

Agents and Multi-Agent Systems

Multi-Agent Systems

2023/2024

Ana Paula Rocha, Henrique Lopes Cardoso

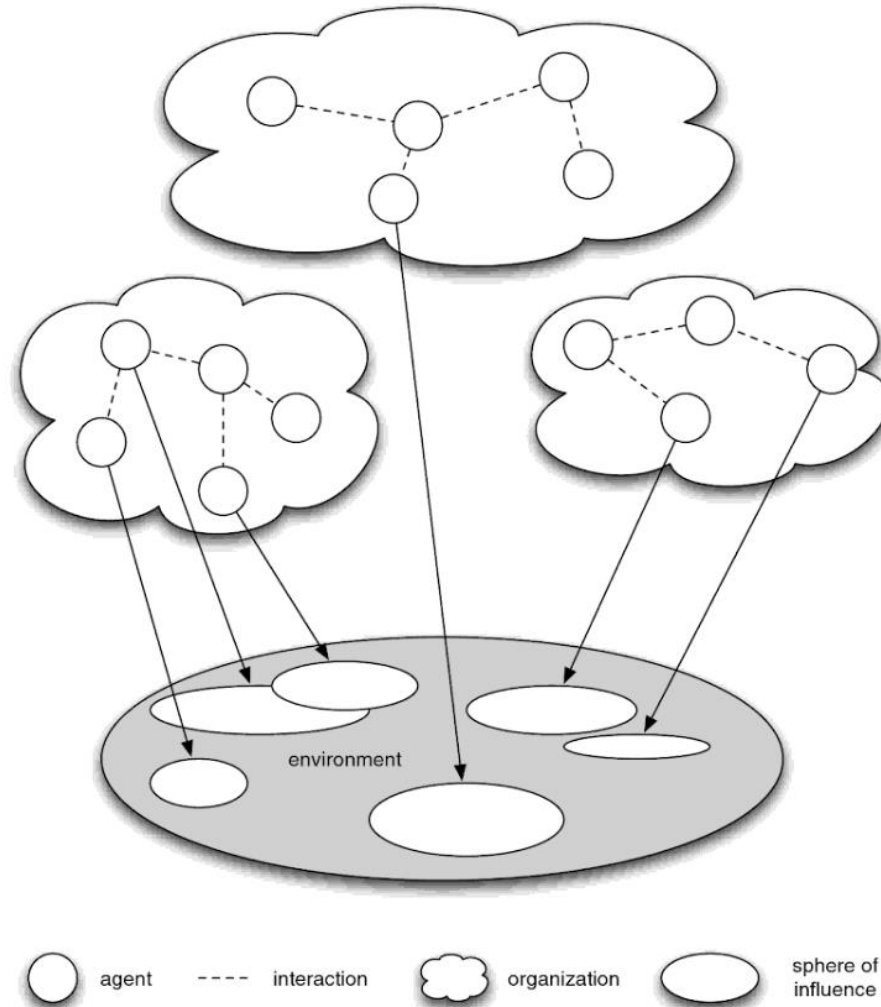
Multi-Agent System

- *“There's no such thing as a single agent system.”*

A *multiagent system* is one that consists of a number of agents, which *interact* with one-another. In the most general case, agents will be acting on behalf of users with different goals and motivations. To successfully interact, they will require the ability to *cooperate*, *coordinate*, and *negotiate* with each other, much as people do.

- ***Social ability***: intelligent agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives.

