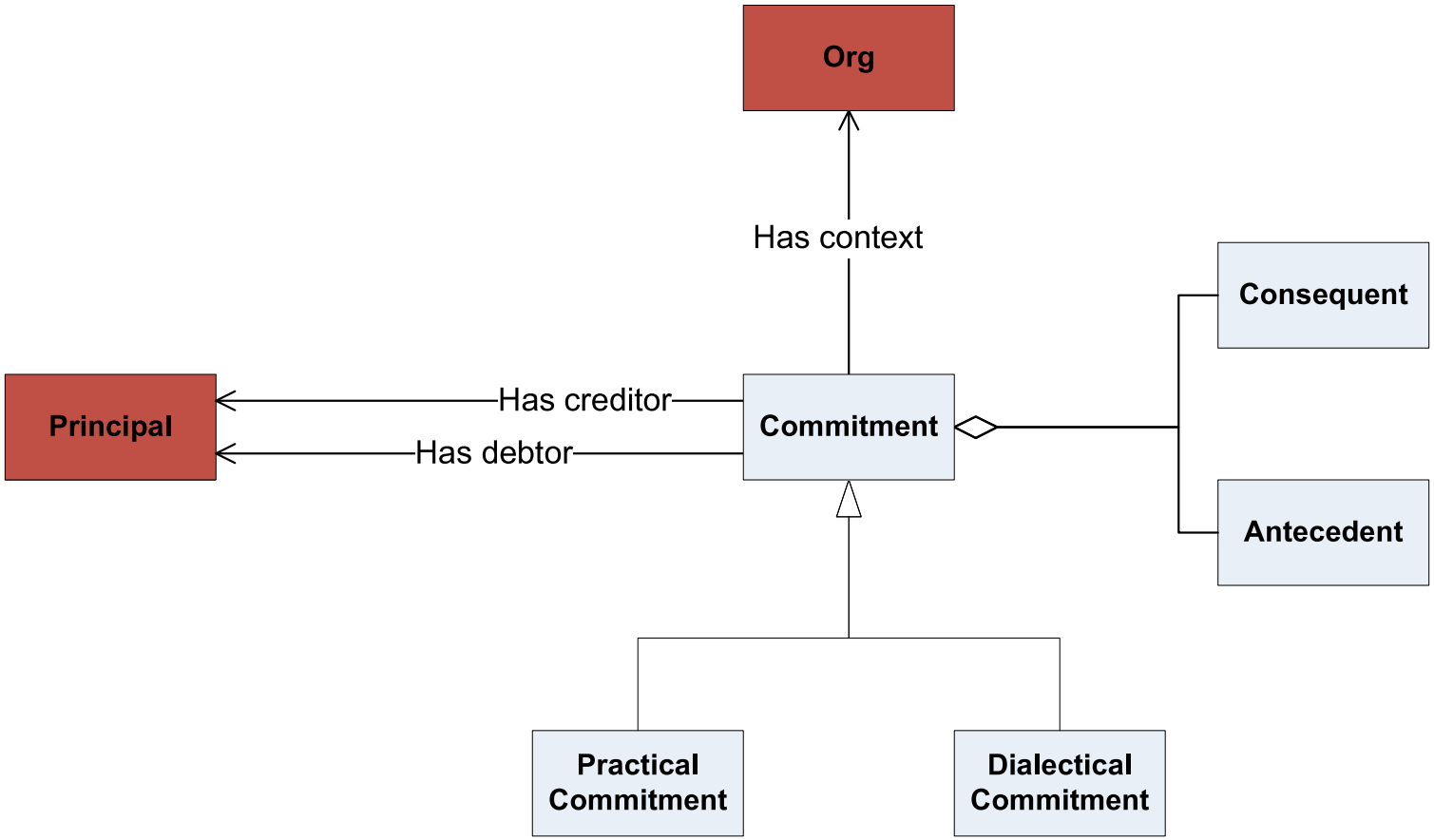
Commitments as Elements of a Contract

A kind of normative relationship: Express meanings of interactions

- Are atoms of contractual relationships
- Enable correctness checking of contracts
- Yield precise meanings and verifiability



