Commitment Alignment

Techniques for Distributed Computing

Amit K. Chopra and Munindar P. Singh

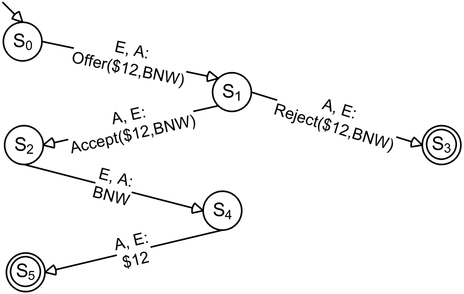
North Carolina State University

November, 2008

Business Interactions

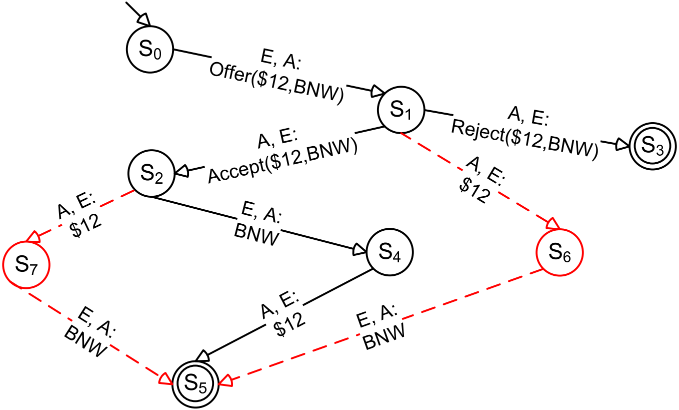
- Flow of messages
- Languages: WS-CDL, ebBP, UML Interaction Diagrams

Example involving EBook and Alice



Applications: TWIST (foreign exchange), HL7 (healthcare), RosettaNet (quote to cash)

Challenge: Flexible Interaction



Problem with current approaches

- No business meaning; compliance = token-matching
- Unnecessarily rigid: *lost business opportunities*!

Business Interactions: A Semantic Approach

Meanings in terms of effects on commitments

Compliance = not violating a commitment

- $\mathsf{C}(\mathit{debtor}, \mathit{creditor}, \mathit{antecedent}, \mathit{consequent})$
  - $\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \$12, \mathit{BNW})$

- DETACH: $\mathsf{C}(x, y, r, u) \wedge r \rightarrow \mathsf{C}(x, y, \top, u)$
  - $\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \$12, \mathit{BNW}) \wedge \$12 \Rightarrow \mathsf{C}(\mathit{EBook}, \mathit{Alice}, \top, \mathit{BNW})$
  - $\mathsf{C}(\mathit{debtor}, \mathit{creditor}, \top, \mathit{consequent})$: unconditional commitment

- DISCHARGE: $u \rightarrow \neg\mathsf{C}(x, y, r, u)$
  - $\mathit{BNW} \Rightarrow \neg\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \$12, \mathit{BNW})$
  - $\mathit{BNW} \Rightarrow \neg\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \top, \mathit{BNW})$

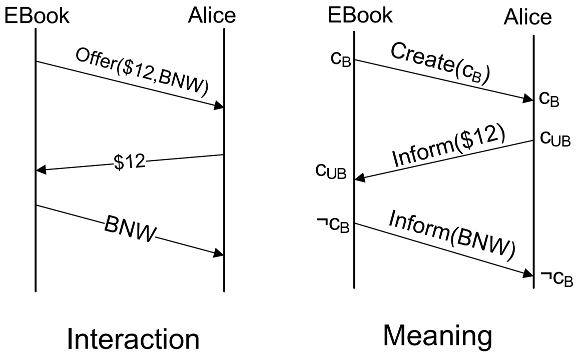
Commitment Operations

Operation	Performer	Effect
$\text{CREATE}(\mathsf{C}(x, y, r, u))$	x	$\mathsf{C}(x, y, r, u)$
$\text{CANCEL}(\mathsf{C}(x, y, r, u))$	x	$\neg\mathsf{C}(x, y, r, u)$
$\text{RELEASE}(\mathsf{C}(x, y, r, u))$	y	$\neg\mathsf{C}(x, y, r, u)$
$\text{DELEGATE}(\mathsf{C}(x, y, r, u), z)$	x	$\mathsf{C}(z, y, r, u)$
$\text{ASSIGN}(\mathsf{C}(x, y, r, u), z)$	y	$\mathsf{C}(x, z, r, u)$

