OWL-P: Protocols for Processes

Toward the Pragmatic Web

Munindar P. Singh

(*Students:* Amit K. Chopra, Ashok U. Mallya)

singh@ncsu.edu

Department of Computer Science

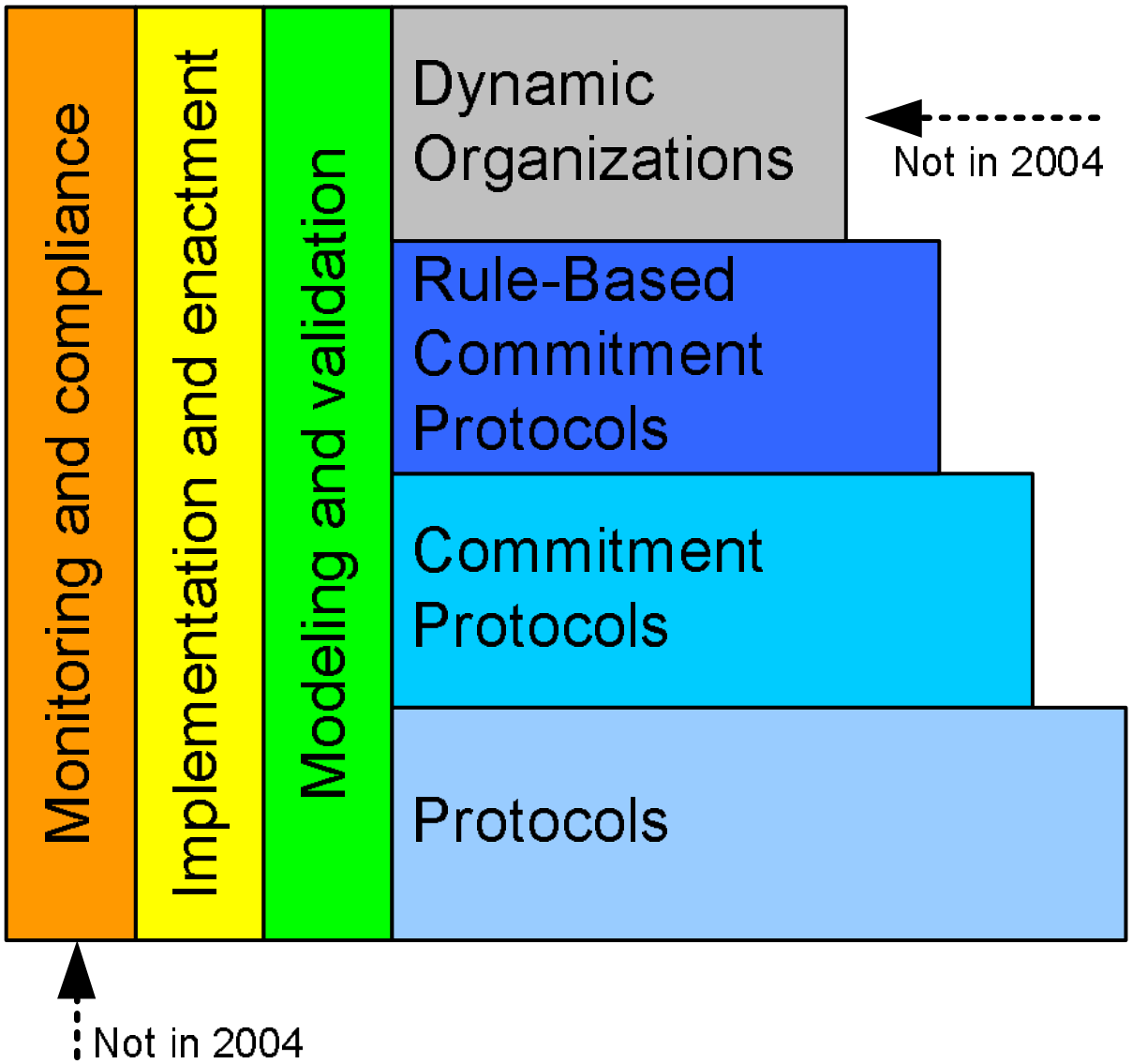
North Carolina State University

http://www.csc.ncsu.edu/faculty/mpsingh/

Why Processes and Protocols?

