CogSysI Lecture 11: Communication

Intelligent Agents
WS 2004/2005

Part III: Multi-Agent Systems

Communication
Multiagent Communication

- Competitive: mechanisms for collective decision making
  - (Voting)
  - Auction
  - Negotiation
  - (Argumentation)

- Cooperative: communication for distributed problem solving
  - Speech acts
  - Agent Communication Languages
  - Ontologies
Collective Decision Mechanisms

Design of a protocol

- Guaranteed success: ensuring that eventually an agreement can be reached
- Maximizing social welfare: total sum of utilities should be maximal
- Pareto efficiency: no other outcome where at least one agent is better off and none is worse off
- Individual rationality: following the protocols in the best interest of negotiation participants
- Stability: providing all agents with an incentive to behave in a particular way (e.g. Nash equilibrium)
- Simplicity: a participant can easily determine the optimal strategy
- Distribution: designed such that there is no 'single point of failure' and to minimize communication between agents
Auctions

- Online auctions are very popular
- simple interaction scenarios $\rightarrow$ easy to automate
- good choice as a simple way for agents to reach agreements, allocating goods, tasks, resources
- Auctioneer agent, bidder agents, a good
- private value vs. public/common value of goods,
- correlated value: value for private factors as well as other agents valuation
Dimensions of Auction Protocols

- Winner determination: first price (highest bid gets good for the bidded amount), second price (highest bidder gets the good, but for price of 2nd highest bid)
- Knowledge of bids: open cry, sealed-bid
- Bidding mechanism: one shot, ascending/descending bids in successive rounds

→ different types of auctions
English Auctions

- “Mother of auctions” (Sothebys)
- first-price, open cry, ascending (starting with a reservation price)
- Dominant strategy: successively bid a small amount more than the current highest bid until price reaches current valuation, then withdraw
- Winner’s curse: uncertainty about the true value (e.g. land speculation), winner might have overvalued the good
Dutch Auctions

- open-cry, descending
- auctioneer starts with an artificially high value
- decreasing value, until someone makes an offer
- no dominant strategy
- also susceptible to winner’s curse
First-Price Sealed-Bid Auctions

- first-price, sealed bid, one-shot
- simplest form of auction
- difference between second-highest and highest bid is wasted money
- best strategy: bid less than true valuation, how much less depends on the other agents, no general solution
Vickery Auctions

- second-price, sealed bid, one-shot
- dominant strategy: bid true valuation
- Because truth telling is the dominant strategy, this form is discussed much in multiagent literature
- BUT: counterintuitive for human bidders
- Possibility of antisocial behavior: own valuation 90 $, guess that another agent will pay 100 $, therefore bid 99 $ such that opponent needs to pay more than necessary
- Commercial situations: one company cannot compete directly but forces other company into bankruptcy
Issues for Auctions

- Expected revenue: Strategies for the auctioneer to maximize his revenue (for risk-neutral bidders, expected revenue is provably identical in all auction types)
- Lies and collusions: coalition between agents (bid small amounts, share win afterwards), place bogus bidders, ...
- Counterspeculation: costs time and money and is risky (compare with meta-level reasoning)
Negotiation

- Auctions are only concerned with allocation of goods.
- When agents must reach agreements on matters of mutual interest, richer techniques are required.
- Negotiation techniques for artificial agents (Rosenschein and Zlotkin, 1994).
Object vs. Agent Communication

- **Object** $o_2$ invokes a public method $m_1$ of object $o_1$ passing argument $arg$
  - $o_2$ communicates $arg$ to $o_1$
  - BUT: the decision to execute $m_1$ lies only with $o_2$

- An autonomous agent has control over its state and its behavior
  - There is no guarantee that another agent really performs an action

- An agent cannot force another agent to perform some action or to change its internal state

- An agent can try to influence another agent by communication

- Communication can change the internal state (belief, desire, intention) of another agent

- Communication as special case of action: **speech acts**
Speech Acts – Austin

Philosopher John Austin (1962)

Some utterances have the same characteristics as actions

E.g., declaring war, declaring a couple as married

Identification of performative verbs, such as request, inform, promise

Aspects of speech acts

“The tea pot is empty”

Locution: How is an utterance formulated

Illocution: What did the speaker mean (“somebody should make new tea”)

Perlocution: How does the utterance influence the receiver
Philosopher John Searle (1969)

Identified properties that must hold for a speech act to succeed

E.g.: **Speaker requests** that **hearer** performs **action**

- Normal I/O conditions: Hearer must be able to hear the request (not deaf, etc.), act must be performed in normal circumstances (not in a movie etc.)

