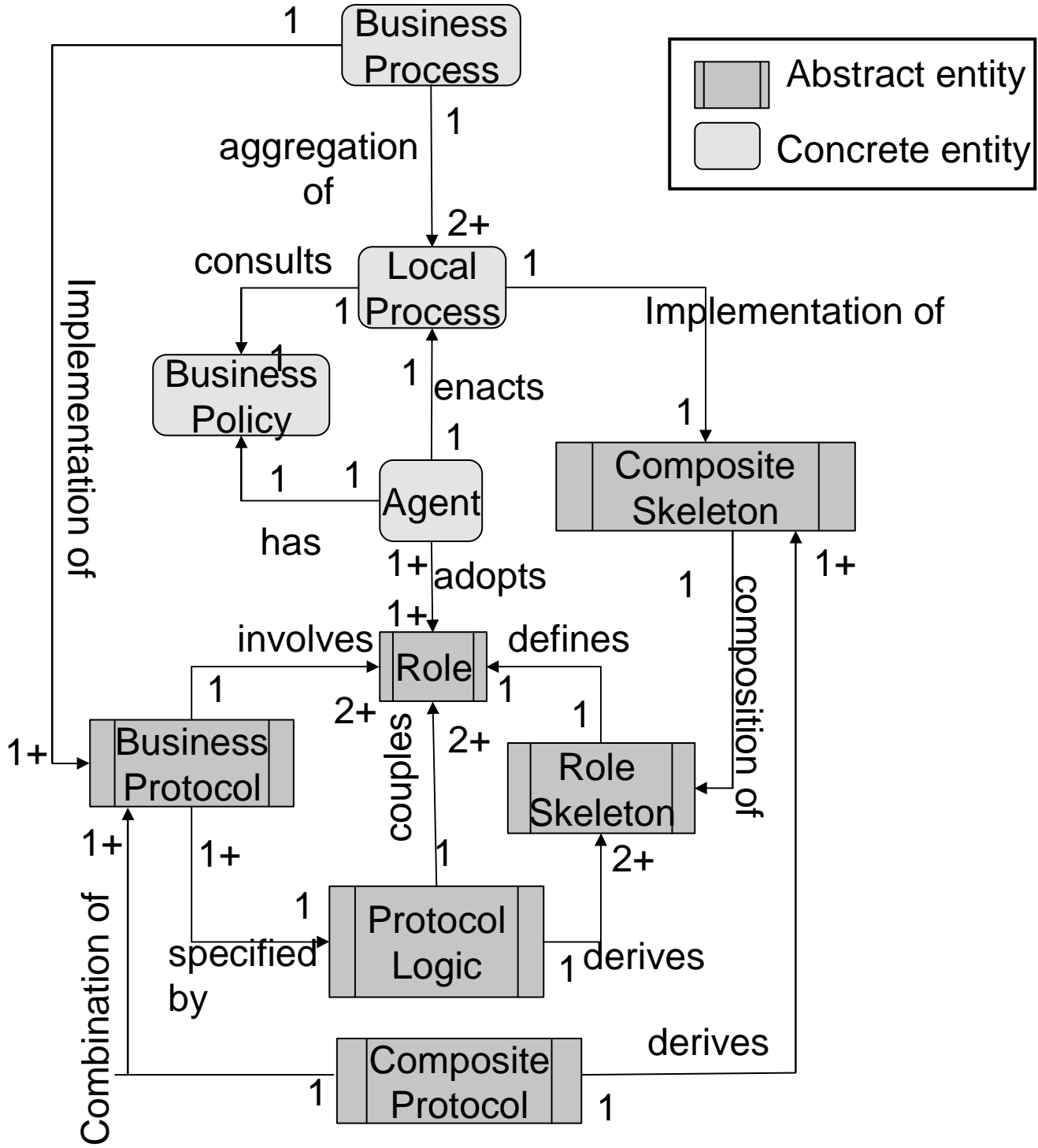


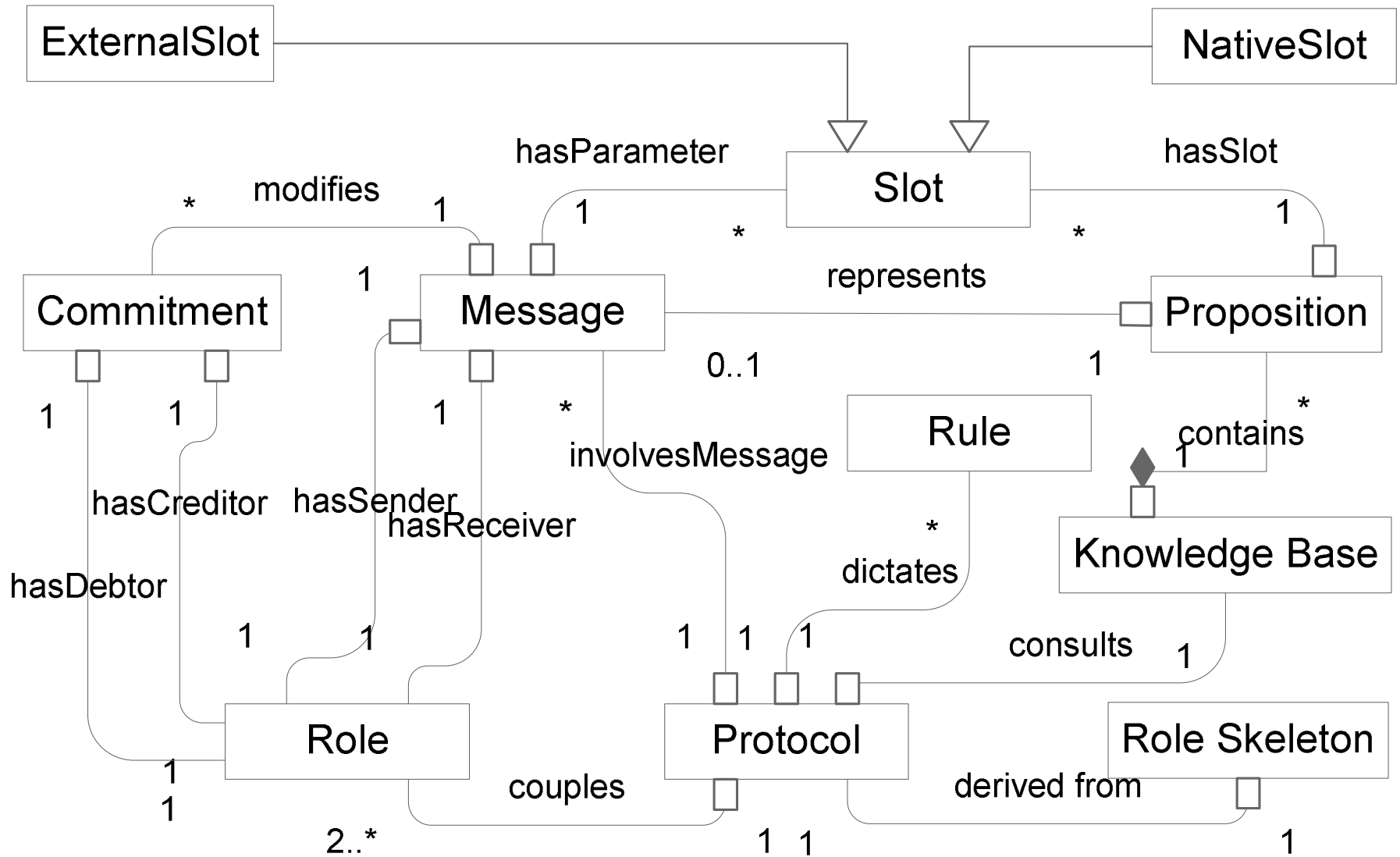
OWL-P: Processes = Protocols + Policies

Munindar P. Singh, Amit K. Chopra, Nirmal Desai, Ashok U. Mallya
{singh, akchopra, nvdesai, aumallya}@ncsu.edu
North Carolina State University, Raleigh, NC.
<http://research.csc.ncsu.edu/mas/>