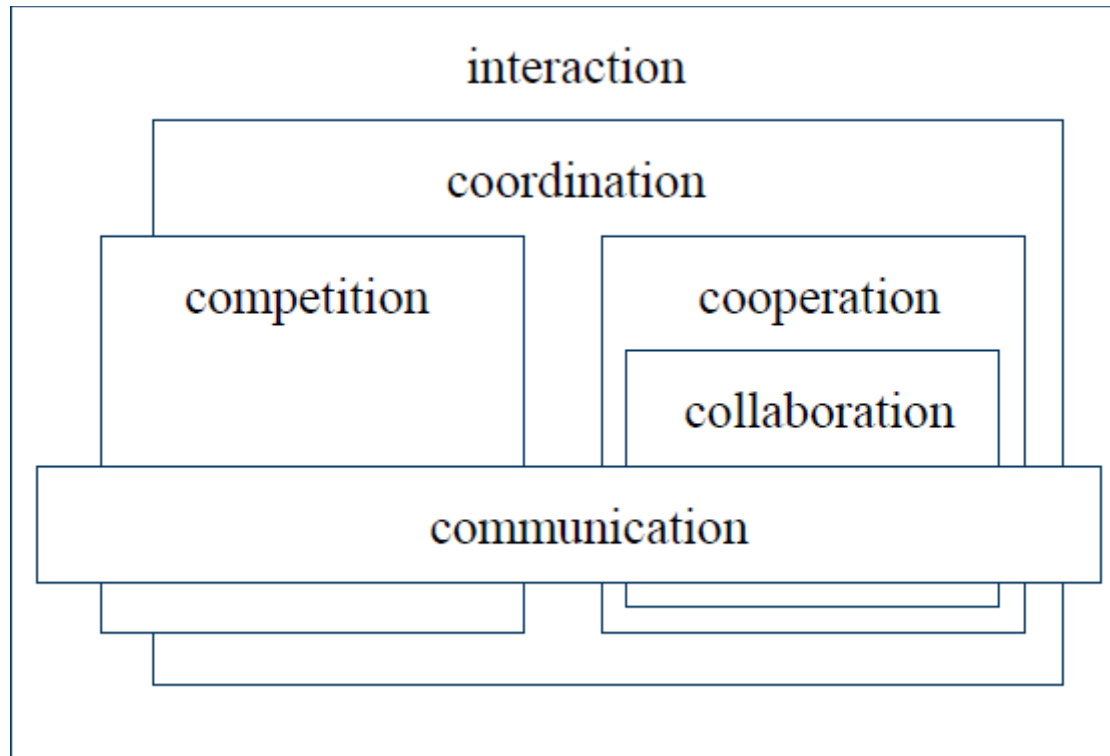
Multi-Agent System



Multi-Agent Systems

COMMUNICATION

Interaction



Communication

- Communication in OOP: method invocation
 - Object o_2 can communicate with object o_1 by invoking a (publicly) available method m_1 in o_1
 - `o1.m1(...)`
 - Object o_1 has **no control** over execution of m_1 : decision is made by o_2
- Communication in an agent-oriented setting
 - Agent a_1 has the capability to perform action α
 - Agent a_2 cannot ‘invoke a method’ in a_1 , since it is an **autonomous** agent
 - a_1 has control over both its **state** and its **behavior**
 - Agent a_2 can **ask** agent a_1 to perform the action, but is up to a_1 to actually perform it or not

“Objects do it for free, agents do it for money!”

Speech Act Theory

- Communicative actions
 - Agents can *influence* other agents
- Multi-agent approaches to communication are based on **speech act theory** [Austin, 1962] [Searle, 1969]
 - How utterances are used to achieve one's intentions
 - Utterances are just like “physical” actions to change the state of the world
 - *Performative verbs: request, inform, promise, ...*

Speech Acts

- Three aspects of a **speech act**

- *Locution*: physical utterance

- “Please make me some tea”

- *Illocution*: intended meaning

- “He requested me to make some tea”

- *Perlocution*: resulting action

- “He got me to make tea”

reflects the sender's
intention

determined by the
receiver's autonomy

- Two parts of a **speech act**

- **Performative** = communicative verb used to distinguish between different “illocutionary forces” (the *type* of speech act)

- Examples: inform, request, enquire, promise, ...