- *EBook* does $\text{DELEGATE}(\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \$12, \mathit{BNW}), \mathit{Charlie}) \Rightarrow \mathsf{C}(\mathit{Charlie}, \mathit{Alice}, \$12, \mathit{BNW})$
- *Alice* does $\text{ASSIGN}(\mathsf{C}(\mathit{EBook}, \mathit{Alice}, \$12, \mathit{BNW}), \mathit{Bob}) \Rightarrow \mathsf{C}(\mathit{EBook}, \mathit{Bob}, \$12, \mathit{BNW})$

Interaction and Meaning: An Example

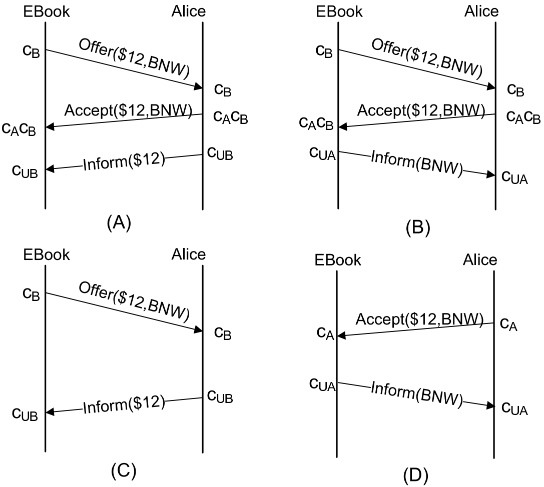
$c_B = C(EBook, Alice, \$12, BNW)$
$c_{UB} = C(EBook, Alice, \top, BNW)$



Each agent infers commitments based solely on the messages it observes

Realizing Flexible Interaction

$$c_A = C(Alice, EBook, BNW, \$12),\ c_{UA} = C(Alice, EBook, \top, \$12)$$
$$c_B = C(EBook, Alice, \$12, BNW),\ c_{UB} = C(EBook, Alice, \top, BNW)$$



(A)

(B)

(C)

(D)

Commitment Alignment

Commitment Alignment

Key aspect of business interoperability

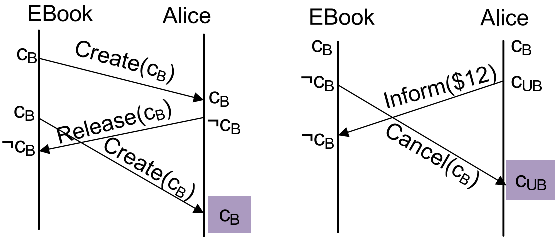
- In all *relevant* executions, if creditor infers a commitment, debtor must also infer that commitment
- *Misalignment*: Alice infers C($EBook, Alice, \$12, BNW$) but EBook does not
- Notice asymmetry
  - Not Misalignment: EBook infers C($EBook, Alice, \$12, BNW$) but Alice does not

Misalignment Cause: Autonomy

An agent can freely send messages

- *Freely send* ⇒ asynchronous messaging

$c_B = \mathrm{C}(EBook, Alice, \$12, BNW)$
$c_{UB} = \mathrm{C}(EBook, Alice, \top, BNW)$



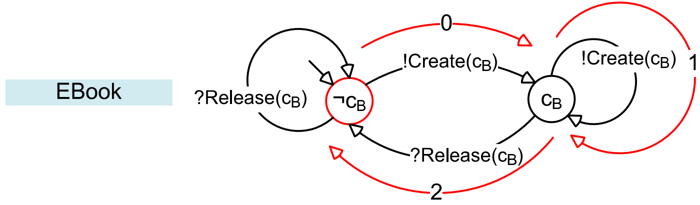
(A)

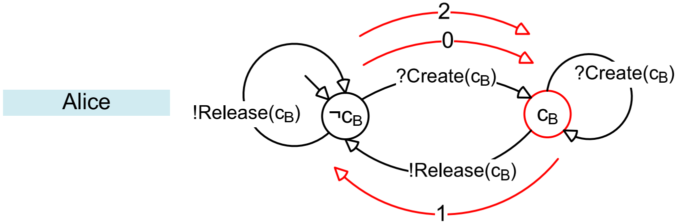
(B)

