Protocols for Processes

Programming in the Large for Open Systems

Munindar P. Singh

(*Students:* Amit K. Chopra, Nirmit V. Desai, 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
  - Any number of products
- Current industry approaches are impoverished: scripting languages
  - No special abstractions for dealing with open systems: autonomy, heterogeneity, dynamism
  - That is, not designed for SOAs

Programming in the Large

About creating large software systems; the main challenges of modern software engineering

- Traditional emphases
  - Built by large teams
  - Long-lived and stateful components
- Proposed emphases
  - Special treatment of open systems: autonomy, heterogeneity, dynamism
  - Long-lived and stateful components interacting in subtle ways

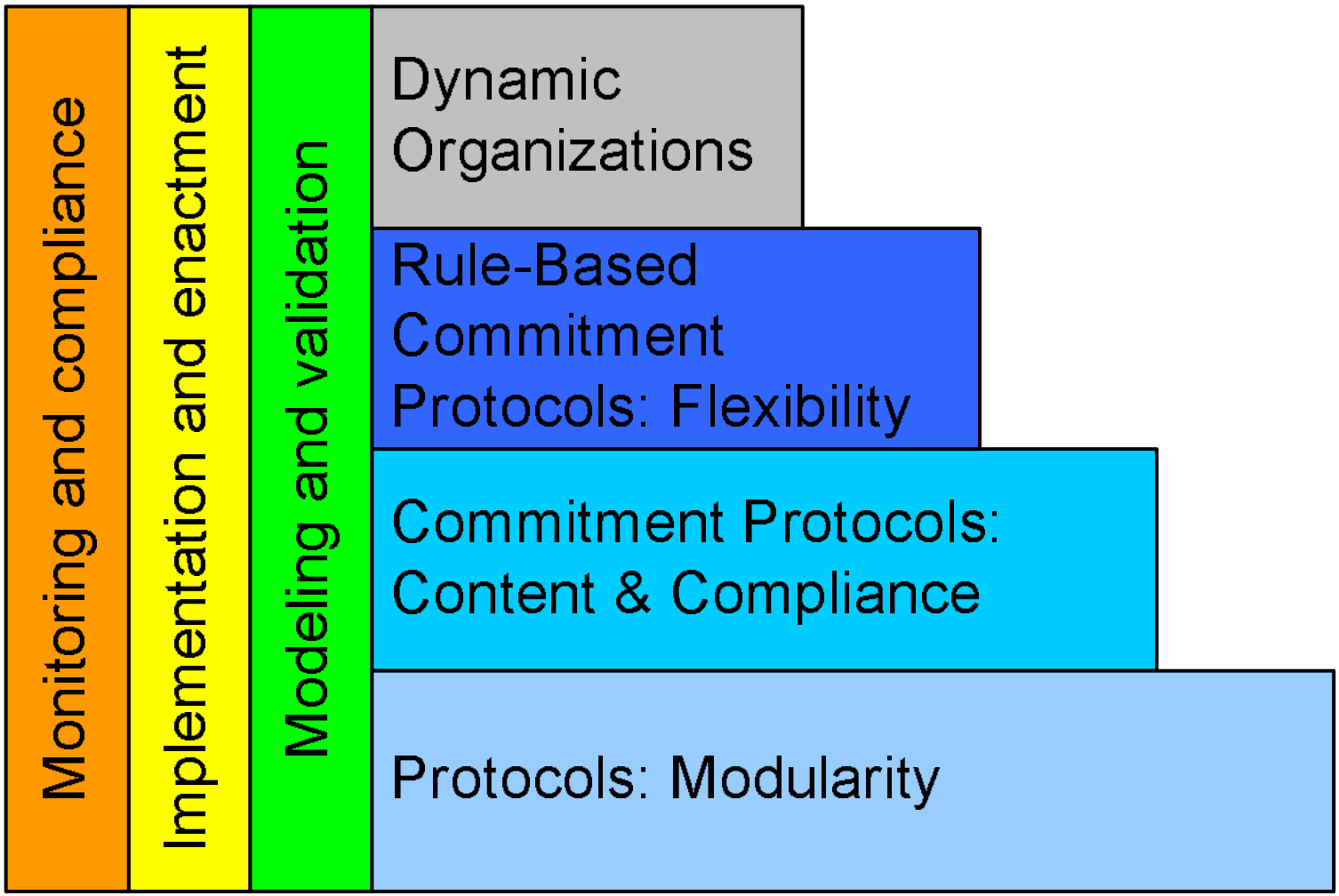
The Essential Tension

- *Reusability* requires
  - Context freedom
  - Encapsulation
- *Usability* (usefulness) requires
  - Context sensitivity
  - Varieties of context include organizations, laws, and the real world
- Main idea
  - The components have a life of their own
  - The interactions are what matter

A Process is . . .

- *Orchestration:* a partial order of actions under the control of a central conductor
  - Akin to a workflow or flow in BPEL
- *Choreography:* an exchange of messages among participants
  - Akin to a conversation as described by WS-Chor
- *Collaboration:* a joint set of activities among business partners
  - Akin to real business; essential for SOAs

