Agent Communication (AI section 2)

SAPIENZA
UNIVERSITÀ DI ROMA

Daniele Nardi

Dipartimento di Ingegneria
Informatica, Automatica e Gestionale
daniele.nardi@dis.uniroma1.it
http://rococo.dis.uniroma1.it

The slides have been prepared using an earlier version by Alexander Kleiner

Summary

Communication protocols
Content sharing

[W] chapter 6 and 7
[WE] chapter 3

Introduction

- Communication in concurrent systems:
  - Synchronization of multiple processes
    - E.g., solving the “lost update scenario”:
- Communication in OOP
  - Method invocation between different modules
    - E.g., object o2 invokes method m1 on object o1 by executing the code o1. m1(arg), where “arg” is the argument to communicate

Communication in MAS

- Autonomous agents have control over both state and behavior
- Methods are executed according to the agent’s self-interest
- Agents can perform communicative actions, i.e. attempt to influence other agents
- Agent communication implies interaction, i.e. agents perform communication acts

Speech Acts I

Communication in MAS is inspired by speech act theory

The theory of speech acts is generally recognized to have begun with the work of the philosopher John Austin: "How to Do Things with Words" (Austin, 1962)

- *Speech act theory* studies the pragmatic use of language
  - an attempt to account for how language is used by people every day to achieve their goals and intentions
- *Speech act theory* treats communication as action
  - speech actions are performed by agents just like other actions, in the furtherance of their intentions

Speech Acts II

- Austin noticed that some utterances are rather like 'physical actions' that appear to *change the state of the world.*
  - declaring war
  - 'I now pronounce you man and wife'
- Austin identified a number of performative verbs, which correspond to various different types of speech acts
  - Examples of performative verbs are *request*, *inform*, and *promise*

Speech Acts III

Searle (1969) identified the following key classes of possible types of speech acts:

- *Representatives:*
  commits the speaker to the truth of an expression, e.g., 'It is raining' (*informing*)
- *Directives:*
  attempts to get the hearer to do something e.g., 'please make the tea' (*requesting*)
- *Commissives:*
  which commits the speaker to do something, e.g., 'I promise to… ' (*promising*)
- *Expressives:*
  whereby a speaker expresses a mental state, e.g., 'thank you!' (*thanking*)
- *Declarations:*
  effect change of state, such as "declaring war" (*declaring*)

Speech Acts IV

Cohen and Perrault (1979) started to modeling speech acts in a planning system (STRIPS formalism)

Speech acts are thus characterized by:

Preconditions:
e.g. to make a request **Speaker** must *believe* that **Hearer** *cando* 'X' and also it must *believe* that **Hearer** *believes* it *cando* 'X'

Effects:
e.g. the result of the request is that **Hearer** *wants* 'X'

Agent Communication Languages

Agent communication languages (ACLs) are standard formats for the exchange of messages

- KSE (Knowledge Sharing Effort) in early 1990s designed two ACLs with different purpose
  - The Knowledge Interchange Format (KIF), a language for expressing content, closely based on First Order Logic
  - The Knowledge Query and Manipulation Language (KQML), which is an 'outer' language for agent communicatio

Knowledge Query and Manipulation Language (KQML) I

- KQML defines *communicative verbs*, or *performatives*, for example:
  - `ask-if` ('is it true that. . . ')
  - `perform` ('please perform the following action. . . ')
  - `tell` ('it is true that. . . ')
  - `reply` ('the answer is . . . ')
- Each message has a performative (the „class" of a message) and a number of parameters

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

Asking about the price of IBM stock

Terminology

KQML II Parameters of messages

Parameter	Meaning
`:content`	content of the message
`:language`	formal language the message is in
`:ontology`	terminology the message is based on
`:force`	will sender ever deny content of message?
`:reply-with`	reply expected? identifier of reply?
`:in-reply-to`	id of reply
`:sender`	sender ID
`:receiver`	receiver ID

KQML III Example dialogs

Dialogue (a)

```
(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)))
```

Talking about motors

Query reference q1

Asking about torque of motor 1

Answer: "It is 12kgf"

Dialogue (b)

```
(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)
```

Streaming of messages, e.g. request all available knowledge

Indication of "End of Stream"