- Heavy interest from IT practitioners.
- Standardization efforts.
- Match with Semantic Web research.
  - Tractable problems with high impact.
  - Great application area for semantics.
- Segue into upcoming research program.

Emphases of this Project: 1



Emphases of this Project: 2

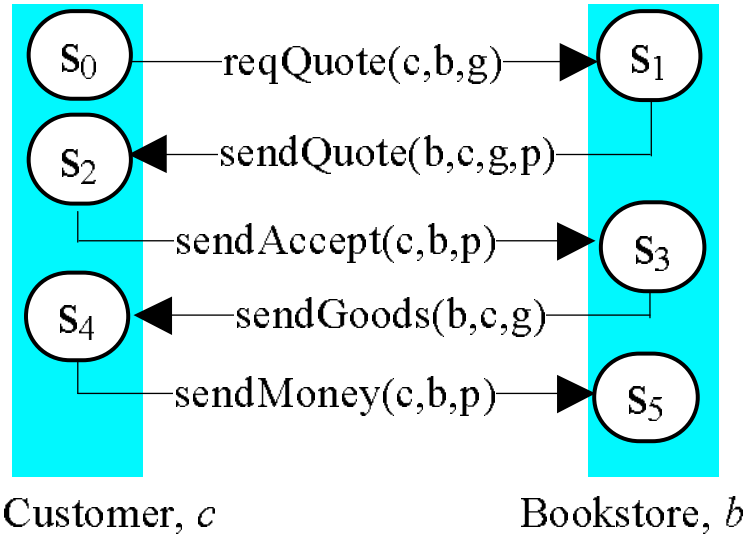
- *Protocols:* Support reuse via *abstraction* and *composition* for process modeling and enactment.
- *Commitments:* Enable flexible modeling and enactment of protocols.
- *Engineering:* Full automation is not needed.
- Tools needed for engineering.
  - Modeling and validation.
  - Implementation and enactment.
  - Monitoring and compliance.

Trends and Assessment

- Increasing # of business protocols.
  - IOTP, Escrow, SET, NetBill, . . .
  - RosettaNet: 107 Partner Interface Processes (PIPs).
  - ebXML Business Process Specification Schema (BPSS).
- Generally highly limited: two party, request-response protocols.
- No commitments; no formal semantics.
- Limited support for modeling or enactment.