Misalignment Cause: Autonomy

Automata view

$c_B = C(EBook, Alice, \$12, BNW)$

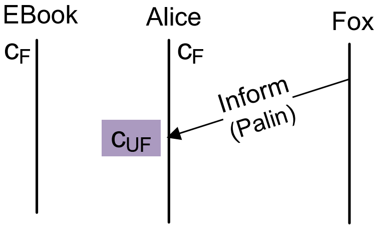




Misalignment Cause: Distribution

Agents may not have vital information

$$c_F = \mathrm{C}(EBook, Alice, Palin, F451)$$
$$c_{UF} = \mathrm{C}(EBook, Alice, \top, F451)$$



Misaligned

Misalignment Cause: Heterogeneity

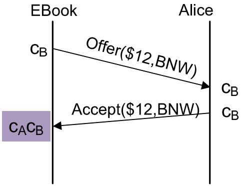
Interfaces of agents are incompatible

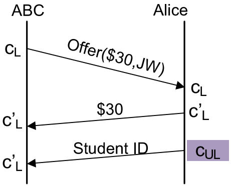
- Developed and evolve independently
- Agents reflect business interests of autonomous parties
- Messages may have incompatible meanings

Heterogeneity Examples

$$c_A = C(Alice, EBook, BNW, \$12)$$
$$c_B = C(EBook, Alice, \$12, BNW)$$

$$c_L = C(ABC, Alice, \$30 \wedge age, JW)$$
$$c'_L = C(ABC, Alice, age, JW)$$
$$c_{UL} = C(ABC, Alice, \top, JW)$$





Interpret *Accept* differently

Interpret *Student ID* differently

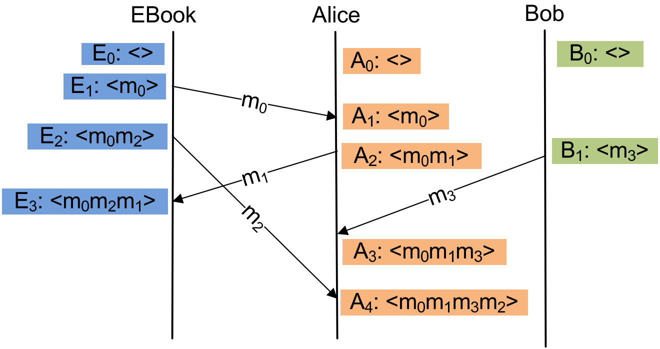
Contributions

Semantics and techniques for alignment

1. Formalize commitment alignment
   - Specify *relevant* executions
2. Handle autonomy: semantics of commitment operations and constraints on autonomy
3. Handle distribution: constraints on agent behavior
4. Handle heterogeneity: semantic interfaces of agents and a decision procedure to verify if agents are compatible

Formalizing Alignment

- Messaging: point to point, ordered, reliable, not “creative”
- Agent observes messages it sends and receives, serially



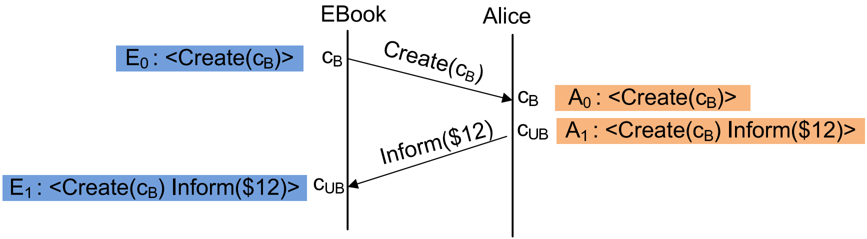
[E_1 A_0 B_0], [E_1 A_1 B_0] : Valid system states (executions)

[E_3 A_1 B_0] : Invalid system state

Relevant Executions: Quiescence

No messages in transit

$$c_B = C(EBook, Alice, \$12, BNW)$$
$$c_{UB} = C(EBook, Alice, \top, BNW)$$



EBook

Alice

E_0 : <Create(c_B)>

c_B

Create(c_B)

c_B

A_0 : <Create(c_B)>

Inform($12)

c_{UB}

A_1 : <Create(c_B) Inform($12)>

E_1 : <Create(c_B) Inform($12)>

c_{UB}

[E_0 A_1] : Not quiescent

[E_1 A_1] : Quiescent

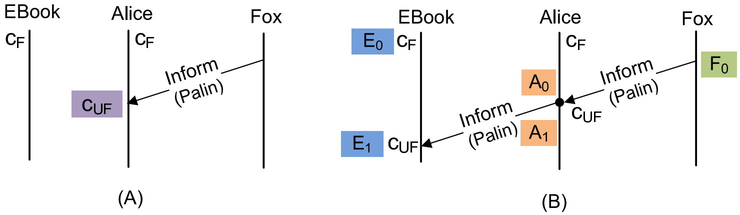
Relevant Executions: Integrity

Vital information is propagated

(Analogous to atomicity)

