UML Extensions for Agents

Jaelson Castro

April 2000

Overview

- UML
- Agent UML (AUML)
  - ✓ Agent Interaction Protocols
  - ✓ Richer Role Specification
  - ✓ Package Extension
  - ✓ Deployment Diagram Extension
- Conclusion

UML

- Static models: class, package diagrams
- Dynamic models:
  - ✓ Interaction diagrams (sequence and collaboration)
  - ✓ State diagrams
  - ✓ Activity diagrams
- Implementation models: component, deployment diagrams
- Object constraint language (OCL)

AUML: Agent UML

- Both FIPA and OMG are exploring and recommending extensions to UML
- Applications:
  - ✓ Specification of Agent Interaction Protocols (AIP)
  - ✓ Richer role specification
  - ✓ Package extension
  - ✓ Deployment diagram extension

Agent Interaction Protocols

- AIP describes a communication pattern as an allowed sequence of messages between agents and the constraints of the content of those messages.
- FIPA has specified many protocols
  - ✓ Request Protocol, Query Protocol, Request-when Protocol, Contract-net Protocol, Iterated-Contract-Net Protocol, Auction-English Protocol, Auction-Dutch Protocol

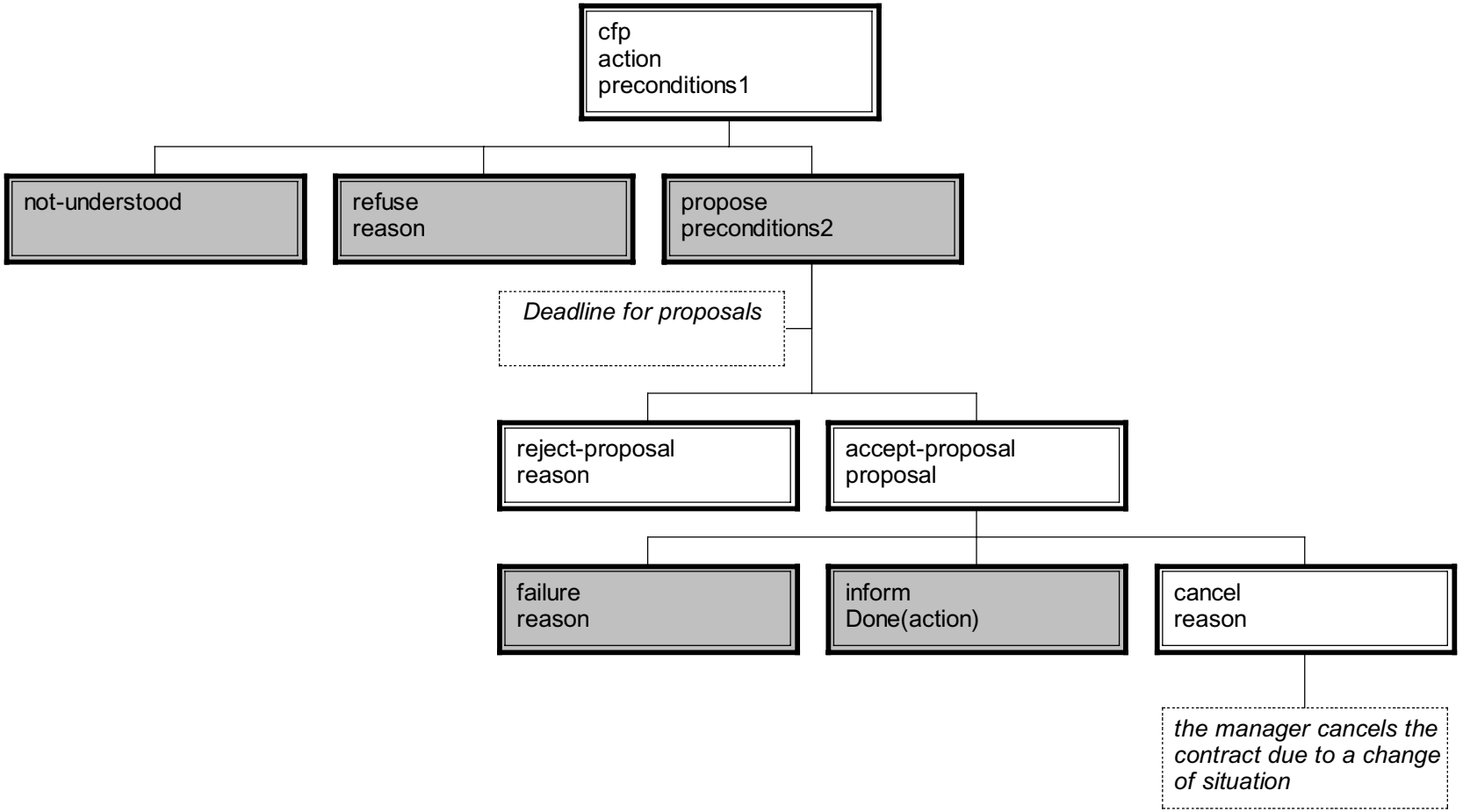
FIPA Contract-net Protocol

- The manager solicits ***proposals*** from other agents by issuing a ***call for proposals***, which specifies the task and any conditions the manager is placing upon the execution of the task.
- Agents receiving the call for proposals are viewed as potential ***contractors***, and are able to generate proposals to perform the task as *propose* acts.
- The contractor's proposal includes the preconditions that the contractor is setting out for the task, which may be the price, time when the task will be done, etc.
- The contractor may ***refuse*** to propose.

FIPA Contract-net Protocol (cont'd)

- The manager receives back replies from all of the contractors, evaluates the proposals and makes its choice of which agents will perform the task. One, several, or no agents may be chosen. The agents of the selected proposal(s) will be sent an acceptance message, the others will receive a notice of rejection.
- Once the manager accepts the proposal the contractor acquires a commitment to perform the task. Once the contractor has completed the task, it sends a completion message to the manager.
- The protocol requires the manager to know when it has received all replies. In the case that a contractor fails to reply with either a *propose* or a *refuse*, the manager may potentially be left waiting indefinitely. To guard against this, the *cfp* includes a deadline by which replies should be received by the manager.