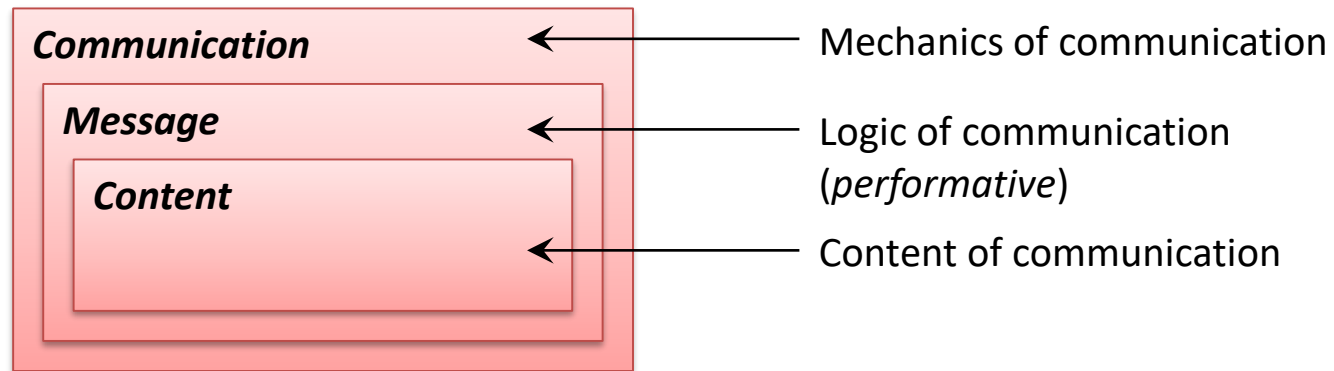
- **Propositional content** = what the speech act is about

- Example: “the window is open”

Agent Communication Languages

- Communication includes two kinds of semantics:
 - Semantics of the communication protocol (which must be domain independent)
 - Semantics of the enclosed message (which typically depends on the domain)
- Agent communication languages – two main efforts:
 - **Knowledge Sharing Effort (KSE)**
 - Knowledge Query and Manipulation Language (KQML): defines an envelope for messages, including the *performative* (message intent)
 - Knowledge Interchange Format (KIF): content language, similar to first-order logic
 - **Foundation for Intelligent Physical Agents (FIPA)**
 - Agent Communication Language (**ACL**): similar to KQML, but with more precise performatives

ACL Message Components



- **Content**: message proper, using a representation language and an ontology
- **Message**: wraps the message content – defines the type of interaction
- **Communication**: low-level parameters, such as the identity of the sender and receiver – “envelope”

ACL Message Components

- Typical message (performative + parameters)

(<performative>

:sender <word>

:receiver <word>

:language <word>

:ontology <word>

:content <expression>

...)

```
(inform
  :sender Amazon
  :receiver ag123
  :language Prolog
  :ontology BookShop
  :content price(AIMA,29.99) )
```

- The semantics of performatives is domain independent
- The semantics of the message is defined by *:content*, *:language*: and *:ontology*

Arguments

Parameter	Category of Parameters
performative	Type of communicative acts
:sender	Participant in communication
:receiver	Participant in communication
:reply-to	Participant in communication
:content	Content of message
:language	Description of Content
:encoding	Description of Content
:ontology	Description of Content
:protocol	Control of conversation
:conversation-id	Control of conversation
:reply-with	Control of conversation
:in-reply-to	Control of conversation
:reply-by	Control of conversation

Performatives

- Two main performatives:
 - **inform**: the sender wants the receiver to believe this content
 - basic mechanism for communicating information
 - **request**: the sender requests the receiver to perform an action
- All other performatives are defined in terms of these two
- **FIPA ACL Semantics**: meaning of **inform** and **request** defined in terms of
 - “feasibility precondition”
 - “rational effect”

FIPA ACL Semantics for Inform and Request

INFORM

- The content is a **statement**
- **Pre-condition:**
 - Sender **believes** the content is true
 - Sender **does not believe** the recipient is aware of whether the content is true
- **Rational effect:**
 - Sender **intends** the recipient to **believe** the content

REQUEST

- The content is an **action**
- **Pre-condition:**
 - Sender **believes** recipient can perform the action
 - Sender **does not believe** that recipient already **intends** to perform action
- **Rational effect:**
 - Sender **intends** the recipient to execute the action

Performatives

Performative	Passing information	Requesting information	Negotiation	Performing actions	Error handling
accept-proposal			×		
agree				×	
cancel		×		×	
cfp			×		
confirm	×				
disconfirm	×				
failure					×
inform	×				
inform-if	×				
inform-ref	×				
not-understood					×
propagate				×	
propose			×		
proxy				×	
query-if		×			
query-ref		×			
refuse				×	
reject-proposal			×		
request				×	
request-when				×	
request-whenever				×	
subscribe		×			

Ontologies

- If two agents are going to communicate about a certain domain, they must **agree on terminology**
- **Ontology**
 - A formal definition of a body of knowledge, involving a taxonomy of class and subclass relations coupled with their definitions
 - A formal specification of a shared conceptualization
- Many developments in ontology languages arise from interest in the semantic web
 - Add information to web pages such that it becomes possible for computers to process them
- Ontology languages
 - XML, OWL, RDF, ...

