Communication among Agents

S. Costantini

Computer Science Dept. Univ. of L'Aquila, Italy

I have obtained this material (that I use for academic purposes only) by assembling and modifying material that I have found on the Web. I thank all the colleagues that put their material on-line and thus contribute to the dissemination on knowledge. I especially thank Prof. Yannis Labrou, Prof. Timothy Finin and Prof. Charles Nicholas for having re-used much of their material. I also thank Dr. Arianna Tocchio (Ph.D.) and Dr. Vasco Gallotti for their help. Of course, this resulting material is free for academic use.

Summary

- Present the general requirements of agent communication languages
- Sketch their conceptual and theoretical basis
- Describe some current languages and their realizations in software implementations
- FIPA standardization efforts

Some key ideas on Agents

- Software agents offer a new paradigm for very large scale *distributed heterogeneous applications*.
- The paradigm focuses on the *interactions* of autonomous, cooperating processes which can adapt to humans and other agents.
- Mobility is an orthogonal characteristic which many, but not all, consider central.
- Intelligence is always a desirable characteristic but is not required by the paradigm.
- The paradigm is still forming.

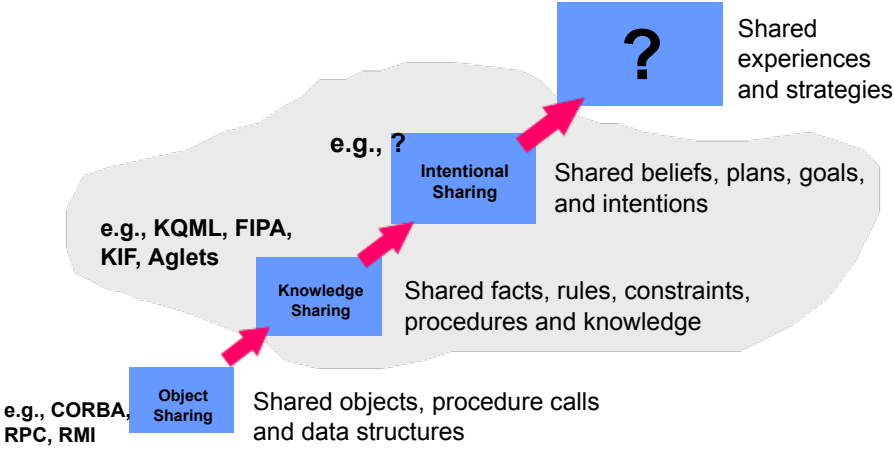
Why is communication important?

- Most (but not all) would agree that communication is a requirement for cooperation.
- Societies can do things that no individual (agent) can.
- Diversity introduces heterogeneity.
- Autonomy encourages disregard for other agents' internal structure. (Gli agenti cercano di capire cosa sanno gli altri)
- Communicating agents need only care about understanding a "common language". (Gli serve un linguaggio comune (anche non noto))

What is communication?

- Communication almost always means "communication in a common language"
- "Language" does not include natural languages only.
- 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, affect one's environment, etc.

Some ACLs (Agent Communication Languages) and their envisaged perspectives



Agent Communication, at the technical level

- Messages are transported using some lower-level transport protocol (SMTP,TCP/IP, HTTP, IIOP, etc.)
- An Agent Communication Language (ACL) defines the types of messages (and their meaning) that agents may exchange.

Agent Communication at the conceptual level

- Agents may engage in "conversations." Conversations follow a "protocol". There can be interaction protocols for, e.g. negotiation, auction, etc.
- Conversation: task-oriented, shared sequences of messages.
- The agent's communicative (as well as non-communicative) behavior is driven by internal higher-level conceptualization goals and strategies.

Agent Communication at the conceptual level

- What does an agent mean to communicate? Just "facts"? No!
- Often, agents have (and wish to exchange) "propositional attitudes"

La Comandante che Ignora l'Ordine

Propositional attitudes

- With respect to a content-bearing proposition (e.g., "*it is raining*"), an agent may associate to it:
  - a propositional attitude, such as believing, asserting, fearing, wondering, hoping, etc.
- <*a*, fear, raining(tnow) >
- Usually, each agent is able to express a finite number of propositional attitudes. (Hanno un numero predefinito di atteggiamenti proposizionali)

Propositional Attitudes and Speech Act Theory (TEORIA DELLA COMUNICAZIONE UMANA)

! DOMANDA DA ORALE

- This suggests to model agent communication by means of "Speech Act Theory", which is a high-level framework to account for human communication.
- Each communication is understood and a particular action called a "speech act".
- Communication involves more than the bare content.

Propositional Attitudes and Speech Act Theory

- Speakers do not just utter true or false sentences
- Speakers perform speech acts:
  - requests, suggestions, promises, threats, etc.
- Every utterance (thing which is said) is a speech act

Speech Act Theory (continued)