Example: Commitment Progression

Via explicit operations or because of logical properties

C(Buyer, Seller, goods, pay): Active and conditional

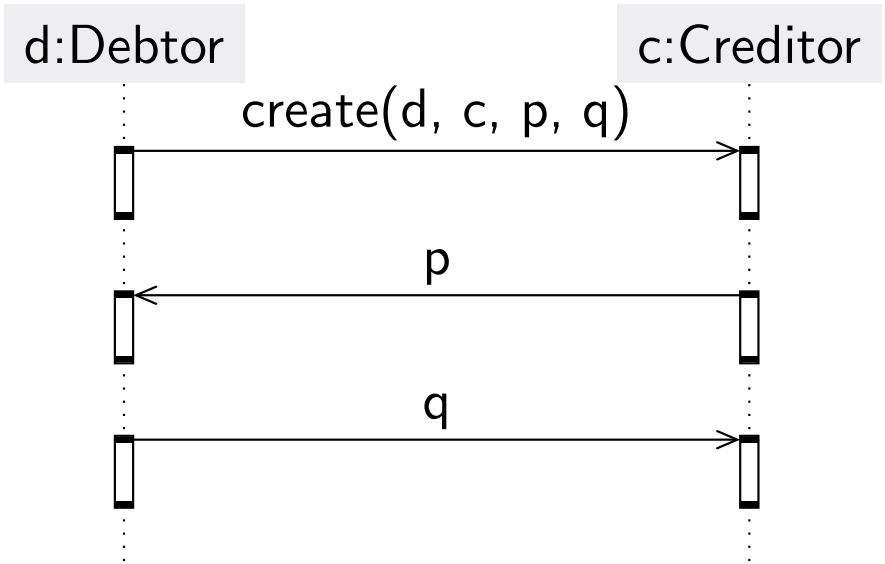
- If *goods* $\wedge$ C(Buyer, Seller, goods, pay) Then
  - Active and detached (or unconditional or base)
  - C(Buyer, Seller, T, pay)
- If C(Buyer, Seller, T, pay) Then
  - If *pay* Then Satisfied
  - If never *pay* Then Violated
- If C(Buyer, Seller, goods, pay) Then
  - If *pay* Then Satisfied
  - If never *pay* and never *goods* Then Expired

Can be nested:

C(Seller, Buyer, pay, C(Shipper, Buyer, T, deliverGoods))

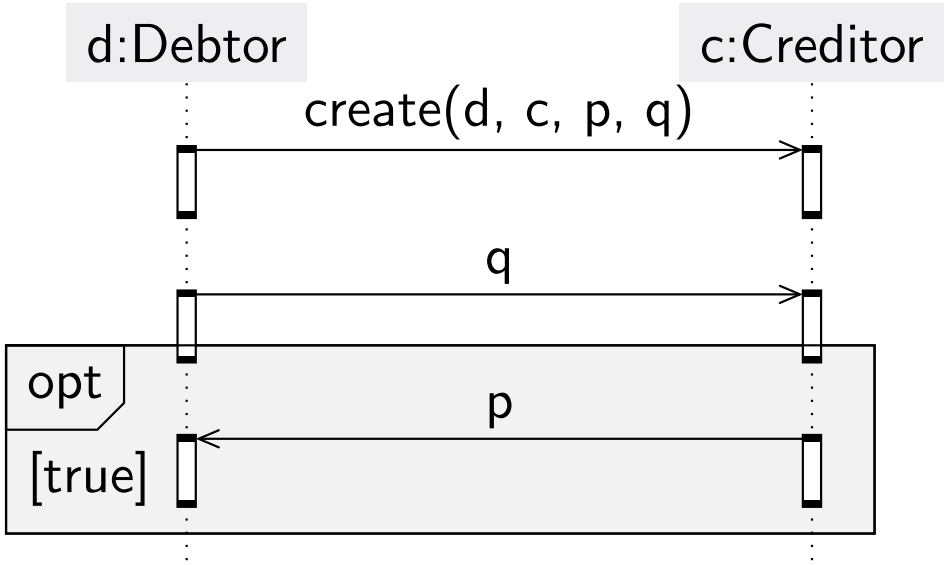
Operationalizing Commitments: Detach then Discharge

C(debtor, creditor, antecedent, consequent)