Ontologies

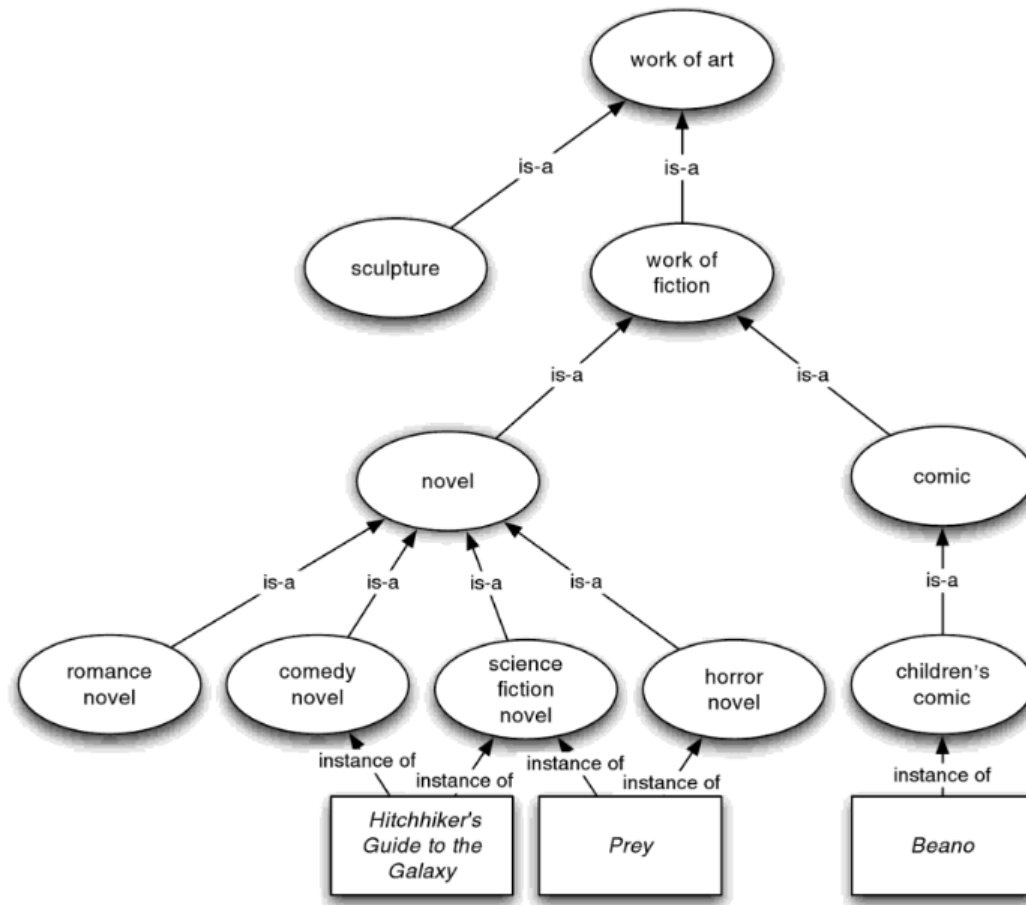


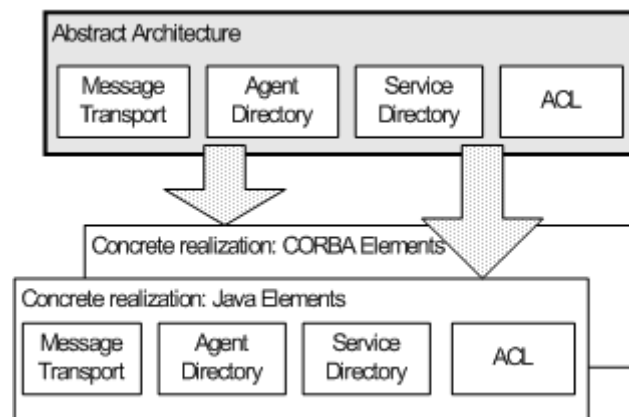
Figure 6.1: A fragment of Bob's knowledge after a conversation about the novel *Prey*. Classes are drawn as ovals, and instances as rectangles. Labels on arrows indicate the nature of the relationship between entities.