Simple Scenario and Example Run

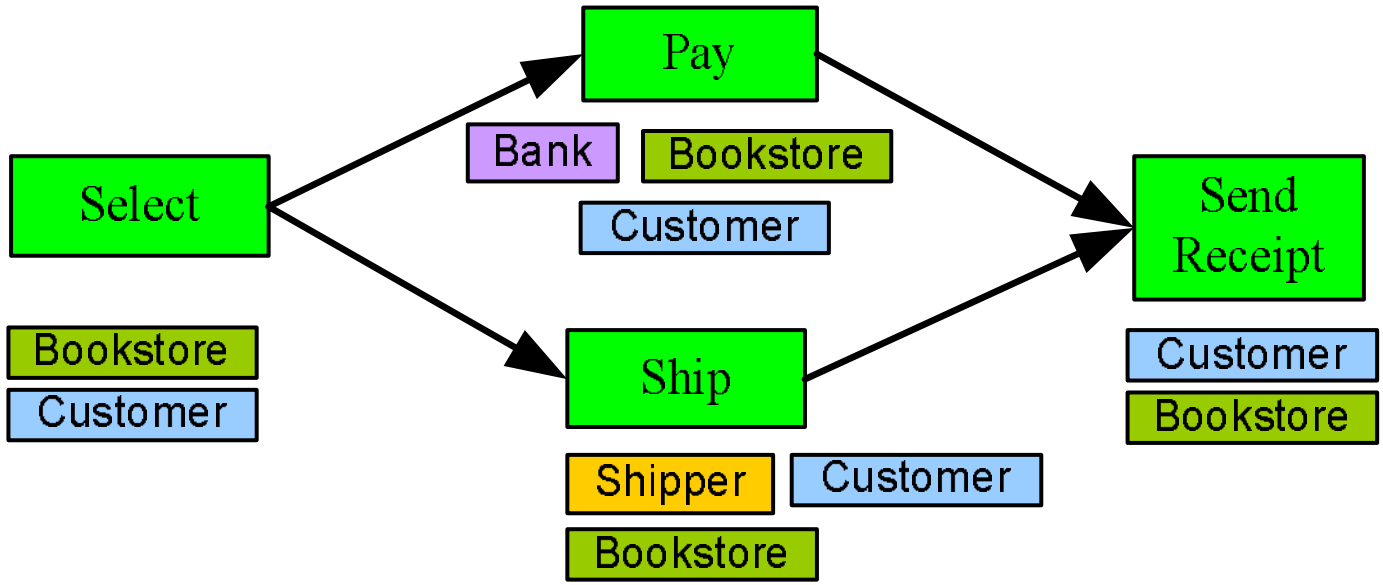
- A customer (C) looks up a book at a vendor (B) and is quoted price and availability.
- C orders the book from B.
- B ships to C.
- C pays B.

