LECTURE 5: COMMUNICATING

An Introduction to Multiagent Systems

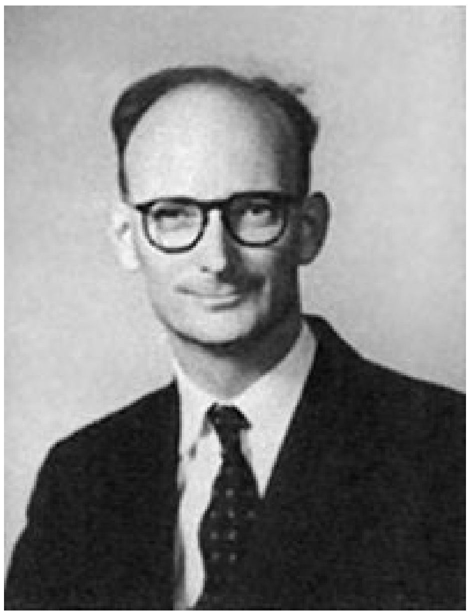
CISC 7412, Fall 2011

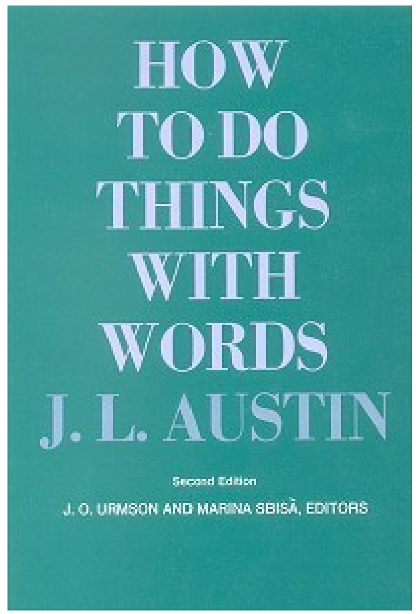
Agent Communication

- In this lecture we will begin to look at *multi* agent aspects.
- The most fundamental thing that agents have to do if they want to interact is to *communicate*.
- There are some limited things that one can do with communication, but they are, well limited.
- Most work on multiagent systems assumes communication.
- You can think of this as a transport layer for all the things we'll talk about in future weeks.

Speech Acts

- We start with this man:





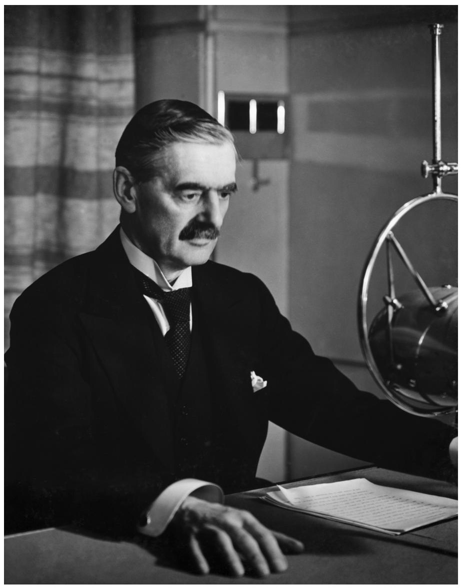
John Langshaw Austin

- In particular his 1962 book *How to Do Things with Words*.

- *How to Do Things with Words* is usually taken to be the origin of *speech acts*
- Speech act theories are *pragmatic* theories of language, that is theories of how language ia *used*.
- Speech act theories attempt to account for how language is used by people every day to achieve their goals and intentions.
- Most treatments of communication in (multi-)agent systems borrow their inspiration from *speech act theory*, doubtless because the “action” part can be tied closely to existing ideas about how to model action.

- Austin noticed that some utterances are rather like 'physical actions' that appear to *change the state of the world*.

- For example Neville Chamberlain saying:



This morning the British Ambassador in Berlin handed the German Government a final note stating that, unless we hear from them by 11 o'clock that they were prepared at once to withdraw their troops from Poland, a state of war would exist between us. I have to tell you now that no such undertaking has been received, and that consequently this country is at war with Germany.

- 11.15 am, September 3rd 1939.

