents such that until it is mutually believed to be irrelevant, agents have a corresponding weak goal.

Agent has a weak goal if it has the goal or believes the goal is irrelevant and has the goal of making this mutually believed.

Stages in Cooperation

- Plan Selection
  - May be individual plan to achieve goal
  - May be group plan
- Intention Adoption
  - If plan is group plan, need to form cooperative intention among group
- Group Action
  - Coordination of individual contributions
  - Work of Kinny et al on Planned Team Activity

Further Reading

Agent Resources

AgentLink
- UMBC AgentWeb
- Journals
- etc

AgentLink
- European Commission funded
- 150 members at 1 January 2002
- Open to Europeans for full membership, others for associate membership
- AgentLink I: 1998-2000
- AgentLink II: 2000-2003

AgentLink website
- Website: www.AgentLink.org
  - People Finder
  - Agent Events
  - Teaching Curricula database
  - Papers clearinghouse
  - Software database
- Documents

AgentLink Publications
- Monthly email update
- AgentLink News (3 times a year)
  - Available from web
  - Print copies produced
- Irregular publications:
  - Books
  - Roadmap

European Agent Systems Summer School
- Utrecht in 1999
- Saarbruecken in 2000
  - 150+ participants from Europe, US, etc
- Prague in July 2001
  - 200 participants
- Bologna in July 2002
  - 150+ participants

UMBC Agent Web
- agents.umbc.edu
- Information
- Resources
- Mailing list
- Announcements
- Conferences
- Tim Finin and Yannis Labrou
Journals
- Autonomous Agents and Multi-Agent Systems (official journal of Autonomous Agents, ICMAS, AgentLink)
- Artificial Intelligence
- Knowledge Engineering Review
- IEEE Transactions on SMC

Conferences
- Autonomous Agents
- ICMAS
- ATAL
- AAMAS
- ESAW
- MAAMAW
- PRIMA
- CIA
- UKMAS

Mailing Lists
- Agents list
  - http://www.cs.umbc.edu/agentslist/
- DAI List
  - DAI-List-Request@ece.sc.edu
- AgentLink
  - Coordinator@agentlink.org

Understanding Agent Systems