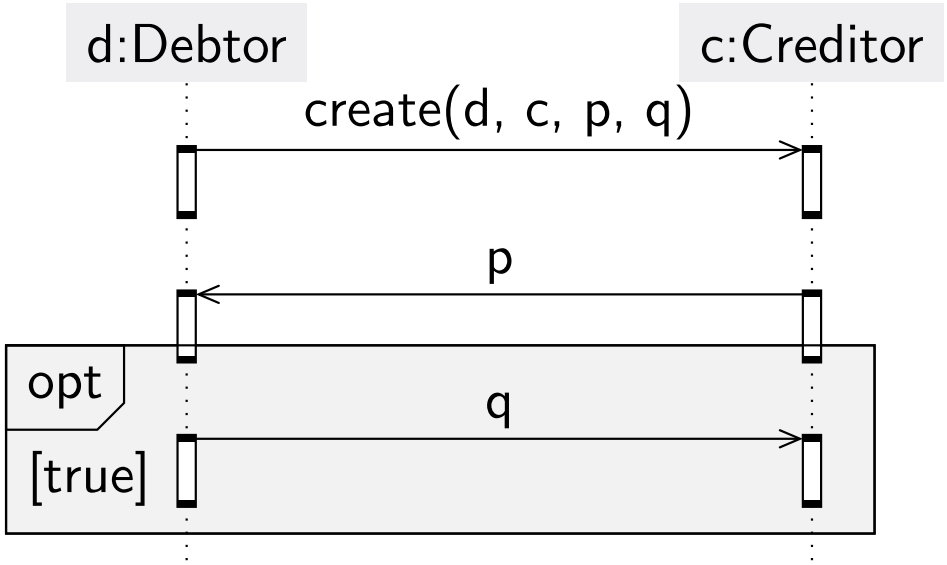
Operationalizing Commitments: Discharge First; Optional Detach

How about this?



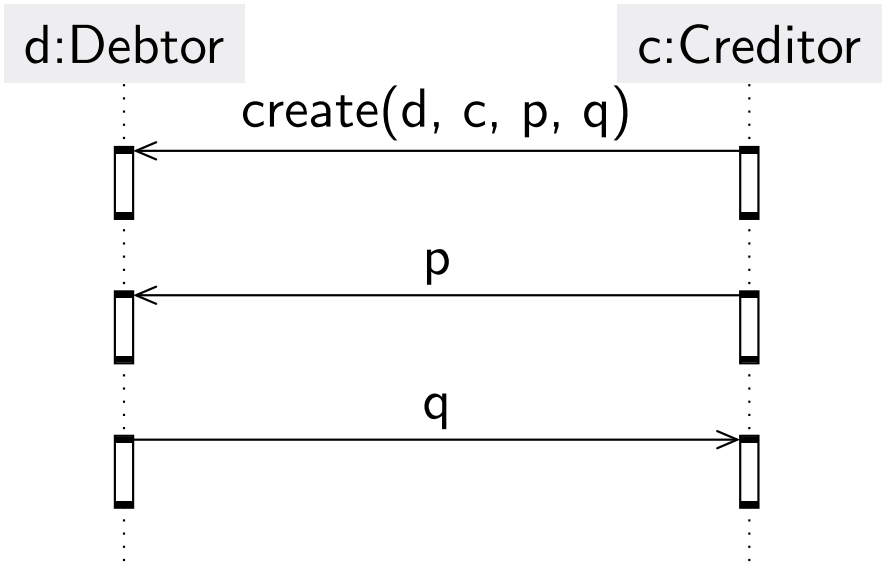
Operationalizing Commitments: Detach First; Optional Discharge

How about this?