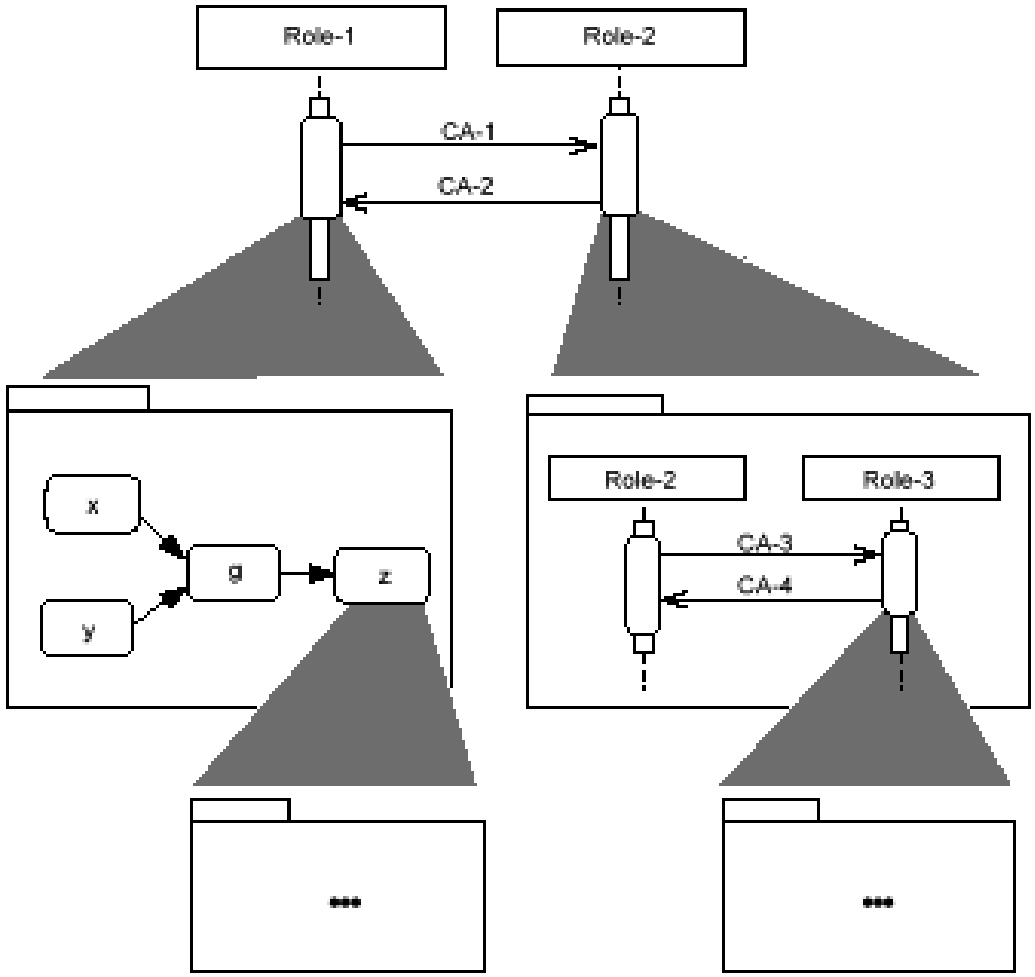
FIPA Notation



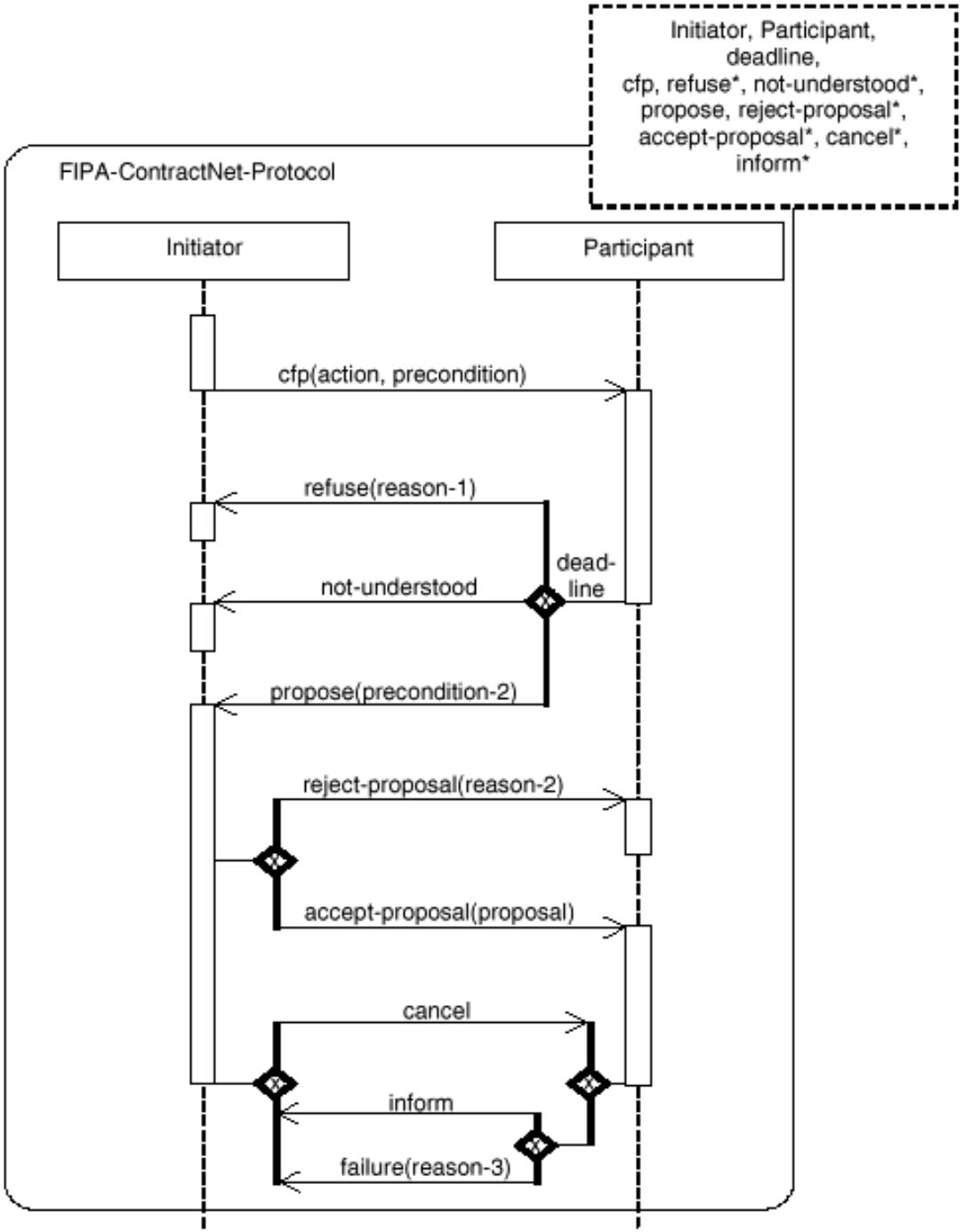
AUML Approach to Protocols

- AUML adopts a layered approach to protocols:
  - ✓ Level 1 - Represent the overall protocol (sequence diagrams, packages, templates)
  - ✓ Level 2- Represent interactions among agents (sequence, collaboration, activity, state diagrams)
  - ✓ Level 3- Represent internal Agent Processing (activity and state diagrams)

A Layered Approach to Protocols



Level 1: Overall protocol

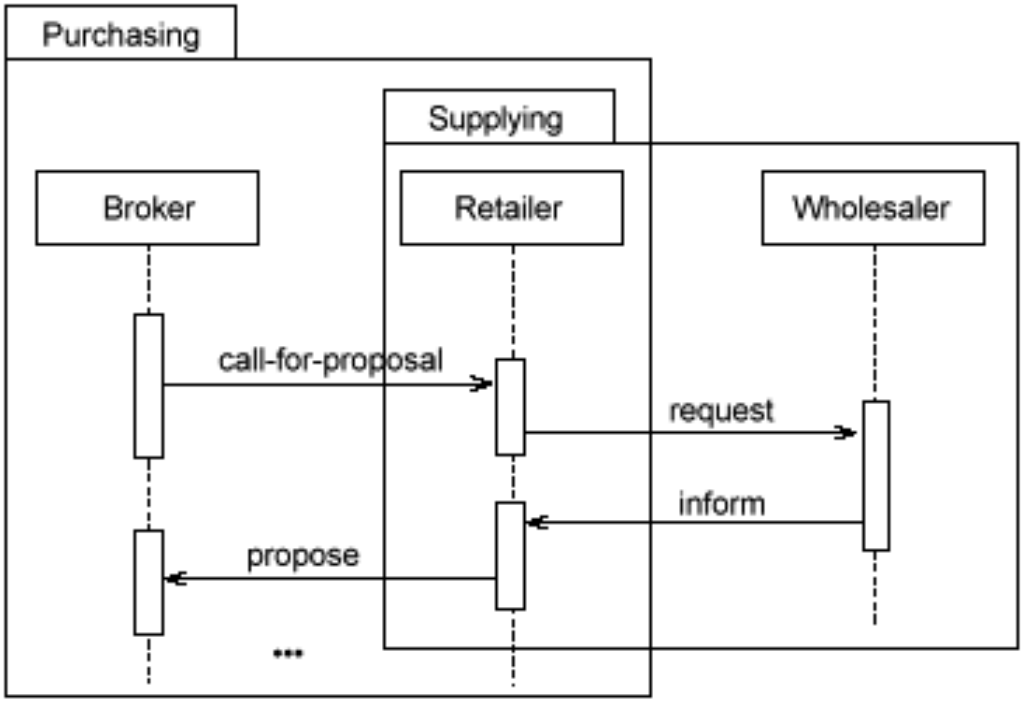


Level 1: Overall Protocol (Packages)

- Packages aggregate modeling elements into conceptual wholes
  - ✓ in UML 1.3 packages only group class diagrams
- Protocols can be codified as patterns of agent interaction
  - ✓ in AUML packages can group sequence diagrams (to model protocol patterns)

Level 1: Using Packages to Express Nested Protocols

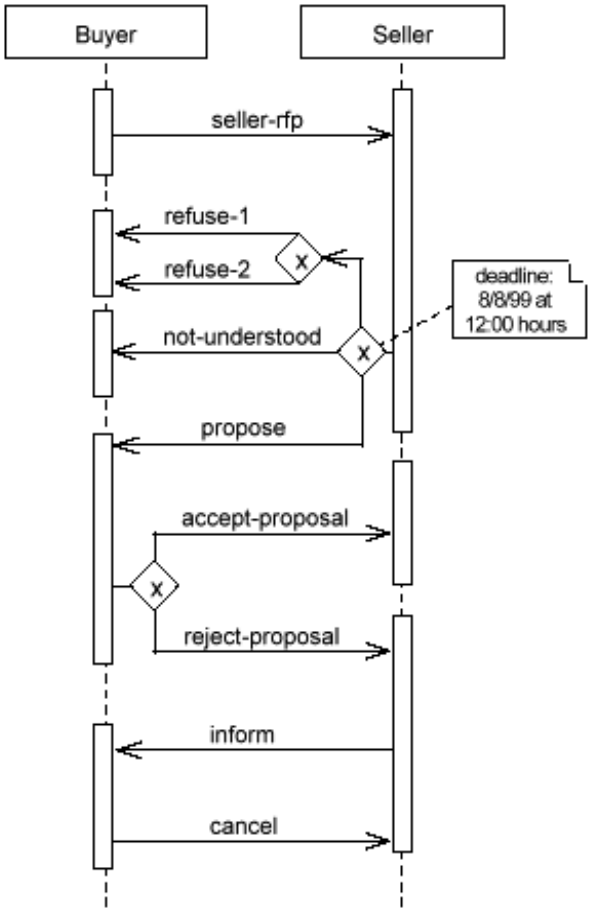
- Purchasing protocol (Broker X Retailer)
- Supplying protocol (Retailer X Wholesaler)