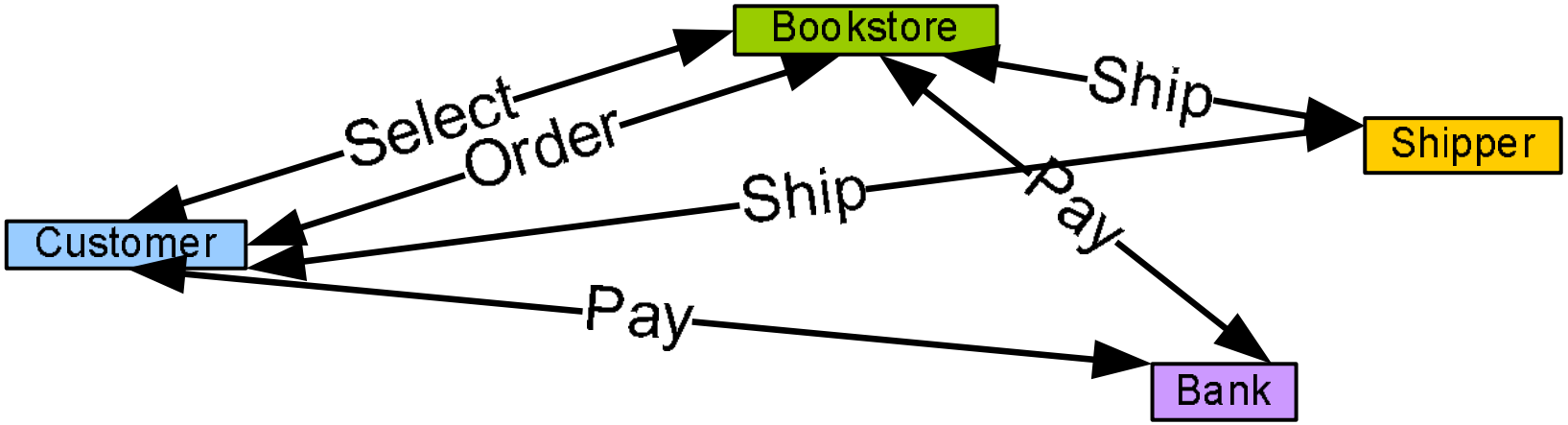


Challenges: Modeling

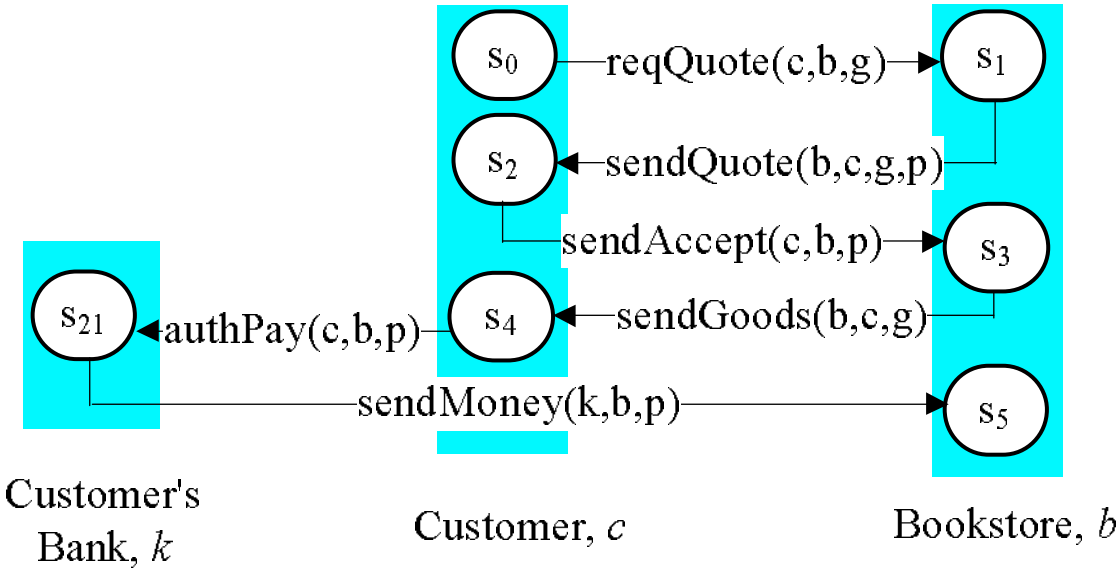
- *Refinement:* pay by credit card versus pay.
- *Extensibility:* verify C's attributes, e.g., age.
- *Adjustment:* receive payment before shipping; receive book before paying.
- Alternative execution examples:
  - B arranges for a shipper (S) to deliver the book to C.
  - C pays via bank (K).
  - Compose a process from the above.

Process View: Global or Protocol?