- Led to:

2nd EXTRA

The San Diego Union

BRITAIN DECLARES WAR

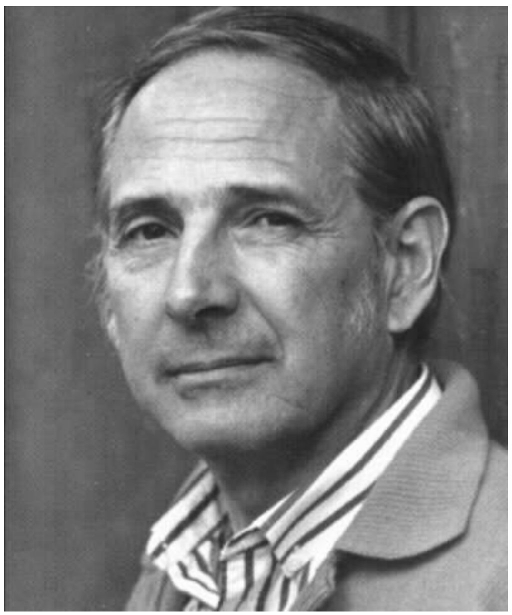
France Ready; To Back Britain

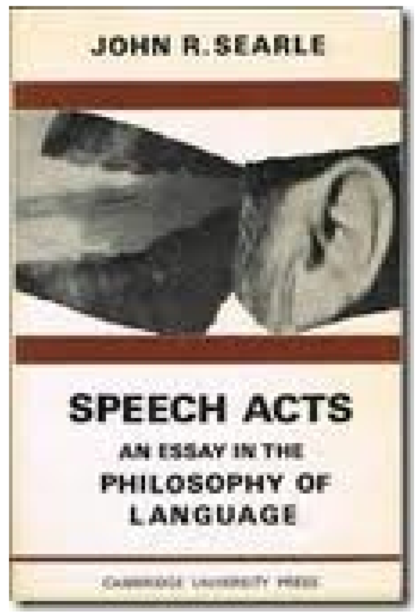
NO GAME OF TAG AS LONDON PRACTICES FOR WAR

'State of War Exists,' Chamberlain Tells World

- Paradigm examples are:
  - declaring war;
  - naming a child;
  - "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.

- The next step was taken by John Searle





who identified various different types of speech act.

- In his 1969 book *Speech Acts: an Essay in the Philosophy of Language* he identified:
  - *representatives*:
    such as *informing*, e.g., 'It is raining'
  - *directives*:
    attempts to get the hearer to do something e.g., 'please make the tea'
  - *commisives*:
    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*:
    such as declaring war or naming.

- 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, . . .)
  - *propositional content*:
    (e.g., “the door is closed”)

- 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?"
- Several speech acts with the same propositional content.

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 *precondition-delete-add* list formalism of planning research.
- Note that a speaker cannot (generally) *force* a hearer to accept some desired mental state.

- Here is their semantics for *request*:

  $request(s, h, \phi)$

  pre:

  – s believes h can do ϕ
  (you don't ask someone to do something unless you think they can do it)

  – s believe h believe h can do ϕ
  (you don't ask someone unless *they* believe they can do it)

  – s believe s want ϕ
  (you don't ask someone unless you want it!)

  post:

  – h believe s believe s want ϕ
  (the effect is to make them aware of your desire)

KQML and KIF

- We now consider *agent communication languages* (ACLs) — standard formats for the exchange of messages.
- One well known ACL is KQML, developed by the ARPA knowledge sharing initiative.

  KQML is comprised of two parts:

  – the knowledge query and manipulation language (KQML); and

  – the knowledge interchange format (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*.

- In order to be able to communicate, agents must have agreed a common set of terms.
- A formal specification of a set of terms is known as a *ontology*.
- The knowledge sharing effort has associated with it a large effort at defining common ontologies — software tools like `ontolingua` for this purpose.
- Chapter 6 of the textbook talks a lot about ontologies — we'll say a bit about them next.