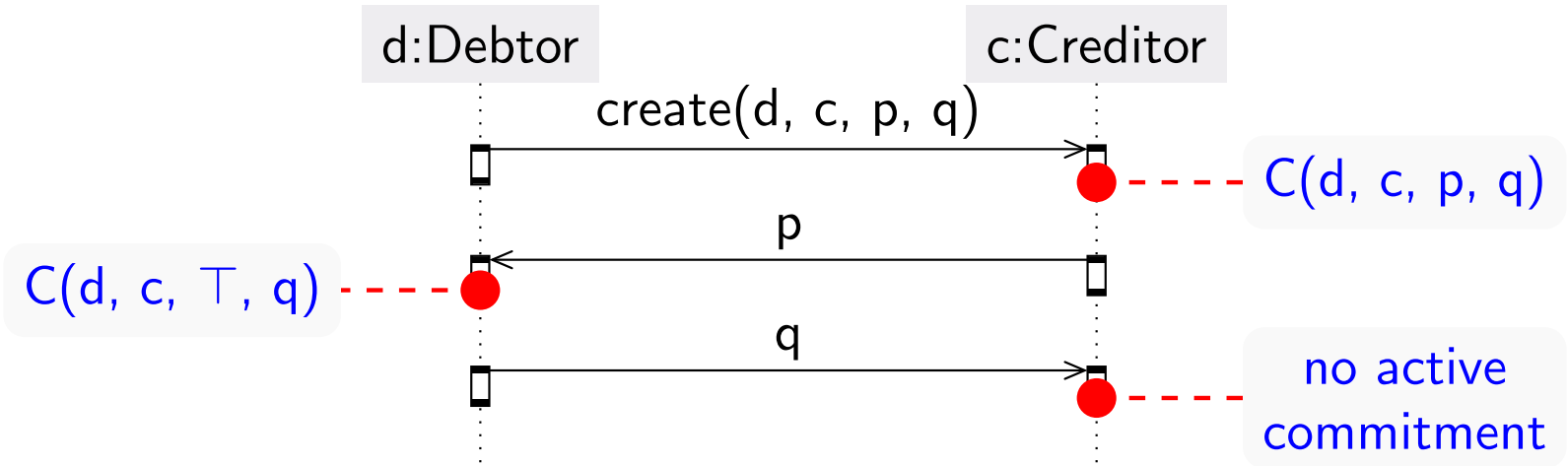
Operationalizing Commitments: Creation by Creditor

C(debtor, creditor, antecedent, consequent)



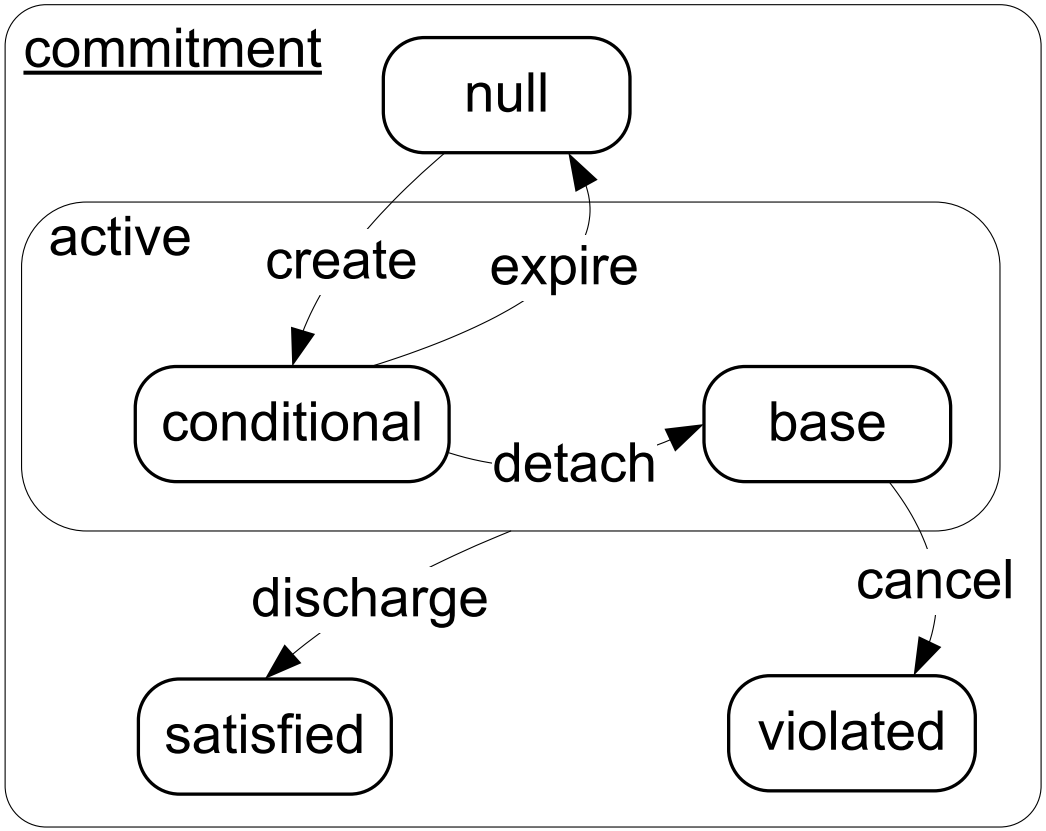
Operationalizing Commitments: Strengthening by Creditor

