Specifying agent interactions using UML:

What does this mean for current and future FIPA standards?

James J. Odell

www.jamesodell.com
## AN AGENT CONVERSATION

<table>
<thead>
<tr>
<th>Seq</th>
<th>Snr</th>
<th>Rcvr</th>
<th>Utterance</th>
<th>Rspnds</th>
<th>Replies</th>
<th>Resolves</th>
<th>Completes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B,C,D</td>
<td>REQUEST: Please send me 50 widgets at your catalog price by next Thursday.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>C</td>
<td>QUESTION: Are you bidding on A's RFQ?</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>B</td>
<td>INFORM: Yes, I am.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>A</td>
<td>REFUSE</td>
<td>3</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>A</td>
<td>PROPOSE (INFORM + REQUEST): How about 40 widgets at catalog price by next Friday?</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>C</td>
<td>REQUEST: Please send me 40 widgets at catalog price by next Friday.</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>A</td>
<td>COMMIT: I plan to send you 40 widgets at catalog price by next Friday.</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>D</td>
<td>A</td>
<td>COMMIT: I plan to send you 50 widgets at catalog price by next Thursday.</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>C</td>
<td>ASSERT: I've found a better supplier, and am not relying on your COMMIT.</td>
<td>7,8</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>C</td>
<td>A</td>
<td>REFUSE: I'm abandoning my COMMIT.</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>D</td>
<td>A</td>
<td>SHIP: Here are your widgets. Please pay me.</td>
<td>1</td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>A</td>
<td>D</td>
<td>ASSERT + REQUEST: You’re five short. Please send the difference.</td>
<td>11</td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>A</td>
<td>SHIP: Here are five more widgets. Please pay me.</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>A</td>
<td>D</td>
<td>PAY</td>
<td>13</td>
<td></td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>
WINOGRAD-FLORES VERSION OF AN AGENT INTERACTION PROTOCOL (AIP)

After Smith and Cohen 1995
1997 FIPA-STYLE
AGENT INTERACTION PROTOCOL (AIP)

FIPA 97 Specification, Version 2.0, Part 2, Agent Communication Language

Specifying agent interactions using UML
FIPA (Federated Intelligent Physical Agents) has been working to develop and promote standardization in the area of agent interoperability since 1996. It has an ongoing work program, meeting around the globe on a quarterly basis, with excess of 50 member organizations. (www.fipa.org)

OMG Agents Working Group recommends standards for agent technology where appropriate—particularly the OMG's Object Management Architecture (OMA). (www.omg.org)
MOF/ADF/BOF ARCHITECTURE

Common Business Objects

Business Object Facility

UML Facility

Meta-Object Facility

CORBA

MetaModel Specialization

Specific

M1

M2

M3

Generic

Mfg

Fin

Med
UML DIAGRAMS

- Dynamic Diagrams
  - Sequence diagrams
  - Collaboration diagrams
  - Activity diagrams
  - Statechart
  - Deployment diagrams
  - Use Case diagrams

- Static Diagrams
  - Class diagrams
  - Packages
  - Component diagrams

So, which will be useful for representing agent interaction protocols?
LEVEL 1

Generic AIP expressed as a template package

FIPA Contract Net Protocol

Initiator

- call-for-proposal
- refuse
- not-understood
- propose
- accept-proposal
- reject-proposal
- inform
- cancel

Participant

Initiator, Participant

Deadline

- call-for-proposal, refuse*, not-understood*, propose, reject-proposal*, accept-proposal*, cancel*, inform*
<table>
<thead>
<tr>
<th>FIPA Contract Net Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buyer, Seller</strong></td>
</tr>
<tr>
<td><strong>8/8/99 at 12:00</strong></td>
</tr>
<tr>
<td>seller-rfp, refuse-1, refuse-2, not-understood, propose, reject-proposal, accept-proposal, cancel, inform</td>
</tr>
</tbody>
</table>
APPLYING A TEMPLATE TO
A PARTICULAR SCENARIO

seller-rfp

refuse-1

refuse-2

not-understood

propose

accept-proposal

reject-proposal

inform

cancel

deadline: 8/8/99 at 2:00 hours

Buyer

Seller
USING PACKAGES TO EXPRESS
“NESTED” PROTOCOL

Purchasing

Broker

Supplying

Retailer

Wholesaler

call-for-proposal

propose

...
LEVEL 2:

Basic format of Sequence Diagrams for agent communication

Agent-1/Role:Class

Agent-2/Role:Class

CA-1

CA-2

And, what is the difference between a role and a class?
Level 2:

And don’t forget that **objects** may play a part in agent communication.

**DEFINITION—reconsidered?**

An agent interaction protocol (AIP) describes a communication pattern as a permitted sequence of messages *among agents* and the constraints on the content of those messages.
CONCURRENT COMMUNICATION

[Diagram showing concurrent communication between agents with various roles and interactions.]
SEQUENCE DIAGRAM

Example with agent instances

A / Customer
A / Negotiator
B / Contractor
B / Competitor Analyzer
C / Contractor1
C / Contractor2
C / Competitor
D / Contractor
D / Debtor

request
refuse
refuse
propose
request
commit
commit
assert
refuse
ship
assert + request
ship
pay

text:

<<role change>>

question
inform

© 2000, James Odell
SEQUENCE DIAGRAM

Alternative with roles as time intervals

(only A’s roles labeled)
SEQUENCE DIAGRAM

Alternative with roles as parallel life-line branches

A: request
   refuse
   request
   commit
   commit
   assert
   refuse
   ship
   assert + request
   ship
   pay

B: propose
   request
   commit

C: question
   inform

D: 

<<role: customer>>  <<role: negotiator>>  <<role: contractor>>  <<role: competitor analyzer>>  <<role: contractor>>  <<role: contractor>>  <<role: competitor>>  <<role: contractor>>  <<role: debtor>>
ROLES AS TYPES

Without individual agents

Customer  Negotiator  Contractor  Competitor Analyzer  Contractor*  Debtor

request  refuse  question  inform

propose  request  commit

commit

assert  refuse

ship

assert + request

ship

pay
AND, CAN FIPA ACL HANDLE

1:M and M:M COMMUNICATIONS?