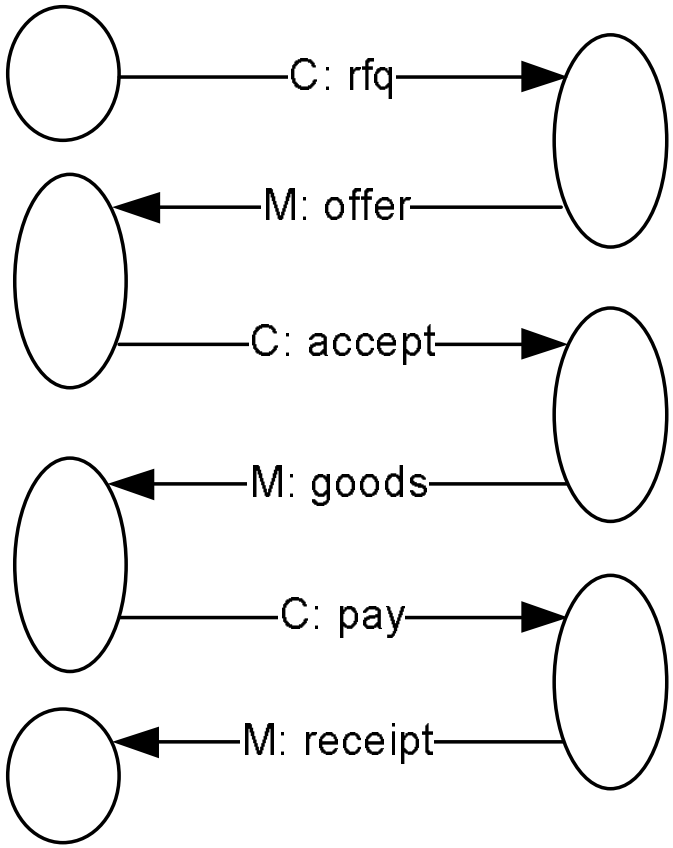
Emphases of Collaboration

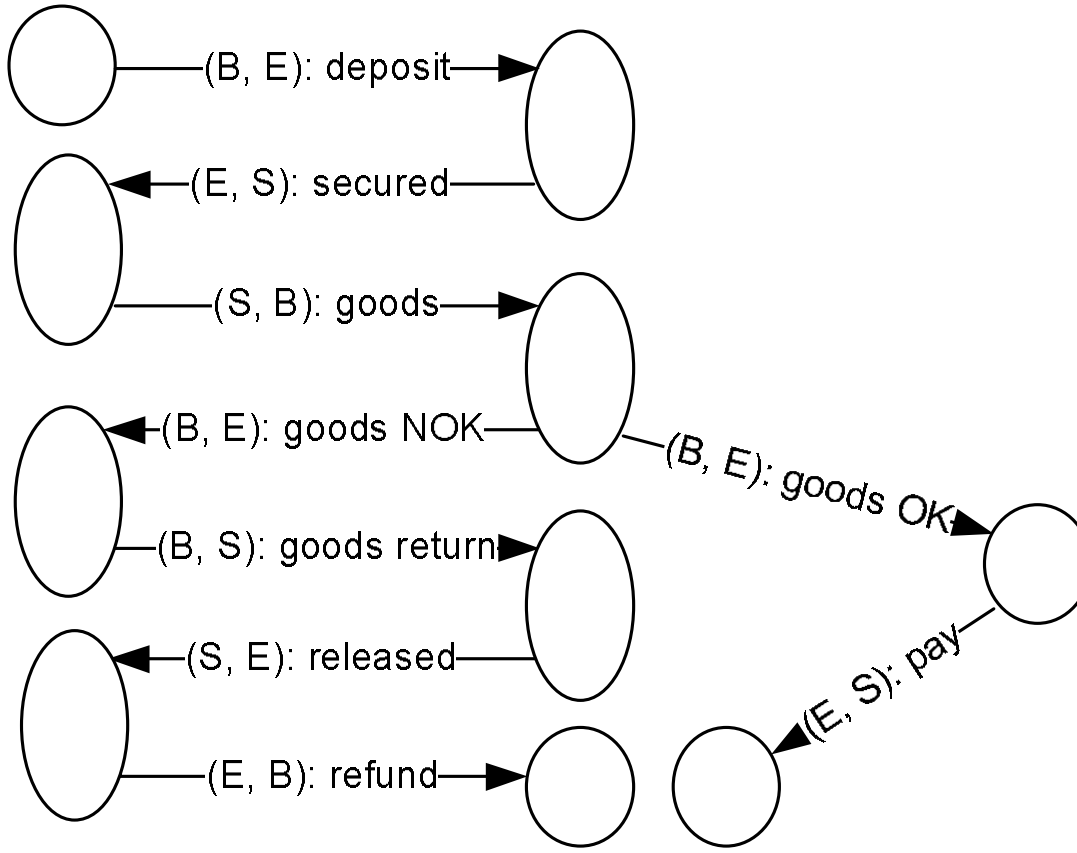


Innovations: 1

- *Protocols:* Conceptually decentralized, reusable, encapsulations of processes
- *Commitments:* Content for protocols
  - Support reuse via abstractions for *refinement* and *aggregation* of protocols
  - What the protocol should accomplish
  - What deviations are legitimate and what aren't
  - Operational semantics for commitments