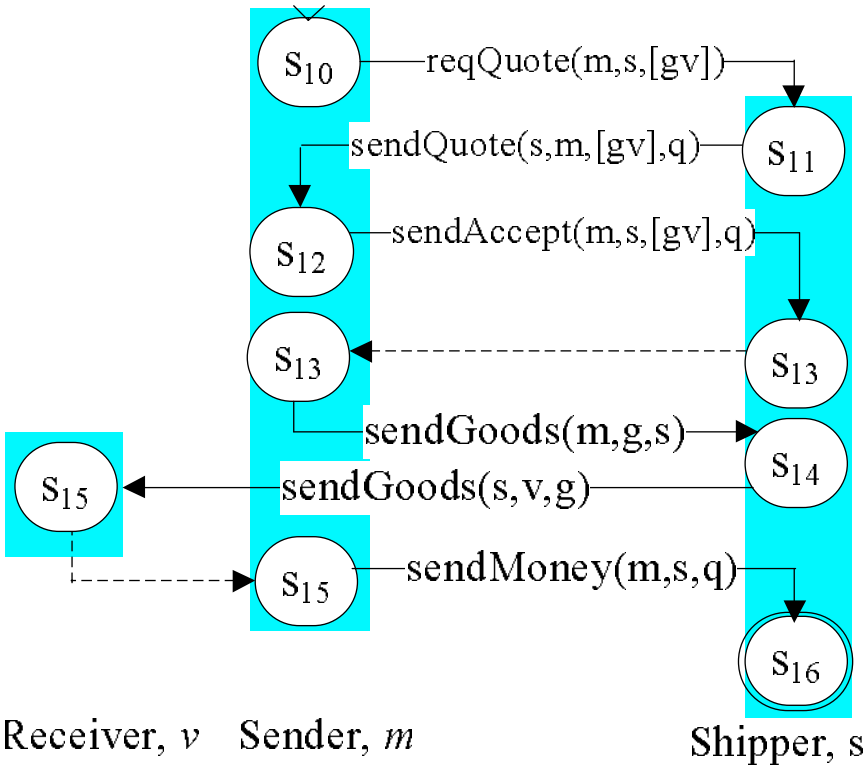




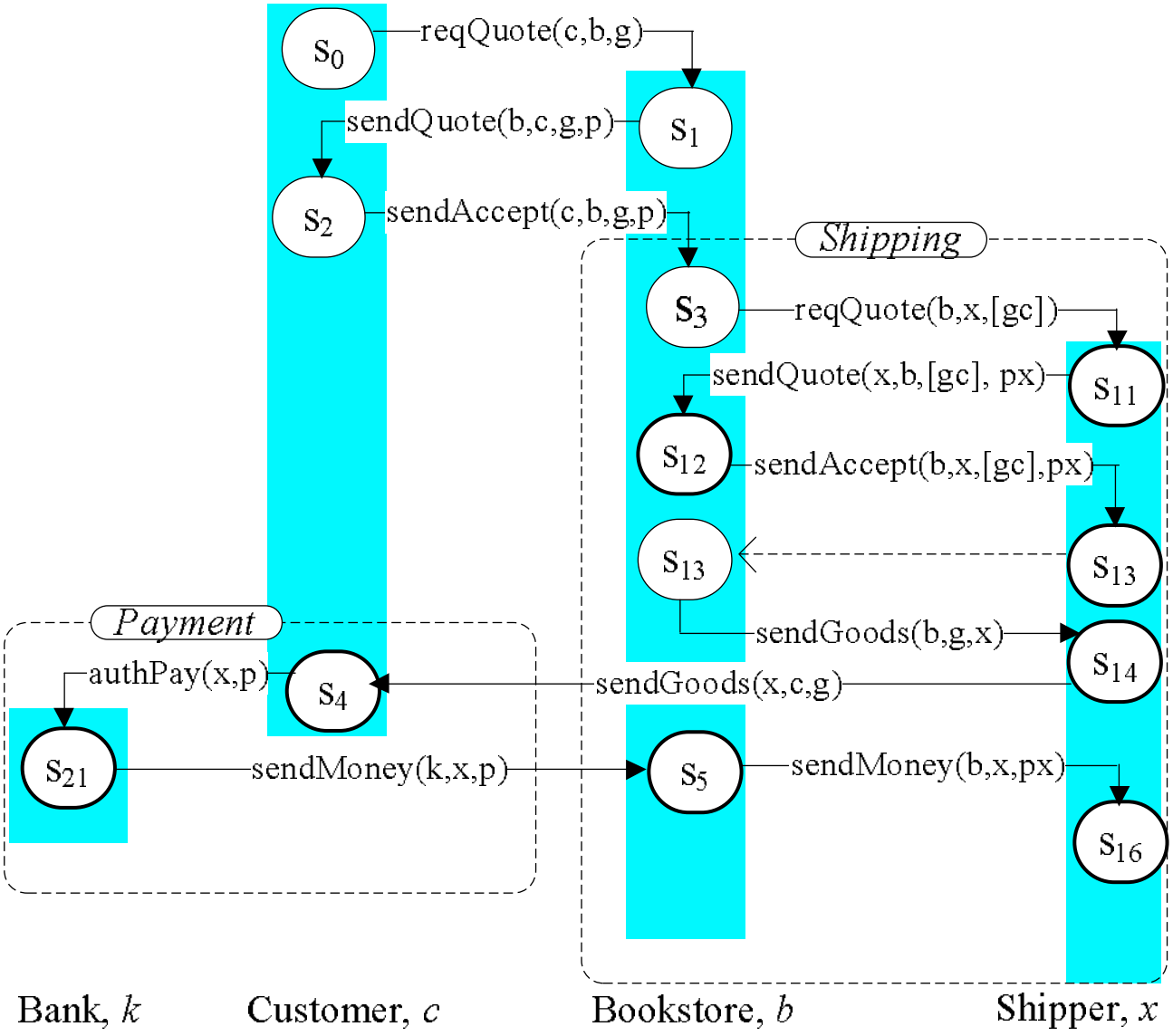
Example Run: Pay via Bank



Example Run: Shipper Protocol



Example Run: Composed Purchase

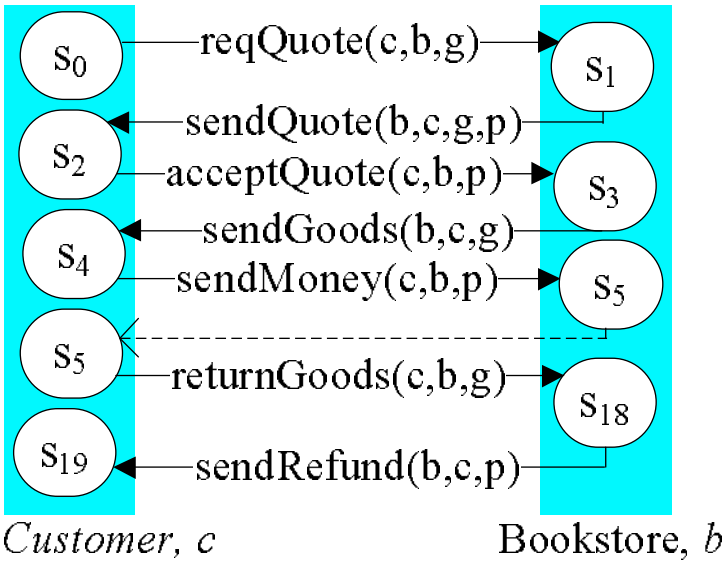


Challenges: Enactment

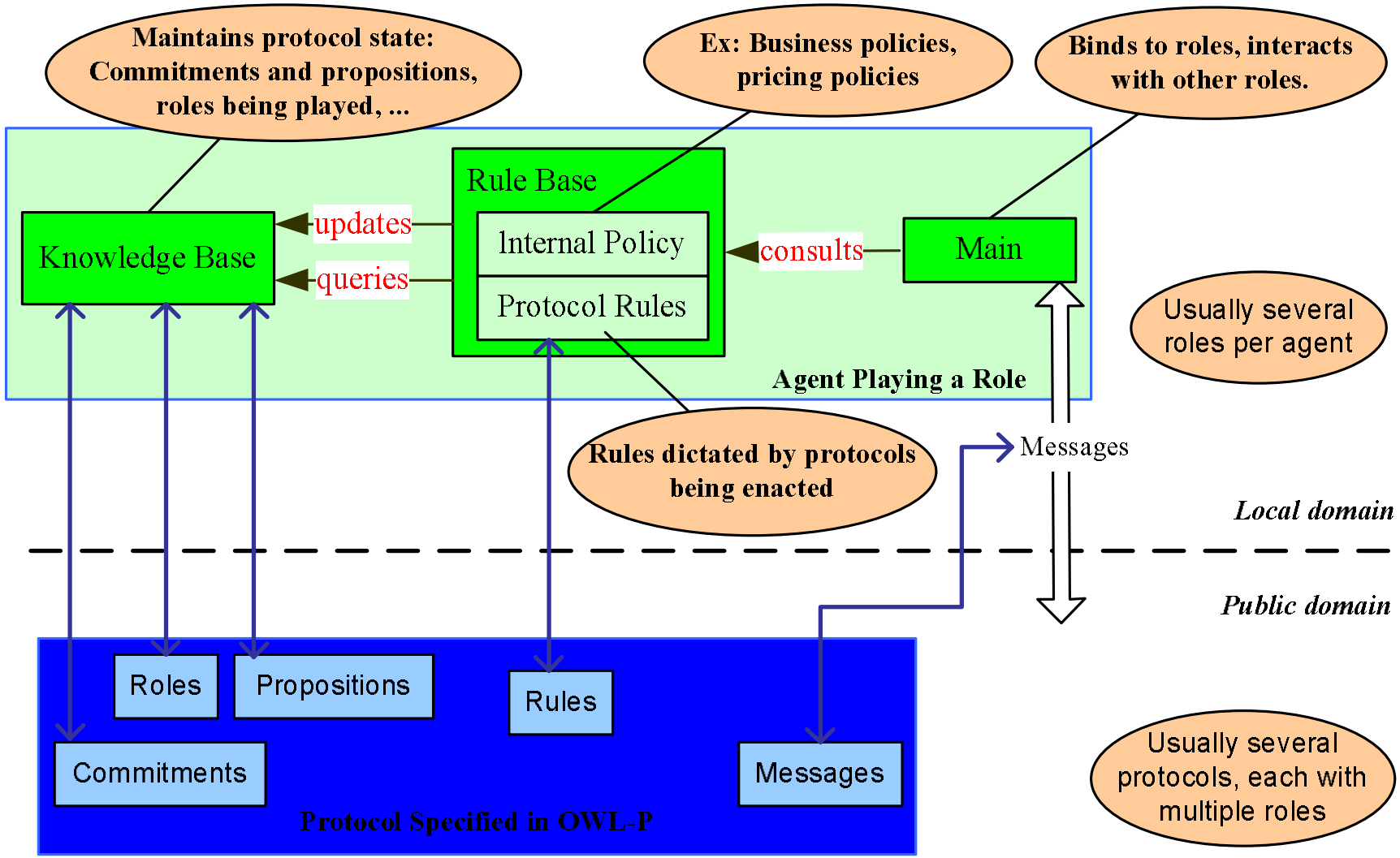
- *Behaving adaptively:* decide dynamically to ship before payment to trusted Cs.
- *Handling exceptions*.
  - External problems: cannot ship book.
  - Detecting violations: no payment; book arrives damaged.
  - Correcting violations: remind, complain, refund, . . .
- *Exploiting opportunities:* combine orders from same C.

Example Run: Return and Refund

Example: Uniform Commercial Code (UCC) allows returns with refunds for goods that are received damaged.



