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

- Rule-Based Commitment Protocols
- Commitment Protocols
- Protocols
- Dynamic Organizations

Not in 2004
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).
- 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

Customer's Bank, $k$

Customer, $c$

Bookstore, $b$

$s_0$ \(\xrightarrow{\text{reqQuote}(c,b,g)} s_1\)

$s_2$ \(\xrightarrow{\text{sendQuote}(b,c,g,p)} s_3\)

$s_3$ \(\xrightarrow{\text{sendAccept}(c,b,p)} s_4\)

$s_4$ \(\xrightarrow{\text{sendGoods}(b,c,g)} s_5\)

$s_5$ \(\xrightarrow{\text{sendMoney}(k,b,p)} s_6\)

$s_6$ \(\xrightarrow{\text{authPay}(c,b,p)} s_7\)

$\text{Bookstore}$,
Example Run: Shipper Protocol

$S_{10}$
- reqQuote($m, s, [gv]$)

$S_{11}$
- sendQuote($s, m, [gv], q$)
- sendAccept($m, s, [gv], q$)

$S_{12}$
- sendGoods($m, g, s$)
- sendGoods($s, v, g$)

$S_{13}$
- sendMoney($m, s, q$)

$S_{14}$

$S_{15}$

Receiver, $v$
Sender, $m$
Shipper, $s$
Example Run: Composed Purchase

Bank, \( k \)  Customer, \( c \)  Bookstore, \( b \)  Shipper, \( x \)
Challenges: Enactment

- **Behaving adaptively:** decide dynamically to ship before payment to trusted Cs.

- **Handling exceptions.**
  - 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

Maintains protocol state: Commitments and propositions, roles being played, ...

Ex: Business policies, pricing policies

Binds to roles, interacts with other roles.

Usually several roles per agent

Local domain

Public domain

Usually several protocols, each with multiple roles

Protocol Specified in OWL-P

DARPA DAML PI Meeting, May 2004 – p.14/17
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