Ontologies

- For agents to communicate, they need to agree on the words (terms) they use to describe a domain.
  - Always a problem where multiple languages are concerned
- For example, if I want to talk about my cat, the way I express the idea:

- Depends on what language understood by the person I'm speaking to:

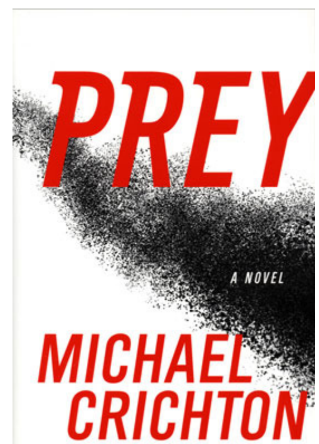
  Cat, chat, gato, . . .

- The role of an ontology is to fix the meaning of the terms used by agents.
- "An ontology is a formal definition of a body of knowledge". (Jim Hendler).
- How do we do this? Typically by defining new terms in terms of old ones:

Alice Did you read "Prey"?

Bob No, what is it?

Alice A science fiction novel. Well, it is also a bit of a horror novel. It is about multiagent systems going haywire.



- What is being conveyed about "Prey" here?
  - It is a novel.
  - It is a science fiction novel.
  - It is a horror novel
  - It is about multiagent systems
- Alice assumes that Bob knows what a "novel" is, what "science fiction" is and what "horror" is.
- She thus defines a new term "Prey" in terms of ones that Bob already knows.
- Notice that we have two kinds of thing:
  - Classes: collections of things with similar properties
  - Instances: specific examples of classes.

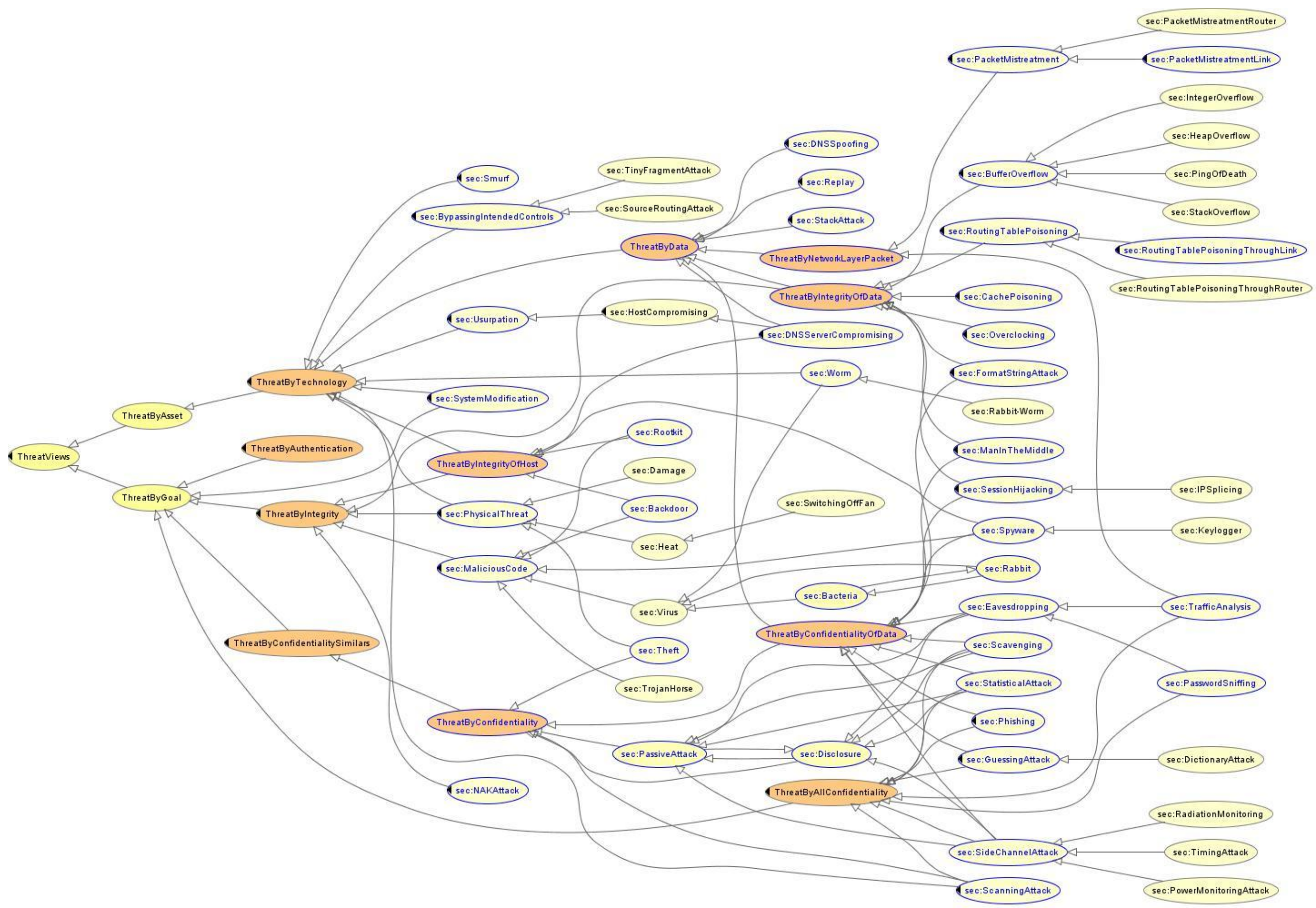
- Part of the reason this interaction works is that Bob has some knowledge that is relevant.
- Bob knows that novels are fiction books
  - "novel" is a subclass of "fiction book"
- Bob knows things about novels: they have authors, publishers, publication dates, and so on.
- Because "Prey" is a novel, it *inherits* the properties of novels.