KQML IV Criticisms

- The basic KQML performative set was overly large and not standardized
  - different implementations of KQML where developed that could not, in fact, interoperate
- The language was missing the performative *commissives*
  - Commissives are crucial for agents coordinating their actions.
- These criticisms led to the development of a new language by the FIPA consortium

Agent Communication Languages (FIPA)

- FIPA (Foundation for Intelligent Physical Agents) is the organization for developing standards in multi-agent systems. It was officially accepted by the IEEE at its eleventh standards committee in 2005
- FIPA's goal in creating agent standards is to promote inter-operable agent applications and agent systems
- FIPA ACL's syntax and basic concepts are very similar to KQML, for example:

```
(inform
      :sender agent1
      :receiver agent2
      :content (price good2 150)
      :language sl
      :ontology hpl-auction
)
```


FIPA ACL Set of Performatives

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

FIPA ACL Performatives Requesting Information

subscribe sender asks to be notified when statement changes

query-if direct query for the truth of a statement

query-ref direct query for the value of an expression

FIPA ACL Performatives Passing Information

`inform`	together with **request** most important performative; basic mechanism for communicating information; sender wants recipient to believe info; sender believes info itself
`inform-ref`	informs other agent about value of expression (in its content parameter); typically content of **request** message (thus asking the receiver to give me value of expression)
`confirm`	confirm truth of content (recipient was unsure)
`disconfirm`	confirm falsity of content (recipient was unsure)

FIPA ACL Performatives Negotiation

`cfp`	call for proposals; initiates negotiation between agents; content-parameter contains action (desired to be done by some other agent) (e.g.: „sell me car") and condition (e.g.: „price < 1000$")
`propose`	make proposal
`accept-proposal`	sender accepts proposal made by other agent
`reject-proposal`	sender does not accept proposal

FIPA ACL Performatives Performing Actions

`request`	issue request for an action
`request-when`	issue request to do action if and when a statement is true
`request-whenever`	issue request to do action if and whenever a statement is true
`agree`	sender agrees to carry out requested action
`cancel`	follows request; indicates intention behind request is not valid any more
`refuse`	reject request

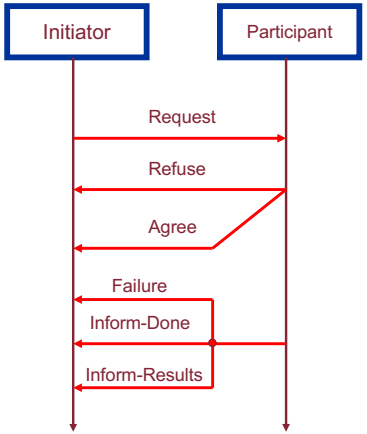
FIPA Interaction Protocols (IPs)

Interaction Protocols (IPs) are standardized exchanges of performatives according to well known situations

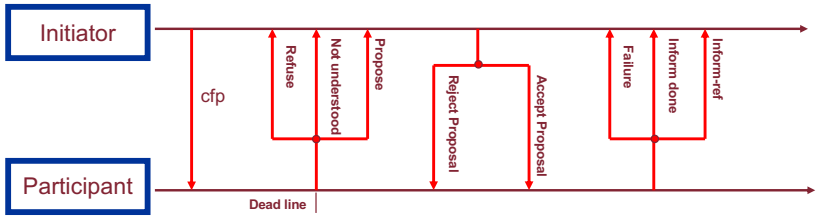
FIPA defined IPs are:

- FIPARequest
- FIPAQuery
- FIPARequestWhen
- FIPAContractNet
- FIPAIteratedContractNet
- FIPAAuctionEnglish
- FIPAAuctionDutch
- FIPABrokering
- FIPARecruiting
- FIPASubscribe
- FIPAPropose