$c_F = \mathrm{C}(\mathit{EBook}, \mathit{Alice}, \mathit{Palin}, \mathit{F451})$

$c_{UF} = \mathrm{C}(\mathit{EBook}, \mathit{Alice}, \top, \mathit{F451})$



$[E_0\,A_0\,F_0]$: Not integral

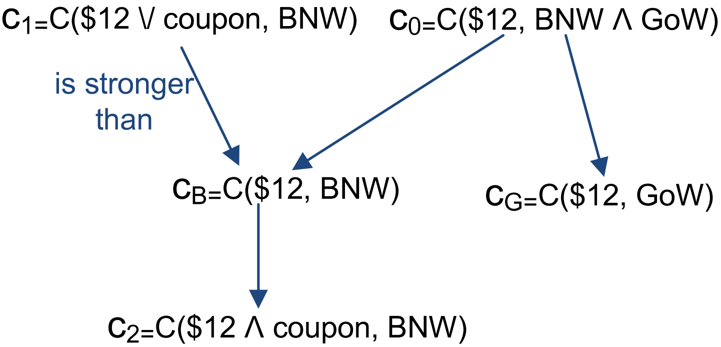
$[E_1\,A_1\,F_0]$: Integral

Handling Autonomy

Sketch

- Define what an agent infers upon observing messages
  - *Create*
  - *Cancel*
  - *Release*
  - *Delegate*
  - *Assign*
  - *Inform*
- Special case: introduce constraints to handle the Cancel-Detach race

Strength of Commitments



c_1=C($12 $\vee$ coupon, BNW)

c_0=C($12, BNW $\wedge$ GoW)

is stronger than

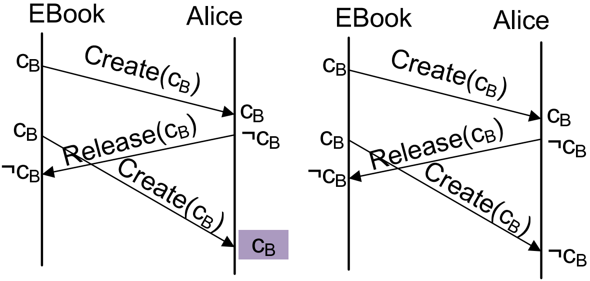
c_B=C($12, BNW)

c_G=C($12, GoW)

c_2=C($12 $\wedge$ coupon, BNW)

Principle of Novel Creation

Create has no effect if a stronger commitment has held before
$c_B = \mathrm{C}(EBook, Alice, \$12, BNW)$

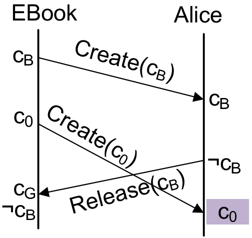


Naïve
Semantics

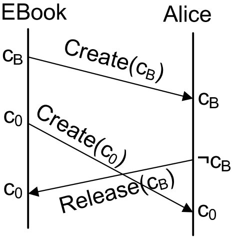
Principle

Principle of Complete Erasure

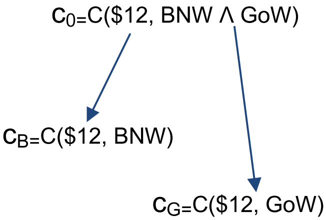
Release or *Cancel* has no effect if a *strictly* stronger commitment holds; if no such commitment holds, then each weaker commitment is released or cancelled, as appropriate



Naïve
Semantics



Principle



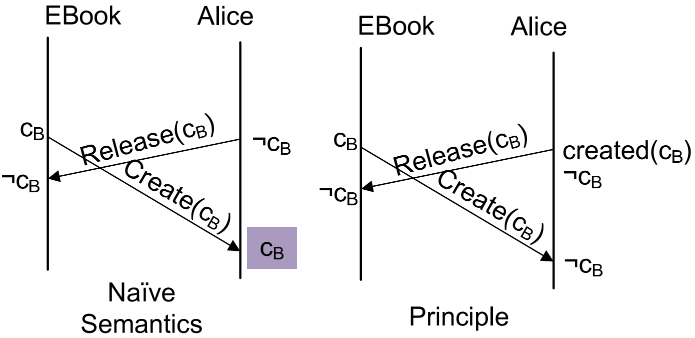
c_0=C($12, BNW ∧ GoW)

c_B=C($12, BNW)

c_G=C($12, GoW)