Architecture



Deliverables

- OWL-P: OWL for protocols.
  - Roles.
  - Messages: content as propositions and commitments.
  - Rules to describe messages and role constraints.
- Autonomous communicating agents (JADE).
- Tool to generate skeletons from OWL-P.
- Rule-based policies that help agents satisfy their protocol roles.
- Methodology to develop agents.

Functionality and IP Status

Open source; on SemWebCentral 6/30 onwards.

- Preliminary versions implemented for OWL-P.
  - Multiagent architecture to enact.
  - Policy-based architecture for each agent.
- Upcoming versions.
  - Incorporate rules better (6/30).
  - Compose protocols (6/30).
  - Fully treat commitments (9/30).
  - Represent quality of service for configuration (9/30) and apply it (12/31).
  - Incorporate policies (12/31).

Papers on this Topic

- "Protocols for Processes: Programming in the Large for Open Systems." *OOPSLA*, Oct 2004.
- "Agent Communication Languages: Rethinking the Principles." *IEEE Computer*, 31(12):40–47, Dec 1998.
- "An Ontology for Commitments in Multiagent Systems." *AI & Law*, 7:97–113, 1999.
- "Reasoning About Commitments in the Event Calculus: An Approach for Specifying and Executing Protocols." *Annals Math & AI*, 42(1-3), 2004.
- "Verifying Compliance with Commitment Protocols." *J. Auton Agents & MAS*, 2(3):217–236, Sep 1999.