Lectures: Communication & Cooperation

Ana Paiva

* These slides are based on the book by Prof. M. Woodridge “An Introduction to Multiagent Systems” and the slides online compiled by Professor Jeffrey S. Rosenschein. Also, some introductions to this lecture were based on a tutorial done by Yannis Labrou on Agent Communication Languages. Modifications introduced by Prof. Ana Paiva, are her sole responsibility.
Importance of Communication

Communication is essential to get things done!

But there are problems...
Aspects of Communication

- Communication almost always means “communication in a common language”
- “Language” does not include natural languages only but synthetic ones.
Language understanding

- Understanding a “common language” means:
  - understanding of its **vocabulary**, i.e., understanding of the meaning of its tokens
  - knowing how to **effectively use the vocabulary** to perform tasks, achieve goals, effect one’s environment, etc.
How can agents speak to each other?
Theoretical Basis for an Agent Communication Language: Speech Act Theory
Speech Acts

- Most treatments of *communication* in (multi-)agent systems borrow their *inspiration from speech act theory*
- Speech act theories are *pragmatic theories* of language, i.e., theories of language use: they attempt to account for *how language is used by people* every day to *achieve their goals* and intentions
• The origin of speech act theories are usually traced to Austin’s 1962 book, *How to Do Things with Words*
Speech Acts

• Austin noticed that *some utterances* are rather *like ‘physical actions’* that appear to *change the state of the world*

• Paradigm examples would be:
  – declaring war
  – christening
  – ‘I now pronounce you man and wife’
But more generally, everything we utter is uttered with the intention of satisfying some goal or intention.

A theory of how utterances are used to achieve intentions is a speech act theory.
Different Aspects of Speech Acts

- From “A Dictionary of Philosophical Terms and Names”:
- “Locutionary act: the simple speech act of generating sounds that are linked together by grammatical conventions so as to say something meaningful. Among speakers of English, for example, ‘It is raining’ performs the locutionary act of saying that it is raining, as ‘Grablistrod zetagflx dapu’ would not.”
Different Aspects of Speech Acts

• “Illocutionary act: the speech act of doing something else – offering advice or taking a vow, for example – in the process of uttering meaningful language. Thus, for example, in saying ‘I will repay you this money next week,’ one typically performs the illocutionary act of making a promise.”
Different Aspects of Speech Acts

- “Perlocutionary act: the speech act of having an effect on those who hear a meaningful utterance. By telling a ghost story late at night, for example, one may accomplish the cruel perlocutionary act of frightening a child.”
Speech Acts

• Searle (1969) identified various different types of speech act:
  – *representatives*: such as *informing*, e.g., ‘It is raining’
  – *directives*: attempts to *get the hearer to do something* e.g.,
    ‘please make the tea’
  – *commissives*: which *commit the speaker* to doing something,
    e.g., ‘I promise to… ’
  – *expressives*: whereby a speaker *expresses a mental state*,
    e.g., ‘thank you!’
  – *declarations*: cause changes in a state of affairs, e.g.,
    declaring war or christening
Speech Acts

• There is some debate about whether this (or any!) typology of speech acts is appropriate

• In general, a speech act can be seen to have two components:
  – a performative verb:
    (e.g., request, inform, promise, …)
  – propositional content:
    (e.g., “the door is closed”)
Speech Acts

• Consider:

  – performative = request
    content = “the door is closed”
    speech act = “please close the door”

  – performative = inform
    content = “the door is closed”
    speech act = “the door is closed!”

  – performative = inquire
    content = “the door is closed”
    speech act = “is the door closed?”
Plan Based Semantics

• How does one define the semantics of speech acts? When can one say someone has uttered, e.g., a request or an inform?

• Cohen & Perrault (1979) defined semantics of speech acts using the STRIPS notation via preconditions and postconditions:
  – “Can-do” preconditions
  – “Want” preconditions
  – Postconditions/effects
Plan-Based Semantics

• Here is their semantics for request:

  \[ \text{request}(s, h, \phi) \]

pre:
  – \( s \) believe \( h \) can do \( \phi \)
    (you don’t ask someone to do something unless you think they can do it)
  – \( s \) believe \( h \) believe \( h \) can do \( \phi \)
    (you don’t ask someone unless \( they \) believe they can do it)
  – \( s \) believe \( s \) want \( \phi \)
    (you don’t ask someone unless you want it!)

post:
  – \( h \) believe \( s \) believe \( s \) want \( \phi \)
    (the effect is to make them aware of your desire)
But how can this be useful in Agents?