Principle of ACCOMMODATION

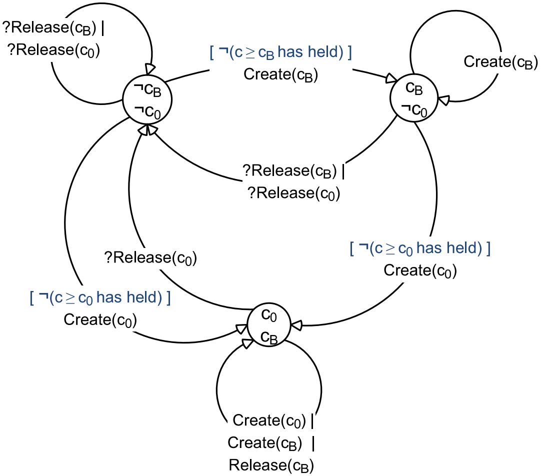
Release or *Cancel* has the effect that each weaker commitment is treated as if it has held before



(Principle of NOVEL CREATION also comes into play)

Principles at Work

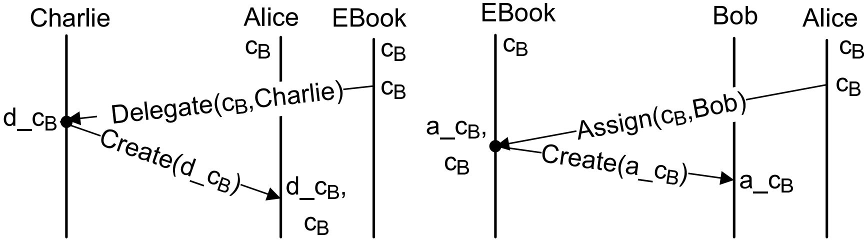
Automata view



Delegate and *Assign*

Multiparty scenarios

$$c_B = \mathrm{C}(EBook, Alice, \$12, BNW)$$
$$d\_c_B = \mathrm{C}(Charlie, Alice, \$12, BNW)$$
$$a\_c_B = \mathrm{C}(EBook, Bob, \$12, BNW)$$



(A) Delegate Pattern

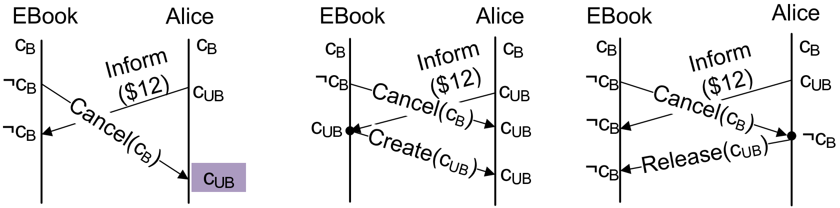
(B) Assign Pattern

Principle of UNIFORM PRIORITY

For every commitment, either its cancellation or its detach must be accorded priority