Multi-Agent Systems

FIPA

Foundation for Intelligent Physical Agents

- FIPA is an IEEE Computer Society standards organization that promotes agent-based technology
- FIPA **Abstract Architecture** Specification



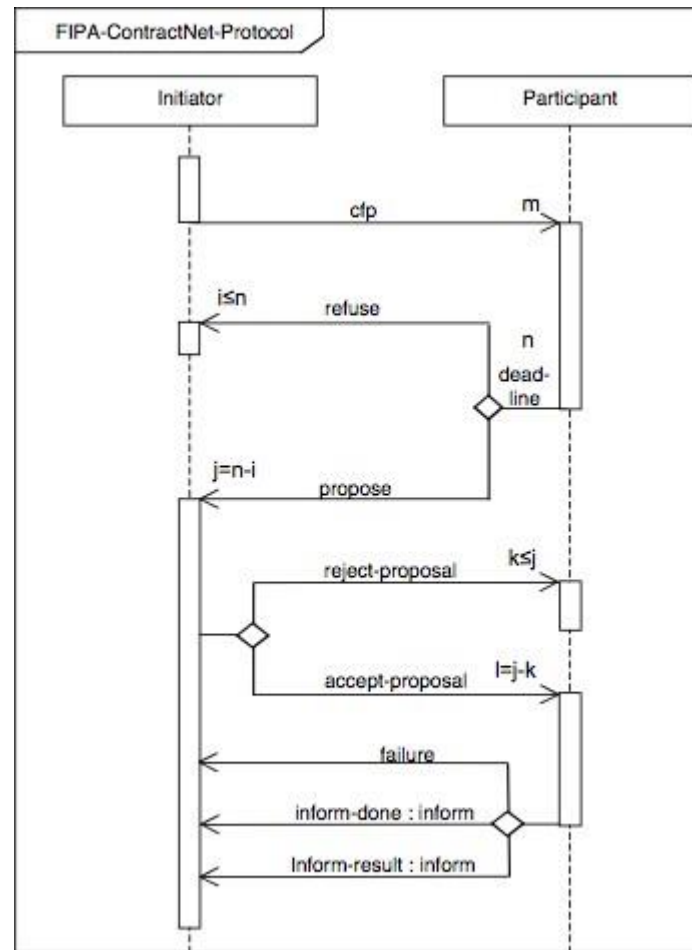
Foundation for Intelligent Physical Agents

- Agent Communication Language (**ACL**) Specifications
 - Message Structure
 - Communicative Act Library
 - Performatives
 - Content Languages
 - FIPA Semantic Language (SL) content language
 - Interaction Protocols
- Agent Management
- Agent Message Transport