- Preparatory conditions: what must be true in the world that speaker correctly chooses the speech act (hearer must be able to perform action, speaker must believe that hearer is able to perform action, hearer does not do the action without request)

- Sincerity conditions: e.g., speaker must really want the action to be performed
Types of Speech Acts

- Representatives: informing
- Directives: requesting
- Commissives: promising
- Expressives: thanking
- Declarations: declaring (change of affairs)
Speech Acts in Planning

- If a planning system requires interaction (with humans, other autonomous agents), the plans must include speech actions.

- How can properties of speech acts be represented in such a way that a planning system can reason about them?

- Cohen and Perrault, 1979

- STRIPS representation with pre- and postconditions, now represented in multi-modal logic special operators for beliefs, abilities, wants.
Speech Acts in Plans

Request(S, H, $\alpha$)
PRE Cando.pr $(S$ Believe $(H$ Cando $\alpha)) \land$

$(S$ Believe $(H$ believe $(H$ Cando $\alpha)))$

Want.pr $(S$ Believe $(S$ Want $requestInstance))$
EFF $(H$ Believe $(S$ Believe $(S$ Want $\alpha)))$

CauseToWant($A_1$, $A_2$, $\alpha$)
PRE Cando.pr $(A_1$ Believe $(A_2$ Believe $(A_2$ Want $\alpha)))$
EFF $(A_1$ Believe $(A_1$ Want $\alpha))$

Inform(S, H, $\phi$)
PRE Cando.pr $(S$ Believe $\phi)$

Want.pr $(S$ Believe $(S$ Want $informInstance))$
EFF $(H$ Believe $(S$ Believe $\phi))$

Convince($A_1$, $A_2$, $\phi$)
PRE Cando.pr $(A_1$ Believe $(A_2$ Believe $\phi))$
EFF $(A_1$ Believe $\phi)$
Speech Acts in Plans cont.

- Request and Inform are basic actions
- Request models only the illocutionary force of the act
- Therefore: CauseToWant as mediating act
  (prelocution; Hearer wants to perform $\alpha$)
- Similarly: Convince is a mediating act for Inform
  (Hearer believes $\phi$)
Agent Communication Languages

- Speech act theory has influenced language development for agent communication
- DARPA-funded Knowledge Sharing Effort (KSE):
  - Knowledge Interchange Format (KIF): Representation of knowledge about a “domain of discourse”
  - Knowledge Query and Manipulation Language (KQML): “Outer” language for agent communication; defines “envelope” format for messages
KIF

(Genesereth and Fikes, 1992)

- Express properties of things in a domain
  Michael is a vegetarian

- Express relationships between things in a domain
  Michael and Janine are married

- Express general properties of a domain
  Everybody has a mother

Closely based on First-Order Logic
Written in a Lisp-like notation
KIF Examples

(= (temperature m1) (scalar 83 Celsius))

; temperature and scalar
; are user-defined functions

(defrelation bachelor (?x) :=
  (and (man ?x) (not (married ?x))))

(defrelation person (?x) :=> (mammal ?x))
KQML

- Message defined by a performative and a number of parameters
- assumption of a virtual knowledge base (VKB)
  - agents treat other agents as if they had some internal representation of knowledge (although they do not know whether this is the case and in what format this knowledge is represented)

(ask-one
 :content (PRICE IBM ?price)
 :receiver stock-receiver
 :language LPROLOG
 :ontology NYSE-TICKS)

CogSysI Lecture 11: Communication – p. 317
## KQML Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>:content</td>
<td>content of message</td>
</tr>
<tr>
<td>:force</td>
<td>whether sender will ever deny content</td>
</tr>
<tr>
<td>:reply-with</td>
<td>whether sender expects a reply and identifier for reply</td>
</tr>
<tr>
<td>:in-reply-to</td>
<td>reference to the identifier of a ’reply-with’</td>
</tr>
<tr>
<td>:sender</td>
<td>sender of the message</td>
</tr>
<tr>
<td>:receiver</td>
<td>intended recipient</td>
</tr>
</tbody>
</table>
Some KQML Performatives