C(debtor, creditor, antecedent, consequent)

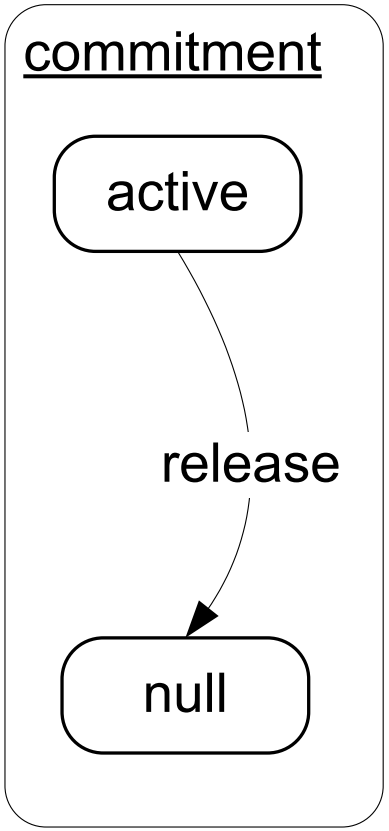


Commitment Life Cycle (and Patterns)

C(debtor, creditor, antecedent, consequent)



(a) Commit



(b) Relieve

Commitment Operations

- *create(C(d, c, p, q))* establishes the commitment
- *detach(C(d, c, p, q))* turns it into a base commitment
- *discharge(C(d, c, p, q))* satisfies the commitment
- *cancel(C(d, c, p, q))* cancels the commitment
- *release(C(d, c, p, q))* releases the debtor from the commitment
- *delegate(z, C(d, c, p, q))* replaces d by z as the debtor
  - d remains ultimately responsible (in our work)
- *assign(w, C(d, c, p, q))* replaces c by w as the creditor

Cupid: Unifying Accountability and Traceability

Computing states of norms over event stores

- Benefits: Basis for
  - Intelligent decision making
  - Key Performance Indicators
- Begin from event schema
  - Keys
  - Distinguished timestamp attribute
- Specify accountability requirements as norms
- Automatically generate SQL schema from event schema
- Automatically generate SQL queries to determine accountability status at specified moment
  - Now
  - A hypothetical moment to help ascribe responsibility

An Information Model and Commitment Specification

```
Quote(mID, cID, qID, itemID, uPrice, t) with key qID
Order(cID, mID, oID, qID, qty, addr, t) with key oID
Payment(cID, mID, pID, oID, pPrice, t) with key pID
Shipment(mID, cID, sID, oID, addr, t) with key sID
Refund(mID, cID, rID, pID, rAmount, t) with key rID
Coupon(cID, mID, uID, oID, rebate, t) with key uID
```

```
commitment DiscountQuote mID to cID
create Quote
detach Order and Payment[, Quote + 10]
  where pPrice >= 0.9 * uPrice * qty
discharge Shipment[,Payment + 5]
```

A DiscountQuote commitment from a merchant to a customer is

- *created* upon Quote;
- *detached* if Order happens and Payment happens within ten days of Quote and is for at least 90% of quoted amount (else *expires*)
- *discharged* if Shipment happens within five days of Payment (else *violated*)

Example: Compensation

Illustrates nesting: A commitment depends upon another commitment's state

```
commitment Compensation mID to cID
create Quote
detach violated(DiscountQuote)
discharge Refund[,violated(DiscountQuote) + 9] where rAmount = pPrice
```

A Compensation commitment is created upon Quote and says that if DiscountQuote is violated, the merchant will refund the payment within nine days of the violation.

