

Agent Communication

Artificial Intelligence: Multiagent Systems – III

Nirav Ajmeri

Department of Computer Science
University of Bristol

TB1 2022-23

Outline

- 1 Review
- 2 Foundations of Agent Communication
- 3 Commitments-Based Approaches

Sociotechnical Systems and Social Norms

- Sociotechnical systems
- Social norms
 - Logical form of a norm
 - Normative specification of an STS
 - Lifecycle of norms (commitment, prohibition, and authorization)

Outline

- 1 Review
- 2 Foundations of Agent Communication
- 3 Commitments-Based Approaches

Speech Acts by Philosopher John Austin

Also known as communicative act theory

Communication is form of an action

- Judge declares a couple married
 - Not merely reporting a fact
 - Brings a fact into existence
- “Marry me” “I do”

Speech Acts (2)

- Phrased in declarative form
 - Performative verbs
- “I declare this couple man and wife”
- “I request that you to marry me” and “I promise that I will marry you”

Informative “Shipment will arrive on Wednesday”

“I inform you that the shipment will arrive on Wednesday”

Directive “Send me the goods”

“I demand that you send me the goods”

Commissive “I will pay you £5”

“I promise that I will pay you £5”

Agent Communication Primitives

- Small number of message types as primitives
- Reasonable but not adequate
 - Multiagent systems have several applications
 - Meanings we need for each are distinct
 - Official meaning may not be sufficient
 - Hardcoding may result in tight coupling

Traditional Software Engineering Approaches

- Sequence Diagrams
- State Transition Diagrams

Evaluation w.r.t Multiagent Systems

- Low level abstractions
- Difficult to design and maintain
- Little flexibility at runtime
- Easy compliance checking but at the cost of flexibility

Artificial Intelligence Approaches

- Knowledge Query and Manipulation Language (KQML)
 - Created by DARPA
 - Agents maintain a knowledge base in terms of belief assertions
 - Assumption: Agents are cooperative and designed by the same designer
- FIPA Agent Communication Language (ACL)
 - Specify a definitive syntax for interoperability
 - Specified the semantics of primitives

Evaluation w.r.t Multiagent Systems

- High level of abstraction
- Curtailed flexibility
- Verifying agent compliance is impossible

Outline

- 1 Review
- 2 Foundations of Agent Communication
- 3 Commitments-Based Approaches**

Commitment

$C(\text{SBJ}, \text{OBJ}, \text{ant}, \text{con})$

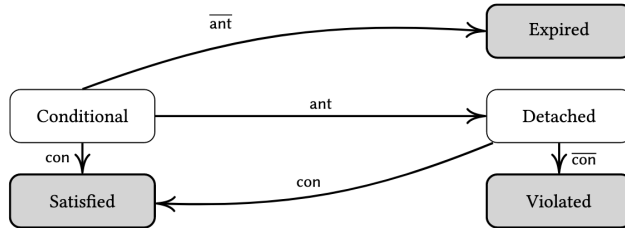
- subject and object are agents
 - Also referred as debtor and creditor
- antecedent and consequent are propositions

$C(x, y, r, u)$

- x is committed to y
- if r holds, then x will bring about u

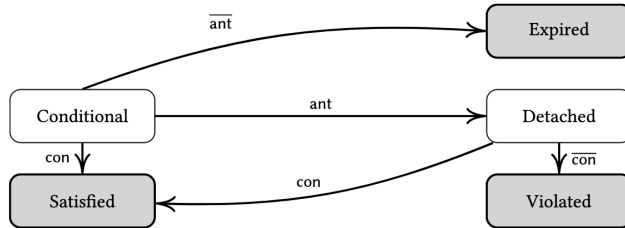
Commitment Life Cycle: Conditional

C(BookCo, Alice, £25, AIBook)



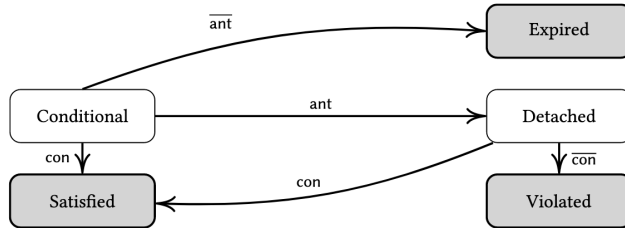
Commitment Life Cycle: Detached

$C(\text{BookCo}, \text{Alice}, \text{£25}, \text{AIBook}) \wedge \text{£25} \implies C(\text{BookCo}, \text{Alice}, \top, \text{AIBook})$



Commitment Life Cycle: Satisfy (or Discharge)

AIBook $\implies \neg C(\text{BookCo}, \text{Alice}, \text{£25}, \text{AIBook}) \wedge \neg C(\text{BookCo}, \text{Alice}, \top, \text{AIBook})$



Commitment Operations

- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold

Commitment Operations

- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold
- $\text{CANCEL}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to not hold

Commitment Operations

- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold
- $\text{CANCEL}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to not hold
- $\text{RELEASE}(x, y, r, u)$: performed by y ; causes $C(x, y, r, u)$ to not hold

Commitment Operations

- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold
- $\text{CANCEL}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to not hold
- $\text{RELEASE}(x, y, r, u)$: performed by y ; causes $C(x, y, r, u)$ to not hold
- $\text{DELEGATE}(x, y, z, r, u)$: performed by x ; causes $C(z, y, r, u)$ to hold

Commitment Operations

- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold
- $\text{CANCEL}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to not hold
- $\text{RELEASE}(x, y, r, u)$: performed by y ; causes $C(x, y, r, u)$ to not hold
- $\text{DELEGATE}(x, y, z, r, u)$: performed by x ; causes $C(z, y, r, u)$ to hold
- $\text{ASSIGN}(x, y, z, r, u)$: performed by y ; causes $C(x, z, r, u)$ to hold

Commitment Operations

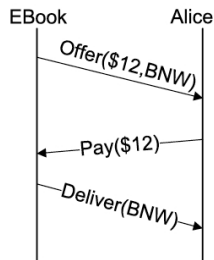
- $\text{CREATE}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to hold
- $\text{CANCEL}(x, y, r, u)$: performed by x ; causes $C(x, y, r, u)$ to not hold
- $\text{RELEASE}(x, y, r, u)$: performed by y ; causes $C(x, y, r, u)$ to not hold
- $\text{DELEGATE}(x, y, z, r, u)$: performed by x ; causes $C(z, y, r, u)$ to hold
- $\text{ASSIGN}(x, y, z, r, u)$: performed by y ; causes $C(x, z, r, u)$ to hold

Bring facts into existence:

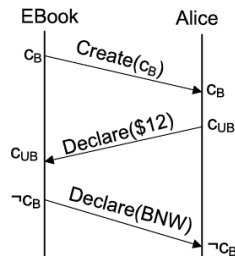
- $\text{DECLARE}(x, y, r)$: performed by x to inform y that the r holds
 - DECLARE is not a commitment operation

Commitment Protocol

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



Messaging



Meaning

Exercise

How could we specify a protocol for payment through a third party? Which commitment operation would be appropriate to capture the payment?