Level 1- Overall Protocol : Templates

- In order for a package to be a true pattern (not simply a reusable component) ***customization*** must be supported
- A template is a parameterized model element whose parameters are bound at model time
- Represented by doted box in upper right corner of the package

Contract Net Protocol



Level 1 Scenario Involving Buyers and Sellers

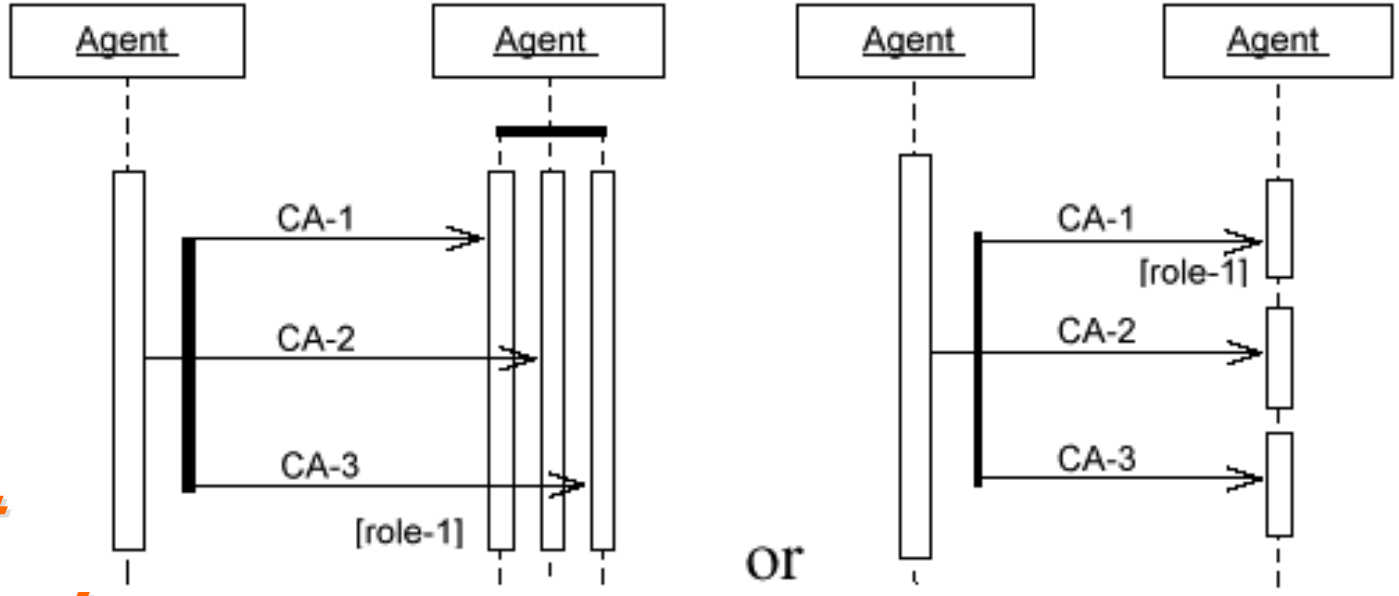
FIPA Contract Net Protocol

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

Level 2: Interactions Among Agents

- Extended sequence diagrams (concurrent threads of interaction)
- Collaboration diagrams
- Activity diagrams

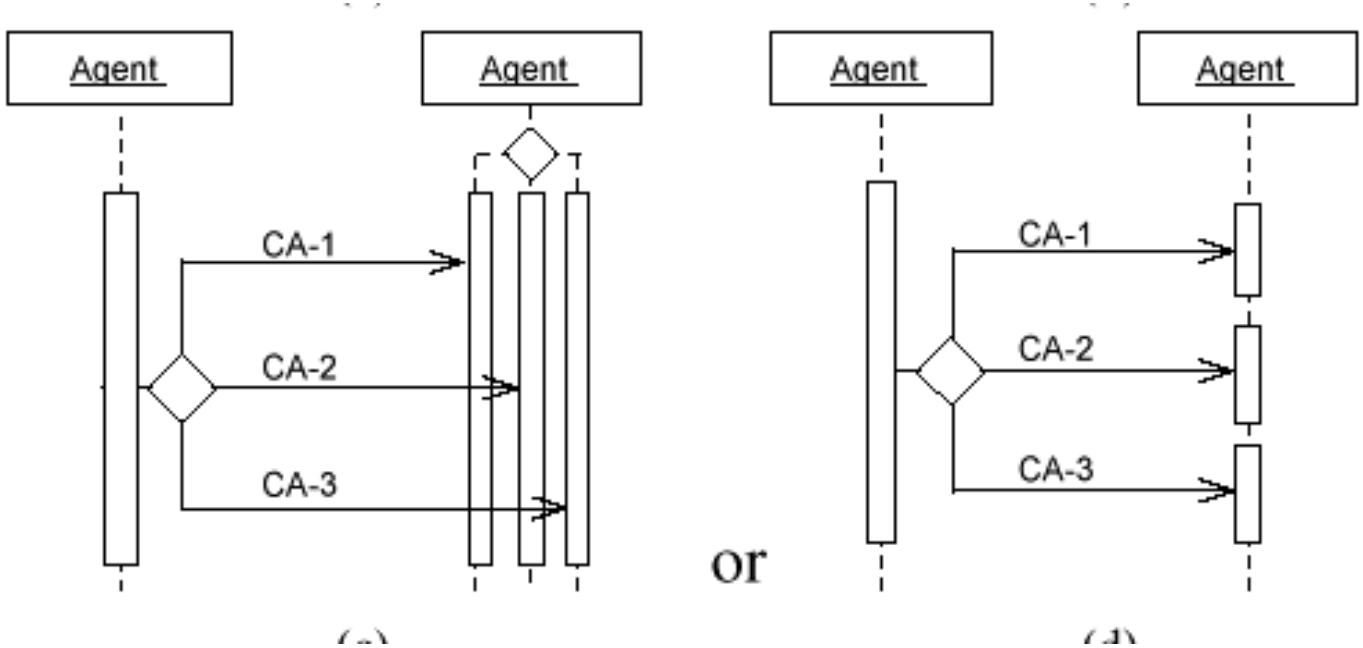
Extended Sequence Diagrams



Concurrent Communication

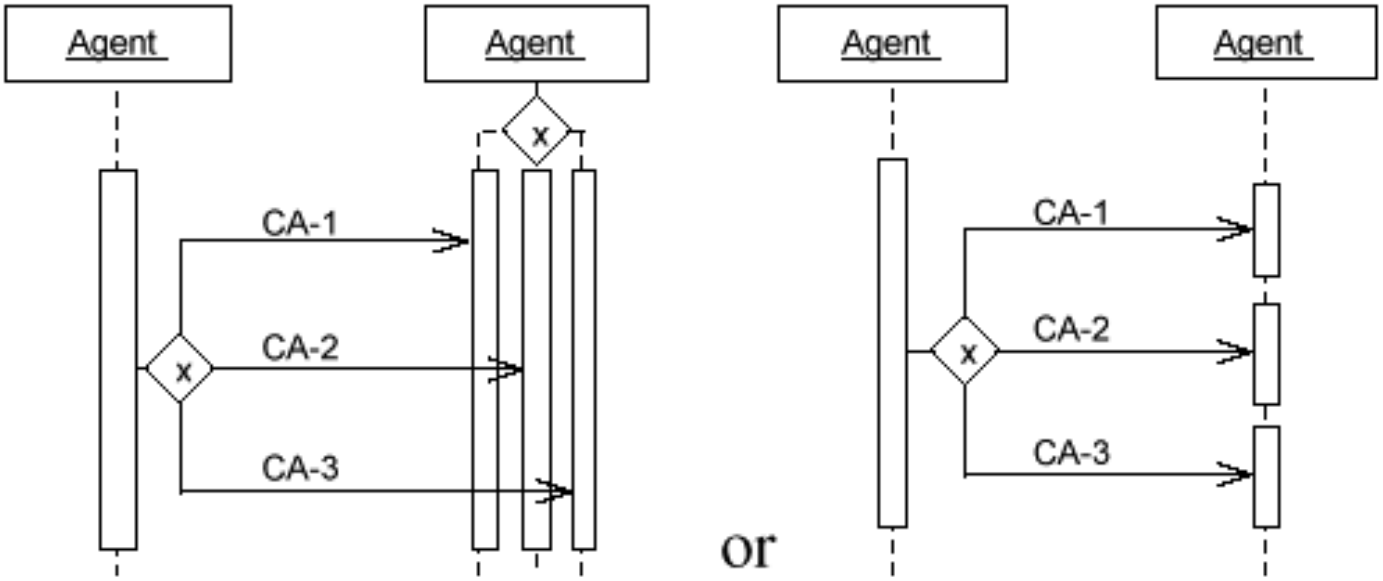
- The multiple vertical bars indicate that the receiving agent is processing several communication threads concurrently.

