- FIPA Mission -

“The promotion of technologies and *interoperability specifications* that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.”
- FIPA History -

• Started work in 1997
• At peak comprised 60 members
• Primary specifications became standard in 2002
• Work ongoing in Modeling, Methodology, Semantics & Services
- Within the Scope of FIPA -

Agent Lifecycle Management
Message Transport
Message Structure
Inter-agent Interaction Protocols
Ontologies
Security
Within the Scope of FIPA

Agent Platform 1

Agent Platform 2

Yellow Pages
White Pages
MTS

Yellow Pages
White Pages
MTS
- Outside the Scope of FIPA -

The Agent
Agent Characteristics -

- Autonomous
- Reactive
- Proactive
- Goal-driven
- Social
- Adaptive
- Cognitive
- The FIPA Agent Platform -

**Agent**

**Non-agent software**

Application

**AGENT**

ACL

**Agent Management System (AMS)**

**Directory Facilitator (DF)**

Message Transport Service

FIPA Agent Platform

- HTTP
- IIOP
- SMTP etc.

**ACL Message Structure Specification**

**Communicative Act Library**

**Interaction Protocol Specifications**

**SL Content Language Specification**

**Agent Message Transport Specification**
The Core Specifications (standard)
<table>
<thead>
<tr>
<th>Agent Service Description</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts 0..n</td>
<td></td>
</tr>
<tr>
<td>contains 0..n</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>contains 0..n</td>
<td></td>
</tr>
<tr>
<td>uses_a</td>
<td></td>
</tr>
<tr>
<td>contains 0..n</td>
<td></td>
</tr>
<tr>
<td>AMS</td>
<td></td>
</tr>
<tr>
<td>Agent Description</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>contains 0..n</td>
<td></td>
</tr>
<tr>
<td>Agent</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>has_a</td>
<td></td>
</tr>
<tr>
<td>contains 0..n</td>
<td></td>
</tr>
</tbody>
</table>
- Agent Management -

Common Operations
REGISTER
DEREGISTER
MODIFY
SEARCH

DF

AMS

Name
Location
Services
Protocols

Ontologies
Lease-time
Scope

Different
Agent
Descriptions

Name
Owner
State

10/05/04
(df-agent-description
 :name (agent-identifier
   :name dummy@foo.com
   :addresses (sequence iiop://foo.com/acc))
 :protocols fipa-request
 :ontologies (set fipa-agent-management)
 :languages (set fipa-sl0)
 :lease-time +000000000T600000000000T )
Agent message transport comprises two levels:

(1) The Message Transport Protocol (MTP) carries out the physical transfer of messages between two ACCs.

(2) The Message Transport Service (MTS) is provided by the AP to which an agent is attached. The MTS supports the transport of FIPA ACL messages between agents on any given AP and between agents on different APs.
- FIPA Message Structure -

1. **Envelopes**
   - Transport Information

2. **Payload**
   - Encoded Message

3. **Message**
   - Message Parameters

4. **Content**
   - Message Content
- FIPA Envelope Parameters -

**Mandatory**

- **to**
  - The intended receiver.
- **from**
  - The sender
- **acl-representation**
  - ACL presentation (e.g. String, XML, Bit-efficient)
- **date**
  - Creation date of the envelope

**Optional**

- **payload-length**
  - Byte length of the payload
- **payload-encoding**
  - ACL language encoding (e.g. US-ASCII, UTF-8)
- **received**
  - Stamp evidencing receipt of the message
- **security-object**
  - Encryption and certification information
FIPA Message Structure

- Envelope
  - Message
    - Content
      - Symbol
      - Interaction Protocol
    - ACL
    - CL
  - Ontology
  - Transport Protocol

- IDL
  - XML
  - bit-eff
- String
  - XML
  - bit-eff

- request, inform, query, contract-net, etc.

- Envelope Encoding Scheme
  - ACL Encoding Scheme
  - CL Encoding Scheme

- IsExpressedIn
- IsTransmittedOver
- BelongsTo
## FIPA ACL Message Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>performative</td>
<td>What action the message performs</td>
</tr>
<tr>
<td>sender</td>
<td>Initiator of the message</td>
</tr>
<tr>
<td>receiver</td>
<td>Recipient of the message</td>
</tr>
<tr>
<td>reply-to</td>
<td>Recipient of the message reply</td>
</tr>
<tr>
<td>content</td>
<td>Content of the message</td>
</tr>
<tr>
<td>language</td>
<td>Language used to express content</td>
</tr>
<tr>
<td>encoding</td>
<td>Encoding used for content</td>
</tr>
<tr>
<td>ontology</td>
<td>Ontology context for content</td>
</tr>
<tr>
<td>protocol</td>
<td>Protocol message belongs to</td>
</tr>
<tr>
<td>conversation-id</td>
<td>Conversation message belongs to</td>
</tr>
<tr>
<td>reply-with</td>
<td>Reply with this expression</td>
</tr>
<tr>
<td>in-reply-to</td>
<td>Action to which this is a reply</td>
</tr>
<tr>
<td>reply-by</td>
<td>Time to receive reply by</td>
</tr>
</tbody>
</table>
- ACL Message Example -

(request
  :sender (:name dominic-agent@whitestein.com:8080)
  :receiver (:name rex-hotel@tcp://hotelrex.com:6600)
  :ontology personal-travel-assistant
  :language FIPA-SL
  :protocol fipa-request
  :content
    (action movenpick-hotel@tcp://movenpick.com:6600
    ))
)
- Communicative Acts -