$c_B = \mathrm{C}(EBook, Alice, \$12, BNW)$

$c_{UB} = \mathrm{C}(EBook, Alice, \top, BNW)$



A) Misaligned

B) Detach priority

C) Cancel priority

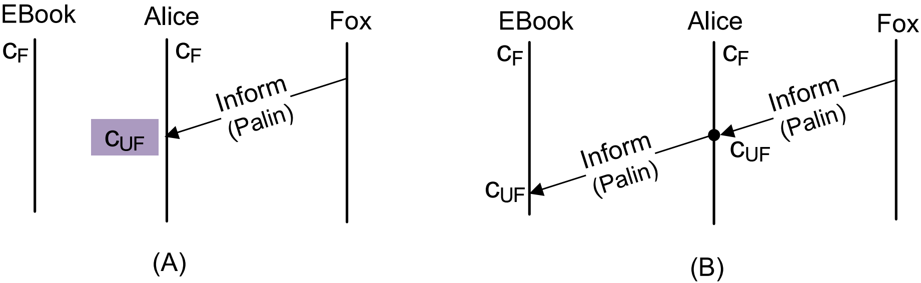
Handling Distribution

Principle of NOTIFICATION

A creditor must notify debtors of detaches

$c_F = C(EBook, Alice, Palin, F451)$

$c_{UF} = C(EBook, Alice, \top, F451)$



Analogously, a debtor must notify creditors of discharges

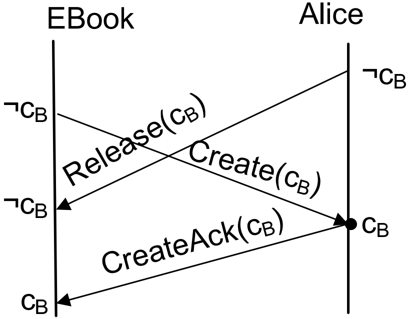
Result

- Proved that under the semantics and constraints, no misalignment occurs because of autonomy or distribution

Related Work: Synchronization-Based Schemes

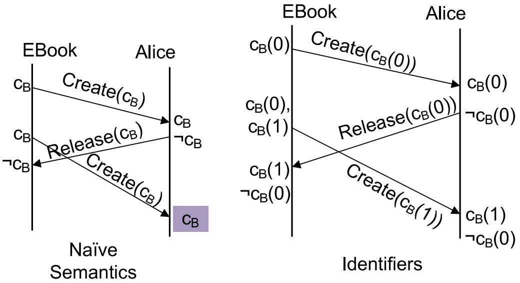
- McBurney and Parsons (AAMAS-03), Paurobally *et al.* (AAMAS-03), Amgoud *et al.* (ECAI-02)
- Our approach is more general

$c_B = \mathsf{C}(EBook, Alice, \$12, BNW)$



Related Work: Commitment Identifiers

- Each commitment is created with a unique identifier, and commitment operations reference those identifiers (Rovastos, AAMAS-07)
- $c_B(0) = \mathrm{C}(id_0, \$12, BNW)$, $c_B(1) = \mathrm{C}(id_1, \$12, BNW)$



Not general:
$\mathrm{C}(id_0, Palin, F451) \wedge \mathrm{C}(id_1, Palin, GoW) \Rightarrow \mathrm{C}(\_, Palin, F451 \wedge GoW)$

Semantic Interface Compatibility

Semantic Interface Compatibility

- Concerns ability of agents of work together based on their specifications
- General idea: agents *cover* each others' *assumptions*
- Traditionally assumptions specified as control and data flow
  - Criteria: absence of deadlocks, schema matching
- Our approach: commitments as assumptions
  - Criterion: alignment of commitments

Interface Specification

Messages an agent expects to send or receive

- Meanings of messages
- Commitments central element of meaning

Rule schema: *Msg* means *Clause*

- *Clause* is a conjunction of commitments and other propositions

Alice
$Offer(EBook, Alice)$ means $C(EBook, Alice, pay, book)$
$Pay(Alice, EBook)$ means pay

Decision Procedure for Interface Compatibility

Our decision procedure checks three kinds of coverage

- Commitment
- Antecedent
- Consequent

Commitment Coverage

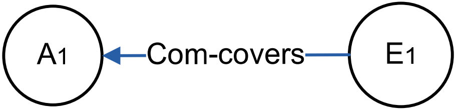
Debtor covers each commitment that creditor assumes

Alice

A_1 : *Offer* means $\mathrm{C}(\mathit{EBook}, \mathit{Alice}, \mathit{pay}, \mathit{book})$

EBook

E_1 : *Offer* means $\mathrm{C}(\mathit{EBook}, \mathit{Alice}, \mathit{pay}, \mathit{book} \wedge \mathit{receipt})$



Antecedent Coverage

Debtor covers creditor's assumptions about the antecedent

Alice

A_2 : *Offer* means $\mathrm{C}(EBook, Alice, pay, book)$

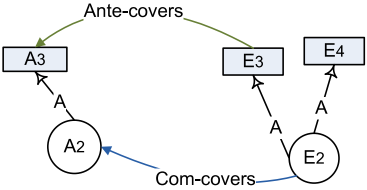
A_3 : *PayCash* means *pay*

EBook

E_2 : *Offer* means $\mathrm{C}(EBook, Alice, pay, book)$

E_3 : *PayCash* means *pay*

E_4 : *PayCredit* means *pay*



Consequent Coverage

Creditor covers debtor's assumptions about the consequent

Alice

A_4 : *Offer* means C(*EBook*, *Alice*, *pay*, *book*)