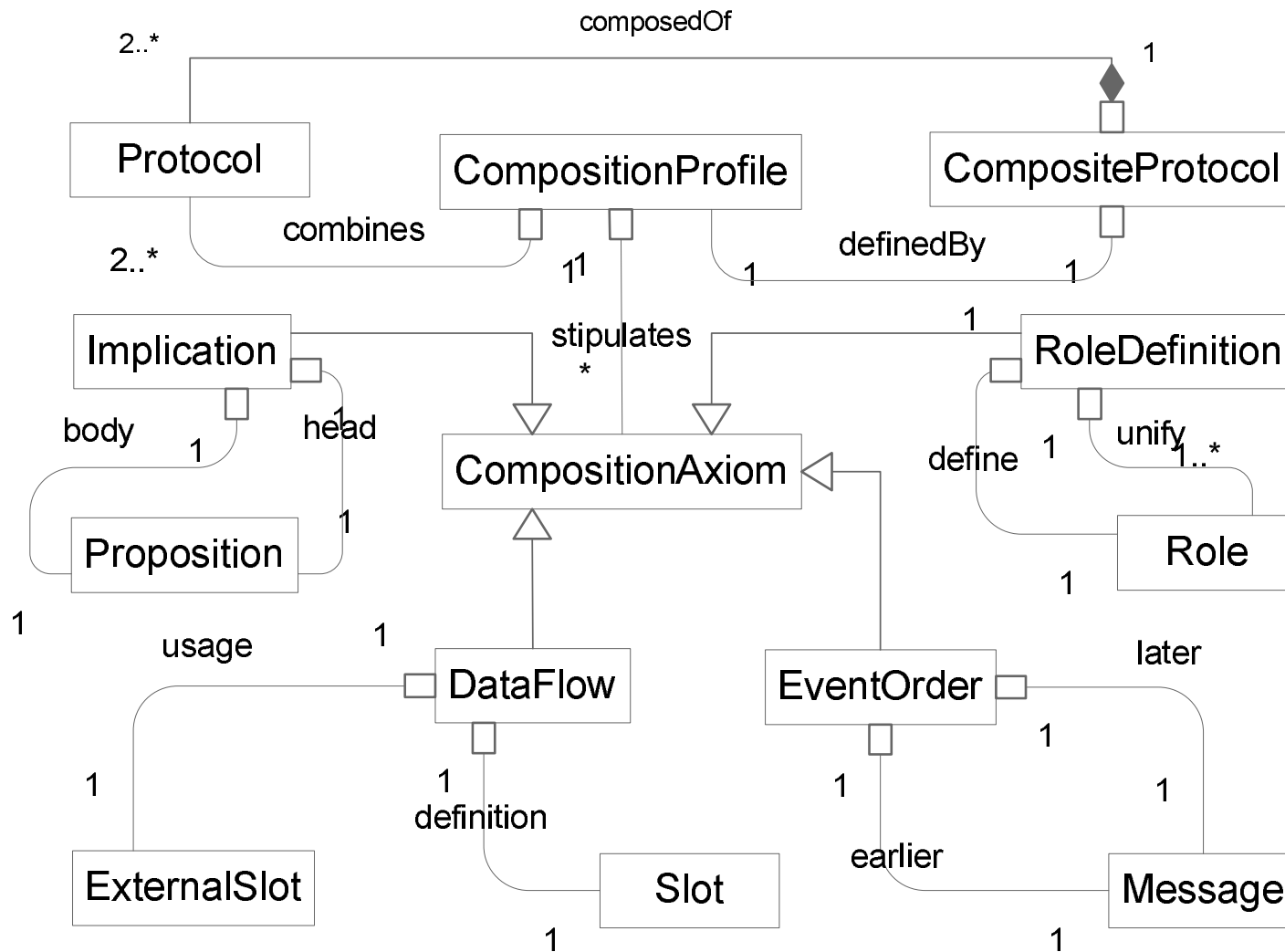
Concepts



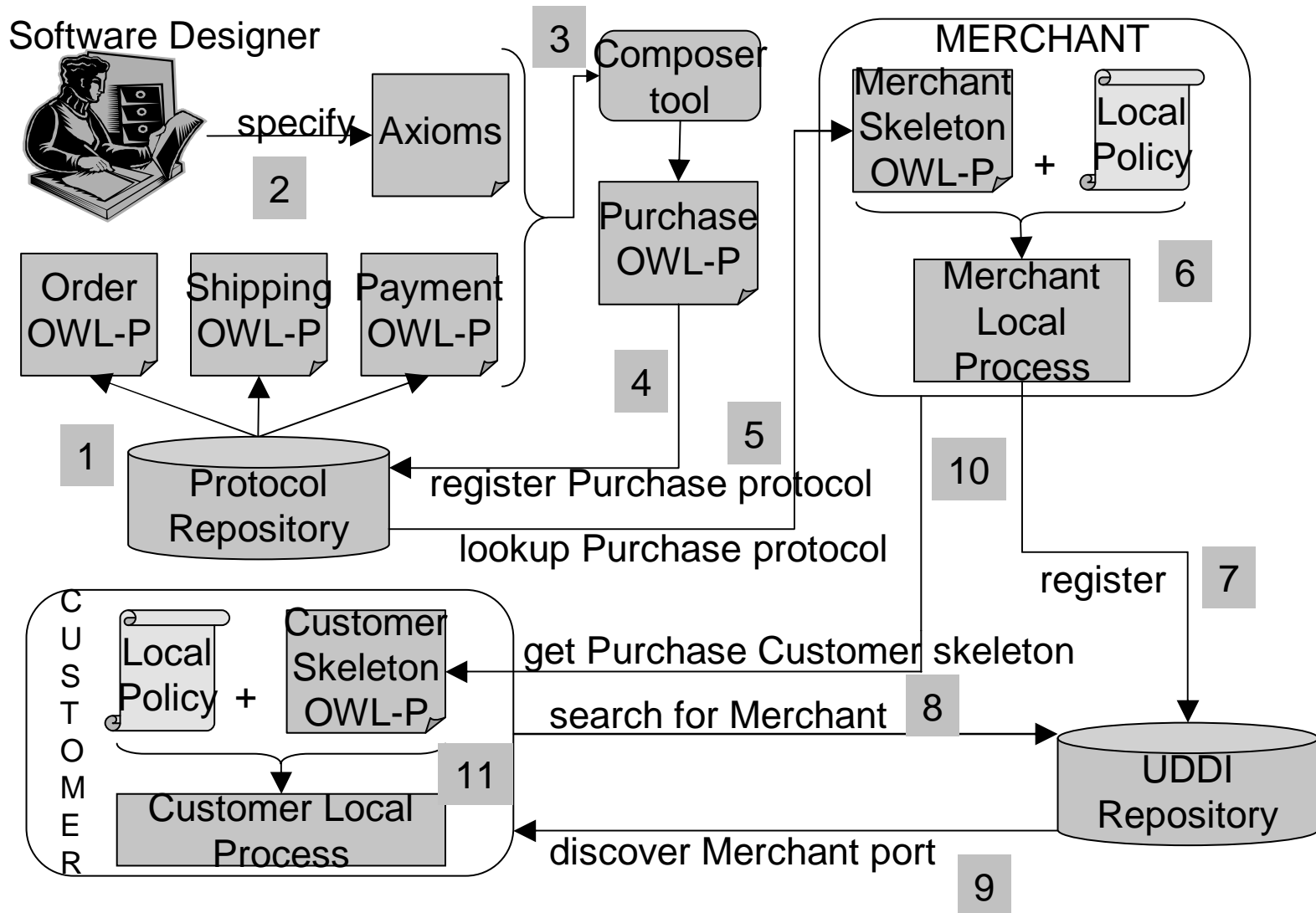
OWL-P Protocol Specification



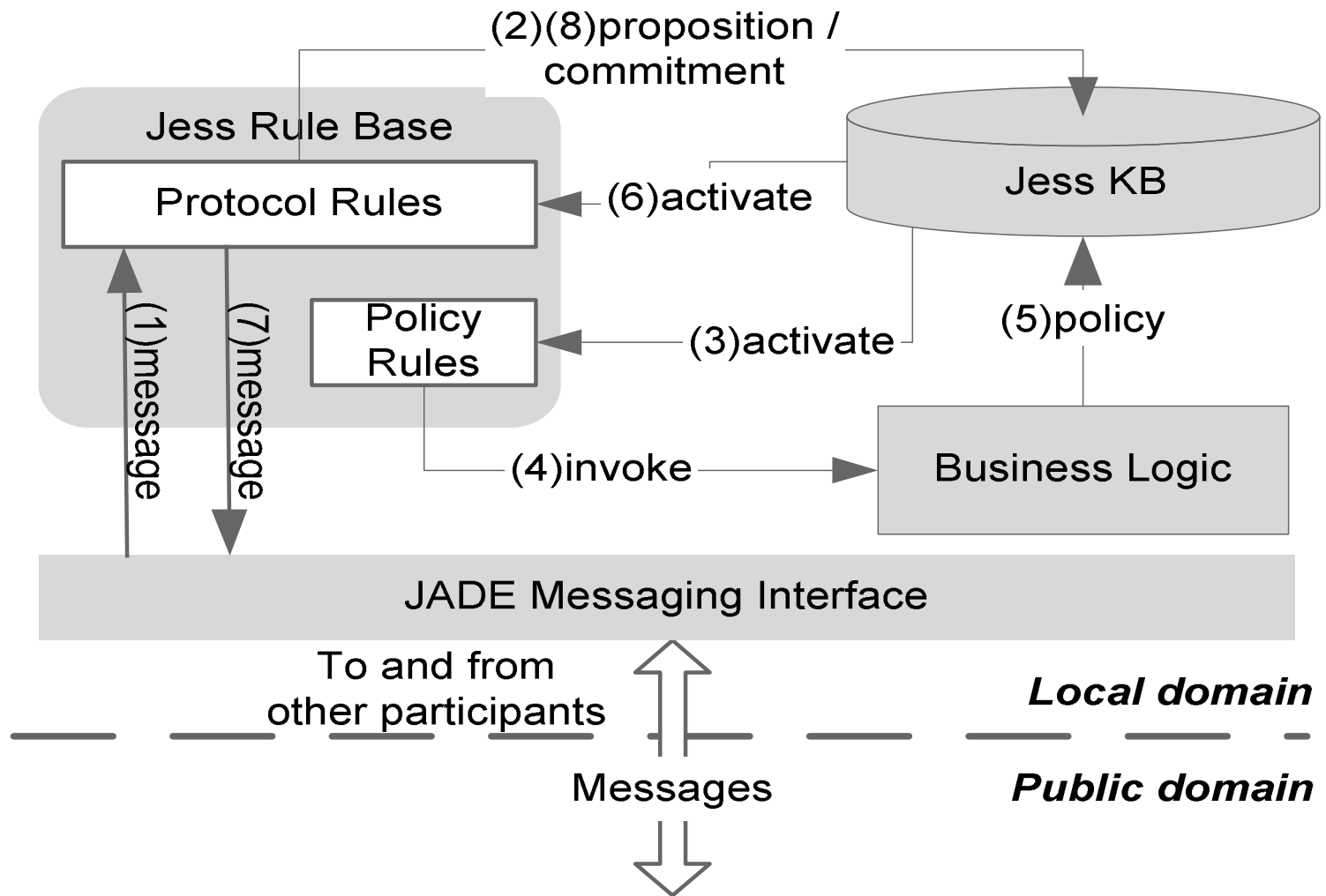
OWL-P Composite Protocol Specification



Process Development Methodology