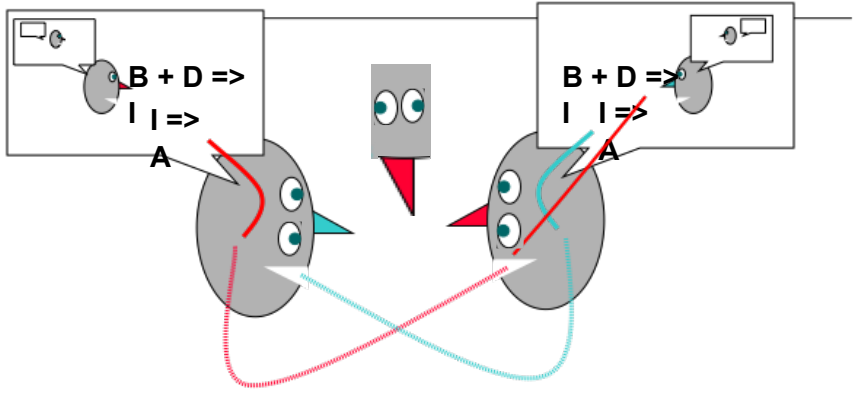
Example: *"Shut the door !"*

- **locution** (fornisce contesto al messaggio)
  physical utterance with context and reference, *i.e.*, who is the speaker and the hearer, which door etc.
- **illocution** (sono le intenzioni del msg)
  the act of conveying intentions, *i.e.*, speaker *wants* the hearer to close the door
- **perlocutions** (conseguenza del msg sui recevent)
  actions that occur as a result of the illocution, *i.e.*, hearer closes the door and the door in thus closed

BDI: a good framework for giving a semantics to agent communication

- BDI architectures describe the internal state of an agent by the mental states of beliefs, goals and intentions
- BDI theories provide a conceptual model of the knowledge, goals, and commitments of an agent
- BDI agents have some (implicit or explicit) representations of the corresponding attitudes

BDI Model and Communication



- Communication is a means to (1) reveal to others what our BDI state is and (2) attempt to affect the BDI state of others.
- Note the recursion: an agent has beliefs about the world, beliefs about other agents, beliefs about the beliefs of other agents, beliefs about the beliefs another agent has about it, ...

ACL Messages

- In any ACL, each message contains at least the following information:
  - Sender
  - Receiver
  - Performative, i.e. propositional attitude such as inform, request, accept, refuse, etc.
  - Content, i.e. the proposition which is the object of the propositional attitude, i.e., agent A informs agent B that "it rains".
  - Ontology: A common vocabulary and agreed upon to describe a subject domain so that the receiver can understand the content. Sometimes there is an implicit shared ontology…

BDI theories, speech acts and ACLs: How do they all fit together?

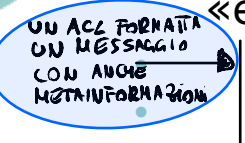
- ACL messages can be understood as speech acts
- Speech acts may be understood in terms of a intentions: e.g., "agent A intends to inform agent B of the fact that..."
- An intentional description makes references to beliefs, desires, intentions and other modalities
- BDI frameworks have the power to describe an agents' behavior, including communicative behavior

Criticism of BDI theories

- BDI logics do not have complete axiomatizations
- they are not efficiently computable
- However, they are useful for conceptualization.
- In practice: semantics of an ACL can be specified in BDI terms, but:
  - semantics is not really implemented
  - the implementation tries however to "simulate" the semantics by respecting its principles.

Principles of ACL's

- Agent Communication Languages aim at providing a general way of encapsulating (in a so-called «envelop»:
  - Performative (propositional attitude)
  - content
  - auxiliary information (sender, receiver, ontology, etc.)



Principles of ACL's

- Agent Communication Languages try to be independent of specific aspects related to:
  - the underlying implementation of message-passing, including network protocols
  - the language/formalism/approach on which agents that adopt the ACL are based

Principles of ACL's

- Then, an Agent Communication Language:
  - can be implemented in different ways on top of different kinds of networks.
  - does not care about the language in which the communicationg agents are written.
  - does not care about the language in which the contents are expressed.
  - does not care about the enclosed ontology (if any).

AGLI ACL NON GLIENE IMPORTA UN CAZZO DI NIENTE DI TUTTA STA ROBA

A popular ACL: KQML

Knowledge Query and Manipulation Language

- KQML includes primitive message types of particular interest for building complex agent architectures (e.g., for mediators, sharing intentions, etc.)
- There are also many dialects and "extended" versions of KQML that include important concepts not addressed in the basic version (e.g., security).

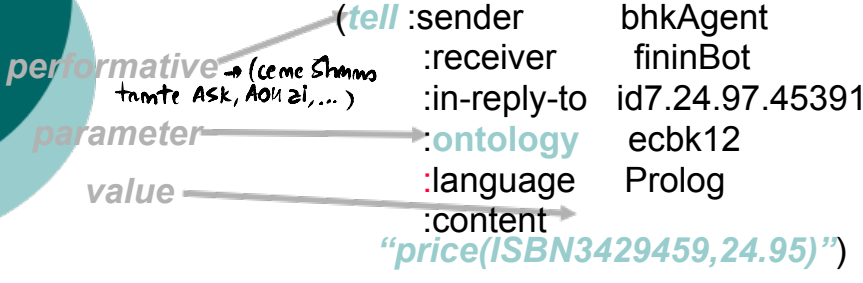
La sintassi non fa il linguaggio

KQML Syntax

- KQML was originally defined as a language with a particular linear syntax which is based on Lisp (i.e., it uses lists as basic structure).
- Alternate syntaxes have been used, e.g., based on SMTP, MIME, HTTP, etc.)
- KQML has also been mapped onto objects and passed from agent to agent as objects

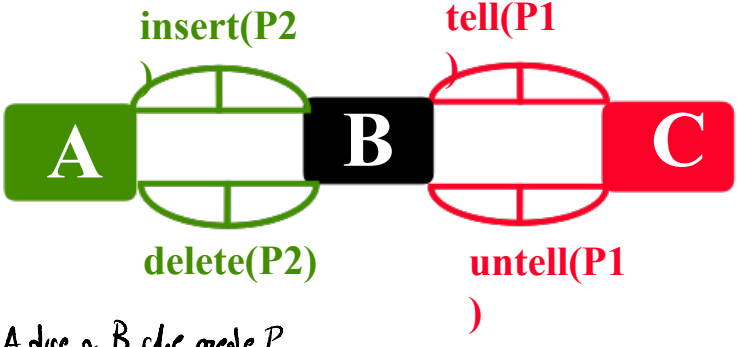
KQML principles do not really depend on syntax.