By creating an “Agent Communication Language based on “speech acts”.”
The “Knowledge Sharing Effort”

- Initiated by DARPA circa 1990
- Sponsored by DARPA, NSF, AFOSR, etc.
- Participation by dozens of researchers in academia and industry.
- Developing techniques, methodologies and software tools for knowledge sharing and knowledge reuse.
- Sharing and reuse can occur at design, implementation or execution time.
Knowledge Sharing Effort

- Knowledge sharing requires communication which requires a “common language”
- We can divide a language into syntax, semantics, and pragmatics
- Some existing components that can be used independently or together:
  - **KIF** - knowledge Interchange Format *(syntax)*
  - **Ontolingua** - a language for defining sharable ontologies *(semantics)*
  - **KQML** - a high-level interaction language *(pragmatics)*
Knowledge Sharing Effort

- **KIF** - knowledge Interchange Format *(syntax)*
- **Ontolingua** - a language for defining sharable ontologies *(semantics)*
- **KQML** - a high-level interaction language *(pragmatics)*

Propositional

Propositional attitudes
KIF: Knowledge Interchange Format

- KIF ~ First order logic with set theory
- An interlingua for encoded declarative knowledge
- Common language for reusable knowledge
  - Implementation independent semantics
  - Highly expressive - can represent knowledge in typical application KBs.
  - Translatable - into and out of typical application languages
  - Human readable - good for publishing reference models and ontologies
This is not a profoundly new idea …

- Vocabulary specification
- Domain theory
- Conceptual schema (plural, noun, meaning, present participle of to be) + -logy: cf. F. ontologie.
Ontologies

Alice: ‘Did you read *Prey*?’
Bob: ‘No, what is it?’
Alice: ‘A science fiction novel. It’s also a bit of a horror novel, actually. It’s about multiagent systems going haywire’

Information:
- *Prey* is a novel
- *Prey* is a science fiction novel
- *Prey* is a horror novel
- The subject of *Prey* is multiagent systems going wrong.
Work of art
- Sculpture

Work of fiction
- Novel
  - Romance Novel
  - Comedy Novel
  - Science Fiction Novel
  - Horror Novel

Instance of
- Hitchhiker’s Guide to the Galaxy
- Prey
There can be many types of ontologies…

**Controlled Vocabulary** use of few predefined keywords to classify entities

**Terms/Glossary**- list of terms and English explanation defining meaning of terms

**Informal is-a taxonomies**- controlled vocabulary organised into an informal hierarchy (eg. Amazon.com)

**Formal**

- **Formal is-a** explicitly definition of subsumption relationships between classes
- **Properties**- allowing for the classes to have properties
- **Value restrictions** give additional information about the relationships
- **Arbitrary Logical Constraints**- constraints on the classes
Systems which communicate and work together must share an ontology.

The shared ontology can be **implicit** or **explicit**.

Implicit ontology are typically represented only by procedures.

Explicit ontologies are (ideally) given a declarative representation in a well defined knowledge representation language.
Three important aspects to explicit ontologies

- **Conceptualization** involves the underlying model of the domain in terms of objects, attributes and relations.
- **Vocabulary** involves assigning symbols or terms to refer to those objects, attributes and relations.
- **Axiomization** involves encoding rules and constraints which capture significant aspects of the domain model.
Ad hoc ontologies in XML

- Simple
- Little effort
- Usually done for a specific purpose
- Used in many applications
The OWL Web Ontology Language is a language for defining and instantiating Web ontologies.

- **OWL Lite** - simplest variant of OWL. Supports only basic ontologies.
- **OWL DL** – extends the properties of OWL Lite. It was chosen to correspond to the formalism of description logic.
- **OWL Full** - very expressive framework, providing features for defining ontologies.
But how can this be useful in Agents?
We now consider *agent communication languages* (ACLs) — *standard formats* for the exchange of messages.

The *best known* ACL is KQML, developed by the ARPA knowledge sharing initiative. KQML is comprised of *two parts*:

- the knowledge query and manipulation language (*KQML*)
- the knowledge interchange format (*KIF*)
KQML and KIF

• **KQML** is an ‘outer’ language, that defines various acceptable ‘communicative verbs’, or **performatives**
  
  Example performatives:
  
  - ask-if (‘is it true that. . . ’)
  - perform (‘please perform the following action. . . ’)
  - tell (‘it is true that. . . ’)
  - reply (‘the answer is . . . ’)

• **KIF** is a **language** for expressing message **content**
Used to state:

- **Properties of things** in a domain (e.g., “Noam is chairman”)
- **Relationships between things** in a domain (e.g., “Amnon is Yael’s boss”)
- **General properties of a domain** (e.g., “All students are registered for at least one course”)
“The temperature of m1 is 83 Celsius”:
(= (temperature m1) (scalar 83 Celsius))