NetBill and Escrow Protocols



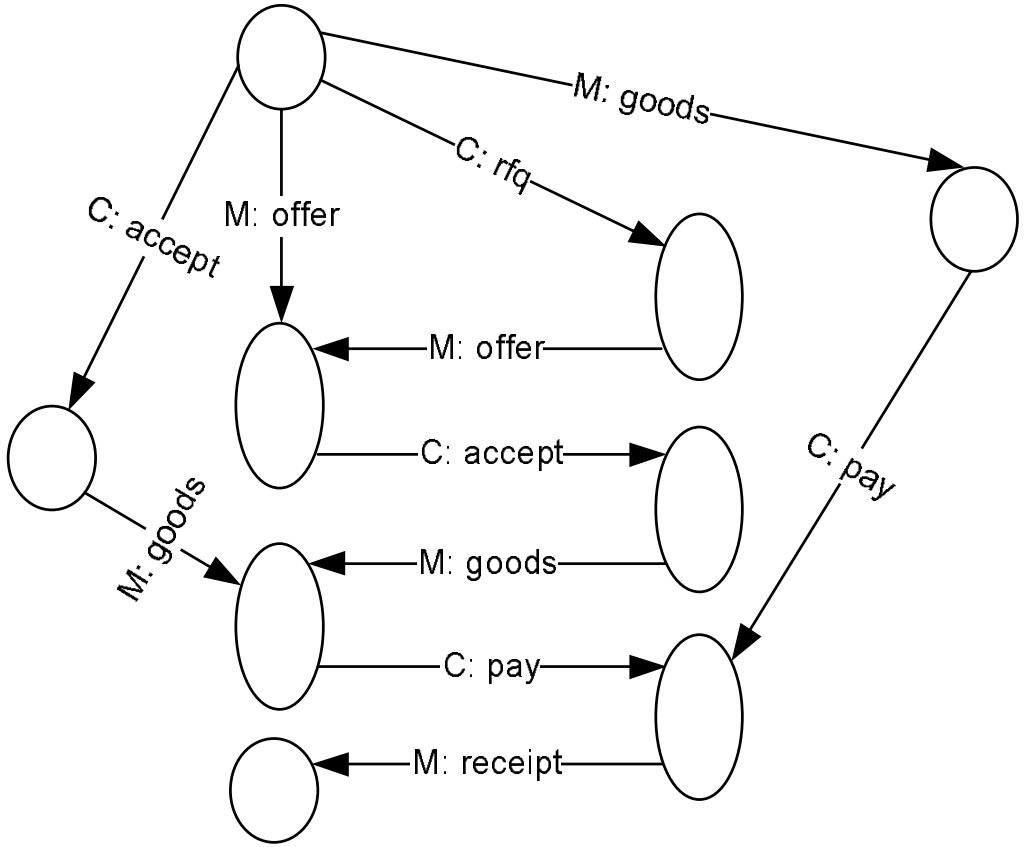


Innovations: 2

- *Rule-Based Reasoning:*
  - Expressing protocols flexibly
  - Accommodating context
  - Deciding specific actions by applying policies
- *Spheres of Commitment:*
  - Modeling organizations
  - Enacting protocols
  - Monitoring and verifying compliance
- Processes $=$ Protocols $+$ Policies

Enhanced NetBill

Compiled from a commitment machine for NetBill



Contributions (In Progress)

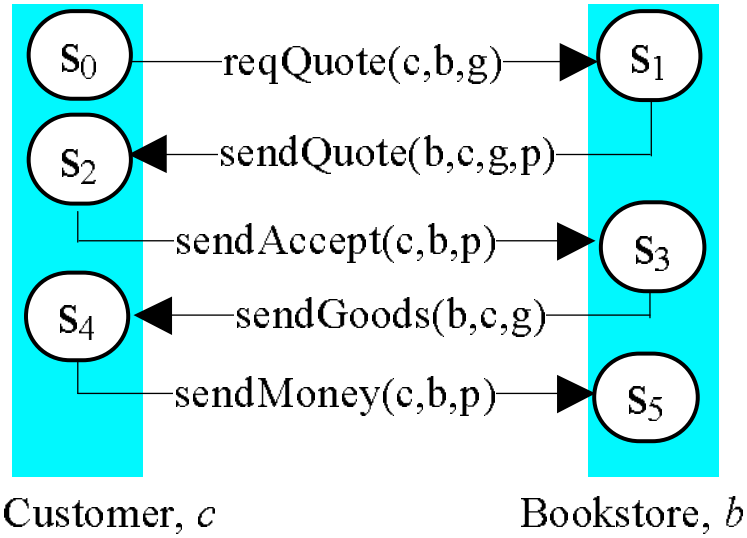
- Specification language for protocols
- Formal semantics based on commitments
- Protocol algebra to support refinement and aggregation
- *Engineering:* not full automation, but tools for
  - Modeling and validation of protocols
  - Modeling and validation of processes
  - Enactment via Spheres of Commitment
  - Monitoring and compliance

Trends and Assessment

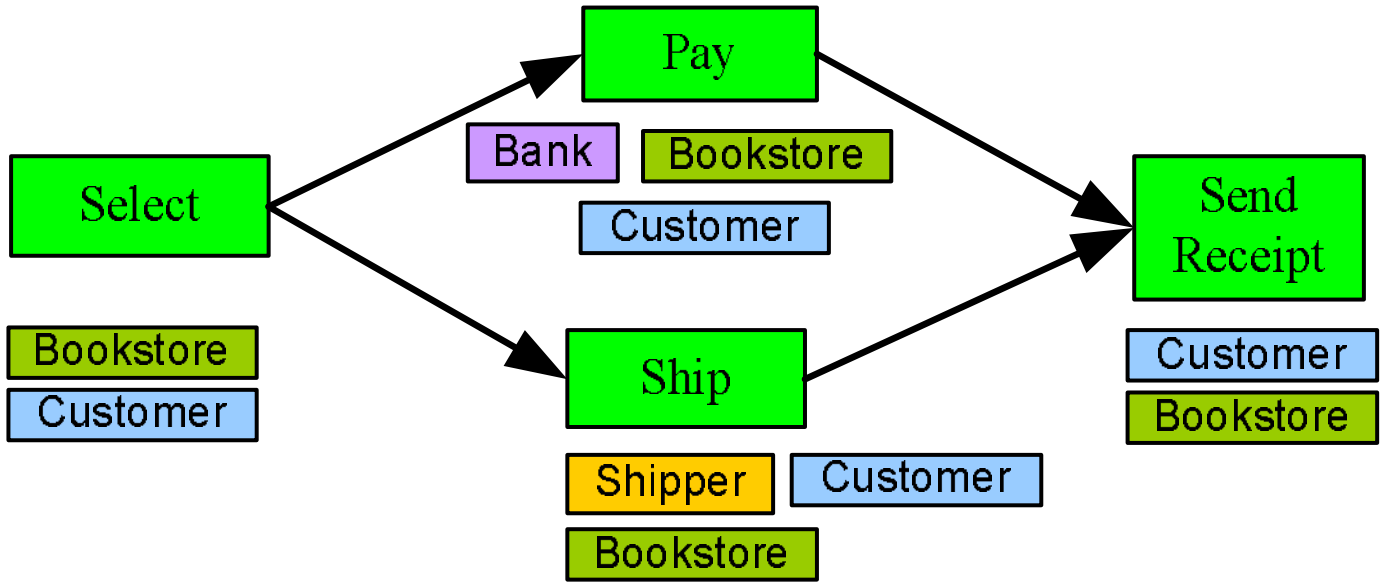
- Increasing # of business protocols
  - IOTP, Escrow, SET, NetBill, . . .
  - RosettaNet: 107 Partner Interface Processes (PIPs)
  - ebXML Business Process Specification Schema (BPSS)
- Intended to be legally binding
- Generally highly limited: two party, request-response protocols
- No commitments;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