Example: a KQML Message



- *speech act* or *performative* : tell
- other possible performatives: ask, reply, subscribe, achieve, monitor, ...

Example of Performatives: Tell, Untell, Insert and Delete



- **tell(P)** says that the sender believes P to be true.
- **untell(P)** means that the sender does not believe P.
- **insert(P)** is a request for the receiver to add P to its *knowledge base"* and **delete(P)** to remove it.

Example: a KQML Message


```
(tell :sender       bhkAgent
      :receiver     fininBot
      :in-reply-to  id7.24.97.45391
      :ontology     ecbk12
      :language     Prolog
      :content
"price(ISBN3429459,24.95)")
```

attribute/value pairs:

- :content *"price(ISBN3429459,24.95)"*
- :language Prolog
- ...

KQML Attributes (or “parameters”)

:sender the actual sender of the performative

:receiver the actual receiver of the performative

:from the origin of the performative in *:content* when forward is used

:to the final destination of the performative in *:content* when forward is used

:in-reply-to the expected label in a response to a previous message (same as the *:reply-with* value of the previous message)

:reply-with the expected label in a response to the current message

:language the name of the representation language of the :content

:ontology the name of the ontology (e.g., set of term definitions) assumed in the :content parameter

:content the information about which the performative expresses

Example: a KQML Message

performative

parameter

value

```
(tell :sender       bhkAgent
      :receiver     fininBot
      :in-reply-to  id7.24.97.45391
      :ontology     ecbk12
      :language     Prolog
      :content
"price(ISBN3429459,24.95)")
```

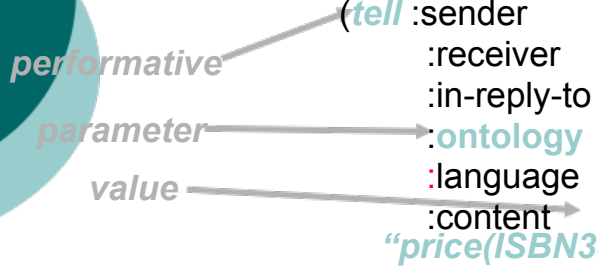
semantics

tell(*i*,*k*, Biϕ) = Bi Biϕ $\wedge$ $\neg$ Bi(Bk Biϕ $\vee$ $\neg$ Bk Biϕ) $\wedge$ ...

sender i beliefs (is aware) to believe ϕ and is convinced that k has no "opinion" about that, i.e., that k neither beliefs nor disbelieves that i believes ϕ

Io ricordo di credere ϕ, chiedo a B cosa crede che io creda e mi baso su quello per decidere (si può andare avanti: B crede di credere che io creda)

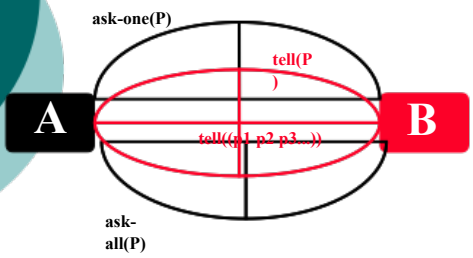
Example: a KQML Message

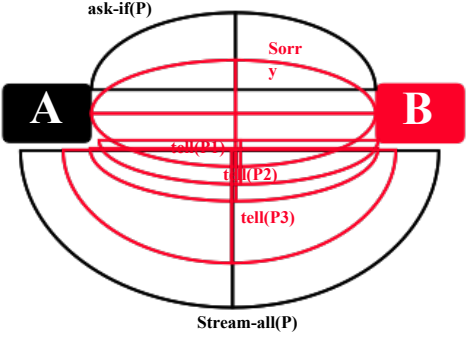


protocol

- what the sender expects the receiver to reply
- there are specific rules (often expressed at finite state automata) that specify how the other agent is allowed to reply

Example of Protocol





- The **ask-one**, **ask-all**, **ask-if**, and **stream-all** performatives provide a basic query mechanism.
- To an **ask** the other agent can reply either with **tell** or **stream-all** (providing the requested information) or with **sorry** if cannot or does not want to provide it

KQML Implementation

- KQML specification includes the description of the underlying software infrastructure. (L'HARDWARE CHE LO SUPPORTA)
- The infrustructure supports basic performatives and protocols, but also higher-level performatives and pre-defined protocols for complex multi-agent activities, e.g.:
  - negotiation
  - advertising
  - brokering