FIPA Interaction Protocols (IPs): Request



FIPA Interaction Protocols (IPs): Contract Net

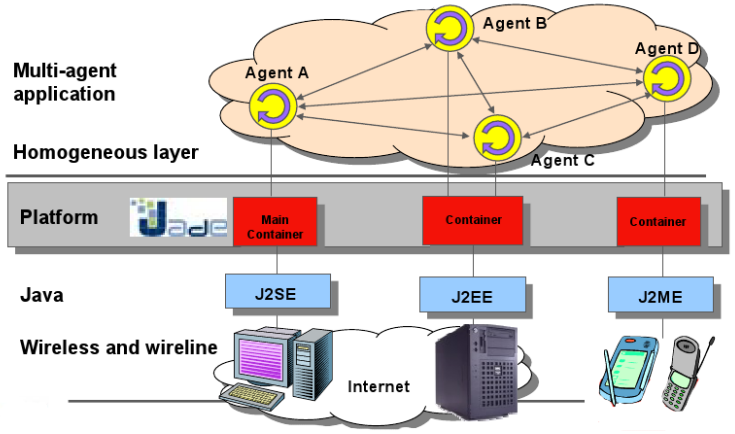


Java Agent Development Framework (JADE)

Open Source project originated by Telecom (TILAB), currently governed by an international board, e.g. Motorola, France Telecom, Whitestein, …

- JADE allows the rapid creation of distributed, multi-agent systems in Java
- High interoperability through FIPA compliance
- JADE includes:
  - A library for developing agents (which implements message transport and parsing)
  - A runtime environment allowing multiple, parallel and concurrent agent activities
  - Graphical tools that support monitoring, logging, and debugging
  - Yellow Pages, a directory where agents can register their capabilities and search for other agents and services

JADE II Connectivity



Agent Communication Languages: commitments

A commitment (M.P. Singh 2008) is an expression of the form:

C*(debtor,creditor,antecedent,consequent)*

where *debtor, creditor* are agents and
antecedent,consequent are propositions

debtor is committed to *creditor* if *antecedent* holds and it will bring about *consequent*

Example: C(*Ebook,Alice,$12,BNW*)

By making *antecedent* true the commitment is detached and by making *consequent* true the commitment is discharged

Agent Communication Languages: commitments II

Commitments operations:

- CREATE
- CANCEL
- RELEASE
- DELEGATE
- ASSIGN
- DECLARE*

DECLARE literally modifies commitments

By using these primitives it is possible to provide a formal characterization of the transactions (e.g. Buy a book).

Agent Communication: contents

Protocols characterize types of messages, but they say nothing about content.

The specification of the content implies that there is a common reference to interpret it.

This means a language: KIF for KQML

and a shared semantics for the sentences (ontology)

Dialogue (a)

```
(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)))
```

Dialogue (b)

```
(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)
```

Knowledge Interchange Format (KIF)

- KIF allows agents to express
  - properties of things in a domain, e.g., “Michael is a vegetarian”
  - relationships between things in a domain, e.g., “Michael and Janine are married”
  - general properties of a domain, e.g., “All students are registered for at least one course” (quantification ∀)
- Examples:
  - “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))`

XML: a step towards syntax + semantics

XML
(eXtensible Markup Language)
In contrast to HTML, whose meta-language mainly describes the page layout, XML allows to tag data with semantics

The Document Type Definition (Schema)
can be used to share the semantics (e.g. need to agree on what 'title' means)

(a) Plain HTML

```
<ul>
   <li><em>Music</em>,
      <b>Madonna<b>,
      USD12<br><p>
   <li><em>Get Ready</em>,
      <b>New Order</b>,
      USD14<br><p>
</ul>
```

(b) XML

```
<catalogue>
   <product type="CD">
      <title>Music</title>
      <artist>Madonna</artist>
      <price currency="USD">12</price>
   </product>
   <product type="CD">
      <title>Get Ready</title>
      <artist>New Order</artist>
      <price currency="USD">14</price>
   </product>
</catalogue>
```

Plain HTML vs. XML

Ontologies (modern interchange languages)

- Ontologies ground the terminology used by the agents
  - For example, an agent wants to buy a screw. But what means then "size"? Is it in inch or centimeter?

OWL (Ontology Web Language) a very expressive tool to describe the knowledge about a domain

RDF (Resource Definition Framework) a language of triples subject-predicate-object, that allows references to external resources to share meaning

Semantic Web

Berners-Lee's vision of the Semantic Web:

I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.

Suggested Reading: "The Web was done by amateurs", Marco Aiello, Springer 2018

Summary

ACLs provide standards for communication among selfish agents, e.g. within an open systems

Work still ongoing:

- on protocols
- on message semantics