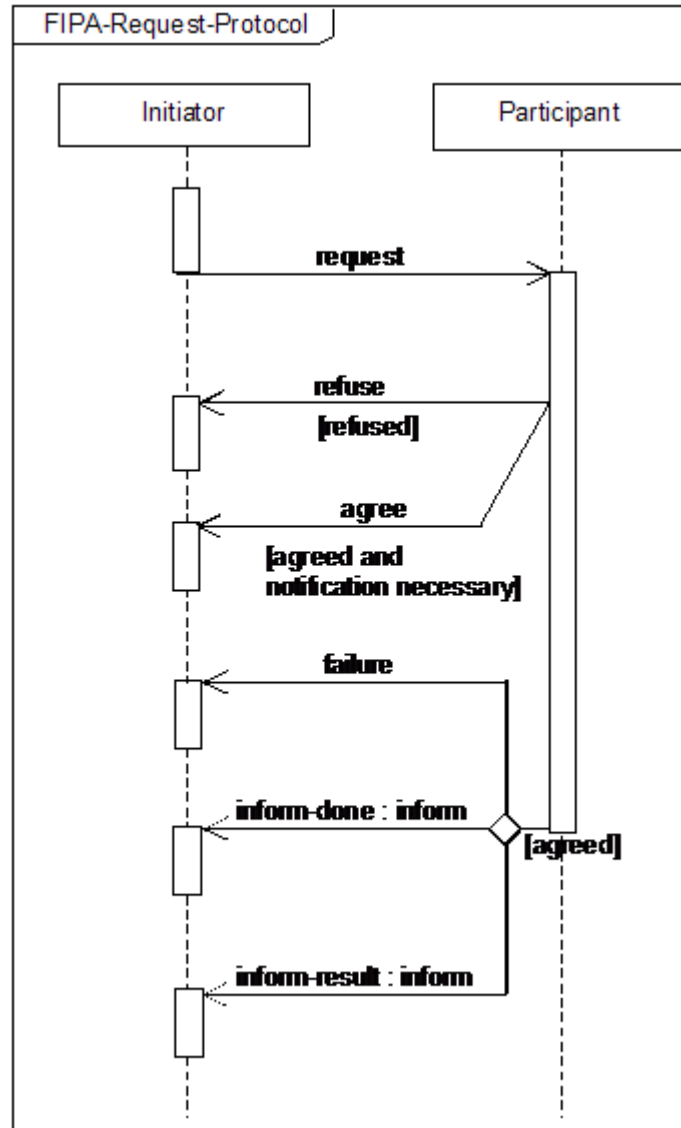
FIPA Interaction Protocols

- **Interaction protocols** define conversations, that is, sequences of messages that together define a semantically meaningful interaction
- One of the most basic and well-known is the **ContractNet** protocol:
 - A manager agent announces a task it wants to assign
 - Responder agents bid for the task execution
 - The manager assigns the task by comparing the bids
 - The assignee finally sends the result of task execution

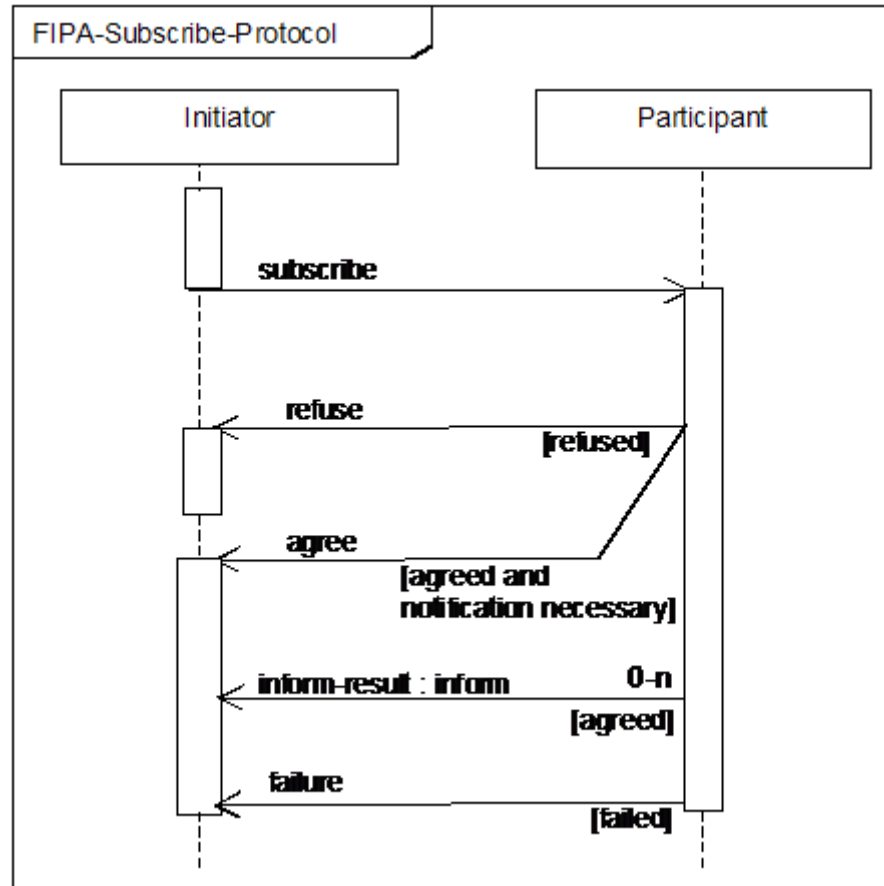
FIPA-ContractNet



FIPA-Request



FIPA-Subscribe



Multi-Agent Systems

MAS DEVELOPMENT METHODOLOGIES AND PLATFORMS

MAS Software Engineering

- **AOSE** (Agent-Oriented Software Engineering)
 - Abstractions: agent, environment, interaction protocol, context, roles, organizations, BDI
 - Methodologies: Gaia, MaSE, Prometheus, Tropos, Porto, ...
- **MAS programming constructs**
 - Agents (internal architecture and building blocks)
 - Infrastructure
 - Environment
 - Interaction artifacts/protocols (communication)
 - Distribution, mobility
- **Development tools**
 - IDE plugins, debugging
 - Agent and MAS visualization