KQML Use

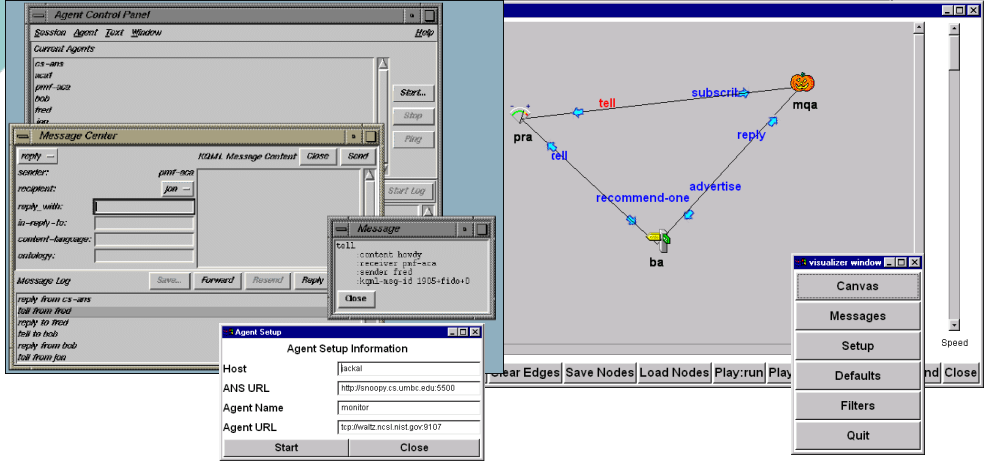
- High-level peformatives allow “Facilitators” and “Brokers” to be easily defined.
- Facilitators are a class of agents that provide communication services such as:
  - message forwarding and broadcasting
  - resource discovery
  - matchmaking
  - content-based routing
  - meta-knowledge queries

KQML Use

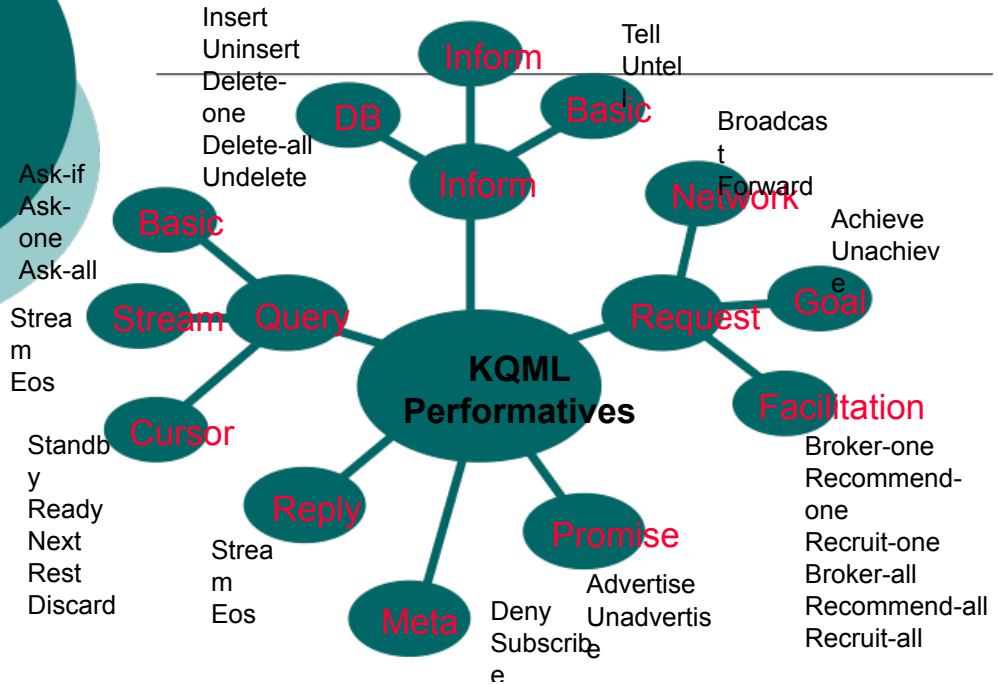
- Performatives of special interest to facilitators are
  - advertise, broker, recruit, recommend, forward, broadcast, etc.
- Brokers are generally considered to focus on matchmaking (Mettamo in communications Routing)
- Facilitators can be intelligent or not
  - Intelligent facilitators use domain knowledge in matching services needs and offers.

KQML Utility Agents: pre-defined facilitators

Standard service agents like Agent Name Server, Logger, Message Router, Authenticator, Broker, Communication Visualizer, Control panel and Proxy Agent aid the development of agent-based systems.



KQML Performatives



Anoter popular ACL: FIPA ACL
What is FIPA

- The Foundation for Intelligent Physical Agents (FIPA) is a non-profit association that has been active for some years and has now become an IEEE Committee.
- FIPA's purpose has been to promote the success of emerging agent-based applications, services and equipment by means of internationally agreed specifications that maximise interoperability across agent-based applications, services and equipment.

www.FIPA.org

Who is FIPA

W LA

Sulle slides ci sono solo i Principi del FIPA

- FIPA has operated through the open international collaboration of member organisations, which are companies and universities active in the agent field.
- Companies: Alcatel, Boeing, British Telecom, Deutsche Telekom, France Telecom, Fujitsu, Hitatchi, HP, IBM, Fujitsu, Hewlett Packard, IBM, Intel, Lucent, NEC, NHK, NTT, Nortel, Siemens, SUN, Telia, Toshiba, etc.
- Universities and Research Institutes: GMD, EPFL, Imperial, IRST, etc.
- Government Agencies: DARPA

FIPA Standard
Agent Communication Language

- Called FIPA ACL
- Based on speech acts
- Messages are actions (communicative actions or "Communicative Acts", or CAs)
- Communicative acts are described in both a narrative form and a formal semantics based on modal logic
- Syntax is similar to KQML
- Specification provides a *normative* description of high-level interaction protocols

Agent-Standardization - FIPA Cooperation between Agents

CAs for Information Exchange

- proposition or reference as content
- Basic CAs:
  - inform
  - query-ref
  - not-understood
- Advanced CAs:
  - inform-if, inform-ref
  - confirm, disconfirm
  - subscribe