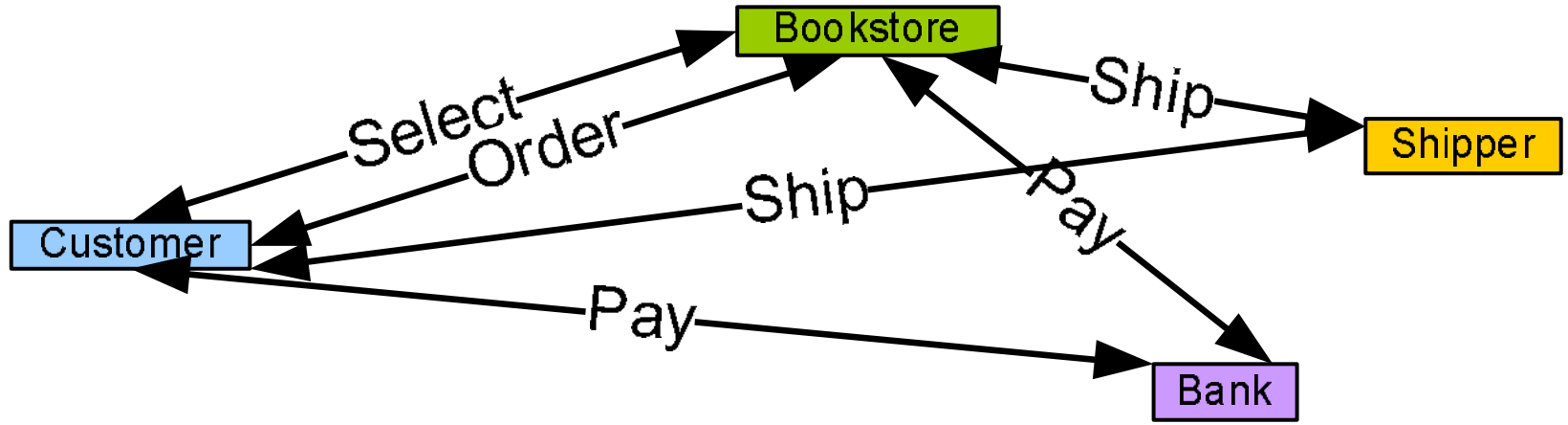
Simple Scenario and Example Run

- 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



Process View: Flow or Protocol





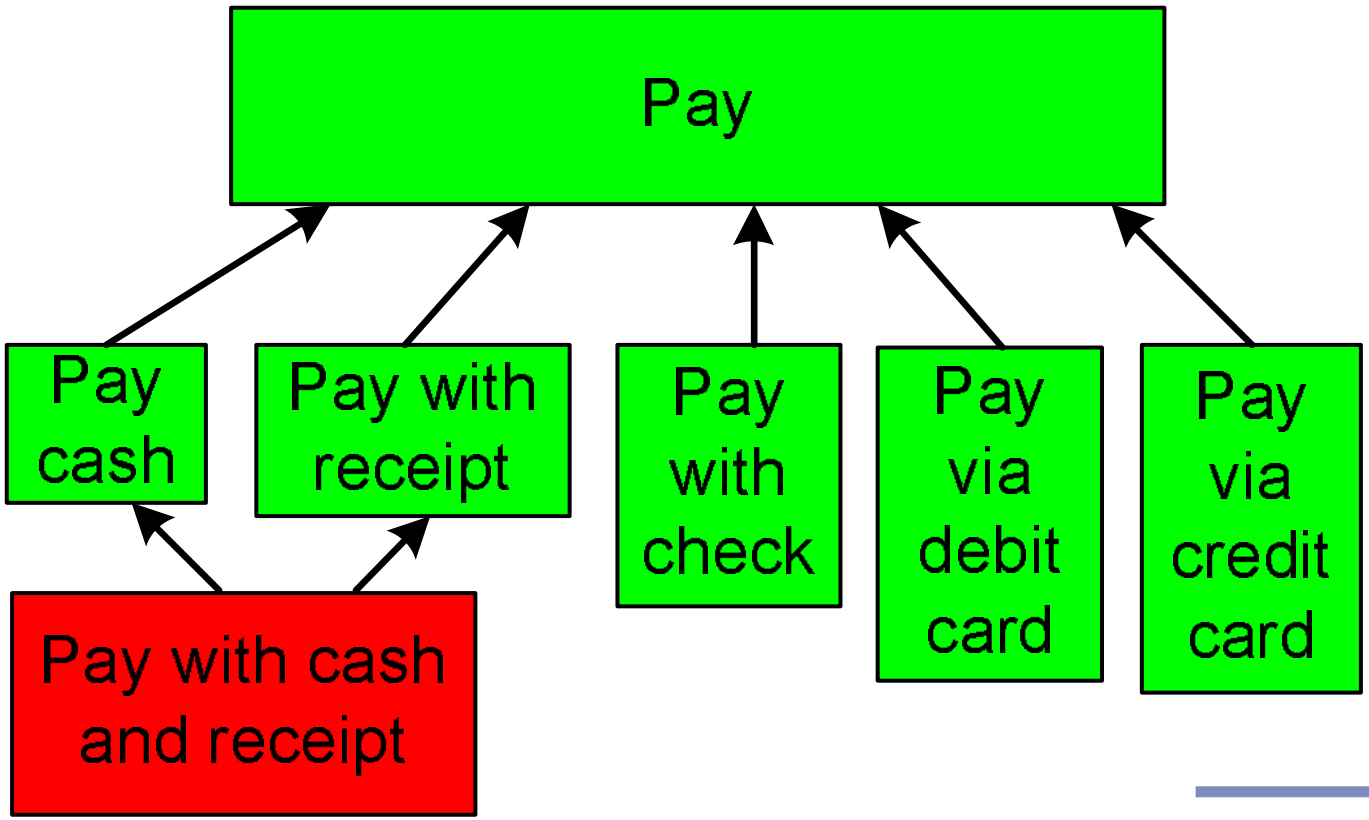
Challenges: Modeling

- *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

Refinement of Protocols

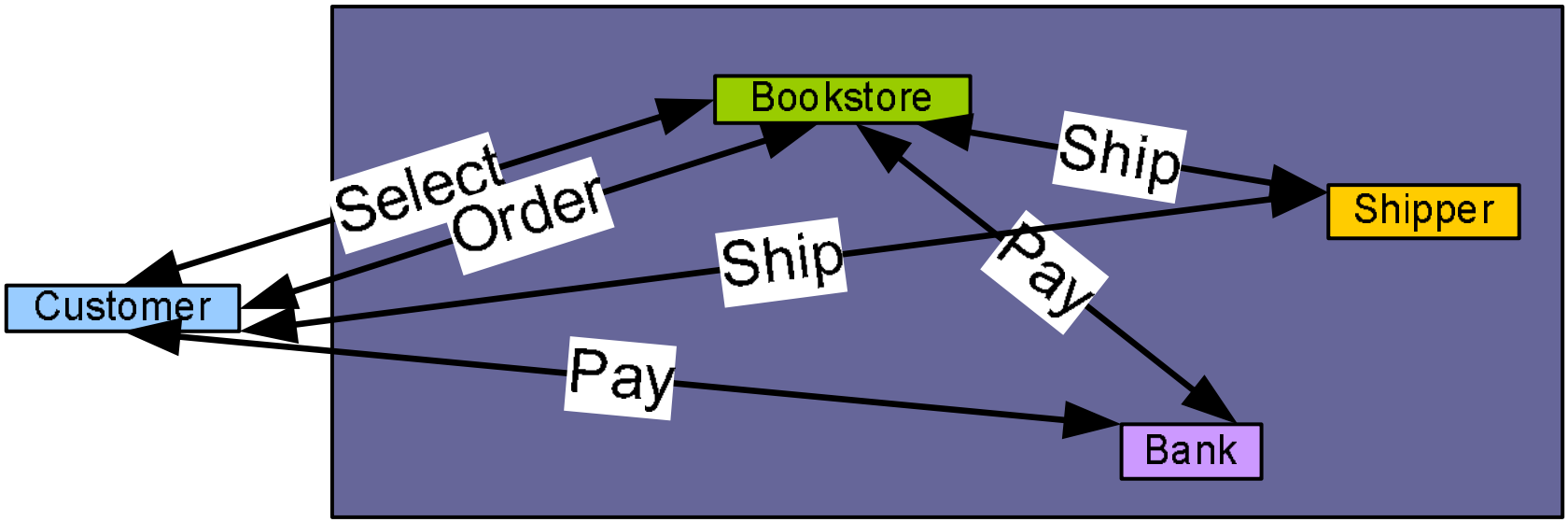
