OWL-P: Processes = Protocols + Policies

Munindar P. Singh

(Students: Amit K. Chopra, Nirmi V. Desai, Ashok U. Mallya)

singh@ncsu.edu

Department of Computer Science
North Carolina State University
http://www.csc.ncsu.edu/faculty/mpsingh/
Problem, Solution, Approach

- **Problem**: Modeling and enacting open business processes
  - Traditional approaches can’t handle autonomy, heterogeneity, dynamism
  - Incorporating context is essential

- **Solution**: Interaction is the key
  - New way of thinking geared toward open systems

- **Approach**: Protocols capture interaction
  - Software engineering: refine, aggregate protocols
  - Agents: flexible enactment
  - Compliance in the face of flexibility
Accomplishments: Intellectual

- Protocols provide interaction-centric modeling, leaving policies to participants
- Commitment semantics yield flexible modeling and enactment
- Theory of protocols supports reusability, refinement, and aggregation of interactions
- Students
  - Three PhD dissertations being supported
  - PhD dissertation involving autonomic service selection based on OWL for QoS being defended on Dec 6.
Accomplishments: Implementational

On SemWebCentral

- OWL-P as an OWL ontology (using SWRL)
  - Roles
  - Messages: content as propositions and commitments
  - Rules to describe messages and roles
- Protégé plugin for OWL-P protocol editor
- Protocol composer and skeleton generator
- Agent-based architecture layered on FIPA
- Rule-based policies that help agents satisfy their protocol roles
Accomplishments: Evangelical

- Papers, tutorials, panels, invited talks
- Trying to reach the software engineering community: well-received at OOPSLA
- Contact with IBM and HP
- Beginning project jointly with IBM on autonomic and agent based business process management
- IEEE Internet Computing track (2005) on *Service-Oriented Computing*
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
  - Autonomy: components have a life of their own
  - 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

- Dynamic Organizations
  - Rule-Based Commitment Protocols: Flexibility
  - Commitment Protocols: Content & Compliance
  - Protocols: Modularity

Monitoring and compliance

Implementation and enactment

Modeling and validation
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
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

- **Slogan:** Processes = Protocols + Policies
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
Vision

■ **Target Audience:** Practitioners

■ Formalization in the background

■ **Engineering:** not full automation, but tools for
  ■ Modeling and validation of protocols
  ■ Modeling and validation of processes
  ■ Generation of software components
  ■ Enactment via Spheres of Commitment
  ■ Monitoring and compliance
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

Diagram:

1. Select
   - Customer
   - Bookstore

2. Pay
   - Bank
   - Bookstore
   - Customer

3. Ship
   - Customer
   - Bookstore
   - Shipper

4. Send Receipt
   - Customer
   - Bookstore

5. Pay
   - Bank

© Singh et al., November 2004 – p.15/2
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

Diagram:

- Pay
  - Pay cash
  - Pay with receipt
  - Pay with check
  - Pay via debit card
  - Pay via credit card
  - Pay with cash and receipt
Aggregation of Protocols

- A simplified protocol may be revealed to a given role
- Decisions could be taken internally but not exposed
Example Run: Pay via Bank

Customer's Bank, k

Customer, c

Bookstore, b

- authPay(c, b, p)
- sendMoney(k, b, p)
- sendAccept(c, b, p)
- sendGoods(b, c, g)
- sendQuote(b, c, g, p)
- reqQuote(c, b, g)

Singh et al., November 2004 – p.19/2
Example Run: Shipper Protocol

1. **Shipper, s**
   - reqQuote(m, s, [gv])
   - sendQuote(s, m, [gv], q)
   - sendAccept(m, s, [gv], q)
   - sendGoods(m, g, s)
   - sendGoods(s, v, g)
   - sendMoney(m, s, q)

2. **Receiver, v**
   - sendGoods(s, v, g)

3. **Sender, m**
   - sendQuote(s, m, [gv], q)
   - sendAccept(m, s, [gv], q)

© Singh et al., November 2004 – p.20/29
Example Run: Composed Purchase

- reqQuote(c,b,g)
- sendQuote(b,c,g,p)
- sendAccept(c,b,g,p)
- sendQuote(x,b,[gc], px)
- sendAccept(b,x,[gc], px)
- sendGoods(b,g,x)
- sendGoods(x,c,g)
- sendMoney(k,x,p)
- sendMoney(b,x,px)

Bank, k  Customer, c  Bookstore, b  Shipper, x
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.
Processes = Protocols + Policies

- **Operational patterns**
  - Time outs, remind, garbage collect, . . .
  - Decisions to manipulate: delegate, assign, . . .
  - Enact protocols dynamically based on agent policies and context

- **Transactional patterns**
  - Induce transactional scopes
  - Apply retry, redo, undo (compensate) where appropriate
  - Enact via Spheres of Commitment
## Remaining Work: Easy

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplify protocol design for business</td>
<td>Libraries of composable protocols</td>
</tr>
<tr>
<td>Produce compliant agents</td>
<td>Refined methodology based on policies</td>
</tr>
<tr>
<td>Make up to date with Rules work and OWL-S</td>
<td>Freshen the work</td>
</tr>
</tbody>
</table>
### Remaining Work: Middlish

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formalize context</td>
<td>Develop operational semantics in $\pi$-calculus</td>
</tr>
<tr>
<td>Protocol compliance</td>
<td>Apply commitment semantics</td>
</tr>
<tr>
<td>Organizational, transactional exception modeling</td>
<td>Exploit Spheres of Commitment</td>
</tr>
<tr>
<td>Optimize role selection based on QoS</td>
<td>Incorporate service quality representations</td>
</tr>
</tbody>
</table>
Papers on this Topic

- Recent papers in ICWS, AAMAS, OOPSLA, ICSOC address parts of the above vision
- Tutorials at WWW, AAMAS, OOPSLA
- Panels at WWW, AAMAS, ICWS
Spheres of Commitment

Buyer

Seller

Operations

Inventory

Packaging

Billing

Shipping

© Singh et al., November 2004 – p.28/29
Contexts as Transformers