Extended Sequence Diagrams



- A decision box will decide which CAs (zero or more) will be sent.
- If more than one CA is sent, the communication is concurrent

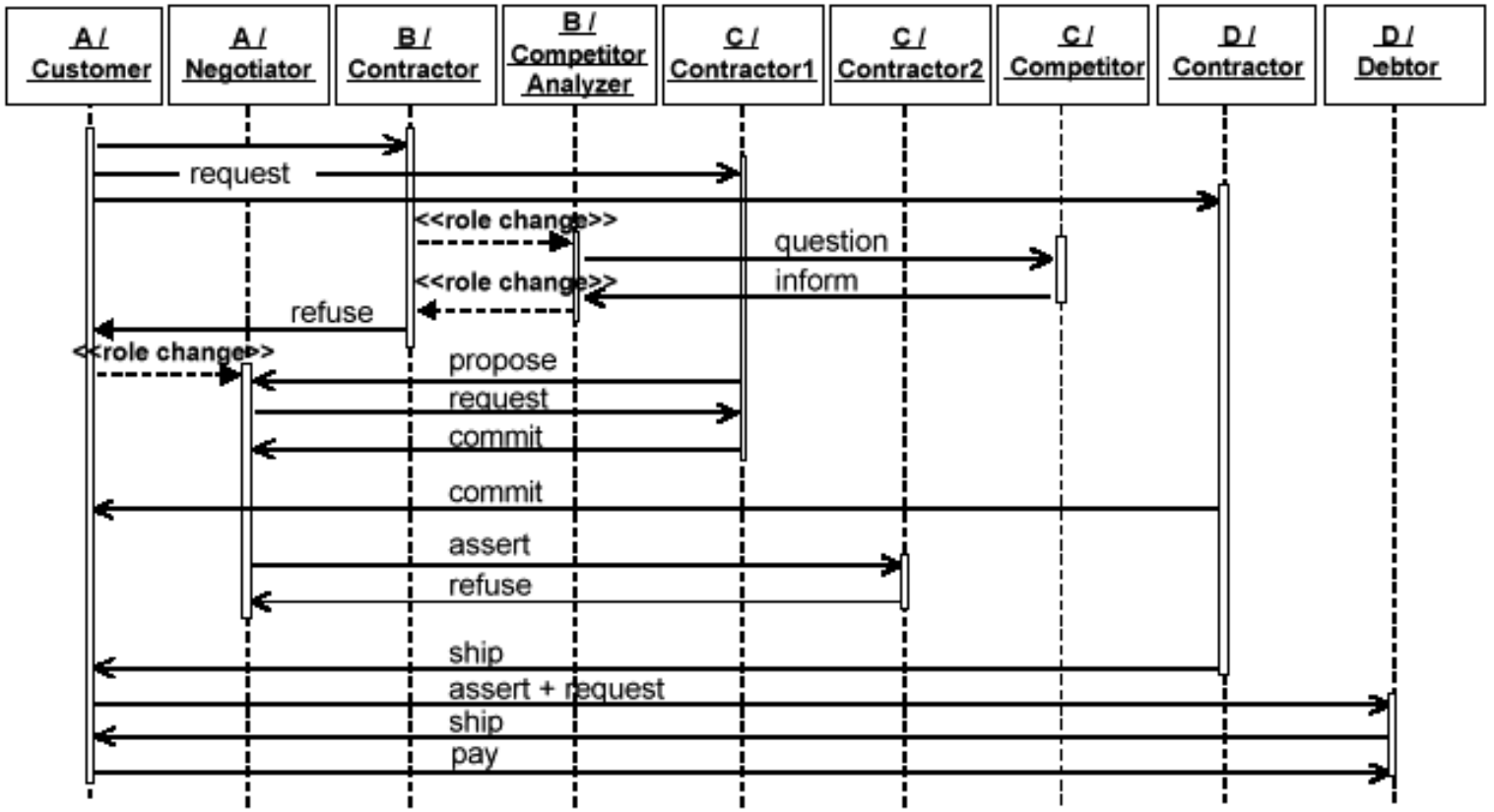
Extended Sequence Diagrams



- Exclusive OR, so exactly one CA will be sent

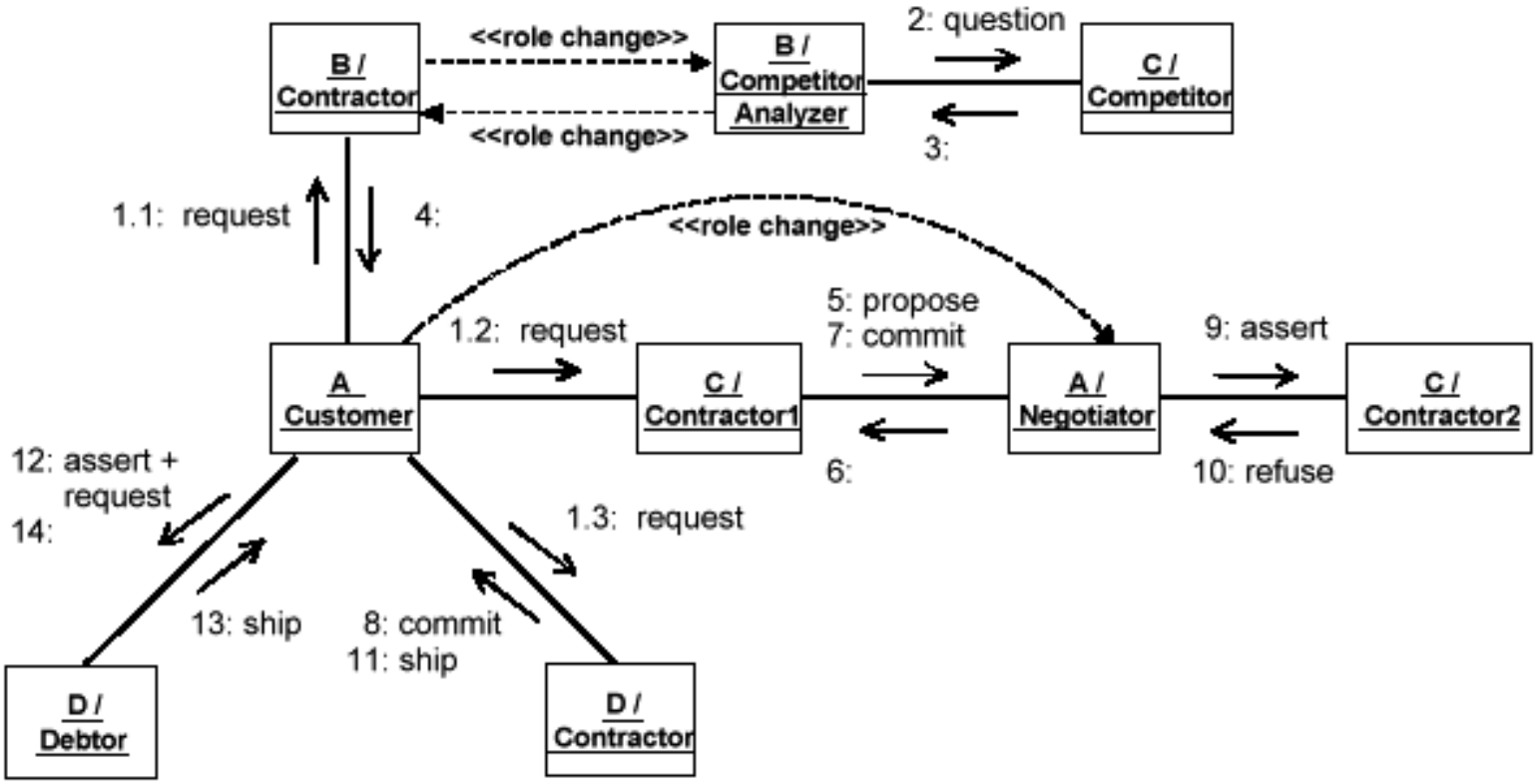
Extended Sequence Diagrams

- *Expressing the roles an agent play: Agent/Role*



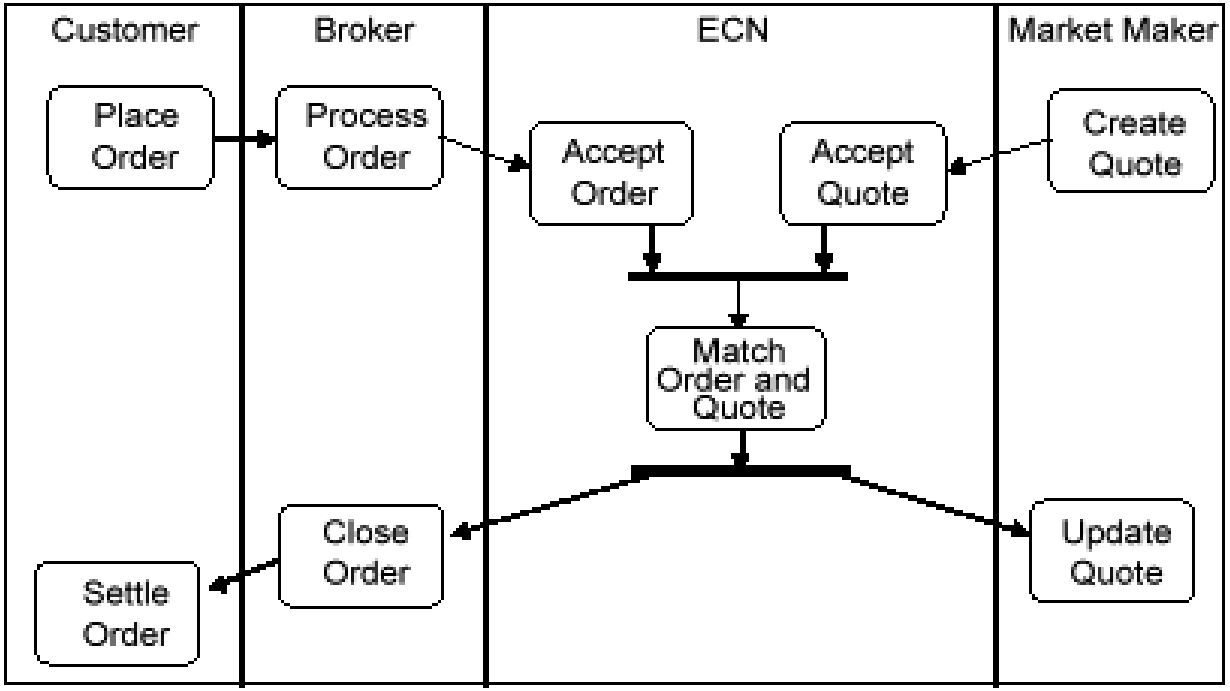