  It has an author, a publisher, a publication date.
- Instances inherit attributes from their classes.

- Classes also inherit.
- Classes inherit attributes from their super-classes.
  - If “novel” is a subclass of “fiction book”, then “fiction book” is a superclass of “novel”
- Fiction books are books.
- Books are sold in bookstores.
- Thus fiction books are sold in bookstores.

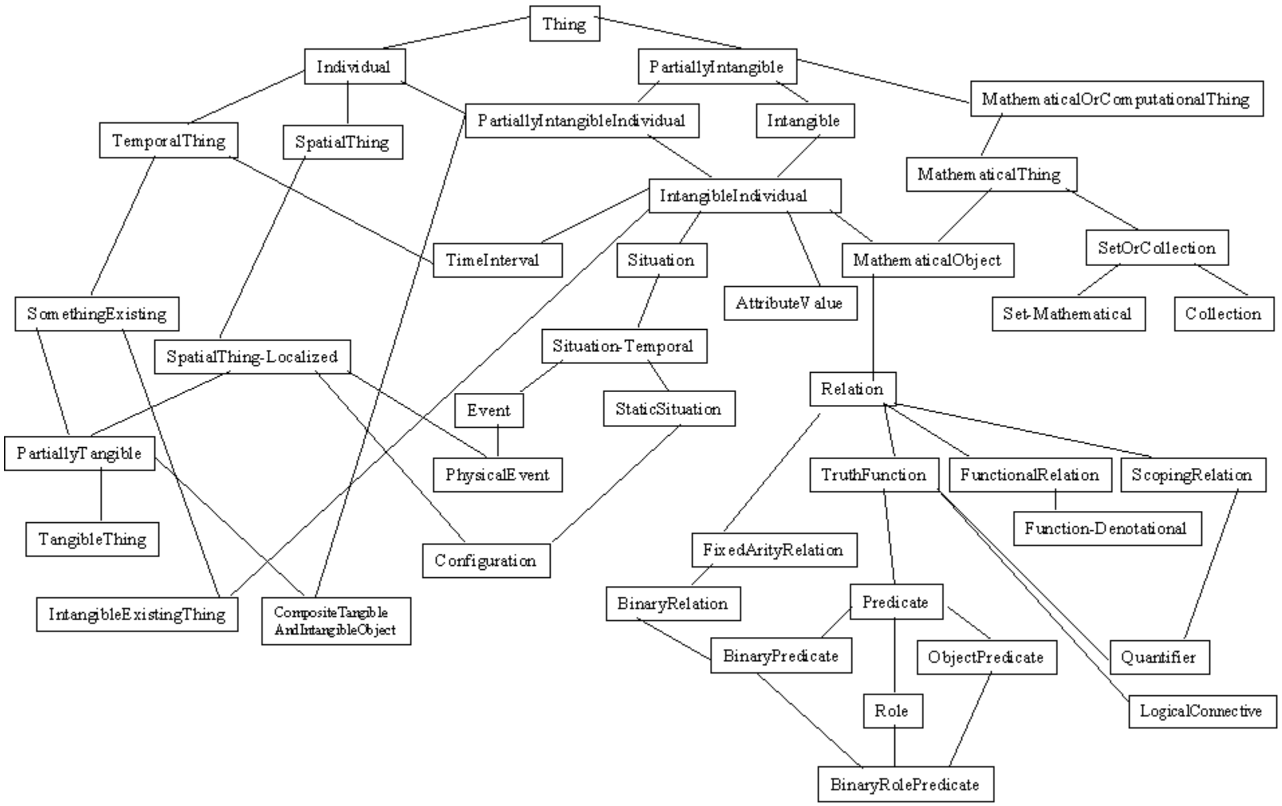
- A lot of knowledge can be captured using these notions.
- We specify which class "is-a" sub-class of which other class.
- We specify which classes have which attributes.
- This structure over knowledge is called an *ontology*.
  - A knowledge base is an ontology with a set of instances.
- A number of ontologies have been constructed.
  - Example on the next slide.