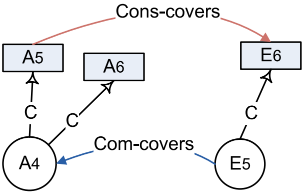
A_5 : *BookShip* means *book*

A_6 : *BookExpedited* means *book*

EBook

E_5 : *Offer* means C(*EBook*, *Alice*, *pay*, *book*)

E_6 : *BookShip* means *book*



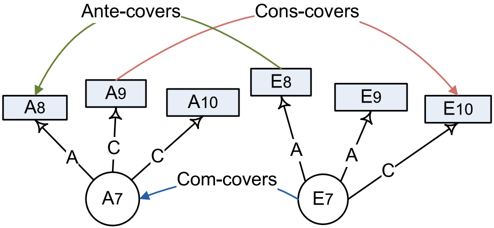
Complete Picture

Alice

A_7 : *Offer* means C(*EBook*, *Alice*, *pay*, *book*)
A_8 : *PayCash* means *pay*
A_9 : *BookShip* means *book*
A_{10} : *BookExpedited* means *book*

EBook

E_7 : *Offer* means C(*EBook*, *Alice*, *pay*, *book*)
E_8 : *PayCash* means *pay*
E_9 : *PayCredit* means *pay*
E_{10} : *BookShip* means *book*



Assembling the Pieces

- If agents are compatible (for the interface language presented here), and the semantics and techniques introduced to handle autonomy and distribution are adopted, then alignment is guaranteed

Summary

Contributions

Semantics and techniques for alignment

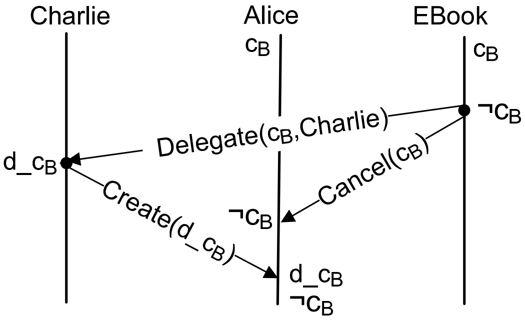
1. Formalized commitment alignment
2. Handled autonomy: semantics of commitment operations and constraints on autonomy
3. Handled distribution: constraints on agent behavior
4. Handled heterogeneity: semantic interfaces of agents and a decision procedure to verify if agents are compatible
5. Guaranteed alignment

Broader Perspective

- Software Engineering
  - Interaction: commitments as abstraction
  - Assumptions: commitments as architectural connectors between components
- Databases: measures for increased concurrency
  - Eventual consistency
  - Unilateral commit
  - Semantics-based consistency
- SOA: protocols, contracts, patterns
  - 2PC-based coordination protocols are inflexible

Delegation Without Responsibility Pattern

$$c_B = \mathrm{C}(EBook, Alice, \$12, BNW)$$
$$d\_c_B = \mathrm{C}(Charlie, Alice, \$12, BNW)$$
$$a\_c_B = \mathrm{C}(EBook, Bob, \$12, BNW)$$

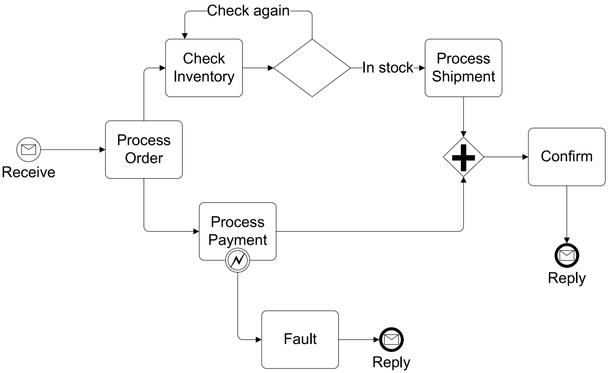


Future Work

- Metacommitments
- Richer language for interfaces, corresponding decision procedures, and tools
- Middleware based on semantics and constraints
- Pattern specification language
- Conformance: alignment preserving substitution

Business Processes: Orchestration

- Implementation of a participant
- Languages: BPMN, BPEL


Check again
Check Inventory
In stock
Process Shipment
Receive
Process Order
Confirm
Process Payment
Reply
Fault
Reply