Properties

- All Cupid queries are *safe*
  - Given any possible model M with finite extensions for base events, the extension of Q relative to M, $[\![Q]\!]$, is finite
- *Well-identified* specifications capture a notion of adequate correlation among the events that in the specification.
- Instances of a *finitely expirable* specification are guaranteed to expire if not detached within a finite amount of time.
- Instances of a *finitely violable* specification are guaranteed to expire if not discharged within a finite amount of time.

Commitment-Based Multiagent Approaches

Give primacy to business meanings of service engagements

- Identify messages
- Identify their meanings in terms of their effect on the social state
  - Creation of the commitments among the participants
  - Manipulation of commitments
  - Changes to parts of the state relevant to commitments
- Instead of explicit state transitions, consider inference on the social state based on the messages

Example Commitment Protocol

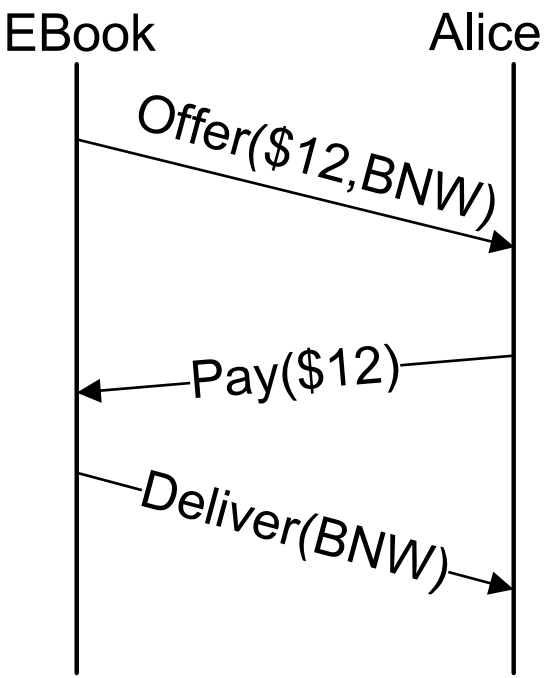
Purely declarative specification

Offer(*mer*, *cus*, *price*, *item*) means CREATE(*mer*, *cus*, *price*, *item*)
Accept(*cus*, *mer*, *price*, *item*) means CREATE(*cus*, *mer*, *item*, *price*)
Reject(*cus*, *mer*, *price*, *item*) means RELEASE(*mer*, *cus*, *price*, *item*)
Deliver(*mer*, *cus*, *item*) means DECLARE(*mer*, *cus*, *item*)
Pay(*cus*, *mer*, *price*) means DECLARE(*cus*, *mer*, *price*)

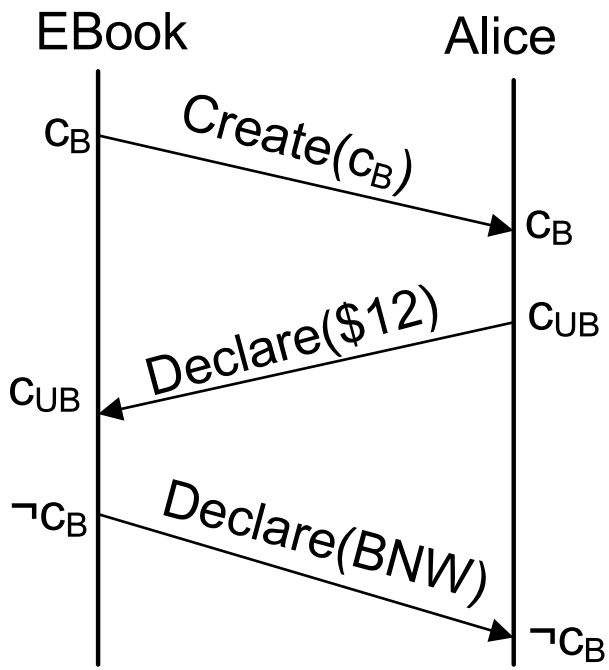
- Specifies how each message affects the social state
  - By acting on a commitment explicitly
  - By bringing about a social fact via DECLARE that may cause commitments to detach or discharge
- The social state is conceptual
- In general, no centralized store of social state
  - Raises the challenge of commitment alignment in distributed systems