Selection criteria for protocols

- *Functional:* pay versus ship
- *Nonfunctional:* payer trusts payee or not

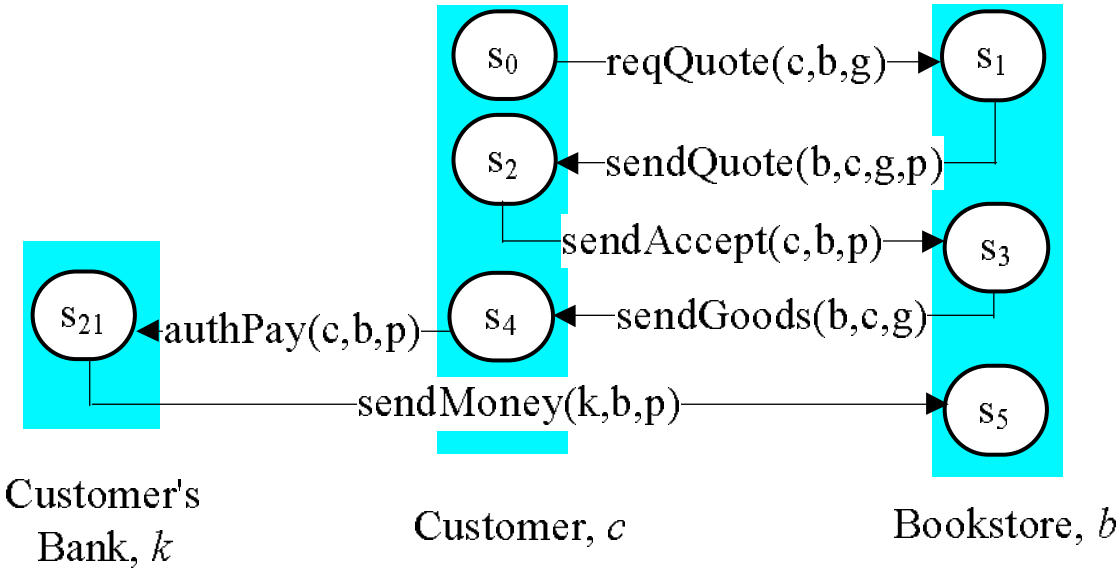


Aggregation of Protocols

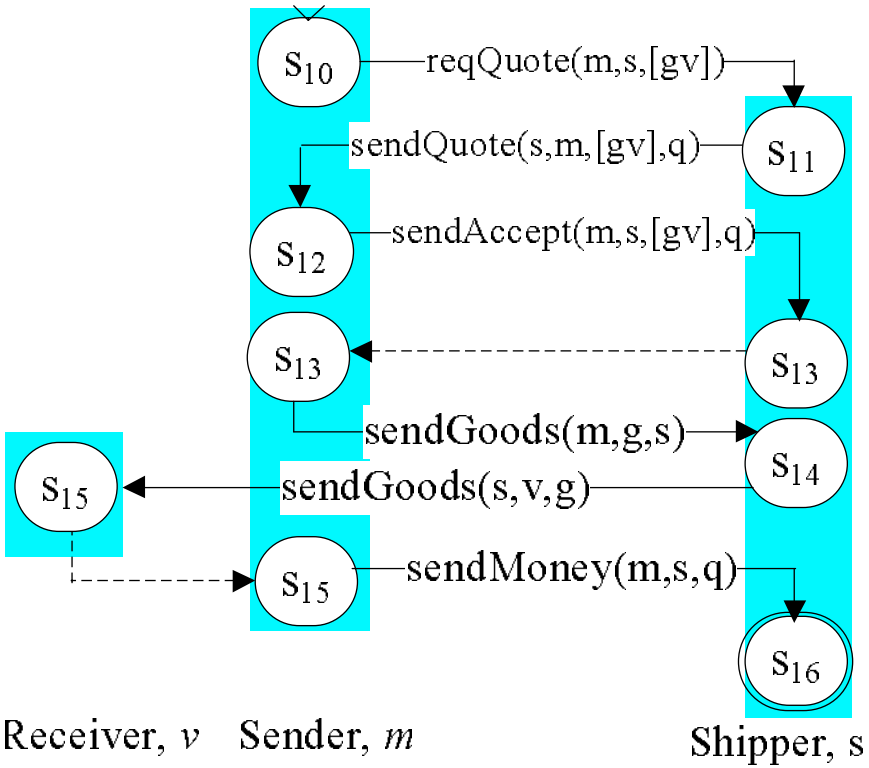
- A simplified protocol may be revealed to a give role
- Decisions could be taken internally but not exposed