(Finin et al., 1993)

<table>
<thead>
<tr>
<th>Performative</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>achieve</td>
<td>S wants R to make something true of their environment</td>
</tr>
<tr>
<td>advertise</td>
<td>S claims to be suited to processing a performative</td>
</tr>
<tr>
<td>ask-about</td>
<td>S wants all relevant sentences in R’s VKB</td>
</tr>
<tr>
<td>ask-all</td>
<td>S wants all of R’s answers to a question</td>
</tr>
<tr>
<td>ask-if</td>
<td>S wants to know whether the answer to C is in R’s VKB</td>
</tr>
<tr>
<td>ask-one</td>
<td>S wants one of R’s answers to C</td>
</tr>
<tr>
<td>broadcast</td>
<td>S wants R to send a performative over all connections</td>
</tr>
<tr>
<td>broker-all</td>
<td>S wants R to collect all responses to a performative</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Some KQML Performatives cont.

...  

eos   end of a stream response to an earlier query  
evaluate S wants R to evaluate C  
monitor S wants updates to R’s response to 'stream-all'  
...

pipe S wants R to route all further performatives to another agent  
recommend-all S wants all names of agents who can respond to C  
...

stream-about multiple response version of 'ask-about'  
stream-all multiple response version of 'ask-all'  
tell S claims to R that C is in S's VKB  
...
Example Dialogues

(evaluate
  :sender A :receiver B
  :language KIF :ontology motors
  :reply-with q1 :content (val (torque m1)))

(reply
  :sender B :receiver A
  :language KIF :ontology motors
  :in-reply-to q1
  :content (= (torque m1) (scalar 12 kgf)))
Example Dialogues cont.

(stream-about
  :sender A :receiver B
  :language KIF :ontology motors
  :reply-with q1 :content m1)

(tell
  :sender B : receiver A
  :in-reply-to q1
  :content (= (torque m1) (scalar 12 kgf)))

(tell
  :sender B : receiver A
  :in-reply-to q1 :content (= (status m1) normal))

(eos
  :sender B : receiver A
  :in-reply-to q1)
From KQML to FIPA ACL

- KQML was taken up by the multiagent systems community in a significant way, BUT
  - Set of performatives was never really fixed, therefore, different KQML implementations were developed which could not interoperate
  - Message transport mechanisms were never precisely defined, again a reason for interoperability problems
  - Semantics of KQML was never rigidly defined; meaning of KQML performatives was only described informally, and therefore open to different interpretations
  - The performative set is rather large and rather ad hoc
  - Performatives for making commitments are missing (important for cooperation)
- Foundation for Intelligent Physical Agents (FIPA) started working standards for multiagent systems, 1995
Ontologies for Agent Communication

Ontology:
- Formal definition of a body of knowledge
- Typically, a taxonomy of class and subclass relations, coupled with definitions of characteristics (attributes, relations) of and between things

Allows for agreement on terminology in agent communication

E.g., agent is buying some engineering item from another agent; must be able to unambiguously specify the desired properties, such as size; meaning of ‘size’, ‘centimeter’ etc.

KIF is a language for defining ontologies

Currently most important: XML (Extensible Markup Language) and DAML (DARPA Agent Markup Language)
XML

Markup Language: Grammar for interspersing documents with markup commands

HTML (Hypertext Markup Language): Markups for displaying documents in a web-browser (separating function and form)

XML is a *meta*-markup language, that is, there is no fixed set of pre-defined tag

Tag based languages describe content ("semantics") rather than form (layout) of data.

XML documents are *trees*: single root element and each element can have an arbitrary number of children.

DAML is based on XML
<rdf:Description rdf:ID="United-Kingdom">
  <rdf:type rdf:resource="GEOREF" />
  <HAS-TOTAL-AREA>
    (* 244820 Square-Kilometers)
  </HAS-TOTAL-AREA>
  <HAS-LAND-AREA>
    (* 241590 Square-Kilometers)
  </HAS-LAND-AREA>
  <HAS-COMPARATIVE-AREA-DOC>
    slightly smaller than Oregon
  </HAS-COMPARATIVE-AREA-DOC>
  ...
</rdf:Description>
Multiagent Systems, Further Aspects

- Cooperative, distributed problem solving
- Modal Logic for MAS (semantics of agent states)
- Agent-oriented software engineering
- Agent platforms