“An object is a bachelor if the object is a man and is not married”:
(defrelation bachelor (?x) :=
  (and (man ?x) (not (married ?x))))

“Any individual with the property of being a person also has the property of being a mammal”:
(defrelation person (?x) :=> (mammal ?x))
KQML and KIF

• In order to be able to communicate, agents must have *agreed on a common set of terms*

• A formal specification of a set of terms is known as an *ontology*

• The knowledge sharing effort has associated with it a large effort at defining common ontologies — *software tools like ontolingua* for this purpose
KQML and KIF

• Example KQML/KIF dialogue…

A to B: (ask-if (> (size chip1) (size chip2)))

B to A: (reply true)
B to A: (inform (= (size chip1) 20))
B to A: (inform (= (size chip2) 18))
• But for all the agents to communicate in the same language, there should have been an agreement about that particular language

• FIPA- Foundation for Intelligent Physical Agents creates “standards” for agents...
FIPA-ACL

• Basic structure is quite similar to KQML:
  – performative
    20 performative in FIPA
  – housekeeping
    e.g., sender, ontology etc.
  – content
    the actual content of the message
• Example:
  (inform
    :sender      agent1
    :receiver    agent5
    :content (price good200 150)
    :language    sl
    :ontology    hpl-auction
  )
<table>
<thead>
<tr>
<th>performative</th>
<th>passing info</th>
<th>requesting info</th>
<th>negotiation</th>
<th>performing actions</th>
<th>error handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept-proposal</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>agree</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>cancel</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cfp</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>confirm</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>disconfirm</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>inform</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inform-if</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inform-ref</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not-understood</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>propose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>query-if</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>query-ref</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>refuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>reject-proposal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request-when</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>request-whenever</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>subscribe</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• For example, the performatives for a negotiation scenario:
  - cfp
  - propose
  - accept-proposal
  - reject-proposal
“Inform” and “Request”

- “Inform” and “Request” are the two basic performatives in FIPA. All others are macro definitions, defined in terms of these.
- The meaning of inform and request is defined in two parts:
  - pre-condition
    what must be true in order for the speech act to succeed
  - “rational effect”
    what the sender of the message hopes to bring about
“Inform” and “Request”

• For the “inform” performative...
The content is a statement. Pre-condition is that sender:
  – holds that the content is true
  – intends that the recipient believe the content
  – does not already believe that the recipient is aware of whether content is true or not

• Effect (on success):
  – recipient will believe the content
“Inform” and “Request”

• For the “request” performative… The content is an action. Pre-condition is that sender:
  – intends action content to be performed
  – believes recipient is capable of performing this action
  – does not believe that receiver already intends to perform action

• Effect (on success):
  – action will be performed
From Communication to Cooperation
What do we need to cooperate?

- Willingness to cooperate
- Common language
- Protocols and mechanisms to guarantee the sustainability of cooperation
Why Cooperation

- Sharing Resources
- Solving a task
- Making a decision
Working Together

• Why and how do agents work together?
• Important to make a distinction between:
  – benevolent agents
  – self-interested agents
Benevolent Agents (the good ones)

- If we “own” the whole system, we can design agents to help each other whenever asked.
- In this case, we can assume agents are *benevolent*: our best interest is their best interest.

*Benevolence simplifies the system design task enormously!*
Yet: Self-Interested Agents

- If agents represent individuals or organizations, (the more general case), then we cannot make the benevolence assumption.
Self-Interested Agents

- Agents will be assumed to act to further their own interests, possibly at expense of others
- Potential for conflict
- May complicate the design task enormously
Achieving Cooperation: Common High-level Protocols

• There is also a need for communication agents to agree on the agent-level protocols they will use.
• The protocol is often conveyed via an extra parameter on a message
  - (ask :from Alice :to Bob … :protocol auction42 …)
Protocols for Agents’ cooperation

“Protocols are structured ways to encode a type of conversation”

• Often formalised as finite state machines (state, transitions) with an initial and a final state
• Protocols simplify the computation by restricting the agent to a limited set of performatives at any given point.
• Protocols also allow a build-in turn-taking mechanism
• Protocols capture the conventional aspects of conversations
Types of Protocols

• Common protocols:
  – Contract net
  – Various auction protocols

• These protocols are often defined in terms of constraints on possible conversations and can be expressed as
  – Grammars (e.g., DFAs, ATNs, …)
  – Petri networks
  – Conversation plans
  – Rules or axioms
Cooperative Distributed Problem Solving (CDPS)

Why?