- An ontology of threats:

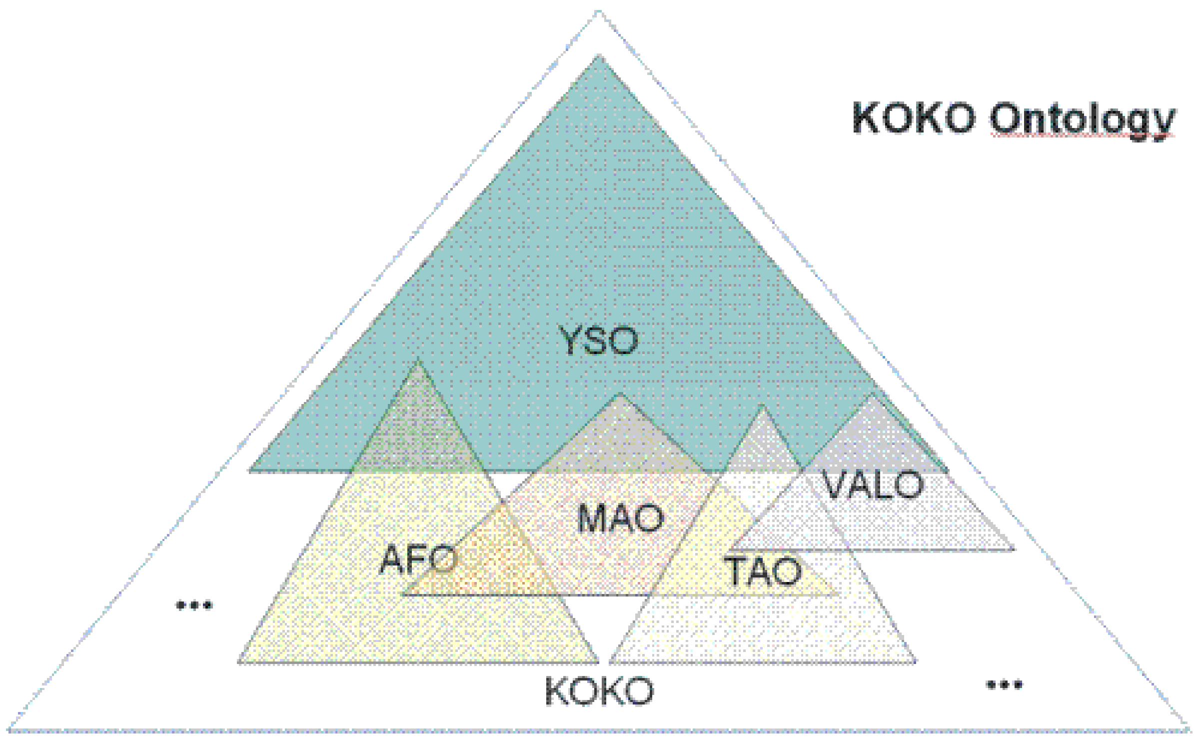


- In general there are multiple ontologies at different levels of detail.
  - Application ontology
    Like the threat ontology
  - Domain ontology
  - Upper ontology
    Contains very general information about the world.
- The more specific an ontology, the less reusuable it is.