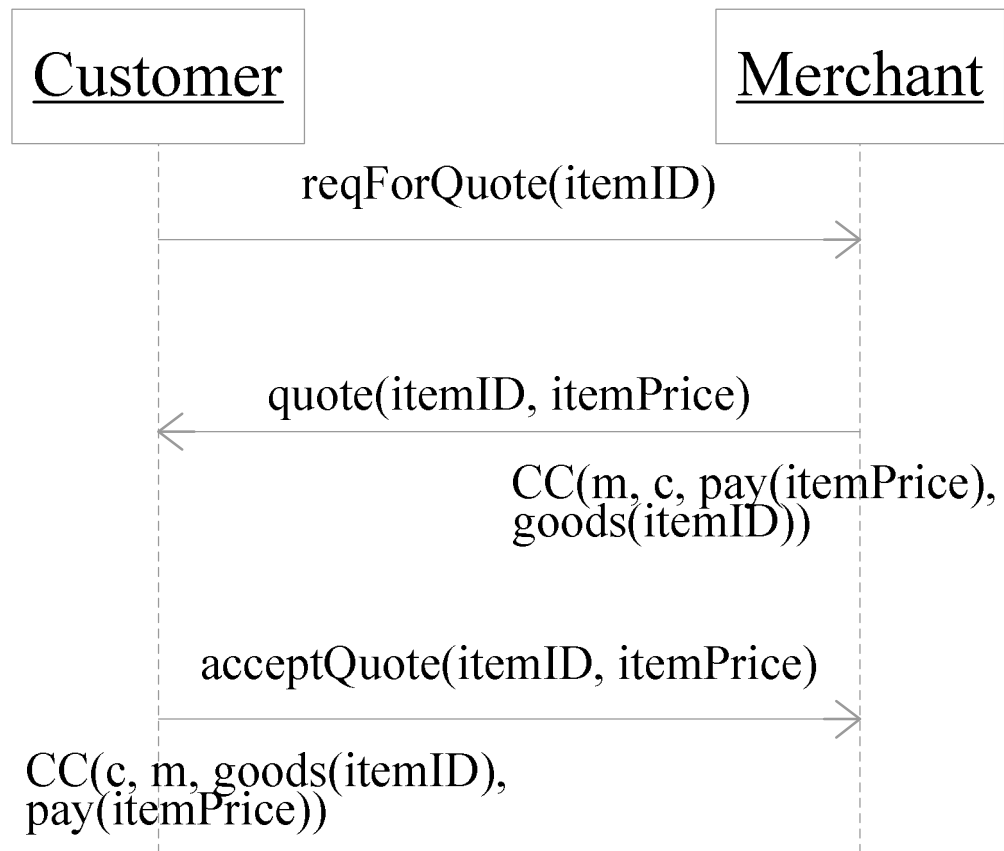
Agent Architecture



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



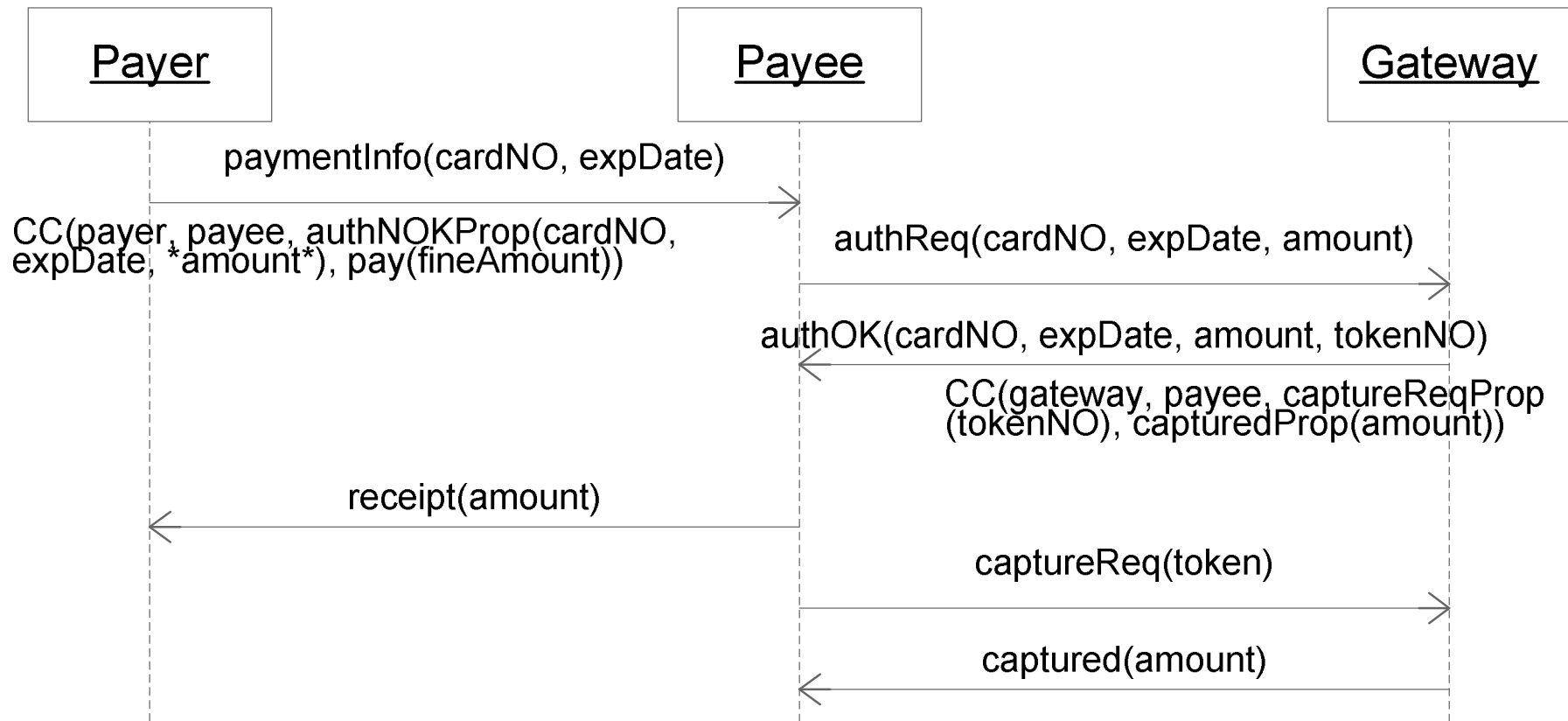
Example

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

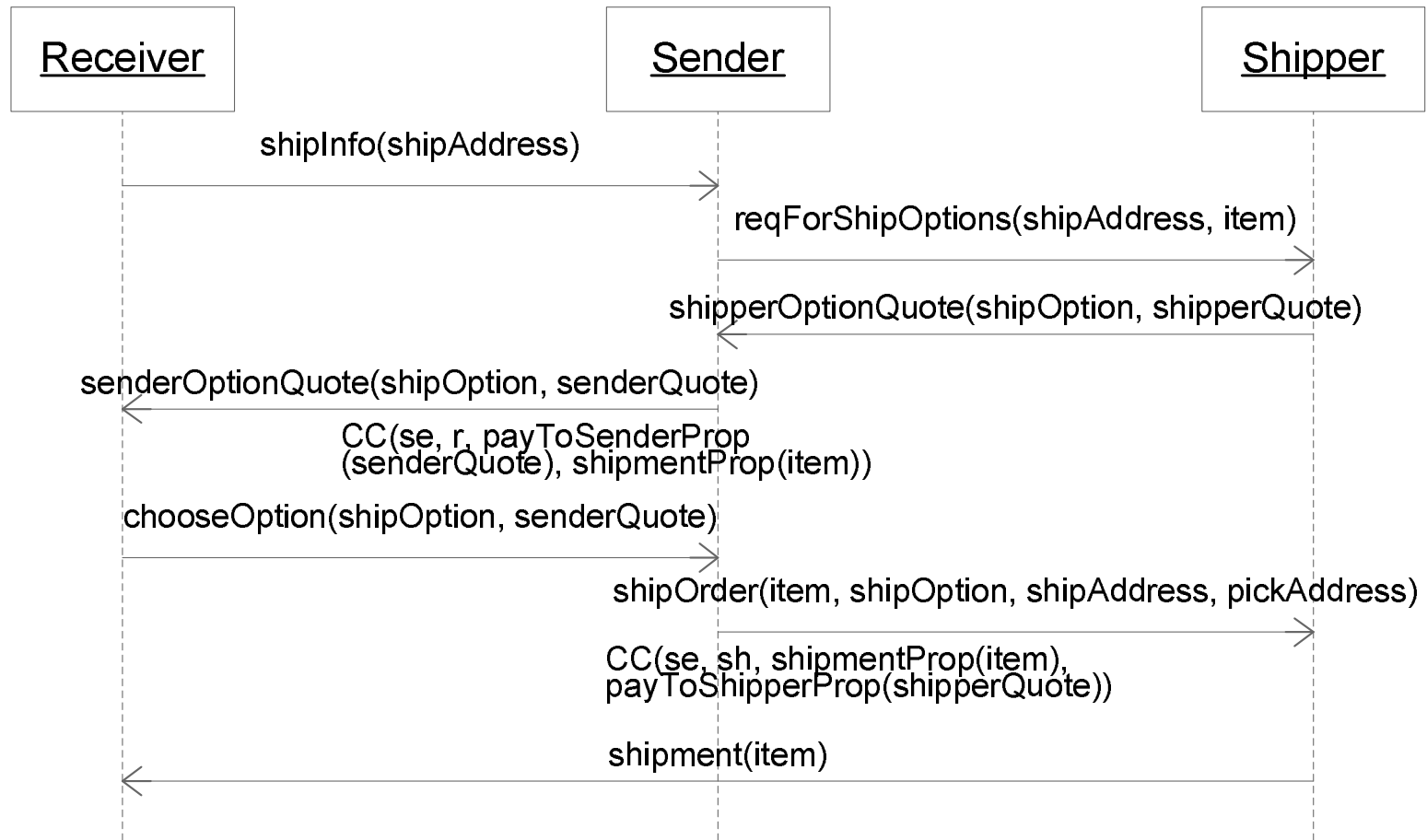
3. contains(KB, reqForQuoteProp(?itemID)) AND
quotePolicy(?itemID, ?itemPrice) →
4. send(M, quote(?itemID, ?itemPrice)) AND
5. createCommitment(M, CC(M, C, pay(?itemPrice),
goods(?itemID)))

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



SHIPPING PROTOCOL



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

Intellectual	Software
<ol style="list-style-type: none">1. Interaction centric modeling is a global spec; models open systems better2. Commitment semantics allow flexible interactions, as in the real-world3. Theory of protocols allows reusability, refinement, and aggregation of interactions	<ol style="list-style-type: none">1. OWL-P Protocol viewer and editor as a Protégé plugin2. Protocol composer tool3. A library for software designers (in future)

Publication

1. Munindar P. Singh, Amit K. Chopra, Nirmitt V. Desai, and Ashok U. Mallya. “Protocols for Processes: Programming in the Large for Open Systems”. In OOPSLA (*Onward!*) 2004.