Representing Agent Interaction Protocols in UML

James J. Odell
H. Van Dyke Parunak
Bernhard Bauer

Agent UML
A Formalism for Specifying Multiagent Software Systems

Bernhard Bauer
Jörg P. Müller
James Odell

Intelligent Autonomous Agents
© Siemens AG
AN AGENT CONVERSATION

<table>
<thead>
<tr>
<th>Seq</th>
<th>Snr</th>
<th>Rcvr</th>
<th>Utterance</th>
<th>Rspnds to</th>
<th>Replies to</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>1</td>
<td>1</td>
<td></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>1</td>
<td>1</td>
<td></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</td>
<td>8</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>1</td>
<td>3</td>
<td></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>11</td>
<td></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>13</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

WINOGRAD-FLORES VERSION OF AN AGENT INTERACTION PROTOCOL (AIP)

After Smith and Cohen 1995
STANDARDIZATION

Analysis and Design Task Force (ADTF) of Object Management Group (OMG)

*Overall goal: vendor-neutral common semantics, meta-model, and abstract syntax for A&D methodologies*

**Mission**

- To enable developers to better understand how to develop applications using OT—thereby growing the market.
- To recommend technology that interoperates across the lifecycle of A&D tools designs/work products.
- To recommend technology that enables reuse of designs and work products developed using A&D tools.
- To recommend technology that has common semantics, meta-model, and abstract syntax for A&D methodologies.
- To leverage existing OMG specifications.

OA&D TF RFP-1 - approved and released June 1996
Submissions due: 17 Jan 1997; Revisions due: 1 September 1997
Selection: 25 September 1997; Adoption: November 1997

Object Management Group
www.omg.org
(508) 820-4300

MOF/OADF/BOF ARCHITECTURE

- **MetaModel Specialization**
  - **Low**
  - **Specific**
  - **Generic**
  - **High**

- **Meta-Object Facility**
- **Common Business Objects**
- **Business Object Facility**
- **OA&D (UML) Facility**
- **Meta-Object Facility**

- **CORBA 3.0**
**UML Diagrams**

- Static Diagrams
  - Class diagrams
  - Packages
  - Component diagrams

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

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

---

**Agents Standardization and Agent UML (AUML)**

- **OMG Agents Working Group** recommends standards for agent technology where appropriate—particularly the OMG's Object Management Architecture (OMA). (www.omg.org)

- **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)
**A COMMON AIP REPRESENTATION**

**UML Sequence Diagrams**

- Initiator
- Participant

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

**FIPA Contract Net Protocol**

---

**INTERACTION PROTOCOLS CAN BE SPECIFIED IN MORE DETAIL**

**Role-1**

- CA-1
- CA-2

**Role-2**

- CA-3
- CA-4

**Role-3**

...
A LAYERED APPROACH TO AIPs

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

USING PACKAGES TO EXPRESS “NESTED” PROTOCOL

- Purchasing
- Supplying
  - Broker
  - Retailer
  - Wholesaler
- call-for-proposal
- request
- propose
- inform
LEVEL 1
Generic AIP expressed as a template package

BINDING THE PARAMETERS OF THE FIPA CONTRACT NET PROTOCOL PACKAGE TEMPLATE

FIPA Contract Net Protocol

Buyer, Seller
8/8/99 at 12:00
seller-rfp, refuse-1, refuse-2, not-understood, propose, reject-proposal, accept-proposal, cancel, inform

Initiator, Participant
Deadline
call-for-proposal, refuse*, not-understood*, propose, reject-proposal*, accept-proposal*, cancel*, inform*

FIPA Contract Net Protocol

contract initiation

Initiator

Participant

call-for-proposal

refuse

not-understood

propose

accept-proposal

reject-proposal

inform

cancel

APPLYING A TEMPLATE TO A PARTICULAR SCENARIO

LEVEL 2:
Basic format of Sequence Diagrams for agent communication

Buyer

Seller

seller-rf

refuse-1

refuse-2

not-understood

propose

accept-proposal

reject-proposal

decline
inform

cancel

deadline: 8/8/99 at 12:00 hours

Agent-1/Role:Class

Agent-2/Role:Class

CA-1

CA-2
CONCURRENT COMMUNICATION
WITH AN AGENT PLAYING
DIFFERENT ROLES

SEQUENCE DIAGRAM

Example
DETECTING ROLES AS TIME INTERVALS
IN A SEQUENCE DIAGRAM

(only A’s roles labeled)

A:

B:

C:

D:

<<role: customer>>

refuse

<<role: negotiator>>

propose

<<role: customer>>

commit

<<role: negotiator>>

refuse

<<role: customer>>

ship

<<role: customer>>

assert + request


ROLES AS PARALLEL BRANCHES
IN A SEQUENCE DIAGRAM

A:

B:

C:

D:

request

question

inform

refuse

propose

commit

assert

ship

assert + request

ship

pay

refuse

propose

commit

commit

assert

refuse

ship

assert + request

ship

pay

question

inform

refuse

propose

commit

commit

assert

refuse

ship

assert + request

ship

pay

refuse

propose

commit

commit

assert

refuse

ship

assert + request

ship

pay
**ROLES AS TYPES**

**WITHOUT INDIVIDUAL AGENTS**

- Customer
- Negotiator
- Contractor
- Competitor Analyzer
- Contractor
- Debtor

**COLLABORATION DIAGRAM**

An alternative representation
ROLES ON MESSAGES
IN A COLLABORATION DIAGRAM

(only roles for A shown)

1.1: request  
<<role: customer>>
1.2: request  
<<role: customer>>
1.3: request  
<<role: customer>>
1.4: refuse  
<<role: customer>>
1.6: propose  
<<role: negotiator>>
1.7: commit  
<<role: negotiator>>
1.8: commit  
<<role: negotiator>>
1.9: assert  
<<role: negotiator>>
1.10: refuse  
<<role: negotiator>>
1.11: ship  
<<role: customer>>
1.12: ship  
<<role: customer>>
1.13: ship

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

ORDER PROCESSING:
ACTIVITY DIAGRAMS WITH SWIMLANES AND OBJECT FLOW
LEVEL 3

Generic AIP expressed as a template package

STATECHART NOTATION

Even another alternative

A state-based speech-act version of the Winograd-Flores protocol in UML statechart notation
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.
OTHER UML EXTENSION

Adding mobility to deployment diagrams.

OTHER UML EXTENSION

Representing cloning, mitosis, and reproduction using sequence and activity diagrams.
**OTHER UML EXTENSION**

Representing parasitic, symbiotic, and emergence relationship using class diagrams

```
Dog               Tree
0..*              1
<<host/parasite>> <<symbiosis>>
Flea              0..*

Epiphyte
```

```
Market           Consumer
0..*             0..*
<<emergence>>
```

---

Representing
Agent Interaction Protocols
(and then some)
in UML

Agent UML
A Formalism for Specifying
Multiagent Software Systems

James J. Odell  Bernhard Bauer
H. Van Dyke Parunak  Jörg P. Müller
Bernhard Bauer  James Odell

For more information, contact
jodell@compuserve.com
www.JamesOdell.com