### 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></td>
<td>7</td>
<td>8</td>
<td>7</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>8</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>
What is AUML — and Why Do I Care?

© James Odell, 2003

Page 3

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
OMG Agent Special Interest Group recommends standards for agent technology where appropriate. (www.omg.org)

FIPA* Modeling Technical Committee - now tasked with developing an AUML standard (www.auml.org)

Others - MESSAGE, PASSI, Tropos, Prometheus, MaSE, …

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

THE FIPA AUML APPROACH

Instead of reliance on the OMG's UML, we intend to reuse of UML wherever it makes sense.

AUML is not restricted to using UML. Other approaches should be used wherever it makes sense.

The general philosophy, then, is:

* When it makes sense to reuse/extend portions of UML, then do it.

* When it doesn't make sense to use UML, use something else or create something new.
Software technology and agents

Software history is one of increasing localization and encapsulation.

<table>
<thead>
<tr>
<th>Monolithic Programming</th>
<th>Modular Programming</th>
<th>Object-Oriented Programming</th>
<th>Agent Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Behavior</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonmodular</td>
<td>Modular</td>
<td>Modular</td>
<td>Modular</td>
</tr>
<tr>
<td>Unit State</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>External</td>
<td>Internal</td>
<td>Internal</td>
</tr>
<tr>
<td>Unit Invocation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>External (CALLed)</td>
<td>External (message)</td>
<td>Internal (rules, goals)</td>
</tr>
</tbody>
</table>

Inspired by: Parunak, H. Van Dyke, Autonomous Agent Architectures: A Non-technical Introduction, Industrial Technology Institute, 10/13/95.

What is AUML — and Why Do I Care?

© James Odell, 2003

OO and agents

UML definition of active object:

- "An object that may execute its own behavior without requiring method invocation."
- "The points at which an active object responds to communications from other objects are determined solely by the behavior of the active object and not by the invoking object."
- "This implies that an active object is both autonomous and interactive to some degree."

– UML 2.0 Superstructure document

OO and the agent-based approach seem to be converging.
**UML 2.0 DIAGRAMS**

- **Structure Diagrams**
  - Class diagrams
  - Composite structure diagrams
  - Component diagrams
  - Packages
  - Deployment diagrams

- **Behavior Diagrams**
  - Sequence diagrams
  - Communication diagrams
  - Activity diagrams
  - State Machine
  - Use Case diagrams

---

**FIRST: A MAJOR DIFFERENCE BETWEEN OBJECT AND AGENT STRUCTURE**

*With agents, nature a source of analogies.*

- symbiosis
- parasitism
- mimicry
- niche formation
- speciation
- sexual and asexual reproduction
- eukaryotic versus prokaryotic cells
- Darwinian evolution
- Lamarckian evolution
- neuron-based systems
- ...

*These analogies mark some of the differences with conventional objects.*
What is AUML — and Why Do I Care?

© James Odell, 2003

Start with a conceptual-level class diagram and identify which objects and links are candidates for agent-hood.

An example with one possible solution.

CLASS DIAGRAM
Agents and/or objects for order processing

COMPOSITE STRUCTURE DIAGRAMS
A composition of interconnected elements that can collaborate.

This:

Can be represented as:
What is AUML — and Why Do I Care?

© James Odell, 2003  

**COMPOSITE STRUCTURE DIAGRAMS**

**Collaborating parts example**

```
+----------------+      +--------------+      +----------------+      +----------------+
| Car            |      | Wheel        |      | Engine         |      | Car            |
| 0..1           |      | 0..1         |      | 2..*           |      | 2              |
| rear wheel     |      | car engine   |      |                |      |                |
| 2              |      | 0..1         |      | 0..1           |      | 1              |
| Wheel          |      | Engine       |      |                |      |                |
```

With class diagram only:
- Graphically less intuitive
- No context. Can imply:
  - "My engine on your wheels."
  - "Requires boats to have wheels"

**Innovations:**
- Composite structures can be instantiated in their own right.
- Agents can be either on or within "cell membranes"
- Replace interface *methods* with interface *agents* (and their responsibilities)
What is AUML — and Why Do I Care?