Agent-Standardization - FIPA Cooperation between Agents

CAs for task delegation

- action-description as content
- Basic CAs:
  - request
  - agree
  - refuse
  - failure
  - not-understood
- Advanced CAs:
  - request-when, request-whenever
  - cancel

Agent-Standardization - FIPA Cooperation between Agents

CAs for Negotiation

- action-description and proposition as content
- Initiating CA
  - Cfp (call for proposals)
- Negotiating CA
  - propose
- Closing CAs
  - accept-proposal
  - reject-proposal

FIPA ACL Example

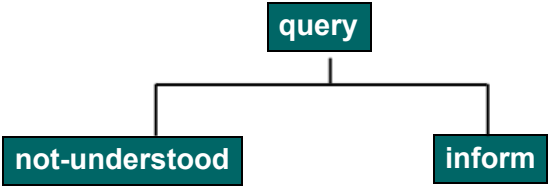
```
(request
   :sender   (:name
   user_agent@bond.mchp.siemens.de:3410)
   :receiver (:name
   hilton_hotel@tcp://hilton.com:5001)
   :ontology fipa-pta
   :language SL
   :protocol fipa-request
   :content
   ( action hilton_hotel@tcp://hilton.com:5001
     ( book-hotel    (:arrival   04/07/1999)
   (:departure 12/07/1999)
     (:infos  (    ))
     )))
```

FIPA ACL

- CAs have their own formal semantics
  - difficult to implement
  - need not be implemented - agent must behave according to semantics
- Interaction protocols (IPs) define structured conversations
  - based on CAs
  - basis for dialogues between agents
  - basic set of pre-defined IPs
  - own IPs can be defined

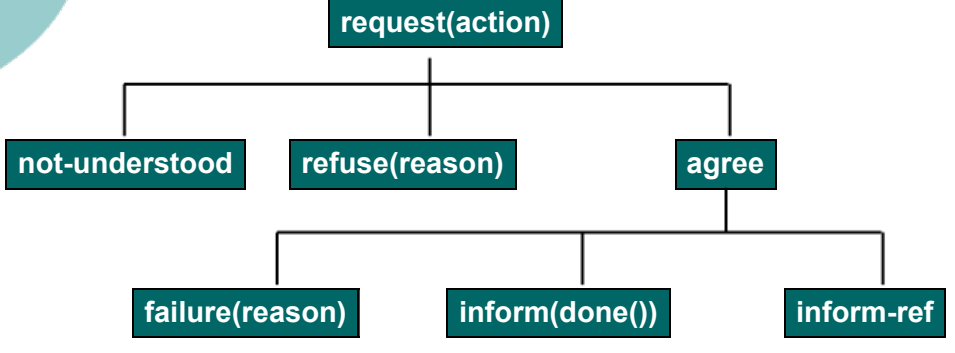
FIPA Performatives for Cooperation among Agents

FIPA-Query (simplified - for information exchange)



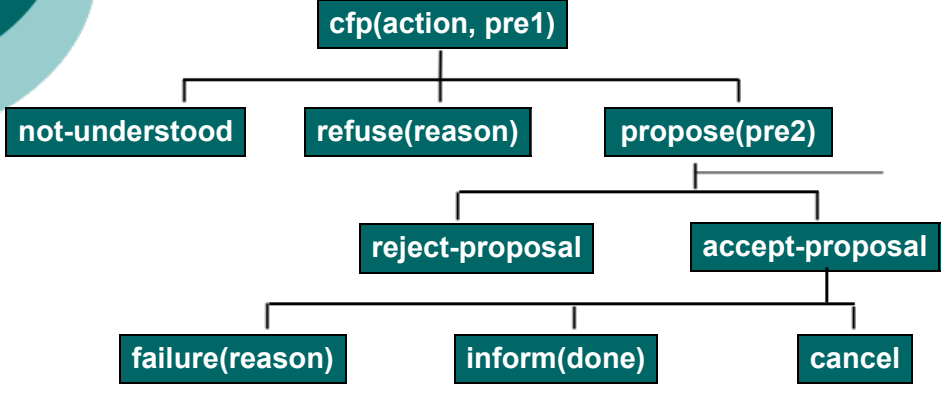
FIPA performatives for Cooperation among Agents

FIPA-Request - for task delegation



FIPA performatives for Negotiation among Agents

FIPA-Contract Net - for negotiation

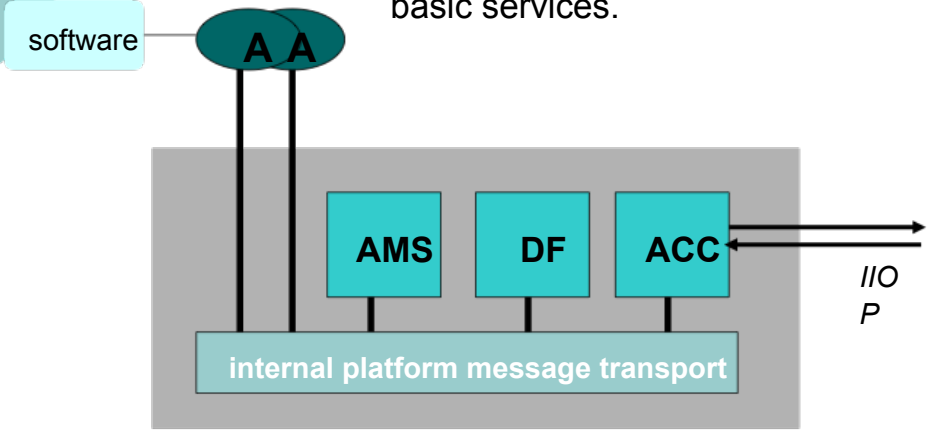


FIPA Agent Platform

- FIPA specification also defines the basic features of the underlying software infrastructure, called “FIPA Agent Platform”.
- It is a guideline for implementation.
- Real implementation follow the definition at least to some extent.