- Drawn from Speech Act theory
- A speaker "utters" speech acts, which are also known as performatives or communicative acts
- ACL messages are modeled after speech acts
- Speech acts may be understood in terms of an intentional level description of an agent
- An intentional description makes references to beliefs, desires, intentions & other modalities
Communicative Act Library (1)

**accept-proposal**: accept a previously submitted proposal

**agree**: agree to perform some action, possibly in the future

**cancel**: cancel some previously requested action

**cfp**: make a call for proposals to perform a given action

**confirm**: inform a receiver that a given proposition is true

**disconfirm**: inform a receiver that a given proposition is false

**failure**: inform another agent that an action was attempted but failed

**inform**: inform a receiver that a given proposition is true

**not-understood**: informs a receiver that sender did not understand

**query-if**: ask another agent whether a given proposition is true

**request**: requests a receiver to perform some action
propose: submit a proposal to perform a certain action
query-ref: ask another agent for the object referred to by a referential expression
refuse: refuse to perform a given action
reject-proposal: reject a proposal during a negotiation
request-when: request a receiver to perform some action when some given proposition becomes true
request-whenever: request a receiver to perform some action as soon as some proposition is true and thereafter each time the proposition becomes true again
subscribe: a persistent intention to notify the sender of a value, and to notify again whenever the value changes
Communicative Act Library (3)

**propagate**: the receiver treats the embedded message as sent directly to it, and must identify the agents denoted by the given descriptor and send the received propagate message to them.

**proxy**: the receiver must select target agents denoted by a given description and to send an embedded message to them.

**subscribe**: a persistent intention to notify the sender of a value, and to notify again whenever the value changes.
Any language can be used as a Content Language, e.g.:

- KIF
- Prolog
- SQL
- Serialized Objects
- Binary Large Objects
- FIPA-SL, FIPA-CCL, FIPA-RDF, FIPA-KIF
FIPA SL content expression has 3 types:

1) Proposition
   - A Wff (well-formed formulae) that can be assigned a truth value in a specific context, e.g., `confirm`.
   - Agent i confirms to agent j that it is, in fact, true that a platypus is a mammal.

```
(confirm
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :content ((is mammal platypus))
  :language fipa-sl )
```
FIPA SL content expression has 3 types:

(2) Action

- Something to be performed, e.g. request.
- Agent i requests agent j (robot) to deliver a box.

(request
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :content ((action (agent-identifier :name j) (deliver box017 (loc 12)))
  :protocol fipa-request
  :language fipa-sl
  :reply-with order567 )
FIPA SL content expression has 3 types:

(3) IRE (Identifying Reference Expression)
- References an object in the domain, e.g. `inform-ref`.
- Agent i requests agent j to tell it the 'capital of Wales'.

(request
 :sender (agent-identifier :name i)
 :receiver (set(agent-identifier :name j))
 :content
 ((action (agent-identifier :name j)
   (inform-ref
    :sender (agent-identifier :name j)
    :receiver (set (agent-identifier :name i))
    :content ((iota ?x (CapitalWales ?x)))))
 :ontology world-politics
 :language fipa-sl ) ))
 :reply-with query0
 :language fipa-sl )

A common vocabulary of agreed upon definitions and relationships between those definitions, to describe a particular subject domain.

E.g.:
Agent-management ontology
UMTS wireless technology ontology
Cinema ontology
Weather ontology
IEEE Standard Upper Ontology
In the previous example the Ontology was *world-politics*. This ontology refers to *ALL* terms in the content expression.

Future work would allow definitions from different ontologies to be associated with different terms in a content expression.

E.g.:  
:ontology (set (world-politics, geography))  
:content ((iota ?x (world-politics:CapitalWales ?x)),  
            iota ?y (geography:LandAreaWales ?y)))
- Interaction Protocols -

- Patterns of message exchange between agents
- Concurrent IPs are called conversations
- Based on communicative acts
- There is a basic set of pre-defined standard IPs
- Ad hoc IPs can be defined
- Communication semantics can be defined at IP level rather than individuals CAs
- Interaction Protocols -

**FIPA defined IPs are:**

- FIPA-Request
- FIPA-Query
- FIPA-Request-When
- FIPA-Contract-Net
- FIPA-Iterated-Contract-Net
- FIPA-Auction-English
- FIPA-Auction-Dutch

- FIPA-Brokering
- FIPA-Recruiting
- FIPA-Subscribe
- FIPA-Propose
- The FIPA-Request Protocol -

Initiator

REQUEST

REFUSE

AGREE

FAILURE

INFORM-DONE

INFORM-RESULT

Participant
The FIPA-Contract-Net IP -

- cfp(action p1)
  - not-understood
  - refuse (reason)
  - propose (p2)
    - refuse (reason)
    - propose (p2)
      - accept proposal
      - inform (done)
      - cancel
    - reject proposal
      - refuse (reason)
      - inform (done)
      - cancel
- Other FIPA Specifications -

- The Abstract Architecture
- Quality of Service
- CCL/KIF/RDF Content Languages
- Agent MTP for WAP
- Network Management and Provisioning
- Ontology service
- Message Buffering Service
- Domains and Policies
- JXTA Discovery Middleware
- Ongoing Work -

- Agent Modelling
- Agent Methodology
- Semantic Framework
- Service Framework
- Security
- Resources -

• FIPA
  • http://www.fipa.org

• Agentcities
  • http://www.agentcities.org/
  • http://www.agentcities.net/

• Agent in General
  • http://www.agentlink.org/