Collaboration Diagrams

- Another way of showing pattern of interaction among agents



Activity Diagrams

- Provides an explicit thread of control
- Useful for complex interaction protocols that involve concurrent processing



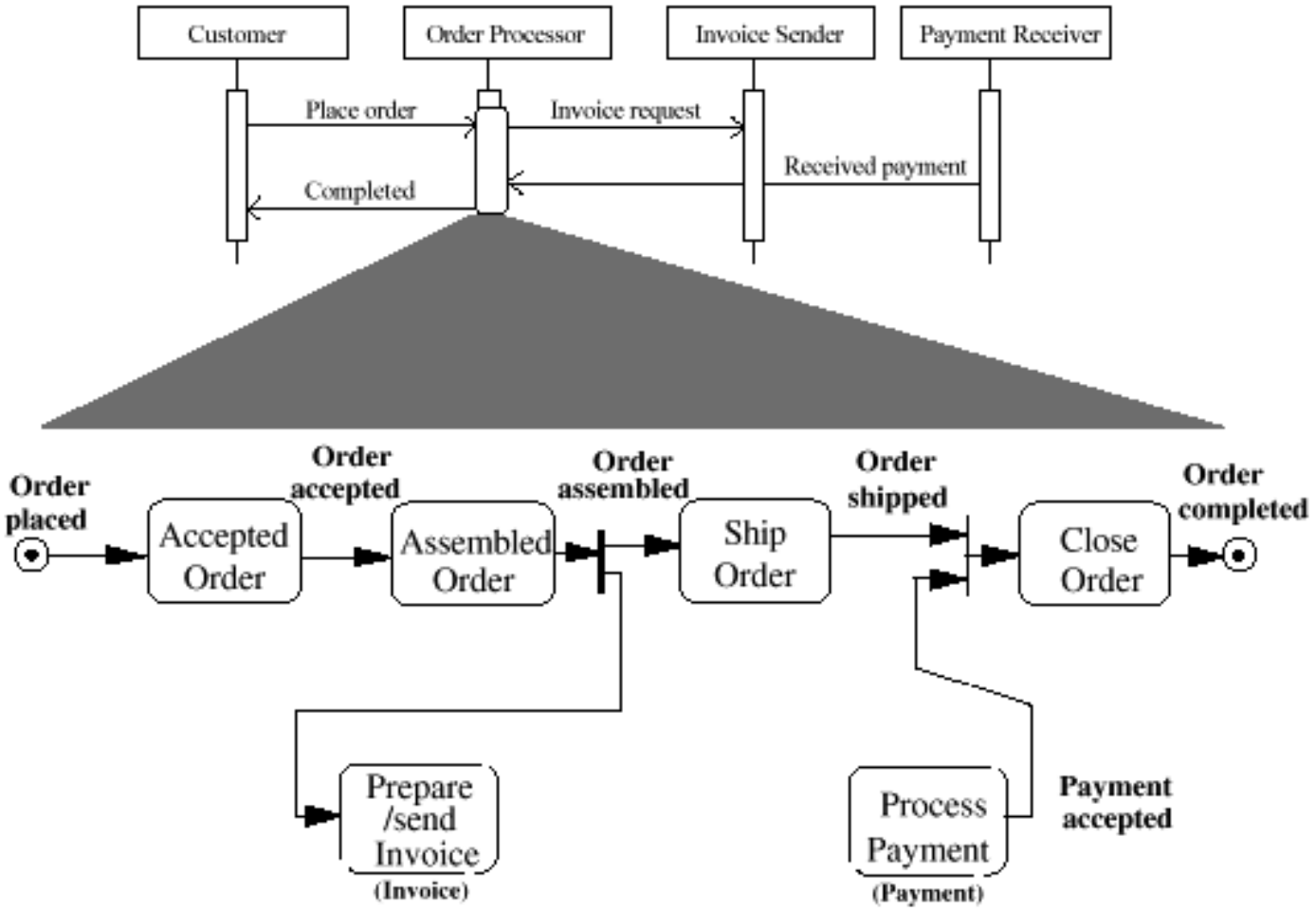
ECN: Electronic Commerce Network Agent

Level 3: Internal Agent Processing

- Specifying the internal processing of agents that are not aggregates
  - ✓ Activities
  - ✓ Statecharts