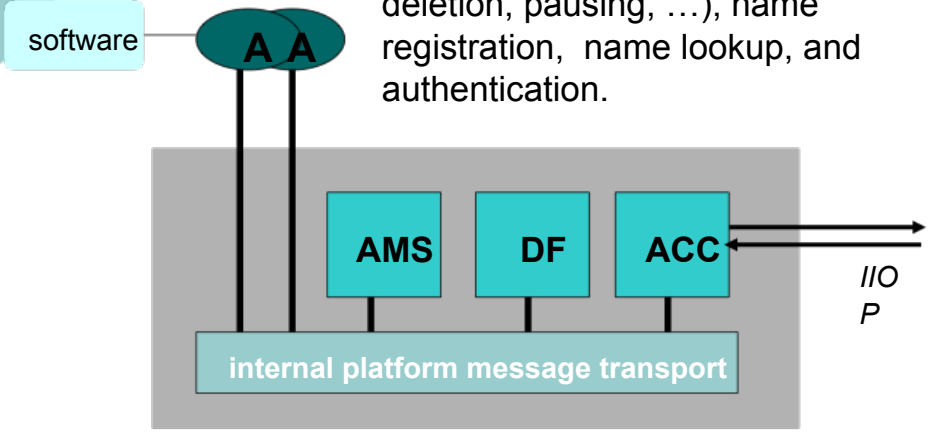
FIPA Agent Platform

Agents belong to one or more agent platforms which provide basic services.



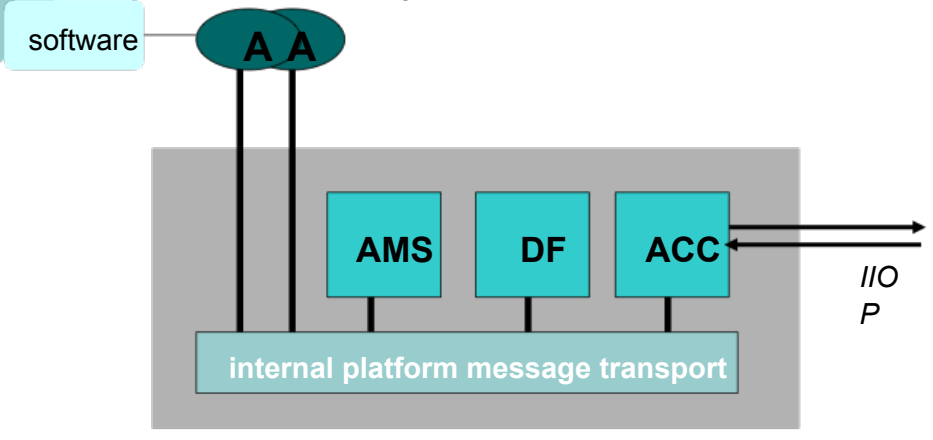
FIPA Agent Platform

The **AMS** (Agent Management System) provides services like lifecycle management (creation, deletion, pausing, …), name registration, name lookup, and authentication.



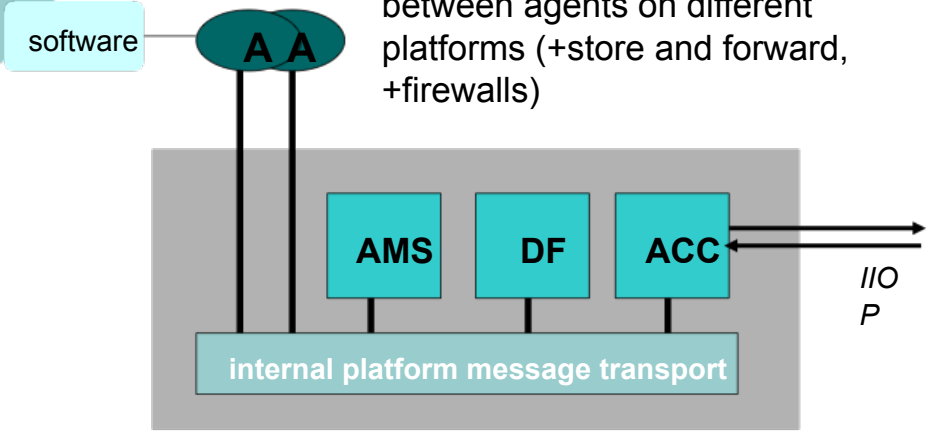
FIPA Agent Platform

The **DF** (Directory Facilitator) provides yellow pages services which describe the attributes and capabilities of agents in the platform.



FIPA Agent Platform

The **ACC** (Agent Communication Channel) accepts and delivers message between agents on different platforms (+store and forward, +firewalls)



FIPA Platform Implementations

- Several platforms have been implemented
  - Zeus (BT)
  - Mecca (Siemens)
  - Spawar
  - JADE (Cselt)
  - Nortel
  - Comtec

and interoperability has been demonstrated.