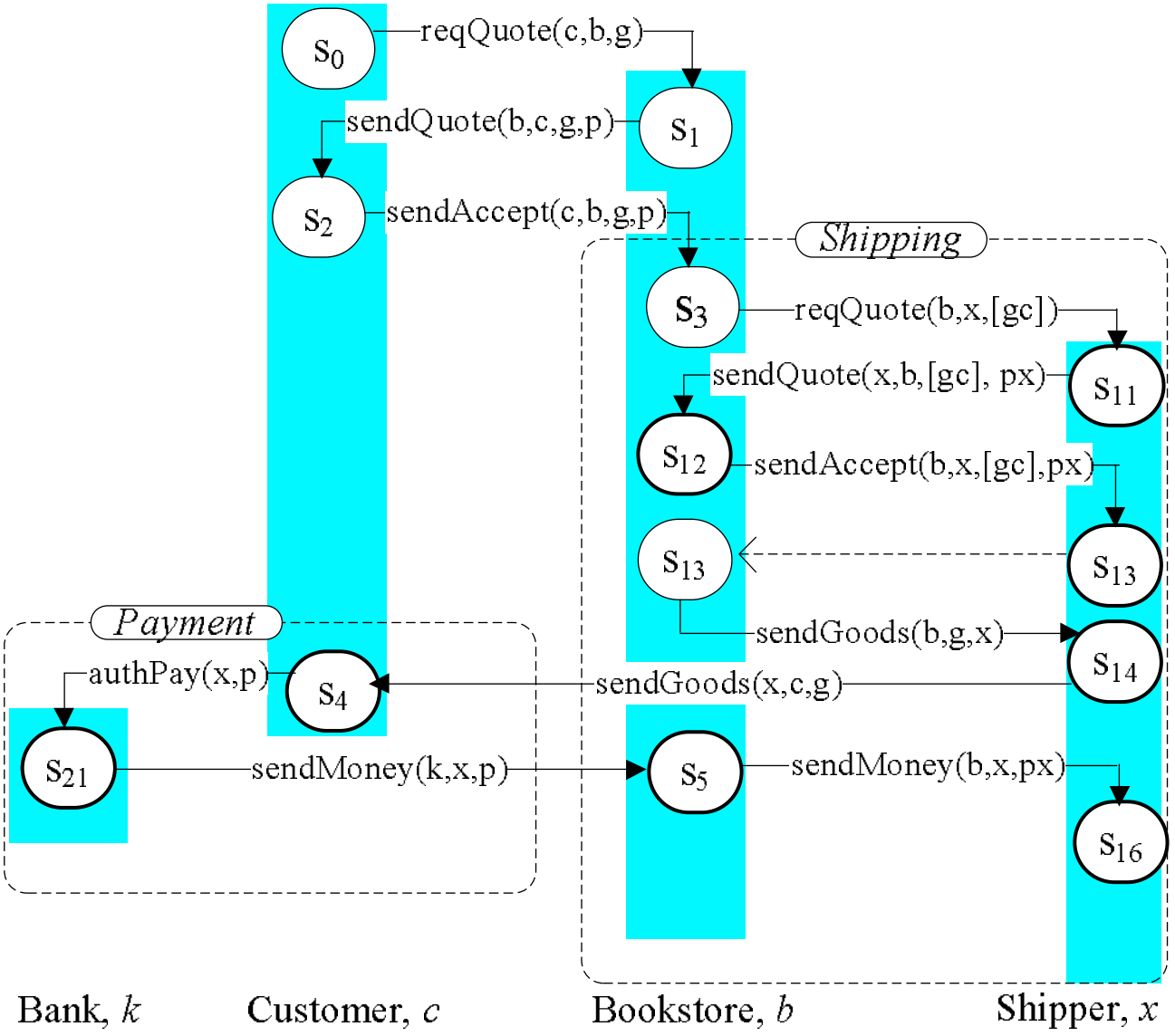
Example Run: Pay via Bank



Example Run: Shipper Protocol



Example Run: Composed Purchase

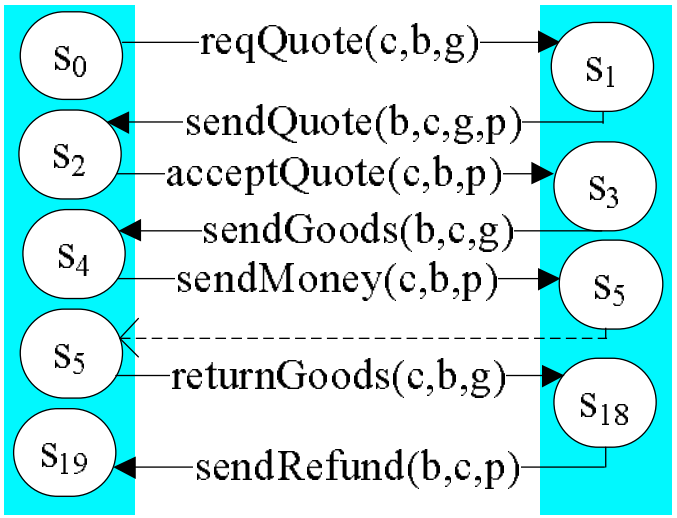


Challenges: Enactment

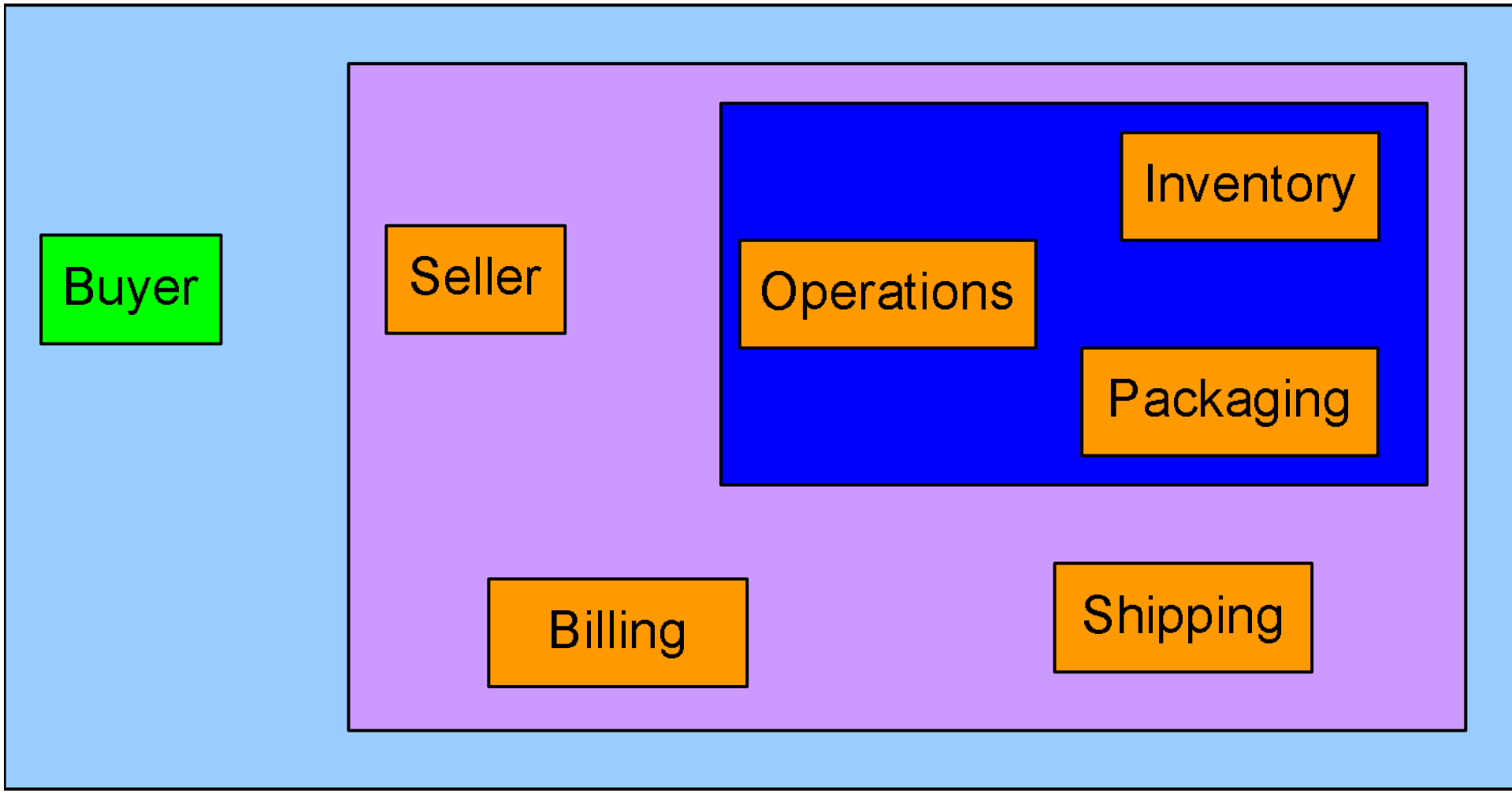
- *Behaving adaptively:* decide dynamically to ship before payment to trusted Cs
- *Handling exceptions*
  - External problems: cannot ship book
  - Context-sensitivity: not legal for kids
  - Detecting violations: no payment; book arrives damaged
  - Correcting violations: remind, complain, refund,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