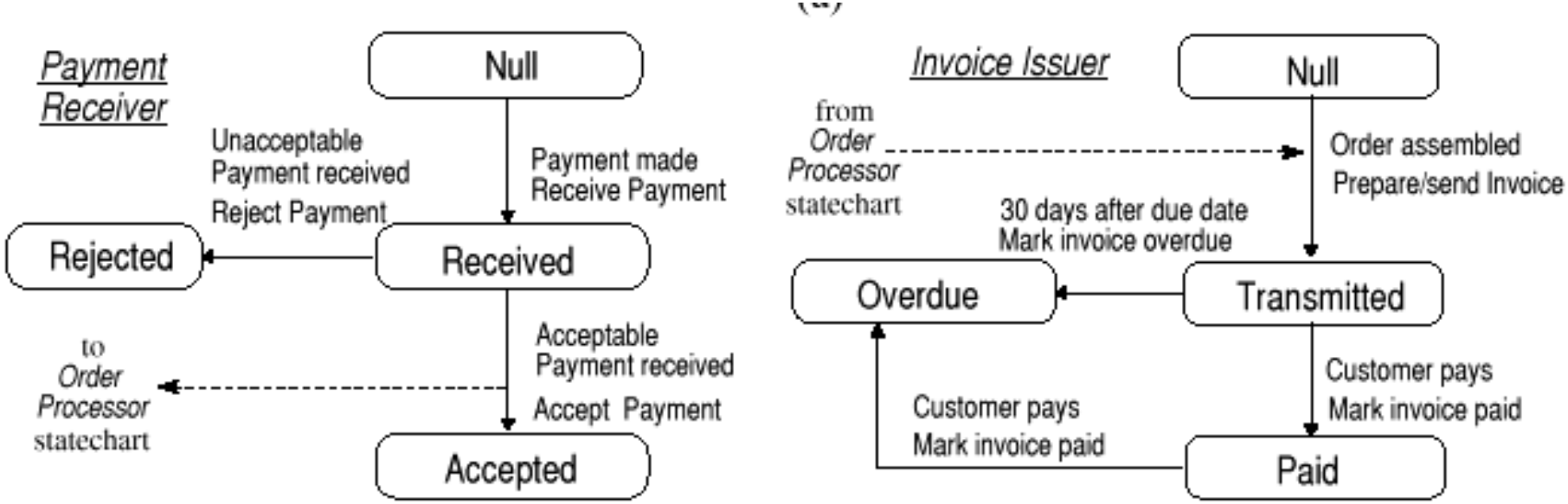
Internal Agent Processing

- Processing within an Order Processor agent



Internal Agent Processing

- Payment Receiver and Invoice Sender Agents



Other AUML Considerations

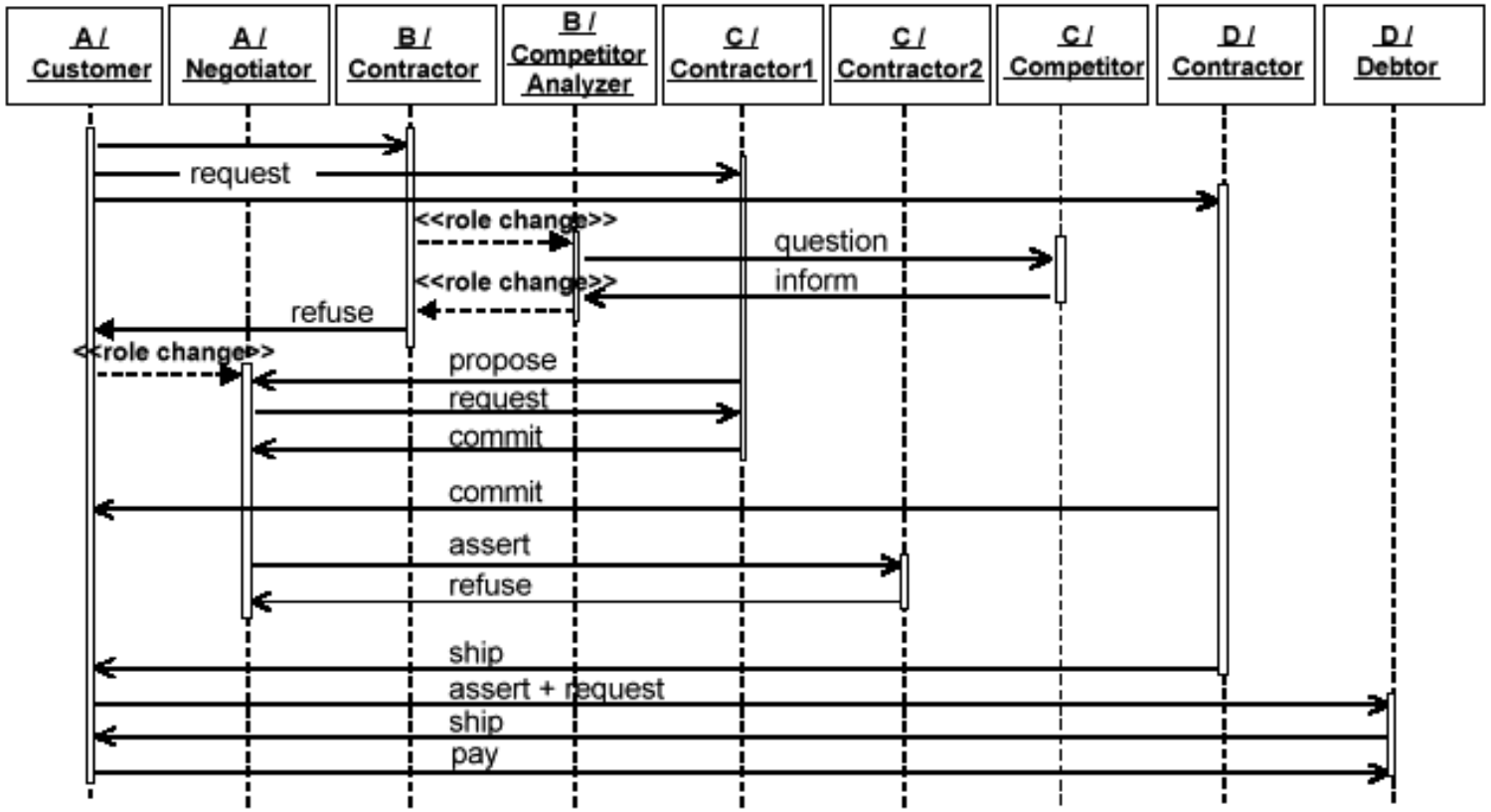
- Richer role specification
- Package extension
- Deployment diagram extension

Richer Role Specification

- Often we need to express the role an agent may play in the course of its interaction with other agents
- If the number of agent and roles increases, UML diagrams become graphically to complex

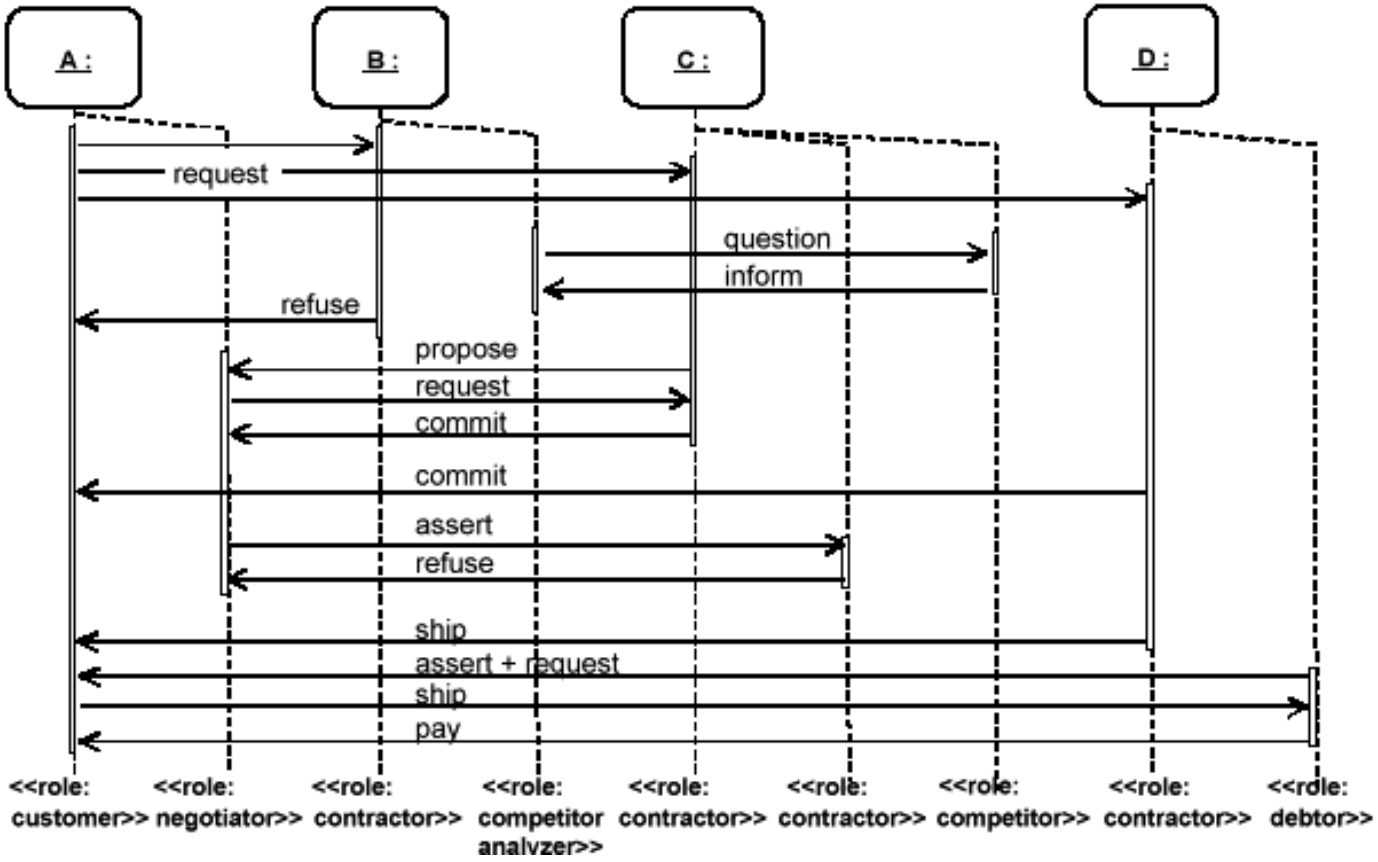
Agent/Role Extension

- Four agents playing 6 Roles:
  - ✓ *Customer, Negotiator, Contractor,*
  - ✓ *Competitor Analyzer, Competitor, Debtor*