© James Odell, 2003

Provided and required interfaces

Specifies a modular unit with well-defined interfaces that is replaceable within its environment.

Component "wiring" symbols provide a more intuitive representation in UML 2.0.

Component internals

White-box view depicts internal assembly and physical containment.

However, this does not fully provide the intuition needed for a composite structure for agents. The component Order and the class Order are not equivalent. For example, what does it mean to "create an instance of Order"?
Packages can specify agent aggregate hierarchies.

- With UML 2.0, composite structure diagrams are generally better for aggregation.
- Packages could be used to express agent environments and subsystems.

**DEPLOYMENT DIAGRAMS**

```
<device>
  :AppServer

<container>
  :J2EEServer
  Order.jar
  ShoppingCart.jar
  Account.jar
  Product.jar
  BackOrder.jar
  Service.jar

<device>
  :DBServer
  OrderSchema.dll
  ItemSchema.dll
```
What is AUML — and Why Do I Care?

© James Odell, 2003

Page 19

DEPLOYMENT DIAGRAMS

For agents

From PASSI

UML DIAGRAMS

Structure Diagrams

- Class diagrams
- Composite structure diagrams
- Component diagrams
- Packages
- Deployment diagrams

Behavior Diagrams

- Sequence diagrams
- Communication diagrams
- Activity diagrams
- State Machine
- Use Case diagrams

© James Odell, 2003

What is AUML — and Why Do I Care?

Page 20
What is AUML — and Why Do I Care?

FIRST: A MAJOR DIFFERENCE BETWEEN OBJECT AND AGENT BEHAVIOR

Using nature a source of analogies

- Interaction
- Collaboration
- Cooperation
- Competition
- ...

“These analogies mark some of the differences with conventional objects.”

“Digital Dealmakers Meet in the Middle”
New York Times, 10 September, 2003

SEQUENCE DIAGRAMS

Two standards support them

Z.120
(11/1999)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS
Formal description techniques (FDT) — Message Sequence Chart

Message Sequence Chart (MSC)

UML 2.0 Superstructure Specification
OMG Adopted Specification
ptc/03-08-02

© James Odell, 2003
**What is AUML — and Why Do I Care?**

© James Odell, 2003

---

**SIMPLE SEQUENCE DIAGRAM**

- Messages have one send event, and one receive event.
  - The send event must occur before the receive event.
  - The send event is the result of an Action
- Events are strictly ordered along a lifeline from top to bottom

---

**DECOMPOSING A LIFELINE**

- We want to look into this lifeline.
- Here is the decomposition.
REFERENCES AND GATES

INTERACTION OVERVIEW
INTERACTION DIAGRAMS IN EARLY AUML

Minor extensions to UML 1.x

INTERACTION DIAGRAMS IN CURRENT AUML

Early AUML extensions have influenced UML 2.0
What is AUML — and Why Do I Care?

© James Odell, 2003

INTERACTION DIAGRAMS IN UML 1.X
Using sequence diagram approach

INTERACTION DIAGRAMS IN UML 2.0
Using interaction overview approach
What is AUML — and Why Do I Care?

© James Odell, 2003

Page 31

Objects may play a part in agent communication

Agent-1:Role  Object:Role  Agent-2:Role

inform msg  request msg  response CA

Dynamic and multiple classification

and role "changes"

Dynamic classification refers to the ability to change classification over time.

Multiple classification refers to the ability to be an instance of more than one classification at a given moment.

instances of Employee

Alice  Jasper  Susan  Paul

Alice  Manager

Salesperson

Unemployed Person
**ROLE “CHANGES”**

Six scenarios are considered to be “changes,” yet semantically they all have a different meaning.

- **Classify** – Add the role of manager to the role of Employee as the result of a promotion.
- **Declassify** - Remove the role of Manager as the result of a demotion.
- **Reclassify** - Change from the role of Employee to the role of Unemployed Person.
- **Activate** - Take up the behaviors of the Manager role as part of the day-to-day business activity.
- **Suspend** – Stop any Manager behavior and take on just those of the Employee role as part of the day-to-day business activity.
- **Shift** - Change from an Employee role to a Pet Owner role, where neither is played at the same time as the other. This is a combination of activating one role while suspending another.