Distinguishing Message Syntax and Meaning

Two views of the same enactment



Messaging



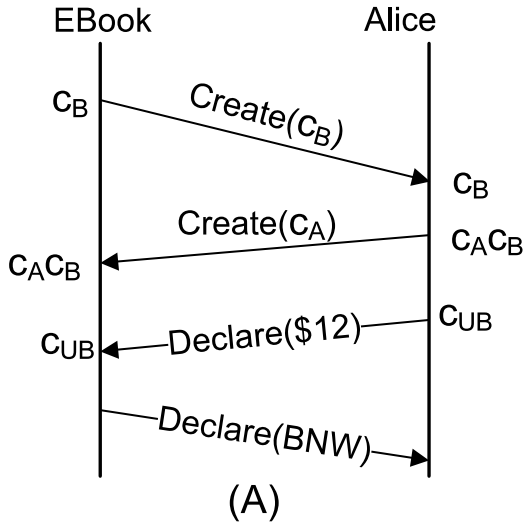
Meaning

Evaluation with respect to MAS

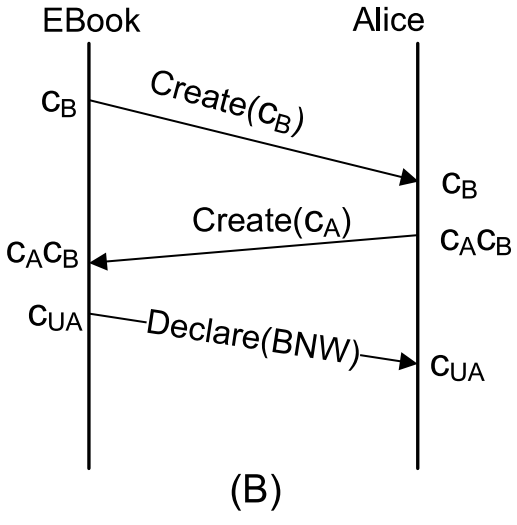
- Compliance: At the business level. A protocol enactment is correct as long as the parties involved do not violate their commitments
- Flexibility: Enhanced by expanding the operational choices for each party, e.g., discharge a commitment when convenient (even sooner); delegate or assign
- Software engineering: Commitments are a high-level abstraction for capturing business interactions
  - Support loose coupling among agents
  - Accommodate the autonomy of each participant

Illustrating Flexible Enactment

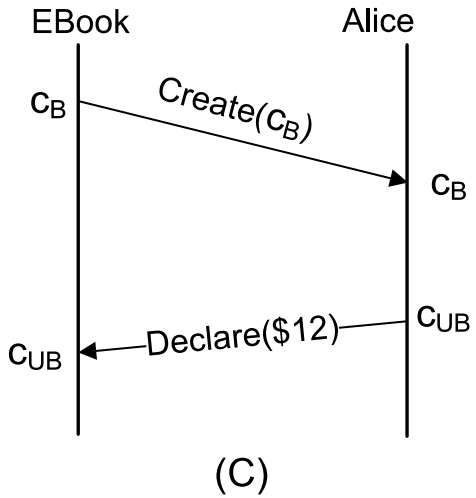
These are compliant executions in terms of commitments, and thus realize the above protocol



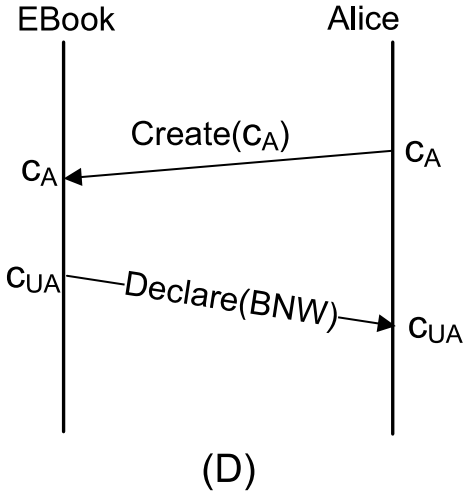
(A)



(B)



(C)



(D)

Comparing Agent Communication Approaches

	Traditional SE	**Traditional AI**	**Commitment Protocols**
Abstraction	control flow	mentalist	business relationship
Compliance	lexical basis	unverifiable	semantic basis
Flexibility	low	low	high
Interoperability	message-level	integration	business-level