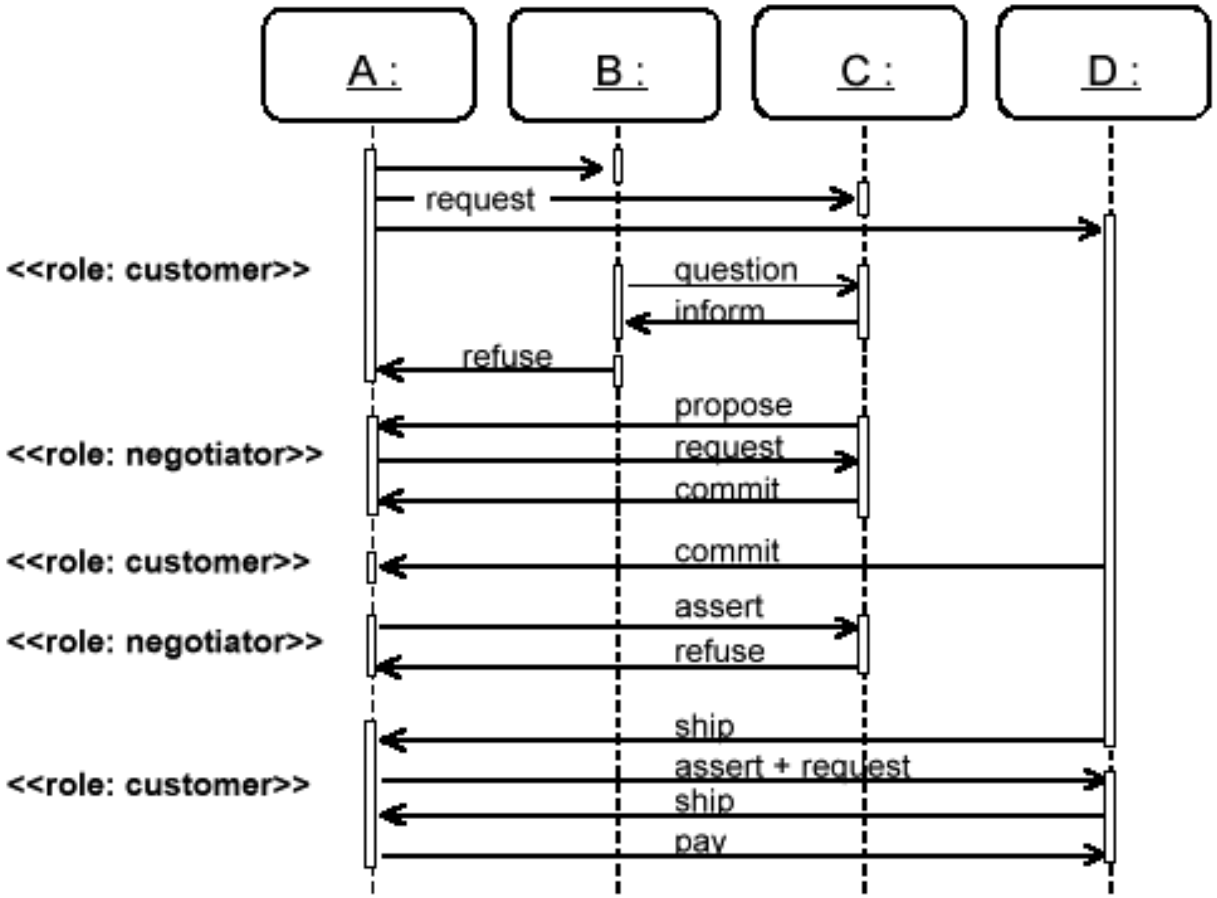
Reducing Visual Complexity

- Each role with its own lifeline



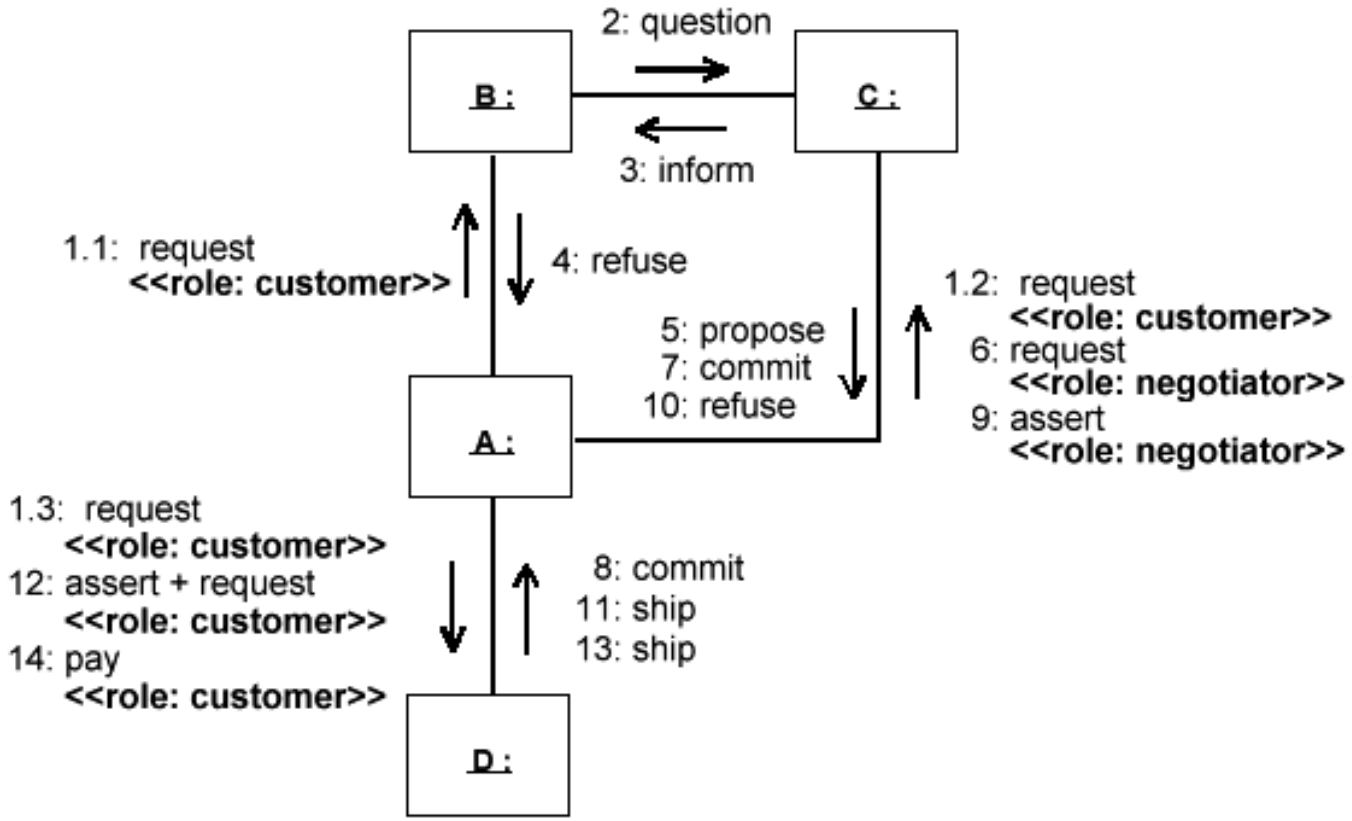
Reducing Visual Complexity

- Each agent with a single lifeline and each activation is labelled with the appropriate role name



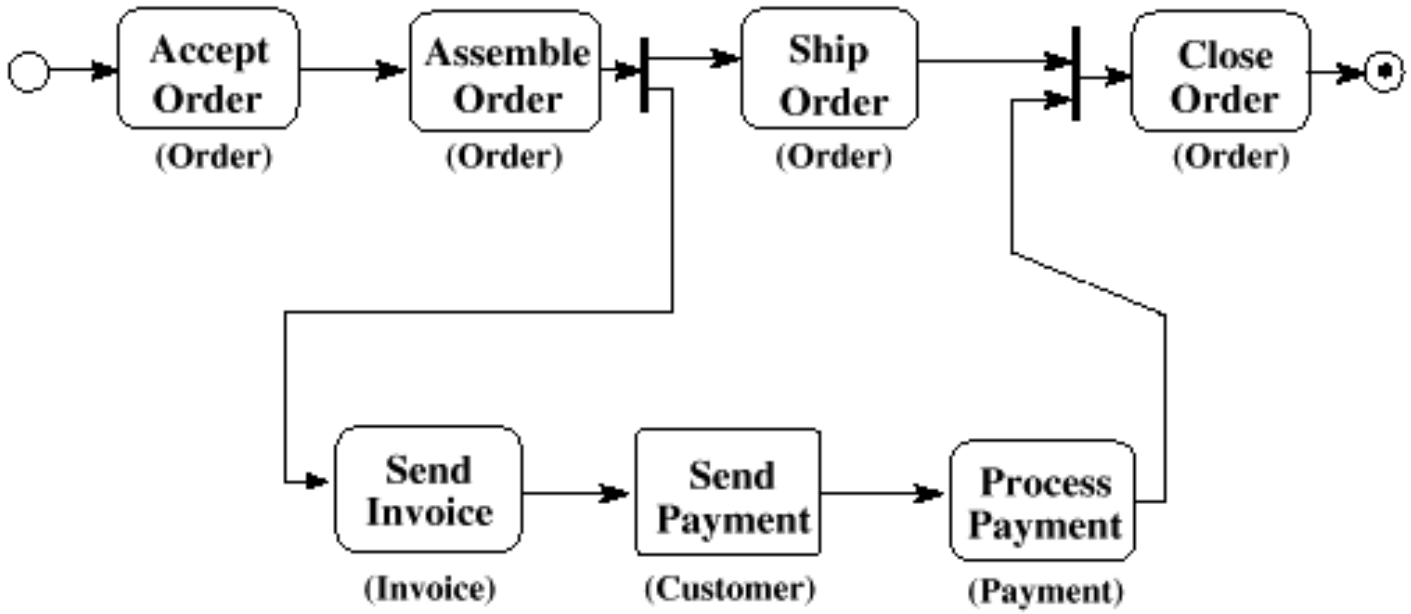
Role and Collaboration Diagrams

- UML has no facility to represent agent roles on interaction lines
- Labelling messages with the role