- The CYC upper ontology:



- Application and domain ontologies will typically overlap:



- How to merge and/or align them is an important problem.

XML

- XML (eXtensible Markup Language) isn't strictly an ontology language, but it is widely used to build simple ontologies.
- Can think of it as an extension to HTML.
  - Allows definition of new tags and document structures.
- In comparison to HTMML, XML makes it possible to include additional information about the content of a document.
- This can then be sued for the kind of reasoning that the *semantic web* is intended to provide.

```
-<ontology>
 -<conceptLevel>
  -<nodes>
   -<node id="1" usTitle="Carreras">
      <attribute id="1" title="Carreras" type="ST_LIST" min="" max=""/>
    </node>
    <node id="2" usTitle="Becas"> </node>
    <node id="3" usTitle="Misiones"> </node>
    <node id="4" usTitle="Area Técnica"> </node>
   </nodes>
  -<relations>
    <relation src="1" dest="2" relID="">dispone</relation>
    <relation src="1" dest="4" relID="">Contiene</relation>
    <relation src="4" dest="2" relID="">dispone</relation>
   </relations>
  </conceptLevel>
 -<conceptInstances>
  -<conceptInstance cid="1" id="1" usTitle="historia utpl">
    <attribute title="Carreras" usValue="Sistemas"/>
   </conceptInstance>
  </conceptInstances>
 -<relationInstances>
   <relationInstance id="1" srcCIID="1" destCIID="3" usRelDataTitle="dispone"/>
   <relationInstance id="2" srcCIID="1" destCIID="2" usRelDataTitle="Contiene"/>
  </relationInstances>
 </ontology>
```

OWL

- OWL is the current standard ontology language.
- In fact we have three languages which are OWL in some sense:
  - OWL Lite, restrictive by computationally efficient.
  - OWL DL, a *description logic* version of OWL, with a ontology-specific features (like the ability to express disjointness of classes).
  - ıOWL Full, highly expressive and very intractable.
- OWL comes with some basic ontology notions (Thing, Class) defined.

```
NS1:geographicCoordinates rdf:nodeID='A179'/>
    <NS1:mapReferences>North America</NS1:mapReferences>
    <NS1:totalArea>9629091</NS1:totalArea>
    <NS1:landArea>9158960</NS1:landArea>
    <NS1:waterArea>470131</NS1:waterArea>
    <NS1:comparativeArea>about half the size of Russia;
        about three-tenths the size of Africa; about half the size of
        South America (or slightly larger than Brazil); slightly larger
        than China; about two and a half times the size of Western Europe
    </NS1:comparativeArea>
    <NS1:landBoundaries>12034</NS1:landBoundaries>
    <NS1:coastline>19924</NS1:coastline>
    <NS1:contiguousZone>24</NS1:contiguousZone>
    <NS1:exclusiveEconomicZone>200</NS1:exclusiveEconomicZone>
    <NS1:territorialSea>12</NS1:territorialSea>
    <NS1:climate>mostly temperate, but tropical in Hawaii and Florida,
        arctic in Alaska, semiarid in the great plains west of the
        Mississippi River, and arid in the Great Basin of the southwest;
        low winter temperatures in the northwest are ameliorated occasionally
        in January and February by warm chinook winds from the eastern slopes
        of the Rocky Mountains
    </NS1:climate>
    <NS1:terrain>vast central plain, mountains in west, hills and low
        mountains in east; rugged mountains and broad river valleys in Alaska;
        rugged, volcanic topography in Hawaii
    </NS1:terrain>}
```

- After that digression, we can return to the KQML/KIF show.
- KQML is an agent communication language. It provides a set of *performatives* for communication.
- KIF is a language for representing domain knowledge. It can be used to writing down ontologies.