Different ACLs: different semantic approaches

- Different approaches to the semantics of an ACL
  - KQML semantics (Labrou 1996)
  - FIPA ACL (FIPA ACL specification)
  - ACL semantics (Cohen & Levesque): tries to model the communication process "in general" focusing on what communication should mean, independently of the particular ACL

The Cohen & Levesque Approach: Rational Agency

- A communicative act is a particular kind of intention
- Intention = Choice + Commitment
- A (partial) theory of rational agency
- Possible-worlds semantics but explicit definition of the intended meaning

Commitments and Intentions

- Internal Commitment to achieve goal p:

  (P-GOAL x p q) =

  (1) (BEL x □p) □

  (2) (GOAL x (LATER p)) □

  (3) [KNOW x (PRIOR [(BEL x p) □ (BEL x □□p) □ (BEL x □q)] =>

  □ [GOAL x (LATER p)])]

meaning

" (1) agent x believes p is currently false

(2) chooses to try to make p true later

(3) and x knows that before abandoning that choice, he must either believe that p is true, or that it will never be true, or that some q (an escape clause) is necessarily true"

The p of P-GOAL

What will just happen i.e., x believes that a will happen next, with escape clause q

Intention

(INTEND x a q) =
(P-GOAL x [DONE x (BEL x (WILL-HAPPEN a)) q])

x has the persistent goal of reaching a state at which it believes that a will happen, after which (state) action a should happen

Intending is a special kind of commitment

The agent is committed to arriving at a state in which he is about to do the intended action next

Thus an agent cannot be committed to doing something accidentally or unknowingly

"I intend for the sun to rise tomorrow" vs
"I intend to get an "A" in this course"

Thoughts on C & L Intention

- Just because an agent intends something, it does not mean that the agent will even attempt to achieve it
  - remember the "escape clause" in the P-GOAL definition
    a "pessimistic" agent might drop all its goals because "the sky is cloudy" or "the weather forecast is bad" or for any other reason

Thoughts on C &L Intention

- The definition of intention does not *guarantee* a causal relationship between the agent's course of action and "action a occurring"
  - the agent is only required to reach a state that the agent believes that will lead to "the action a occurring". There are possible failures, and there is the escape clause.

ACLs a la Cohen & Levesque

- Communicative acts (CAs) are attempts to communicate
- C&L define CAs as attempts that involve two (or more) rational agents (teams)
- Interesting work that focuses on defining rational agents and describing team formation.

Semantics for INFORM

Precondition

{INFORM speaker listener p} = The "honest effort"

{ATTEMPT speaker listener
(know listener p)
[BMB listener speaker *(BMB = Broadcast/make public)*
(P-GOAL speaker (KNOW listener (KNOW speaker p)))]}

- An INFORM is defined as an attempt in which to make an "honest effort", the speaker is committed to making public that he is committed to the listener's knowing that he (the speaker) knows p.

KQML Semantics (Labrou 1996)

- *Preconditions* indicate the necessary state for an agent in order to send a performative and for the receiver to accept it and successfully process it.
- *Postconditions* describe the states of both interlocutors after the successful utterance of a performative (by the sender) and after the receipt and processing (but before a counter utterance) of a message (by the receiver).
- *Preconditions* indicate what can be assumed to be the state of the interlocutors involved in an exchange. Similarly, the *postconditions* are taken to describe the states of the interlocutors assuming the successful performance of the communication primitive

Semantics for TELL

TELL(A,B,X)

- A states to B that A believes X to be true (for A).
- bel(A,X)
- Pre(A): bel(A,X) □ know(A,want(B,know(B,S)))
  where S may be bel(B,X) or NOT(bel(B,X))

  Pre(B): intend(B,know(B,S))
- Post(A): know(A,know(B,bel(A,X)))

  Post(B): know(B,bel(A,X))
- Completion: know(B,bel(A,X))
- The completion condition and postconditions hold unless a SORRY or ERROR suggests B's inability to properly acknowledge the TELL.

Outline of FIPA ACL Semantics

- A primitive's meaning is defined in terms of FPs and REs
- The Feasibility Preconditions (FP) of a CA define the conditions that ought to be true before an agent may plan to execute the CA
- The Rational Effect (RE) is the effect that an agent hopes to bring about by performing an action (but with no guarantee that the effect will be achieved)
- The FPs and the REs involve agents state descriptions

An example of FIPA ACL semantics (inform)

<*i*, inform(*j*, □)>

FP: Bi□ □ □ Bi(Bj□ □ Bj□□)

RE: Bj□

Compare with KQML: Bi Bi □ □ Bi□ □ □ Bi(Bj□ □ Bj□□)

Agent i informs agent j that (it is true that) it is raining today:

```
(inform
   :sender i
   :receiver j
   :content "weather(today,raining)"
   :language Prolog
   :ontology weather42)
```

Comparison of KQML tell and FIPA ACL inform

- The difference is only observable in the semantics
- Syntactically the two messages in KQML and FIPA are almost identical
- Both languages make the same basic assumption of non-commitment to a content language
- Semantically they differ at two levels:
  - different ways to describe the primitive, i.e., pre-, post-, completion conditions for KQML, FPs and REs for FIPA ACL
  - different logic language to describe the propositional (mental) attitudes, e.g. belief

ow do KQML and FIPA ACL differ?

- Different semantics
- Different treatment of the “administration primitives”; in FIPA ACL register, unregister, etc., are treated as requests for action with reserved (natural language) meaning
- No “facilitation primitives”, e.g., broker, recommend, recruit, etc., in FIPA ACL

Which ACL should I use?