In FIPA ACL, the sender is often required to believe something about the receivers:
e.g., “If an agent performs an inform act with … three agents as receiver, it denotes that the sender intends each of these agent to come to believe the content of the message.”

Also, “the sending agent intends that the receiving agent also comes to believe that the proposition is true” and “does not already believe that the receiver has any knowledge of the truth of the proposition.”

However, in a broadcast, you cannot know who the hearer is:

- ”Beware of dog.”
- “Bob, take attendance.”
- “One of you, get a slice of pie.”
- “John & Betty request the pleasure of your company.”

We need more precise group semantics.
(Phil Cohen, Oregon Graduate Institute)
ROLES ON COMMUNICATION ACTS
IN A COLLABORATION DIAGRAM

(only roles for A shown)

1.1: request <<role: customer>>
1.3: request <<role: customer>>
12: assert + request <<role: customer>>
14: pay <<role: customer>>

2: question
3: inform
4: refuse
5: propose
7: commit
10: refuse
8: commit
11: ship
13: ship
1.2: request <<role: customer>>
6: request <<role: negotiator>>
9: assert <<role: negotiator>>

© 2000, James Odell

Specifying agent interactions using UML
ACTIVITY DIAGRAMS

Another alternative representation

Place Order → Process Order
Accept Order → Accept Quote
Create Quote

Match Order and Quote

Close Order → Update Quote
Settle Order
ORDER PROCESSING:
Activity diagrams with swimlanes

Customer
- Place Order
  - Process Order
  - Accept Quote
  - Match Order and Quote
  - Close Order
  - Settle Order

Broker
- Process Order
  - Accept Order
  - Accept Quote
  - Match Order and Quote
  - Update Quote

ECN
- Accept Order
  - Accept Quote
  - Match Order and Quote

Market Maker
- Create Quote
  - Update Quote
ORDER PROCESSING:

Activity diagrams with swimlanes and object flow

Customer
- Place Order
  - Order [requested]
  - Order [filled]
  - Order [closed]
  - Settle Order

Broker
- Process Order
  - Order [unfilled]
  - Close Order

ECN
- Accept Order
  - Order [accepted]
- Accept Quote
  - Quote [accepted]
  - Quote [unfilled]

Market Maker
- Create Quote
- Quote [unfilled]
- Update Quote
  - Quote [updated]
STATECHART NOTATION

Yet another alternative

A state-based speech-act version of the Winograd-Flores protocol in UML statechart notation
INTERACTION PROTOCOLS CAN BE SPECIFIED IN MORE DETAIL

Role-1

Role-2

CA-1

CA-2

CA-3

CA-4

x

y

g

z

...
**LEVEL 3**

Generic AIP expressed as a template package

- Customer
- Order Processor
- Invoice Sender
- Payment Receiver

Order placed → Accepted Order

Order accepted → Assembled Order

Assembled Order → Ship Order

Ship Order → Close Order

Order shipped → Process Payment

Payment accepted → Order completed

Order accepted → Payment accepted

Order accepted → Invoice request

Received payment → Completed

Invoice request → Completed

Place order → Completed
REPRESENTING AGENT INTERACTION PROTOCOLS IN UML

- Level 1: Representing the overall protocol
- Level 2: Representing interaction among agents
- Level 3: Representing internal agent processing
OTHER UML EXTENSION

Package specifying agents instead of operations as interface points.

- Common Function Agents
  - Process Planning Agent
  - Capacity Manager Agent
  - Dispatch Agent
  - Account Manager Agent

- Resource Agent
  - 1..* Account Manager Agent
  - 0..* Dispatch Agent

- Resource
  - 1..* Resource Agent

- Manufacturing Cell
  - aggregate
  - Negotiator Agent

- Interface agent

- Telescoping agent packages
OTHER UML EXTENSION

Adding mobility to deployment diagrams.
OTHER UML EXTENSION

Representing cloning, mitosis, and reproduction using sequence and activity diagrams

**Sequence Diagram**

**Dr_Frankenstein:**

**Dolly:** Sheep

get specs

<<clone>>

**Dolly2:** Sheep

**Class Diagram**

**Dr_Frankenstein:**

**Dolly:** Sheep

<<prototype>>

<<clone>>

**Dolly2:** Sheep

**Sequence Diagram**

**A:** Amoeba

<<mitosis>>

**A1:** Amoeba

**A2:** Amoeba

**Mitosis**

**Collaboration Diagram**

**John:** Starfish

**Janet:** Starfish

<<reproduction>>

**Junior:** Starfish

**Bisexual**
Representing parasitic, symbiotic, and emergence relationship using class diagrams

**OTHER UML EXTENSION**

- **Host/Parasite**: Dog → Flea
  - Dog: 0..*
  - Flea: 0..*

- **Symbiosis**: Tree → Epiphyte
  - Tree: 1
  - Epiphyte: 0..*

- **Emergence**: Market → Consumer
  - Market: 0..*
  - Consumer: 0..*
Specifying agent interactions using UML:

What does this mean for current and future FIPA standards?

James J. Odell

www.jamesodell.com