  KIF is based on first-order logic.
- Given that, let's look at some examples.

KQML/KIF dialogue I

```
A to B:  (ask-if
           (> (size chip1) (size chip2)))
B to A:  (reply true)
B to A:  (tell (= (size chip1) 20))
B to A:  (tell (= (size chip2) 18))
```

KQML/KIF dialogue II

```
(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))
)
```


KQML/KIF dialogue II (continued)

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

(eos
  :sender      B
  :receiver    A
  :in-reply-to q1
)
```

FIPA

- More recently, the Foundation for Intelligent Physical Agents (FIPA) started work on a program of agent standards — the centrepiece is an ACL.
- Basic structure is quite similar to KQML:
  - *performative*;
    20 performative in FIPA.
  - *housekeeping*;
    e.g., sender etc.
  - *content*
    the actual content of the message.

- Example

```
(inform
  :sender        agent1
  :receiver      agent5
  :content       (price good200 150)
  :language      sl
  :ontology      hpl-auction
)
```

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			

"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.

- 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.
- Note that the speaker only has to *believe* that what he says is true.

- Again Chamberlain provides an example, saying, a few months before the previous example:

My good friends this is the second time in our history that there has come back from Germany to Downing Street peace with honor. I believe it is peace in our time.

- He was wrong, but he seems to have believed what he said.

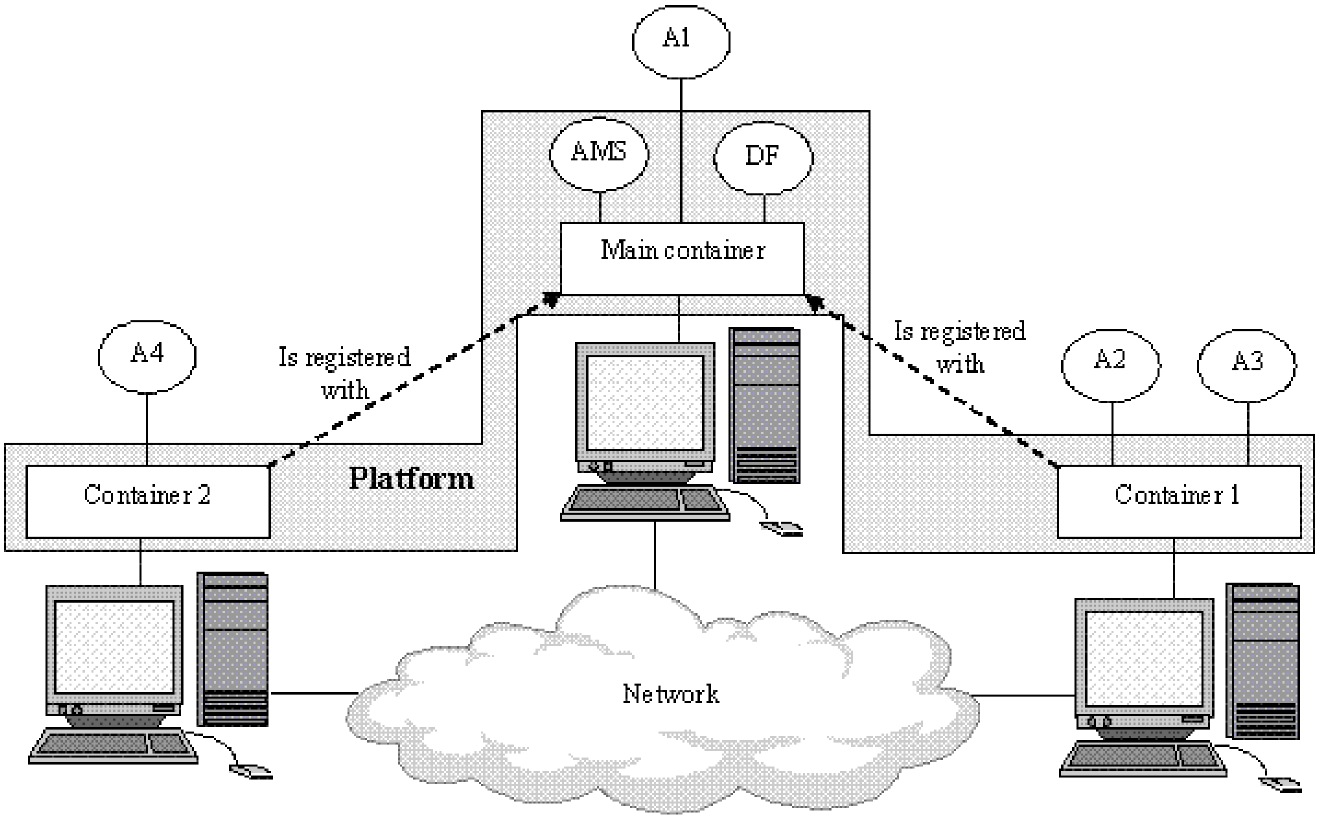
- For the “request” performative. . .
- The content is an *action*.
- Pre-condition is that sender:
  - intends action content to be performed;
  - believes recipient is capable of performing this action;
  - does not believe that recipient already intends to perform action.
- The last of these conditions captures the fact that you don’t speak if you don’t need to.

- Other performatives are:
  - `propose`
    One agent makes a proposal to another.
  - `accept-proposal`
    One agent states that it accepts a proposal made by another agent.
  - `reject-propose`
    One agent rejects a proposal previously made by another agent.
- The syntax of these is similar to that of inform.

JADE

- The FIPA ACL provides a language for writing messages down.
  - It says nothing about how they are passed between agents.
- Several software platforms have been developed to support ACL-based communication.
  - One of the most widely used is JADE.
- Provides transparent (from the perspective of the agent designer) transport of ACL messages.

- In JADE, agents are Java threads running in a “container”.



- All containers register with the *main container*