Example Run: Return and Refund

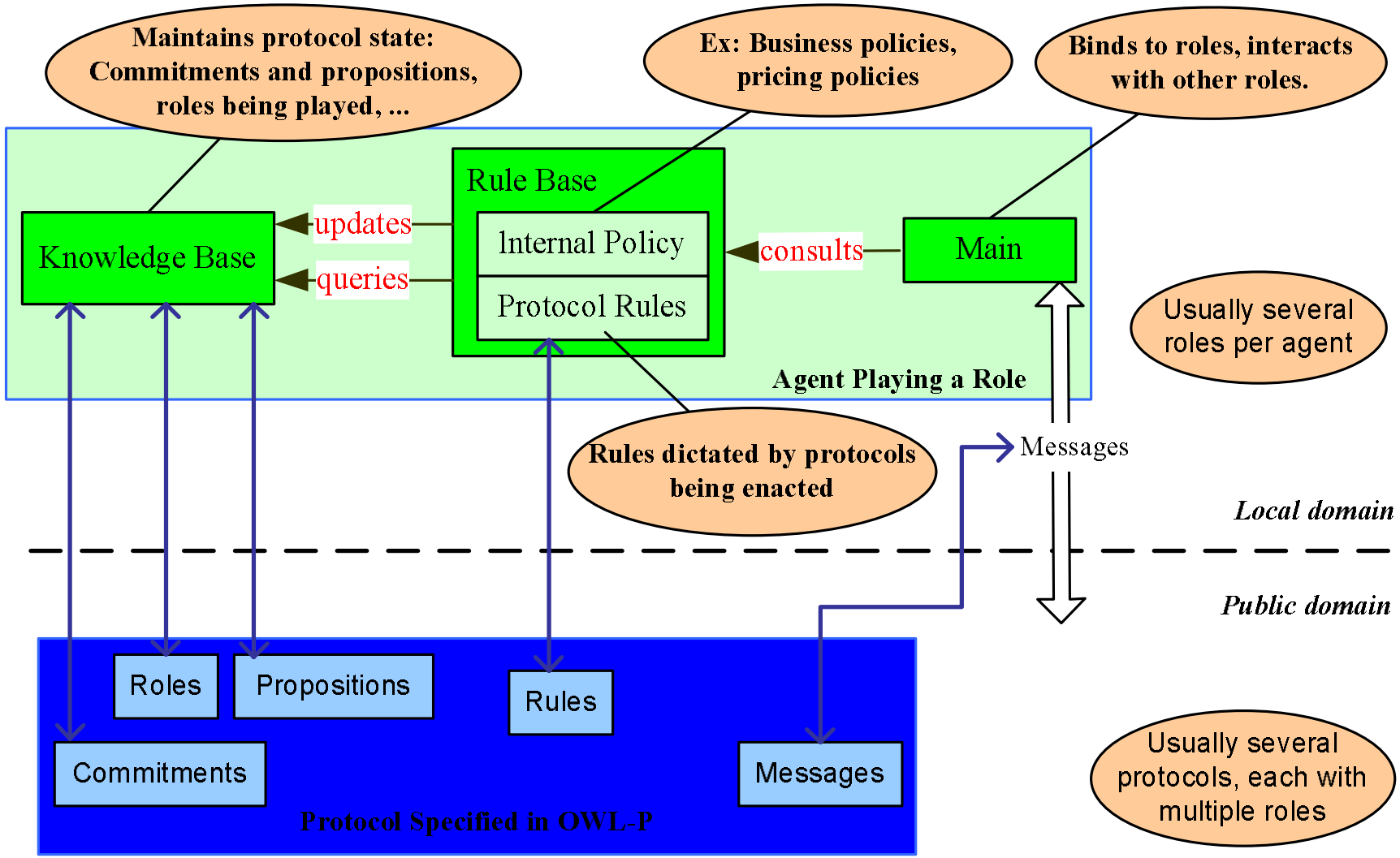
Example: Uniform Commercial Code (UCC) allows returns with refunds for goods that are received damaged



Spheres of Commitment



Architecture



Ongoing Work

- A language, OWL-P, OWL for Protocols
  - Roles
  - Messages: content as propositions and commitments
  - Rules to describe messages and roles
- Tool to generate skeletons from OWL-P
- Operational semantics in π-calculus
- Rule-based policies that help agents satisfy their protocol roles
- Protocol algebra to support refinement and aggregation

Processes = Protocols + Policies

- Operational patterns
  - Time outs, remind, garbage collect, . . .
  - Decisions to manipulate: delegate, assign, . . .
  - Winograd & Flores and other such
- Methodologies, e.g., enhancing Tropos:
  - Cover functional reqs via protocols
  - Refine protocols for nonfunctional reqs
  - Enact protocols dynamically based on agent policies and context

Papers on this Topic

- Newer papers in ICWS, ICSOC, AAMAS address parts of the above vision
- "Agent Communication Languages: Rethinking the Principles." *IEEE Computer*, 31(12):40–47, Dec 1998
- "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. Autonomous Agents & MAS*, 2(3):217–236, Sep 1999
- "An Ontology for Commitments in Multiagent Systems." *AI & Law*, 7:97–113, 1999