MAS Development

- Some examples of platforms...

- JADE
- Jadex
- Cougaar
- Brahms
- SPADE



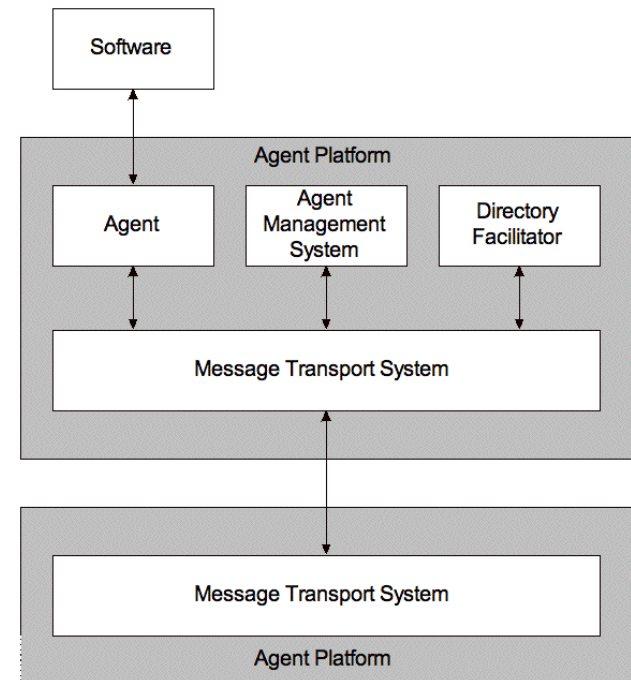
- ...and languages...
 - Jason (AgentSpeak)
 - 2APL
 - Concurrent MetateM



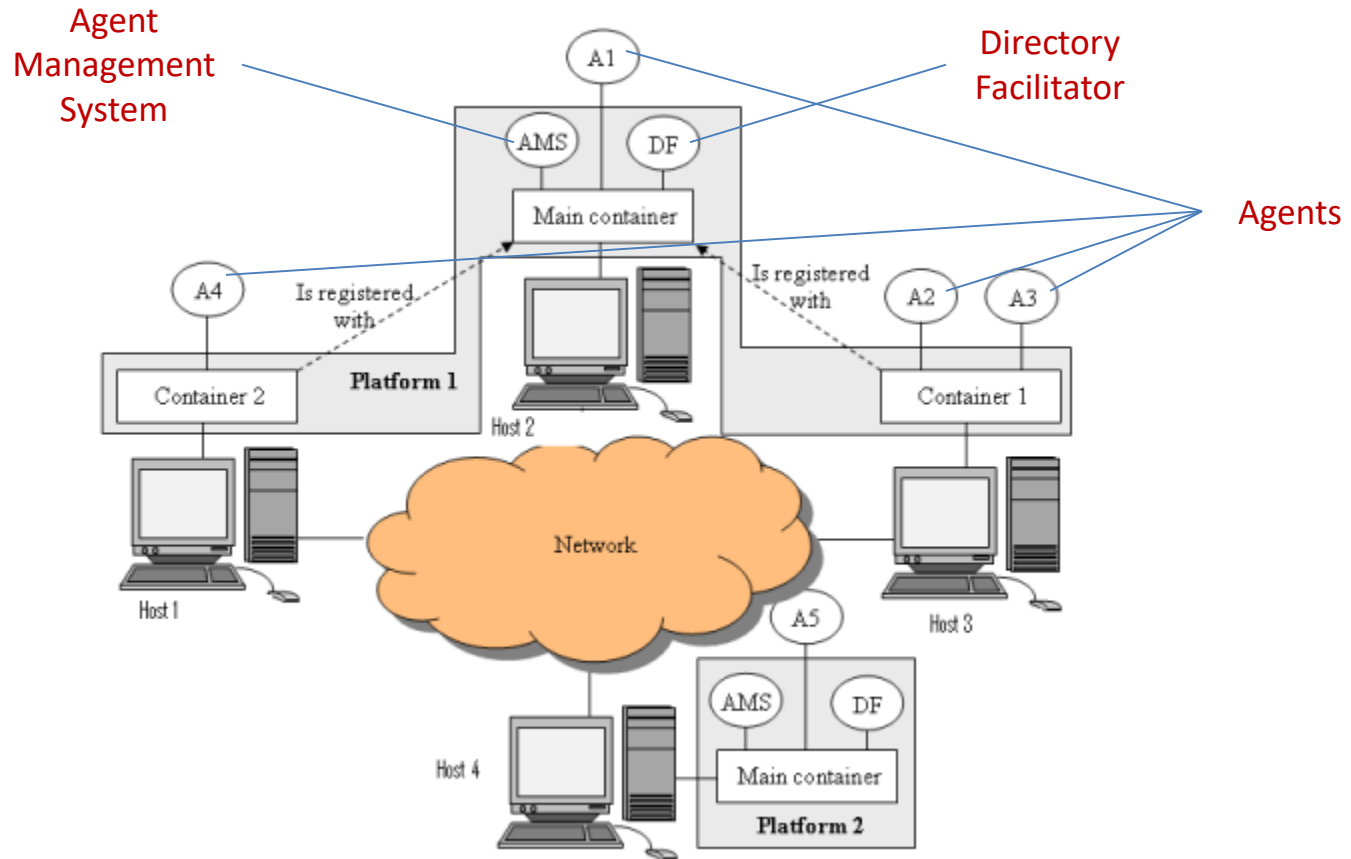
- ...and organizational/environment modeling and programming
 - Moise, CArtaGo, JaCaMo