- The main container does the following:
  - Maintains the *container table* which lists all the containers and their contact information.
  - Maintains a list of all the agents in the system (including location and status).
  - Hosts the *agent management system* (AMS) which names agents as well as creating and destroying them.
  - Hosts the *directory facilitator* which provides a yellow pages allowing agents to be identified by the services they provide.
- See `http://jade.tilab.com/` for more details.

Alternative semantics

- There is a problem with the "mental state" semantics that have been proposed for the FIPA ACL.
- (This also holds for KQML).
- How do we know if an agent's locutions conform to the specification?
- As Wooldridge pointed out, since the semantics are in terms of an agent's internal state, we cannot *verfiy* compliance with the semantics laid down by FIPA.
- In practice, this means that we cannot be sure that a agent is being sincere.
- (Or, more importantly, we cannot detect if it is being insincere).

- This was exactly Chamberlain's problem.

- The people he was talking to lied to him.

- Singh suggested a way around this.
- Rather than define the conditions on a locution in terms of an agent's mental state, base it on something external to the agent.
- Move from a "mentalistic" semantics to a *social* semantics.
- How?
- Take an agent's utterances as *commitments*.
- But what does it mean to say that "if an agent utters an `inform` then it is committing to the truth of the proposition that is the subject of the utterance"?
- Doesn't stop an agent lying, but it allows you to detect when it does
  - For example when they say they want peace but then go and invade Poland.

Contestability semantics

- If an agent asserts that a proposition is true, then it is committing to *defend* that proposition.
- Any asserted proposition can be contested, and the assertor will have to provide an argument that supports it.
- If ever agent only asserts propositions for which it has an IN argument, and every agent only accepts propositions for which it is given an IN argument, then communication is *rational*.

  (We will talk about argumentation in a later lecture.)
- Agents can lie, but only if they have good reasons to support the untruths they tell.
- If agents lie, they run the risk of being caught out (because they have to justify what they say).

Summary

- This lecture has discussed some aspects of communication between agents.
- It has focussed on the interpretation of locutions/performatives as speech acts, and some suggestions for what performatives one might use.
- There is much more to communication that this...
- ...but this kind of thing is required as a "transport layer" to support the kinds of thing we will talk about later.