*composite operations

---

**AUML ROLE-CHANGE NOTATION**

Classification, declassification, and reclassification

(a) Classification  
(b) Declassification  
(c) Reclassification
AUML ROLE-CHANGE NOTATION

UML and AUML notation for expressing activate, suspend, and shift

(a) Activate

(b) Suspend

(c) Shift (asynchronous)

(d) Shift (synchronous)

Or, use the stereotypes: «activates», «suspends», and «shifts».

AUML METAMODEL EXCERPT
What is AUML — and Why Do I Care?

COMMUNICATION DIAGRAM

An alternative representation

ACTIVITY DIAGRAMS

Another alternative representation
ORDER PROCESSING:
Activity diagrams with role-based partitions

ORDER PROCESSING:
Activity diagrams with partitions and object flow
**STATECHART NOTATION**

Yet another alternative

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

---

**UML 2.0 DIAGRAMS**

- Structure Diagrams
  - Class diagrams
  - Composite structure diagrams
  - Component diagrams
  - Packages
  - Deployment diagrams

- Behavior Diagrams
  - Sequence diagrams
  - Communication diagrams
  - Activity diagrams
  - State Machine
  - Use Case diagrams
SOME DEFICIENCIES
OF UML DIAGRAMS FOR AGENTS

UML has no “off-the-shelf” constructs to express:

- goals
- agent
- group
- multicasting
- generative functions, such as cloning, birthing, reproduction
- parasitism and symbiosis
- emergent phenomena
- ...

TROPOS

Symbols

Actor diagram
PROMETHEUS

Symbols

System overview diagram

MaSE

Role model diagram
What is AUML — and Why Do I Care?

© James Odell, 2003

MESSAGE
Symbols

Agent
Role
Organization
Goal
Task
Service
Interaction
Resource

MESSAGE
Diagram examples

Structural relationship diagram
Acquaintance diagrams
Goal implication diagram
Agent diagram
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, Oregan Graduate Institute)

---

**GENERATIVE FUNCTIONS:**

Cloning

Dr. Frankenstein: Scientist

<table>
<thead>
<tr>
<th>Dolly: Sheep</th>
</tr>
</thead>
<tbody>
<tr>
<td>get specs</td>
</tr>
<tr>
<td>&lt;&lt;clone&gt;&gt;</td>
</tr>
</tbody>
</table>

Dolly2: Sheep

Sequence Diagram

Class Diagram

Dolly: Sheep

Dolly2: Sheep

<<prototype>>

<<clone>>
**What is AUML — and Why Do I Care?**

© James Odell, 2003

---

**GENERATIVE FUNCTIONS:**

Mitosis and bisexual

- **Mitosis**
  - A: Amoeba
  - A1: Amoeba
  - A2: Amoeba

- **Bisexual**
  - John: Starfish
  - Janet: Starfish
  - Junior: Starfish

---

**Innovations:**

- stereotypes
- synchronization bars (native to Activity Diagrams) on Sequence and Collaboration Diagrams

---

**EXTENDING CLASS DIAGRAMS TO EXPRESS PARASITISM AND SYMBIOSIS**

- **Dog**
  - 0..*
  - **Flea**
    - 0..*

- **Tree**
  - 1
  - **Epiphyte**
    - 0..*
AUML APPROACH

- Instead of reliance on the OMG's UML, we intend to reuse of UML wherever it makes sense.
  - We need to capitalize on the current knowledge and momentum of the IT industry is a major consideration.
- AUML is not restricted to using UML. Other approaches should be used wherever it makes sense.
  - We need to work with other agent-modeling efforts now — before much more work has been done.

http://www.auml.org

What is It — and Why Do I Care?

James Odell
James Odell Associates
3646 W. Huron River Drive
Ann Arbor, Michigan 48103-9489 USA
telephone: +1 (734) 994-0833
e-mail: email@jamesodell.com
website: www.jamesodell.com

Chair, OMG A&D “UML” Task Force
Chair, FIPA Agent Modeling Technical Committee
OMG-Certified UML Advanced Professional