OWL-P: Processes = Protocols + Policies

Munindar P. Singh, Amit K. Chopra, Nirmit Desai, Ashok U. Mallya
{singh, akchopra, nvdesai, aumallya}@ncsu.edu
North Carolina State University, Raleigh, NC.
http://research.csc.ncsu.edu/mas/
OWL-P Composite Protocol Specification
Process Development Methodology

1. Protocol Repository
   - get Purchase Customer skeleton
   - register Purchase protocol
   - lookup Purchase protocol

2. Software Designer
   - specify

3. Axioms
   - Composer tool
   - Purchase OWL-P

4. Order OWL-P
   - Shipping OWL-P
   - Payment OWL-P

5. register Purchase protocol

6. MERCHANT
   - Merchant Skeleton
   - Local Policy

7. register

8. search for Merchant

9. discover Merchant port

10. Merchant Local Process
    - Merchant Skeleton
    - Local Policy

11. Customer Local Process
    - Customer Skeleton
    - Customer OWL-P
Examples

1. Examples of Order, Shipping and Payment Protocols

2. Examples of composition axioms to compose them into a Purchase composite protocol

3. A composer tool processes these axioms and generates the composite protocol

4. Available at:
   http://biber.csc.ncsu.edu/owl.html
ORDER PROTOCOL

Customer

reqForQuote(itemID)

Merchant

quote(itemID, itemPrice)

CC(m, c, pay(itemPrice), goods(itemID))

acceptQuote(itemID, itemPrice)

CC(c, m, goods(itemID), pay(itemPrice))
Example

1. contains(KB, startProp) AND rfqPolicy(?itemID) ➔
2. send(C, requestForQuote(?itemID))

3. contains(KB, reqForQuoteProp(?itemID)) AND quotePolicy(?itemID, ?itemPrice) ➔
4. send(M, quote(?itemID, ?itemPrice)) AND
5. createCommitment(M, CC(M, C, pay(?itemPrice), goods(?itemID)))

6. contains(KB, quoteProp(?itemID, ?itemPrice)) AND acceptPolicy(?itemID, ?itemPrice) ➔
7. send(C, acceptQuote(?itemID, ?itemPrice)) AND
8. createCommitment(C, CC(C, M, goods(?itemID), pay(?itemPrice)))
PAYMENT PROTOCOL

**Payer**
- paymentInfo(cardNO, expDate)
- CC(payer, payee, authNOKProp(cardNO, expDate, *amount*), pay(fineAmount))
- receipt(amount)

**Gateway**
- authReq(cardNO, expDate, amount)
- authOK(cardNO, expDate, amount, tokenNO)
- CC(gateway, payee, captureReqProp(tokenNO), capturedProp(amount))
- captureReq(token)
- captured(amount)

**Payee**
- authOK(cardNO, expDate, amount, tokenNO)
- CC(gateway, payee, captureReqProp(tokenNO), capturedProp(amount))
- captureReq(token)
- captured(amount)
COMPOSITION AXIOMS

1: roleDefinition(define: Purchase.customer, unify: Order.customer, unify: Shipping.receiver, unify: Payment.payer)

2: roleDefinition(define: Purchase.merchant, unify: Order.merchant, unify: Shipping.sender, unify: Payment.payee)

3: roleDefinition(define: Purchase.gateway, unify: Payment.gateway)

4: roleDefinition(define: Purchase.shipper, unify: Shipping.shipper)

5: dataFlow(definition: Order.itemId, usage: Shipping.item)

6: dataFlow(definition: Order.itemPrice, usage: Payment.amount)

7: implication(body: Shipping.shipmentProp, head: Order.goods)

8: implication(body: Payment.authOKProp, head: Order.pay)

9: eventOrder(earlier: Payment.authOKProp, later: Shipping.shipOrderProp)
## Contributions

<table>
<thead>
<tr>
<th>Intellectual</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Interaction centric modeling is a global spec; models open systems better</td>
<td>1. OWL-P Protocol viewer and editor as a Protégé plugin</td>
</tr>
<tr>
<td>2. Commitment semantics allow flexible interactions, as in the real-world</td>
<td>2. Protocol composer tool</td>
</tr>
<tr>
<td>3. Theory of protocols allows reusability, refinement, and aggregation of</td>
<td>3. A library for software designers (in future)</td>
</tr>
<tr>
<td>interactions</td>
<td></td>
</tr>
</tbody>
</table>