- Neither global control nor global data storage — no agent has sufficient information to solve entire problem
- Control and data are distributed
Example: building a house
Example: solving a crime
Task Sharing and Result Sharing

• Two main modes of cooperative problem solving:
  – task sharing: components of a task are distributed to component agents
  – result sharing: information (partial results, etc.) is distributed
Four Phases to achieve a Solution in CDPS

1. **Problem Decomposition** - the problem needs to be decomposed into smaller parts to be carried out by the different agents.

2. **Sub-problem distribution** - the sub-problems need to be distributed amongst the different agents (task allocation).

3. **Sub-problem solution** - the solution to the subproblems are achieved to be aggregated.

4. **Answer synthesis** - the aggregation of the solutions will lead to the generation of a final solution.
A well known task-sharing protocol for task allocation is the contract net.
The Contract Net

• An approach to *distributed problem solving*, focusing on task distribution

• Task distribution viewed as a kind of contract negotiation

• The “Protocol” specifies also some *content* of communication, and not just its form.
Contract-Net - Roles

- **Manager** – responsible for monitoring the execution of the task.

- **Contractor** – responsible for executing the task.

- An agent can take both roles in a dynamic way during the execution of a task.
Contract Net

- The collection of nodes is the “contract net”
- Each node on the network can, at different times or for different tasks, be a manager or a contractor
- When a node gets a composite task (or for any reason can’t solve its present task), it breaks it into subtasks (if possible) and announces them (acting as a manager), receives bids from potential contractors, then awards the job (example domain: network resource management, printers, …)
The Contract Net

Phases:
1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
• In this stage, an agent recognizes it has a problem it wants help with. Agent has a goal, and either…
  – realizes it cannot achieve the goal in isolation — does not have capability
  – realizes it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc.)
Let’s imagine a bear that wants to eat some nice food from the North Pole. He recognises he cannot do the task.
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
• In this stage, the agent with the task sends out an *announcement* of the task which includes a *specification* of the task to be achieved.
Announcement

• Specification must encode:
  – description of task itself (maybe executable)
  – any constraints (e.g., deadlines, quality constraints)
  – meta-task information (e.g., “bids must be submitted by…”)

• The announcement is then broadcast
Node Issues Task Announcement
Idle Node Listening to Task Announcements
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
Bidding

• Agents that receive the announcement decide for themselves whether they wish to *bid* for the task

• Factors:
  – agent must decide whether it is capable of expediting task
  – agent must determine quality constraints & price information (if relevant)

• If they do choose to bid, then they submit a *tender*
The agents evaluate the announcement and decide if they can do a bid to the manager.
Manager listening to bids
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
Awarding & Expediting

- Agent that sent task announcement must choose between bids & decide who to “award the contract” to.
- The result of this process is communicated to agents that submitted a bid.
- The successful contractor then expedites the task.
- May involve generating further manager-contractor relationships: sub-contracting.
The manager selects the most adequate bid according to his criteria.
Manager Making an Award
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
Example: Contract Net

The Contractor executes the task and delivers the results to the manager.
The Contract Net

Phases:

1. Recognition
2. Announcement
3. Bidding
4. Awarding
5. Expediting
Issues for Implementing Contract Net

• How to...
  – …specify *tasks*?
  – …specify *quality of service*?
  – …select between competing offers?
  – …differentiate between offers based on multiple criteria?
Messages for the Contract Net

- **Request-for-bids(T,X)** is a message used by the Manager for the potential contractors asking for a Bid. It must include the description of the task.

- **Propose(T,O)** is the response positive by the contractor in relation to the announcement for the execution of a task. The proposal has a specification of the offer T.
Not-interested(T,Y) is the negative response of a contractor saying that is not interested in the execution of a task.

Award(T,C,X) is a message of the Manager (X) to the contractor (C) awarding it the execution of the task.
• **Accept(T,X)** is the positive reply to the contractor Y in reply to the offer of the manager for the execution of a task.

• **Refuse(T,Y)** is the negative reply of the contractor to the Manager’s offer to execute the task.
FIPA ACL

FIPA-Contract Net - for negotiation

cfp(action, pre1)

not-understood
refuse(reason)
propose(pre2)

reject-proposal
accept-proposal

failure(reason)
inform(done)
cancel

Deadline
Advantages

- Good solution to the connection problem
- Adequate to problems of hierarchical structure

Disadvantages

- A task may not be allocated if a qualifies agent is occupied.
- For a large number of agents it generates a large amount of messages. (although the eligibility specification may reduce the messages in reply to the announcement)
Other Protocols for working together

- There is a variety of protocols:
  - Negotiation protocols:
    - One-to-one negotiation (Monotonic Bargaining, interest-based negotiation),
    - Multilateral negotiation (Auctions)
    - Argumentation

- There is also hybrid approaches:
  - Micro-protocols
  - Dialogue games
Questions???