JADE: a FIPA-compliant agent platform

- **FIPA-compliant**
 - Agent Platform
 - Agent Management System (**AMS**)
 - Directory Facilitator (**DF**)
 - Message Transport System (**MTS**)
 - Agent Communication Language (**ACL**)
 - Interaction Protocols

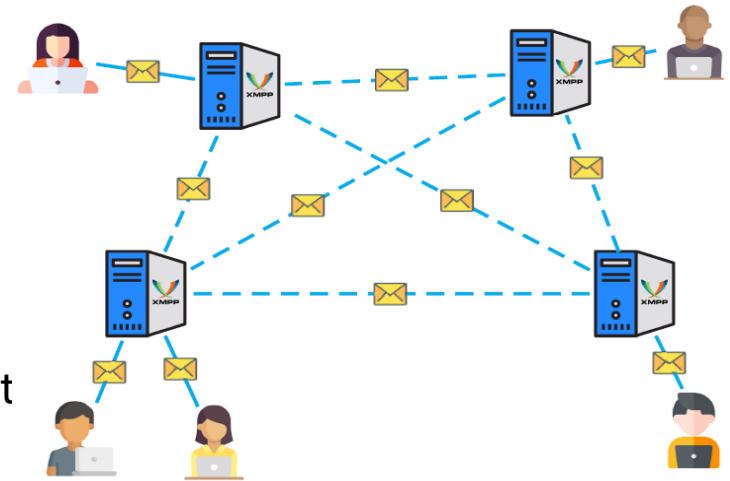


JADE Architecture



SPADE: Smart Python Agent Development Environment

- Multi-agent platform
- Based on the XMPP protocol
 - XMPP server as middleware, enables agent communication, discovery, and management
- Agent model based on behaviours
- FIPA-compliant
- Agent Communication Language (ACL)
- Web-based interface



Further Reading

- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*, 2nd ed., John Wiley & Sons: Chap. 7
- Austin, J. L. (1962). *How to do things with words*. Oxford University Press.
- Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge University Press.
- Shoham, Y. (1993). *Agent-oriented programming*. Artificial Intelligence 60(1), 51-92.
- FIPA: <http://www.fipa.org/>
- JADE: <https://jade.tilab.com/>
- SPADE: <https://spade-mas.readthedocs.io/en/latest/>