Outline

Challenges of Electronic Business

Specification Approaches

Commitments

Architecture in IT

Contracts and Governance

XML Concepts and Techniques

XML Modeling and Storage

Summary and Directions
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

![Diagram of commitments and relationships]
Example: Commitment Progression
Via explicit operations or because of logical properties

\[ C(\text{Buyer}, \text{Seller}, \text{goods}, \text{pay}) : \text{Active and conditional} \]

- If \( \text{goods} \land C(\text{Buyer}, \text{Seller}, \text{goods}, \text{pay}) \) Then
  - Active and detached (or unconditional or base)
  - \( C(\text{Buyer}, \text{Seller}, T, \text{pay}) \)
- If \( C(\text{Buyer}, \text{Seller}, T, \text{pay}) \) Then
  - If \( \text{pay} \) Then Satisfied
  - If never \( \text{pay} \) Then Violated
- If \( C(\text{Buyer}, \text{Seller}, \text{goods}, \text{pay}) \) Then
  - If \( \text{pay} \) Then Satisfied
  - If never \( \text{pay} \) and never \( \text{goods} \) Then Expired

Can be nested:
\[ C(\text{Seller}, \text{Buyer}, \text{pay}, C(\text{Shipper}, \text{Buyer}, T, \text{deliverGoods})) \]
Operationalizing Commitments: Detach then Discharge

\[ C(\text{debtor, creditor, antecedent, consequent}) \]
Operationalizing Commitments: Discharge First; Optional Detach

How about this?

```
create(d, c, p, q)
```

- `d`: Debtor
- `c`: Creditor
- `opt [true]`
- `p`
Operationalizing Commitments: Detach First; Optional Discharge

How about this?

d: Debtor

create(d, c, p, q)

p

opt

[true]

q

c: Creditor
Operationalizing Commitments: Creation by Creditor

\[ C(\text{debtor}, \text{creditor}, \text{antecedent}, \text{consequent}) \]
Operationalizing Commitments: Strengthening by Creditor

\[ C(\text{debtor}, \text{creditor}, \text{antecedent}, \text{consequent}) \]
Commitment Life Cycle (and Patterns)

C(debtor, creditor, antecedent, consequent)

(a) Commit

(b) Relieve
Commitment Operations

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