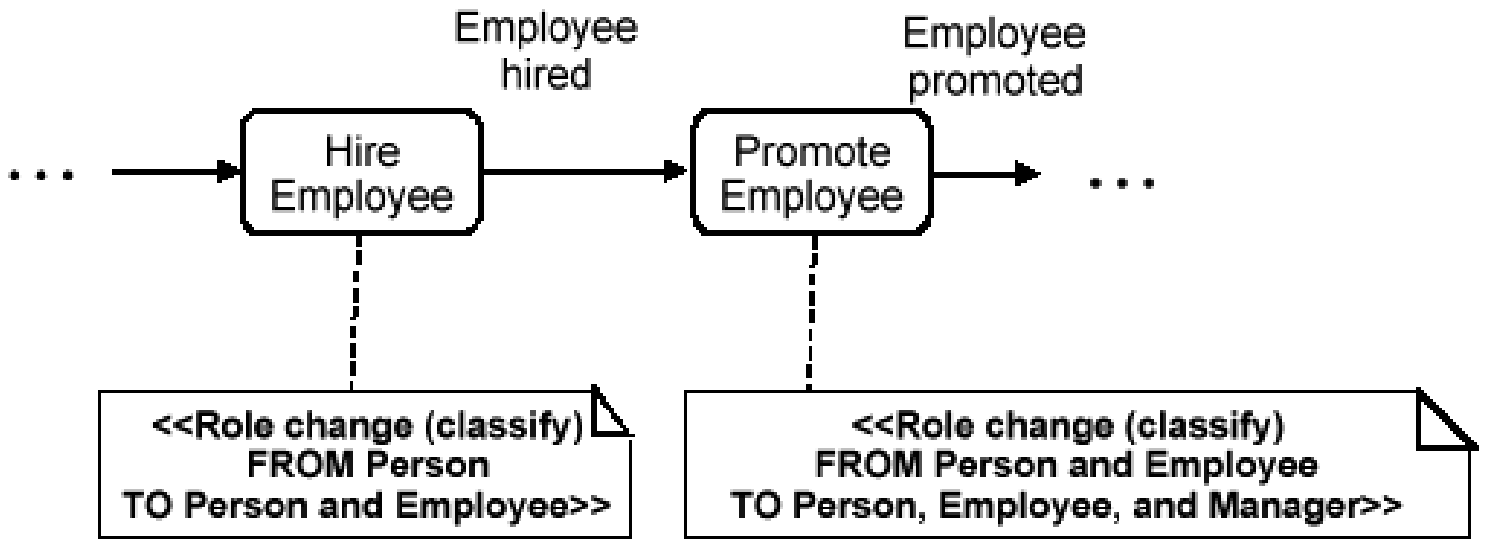
Role and Activity Diagram

- Roles can be associated with activities



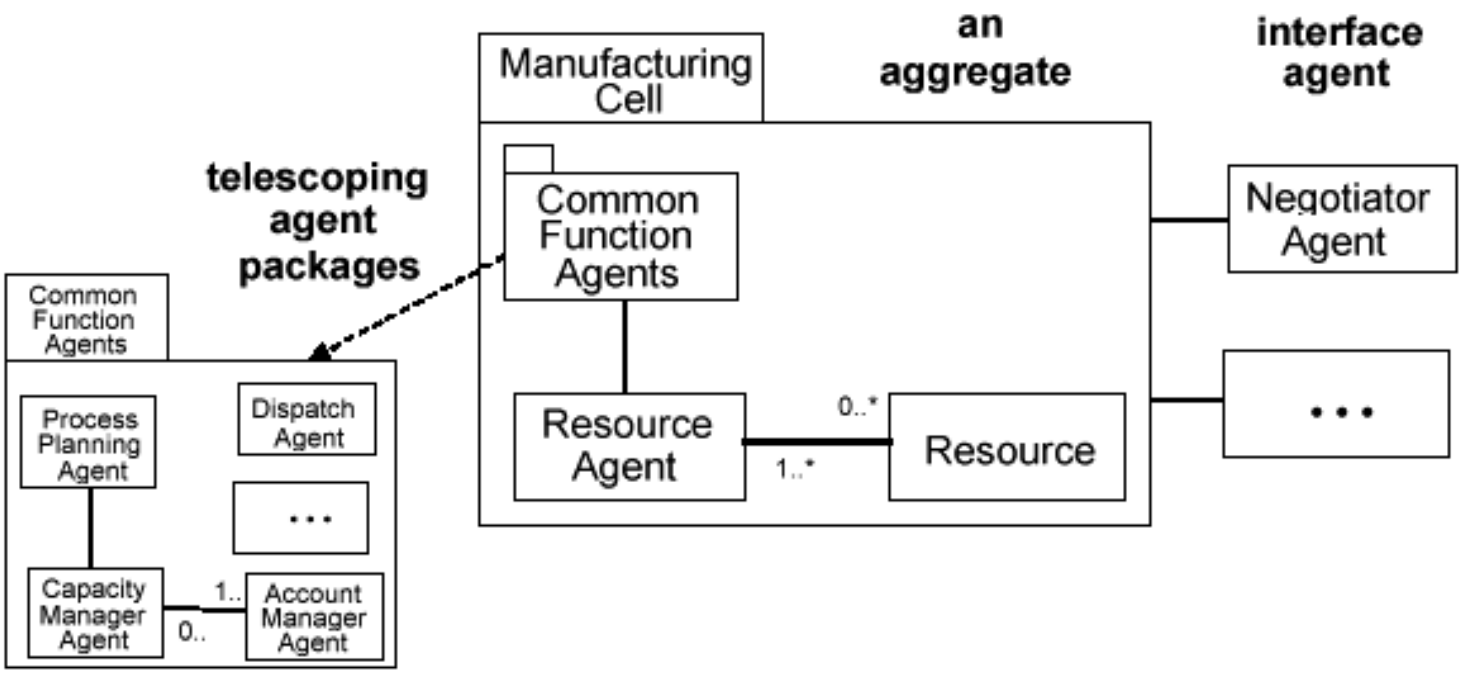
Role and Activity Diagram

- Roles can be represented by notes



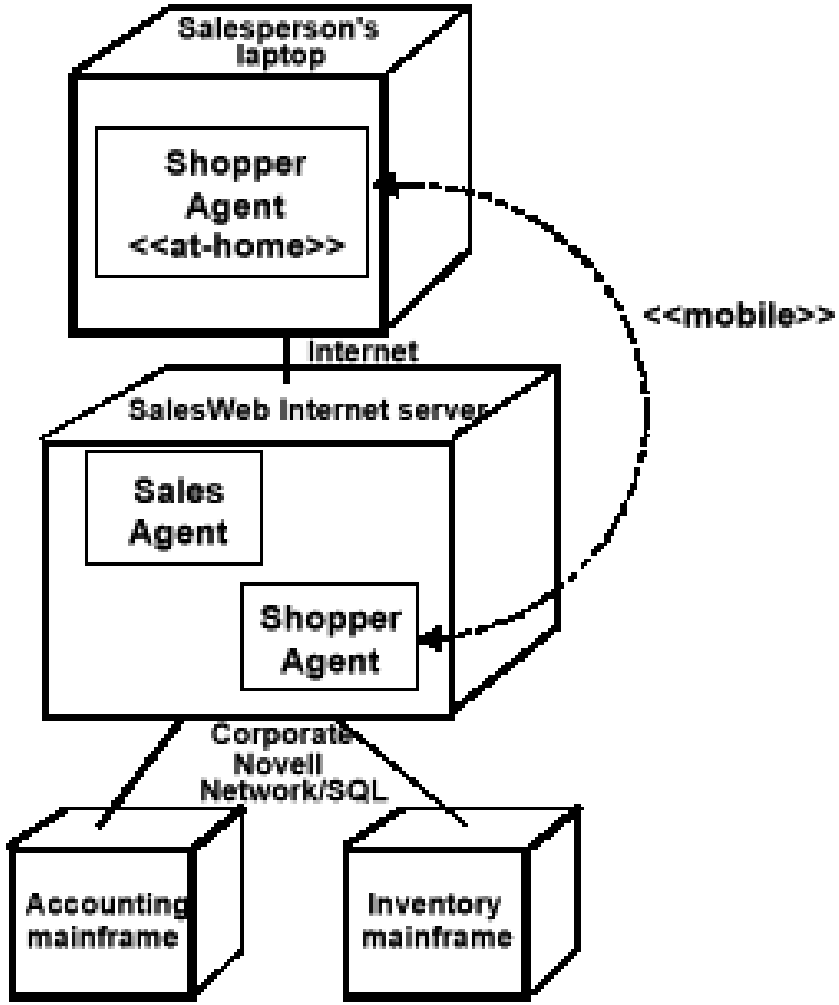
Package Extensions

- The interface can be an agent itself



Deployment Diagram Extensions

- Indication of mobility paths and at-home declarations



Conclusions

- (A)UML provides tools for
  - ✓ Specifying agent interaction protocols
  - ✓ Representing the internal behavior of an agent
  - ✓ Representing role specification, packages with agent interfaces, deployment diagrams indicating mobility, etc.

References

- James Odell, H. Van Dyke Parunak, Bernhard Bauer. *Representing Agent Interaction Protocols in UML*, 1999.
- James Odell, H. Van Dyke Parunak, Bernhard Bauer. *Extending UML for Agents*, 2000.
- James Odell, Conrad Bock. OMG document ad/99-12-01. *Suggested UML Extensions for Agents.* Submitted to the OMG's Analysis and Design Task Force (ADTF) in response to the Request of Information (RFI) entitled "UML2.0 RFI". Dec. 1999.
- Bernhard Bauer. OMG document ad/99-12-03. *Extending UML for the Specification of Agent Interaction Protocols.* FIPA submission to the OMG's Analysis and Design Task Force (ADTF) in response to the Request of Information (RFI) entitled "UML2.0 RFI". Dec. 1999.