- Programmers do not care about semantics and their details.
- As long as the agent does not *implement* modalities (belief, intention, etc.) the semantic differences are irrelevant to the developer.
- The similar syntax guarantees that a developer will not have to alter the code that receives, parses and sends messages.
- The code that processes the primitives should change depending on whether the code observes the proper semantics.
- Most widely adopted ACL: FIPA

Conversations

- Conversations define allowed/useful/desirable sequences of messages for particular tasks and indicate where/how messages "fit" in exchanges.
- Desiderata:
- Allow more intuitive and convenient method for handling messages in context.

Addressing the shortcomings of the semantics with conversations

- Both KQML and FIPA ACL include specifications for conversations (or conversation protocols)
- Conversations are not part of the semantic definition of the ACL
- Conversations shift the focus to an agent's observable behavior
- Programmers might find conversations more useful than formal semantics
- The meaning of primitives is often context/situation dependant and conversations can accommodate context

The Contract Net Protocol

- Is an important example of a widely-used negotiation protocol that can be implemented via the main ACL's
- An important generic protocol, widely used, employing a centralized "manager" agent
- There can be several participant agents who are available to perform tasks in view of some "common interest"

The Contract Net Protocol

- The manager is able to communicate
  - peer-to-peer: a message is sent to a single agent
  - multicast: a message is sent to a selected group of agents
  - broadcast: a message is sent to all participating agents
- The manager communicates that a certain task with certain features is available, and asks for "proposals" that should come within a deadline by agent with the "right" features.

The Contract Net Protocol

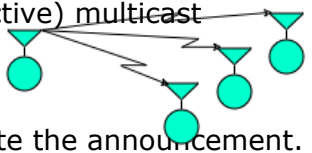
Some of the eligible agents will respond by sending a "bid" in time, i.e., a proposal to perform the task under certain conditions that could be, e.g.:

- a certain time is required
- some resources should be allocated
- a certain reward is expected

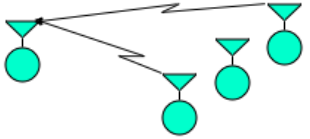
The Contract Net Protocol

An important generic protocol

A manager announces the existence of tasks via a (possibly selective) multicast



- Agents evaluate the announcement. Some of these agents submit bids



- The manager awards a contract to the most appropriate agent

- The manager and contractor communicate privately as

Task Announcement Message

- Eligibility specification: criteria that a node must meet to be eligible to submit a bid
- Task abstraction: a brief description of the task to be executed
- Bid specification: a description of the expected format of the bid
- Expiration time: a statement of the time interval during which the task announcement is valid

Bid and Award Messages

- A bid consists of a *node abstraction*—a brief specification of the agent's capabilities that are relevant to the task
- An award consists of a *task specification*—the complete specification of the task

Applicability of Contract Net

The Contract Net is

- a high-level communication protocol
- a way of distributing tasks
- a means of self-organization for a group of agents

Best used when

- the application has a well-defined hierarchy of tasks
- the problem has a coarse-grained decomposition
- the subtasks minimally interact with each other, but cooperate when they do

Using XML to describe ACL messages (continued)

- Syntactically, ACL messages can be thought as having an “abstract syntax”.
- The abstract syntax “allows” for multiple **syntactic representations** or **encodings**.

An example

```
(inform
 :sender   jklabrou
 :receiver finin
 :content
    (CPU libretto50
                pentium)
 :ontology laptop
 :language kif)
```

```
<!DOCTYPE fipa_acl SYSTEM "fipa_acl.dtd">
<message>
  <messagetype>
     inform
  </messagetype>
  <messageparameter>
     <sender link="http://www.cs.umbc.edu/~jklabrou">
     finin
     </sender>
  </messageparameter>
  <messageparameter>
     <receiver link="http://www.cs.umbc.edu/~finin/">
     finin
     </receiver>
  </messageparameter>
  <messageparameter>
     <ontology link="http://www.cs.umbc.edu/~jklabrou/ontology/laptop.html">
     laptop
     </ontology>
  </messageparameter>
  <messageparameter>
     (CPU libretto50 pentium)
     </content>
  </messageparameter>
  <messageparameter>
     <language link="http://www.stanford.edu/kif.html">
     kif
     </language>
  </messageparameter>
</message>
```

Comments on the XML-encoding of ACL messages

- The content itself of the ACL message could have been encoded in XML
- The XML-encoding enhances the canonical syntactic encoding:
  - it contains parsing information
  - parameter values are not strings but links
- The XML-encoding make ACL messages more “web-friendly”

Observation: in the Object-Oriented approach, how do distributed objects interact?

- Approaches to sharing objects in a distributed system have been evolving over the last years.
  - CORBA
  - Distributed Computing Environment (DCE) developed by the Open Group in the early 90's
  - Java
    - RMI
    - Enterprise Java Beans (EJB)
    - Jini
  - OLE/COM/DCOM/ActiveX (Microsoft)
  - SOAP
- All these can be seen as the "object-oriented alternative" to ACL's

What's Needed Tomorrow

- **Further develop semantics of ACLs**
  Common content languages and ontologies
- **Agent ontologies**
  Sharable ontologies for agent properties, behavior, etc
- **Better handle on metadata**
  Abstractable and applicable to many content languages

What's Needed Tomorrow

- **Declarative and learnable protocols**
  Languages for defining higher-level protocols based on more primitive ones
- **Practical agent knowledge sharing**
  "Social" mechanisms for distributing information and knowledge
- **Frameworks for controlling collections of agents**
  E.g., artificial markets, natural selection, etc.