Agent Communication

Artificial Intelligence: Multiagent Systems – III

Nirav Ajmeri

School of Computer Science
University of Bristol

TB1 2023/24

Outline

Sociotechnical Systems and Social Norms

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

Outline

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 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
  - Hard-coding 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 run-time
- Easy compliance checking but at the cost of flexibility

Artificial Intelligence Approaches

- Knowledge Query and Manipulation Language (KQML)
  - Created by the United States' Defense Advanced Research Projects Agency (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

Commitment

C($\textsc{sbj}$, $\textsc{obj}$, ant, con)

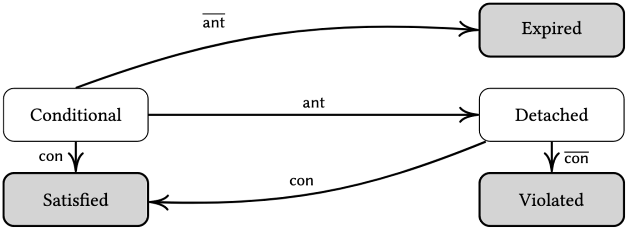
- subject and object are agents
  - Also referred as debtor and creditor
- antecedent and consequent are propositions

C(x, y, r, u)

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

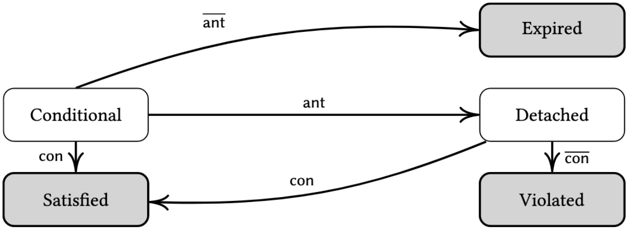
Commitment Life Cycle: Conditional

C(BookCo, Alice, £25, AIBook)



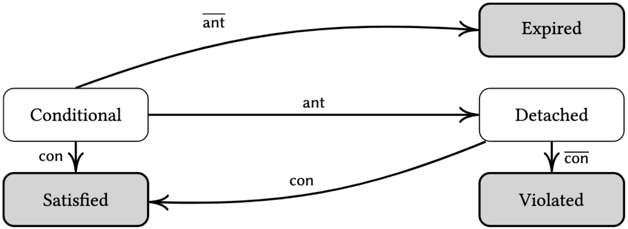
Commitment Life Cycle: Detached

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



Commitment Life Cycle: Satisfy (or Discharge)

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



Commitment Operations

- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold

Commitment Operations

- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold
- CANCEL(SBJ, OBJ, ant, con): performed by SBJ; causes C (SBJ, OBJ, ant, con) to not hold

Commitment Operations

- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold
- CANCEL(SBJ, OBJ, ant, con): performed by SBJ; causes C (SBJ, OBJ, ant, con) to not hold
- RELEASE(SBJ, OBJ, ant, con): performed by OBJ; causes C (SBJ, OBJ, ant, con) to not hold

Commitment Operations

- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold
- CANCEL(SBJ, OBJ, ant, con): performed by SBJ; causes C (SBJ, OBJ, ant, con) to not hold
- RELEASE(SBJ, OBJ, ant, con): performed by OBJ; causes C (SBJ, OBJ, ant, con) to not hold
- DELEGATE(SBJ1, OBJ, SBJ2, ant, con): performed by SBJ1; causes C (SBJ2, OBJ, ant, con) to hold

Commitment Operations

- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold
- CANCEL(SBJ, OBJ, ant, con): performed by SBJ; causes C (SBJ, OBJ, ant, con) to not hold
- RELEASE(SBJ, OBJ, ant, con): performed by OBJ; causes C (SBJ, OBJ, ant, con) to not hold
- DELEGATE(SBJ1, OBJ, SBJ2, ant, con): performed by SBJ1; causes C (SBJ2, OBJ, ant, con) to hold
- ASSIGN(SBJ, OBJ1, OBJ2, ant, con): performed by OBJ1; causes C (SBJ, OBJ2, ant, con) to hold

Commitment Operations

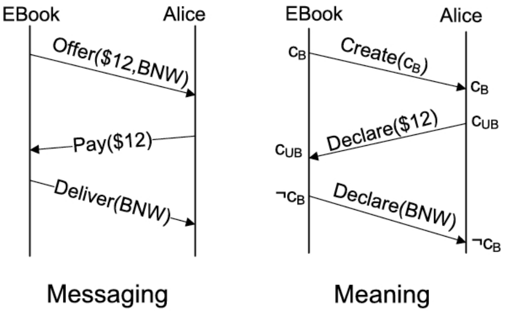
- CREATE(SBJ, OBJ, ant, con): performed by SBJ; causes C(SBJ, OBJ, ant, con) to hold
- CANCEL(SBJ, OBJ, ant, con): performed by SBJ; causes C (SBJ, OBJ, ant, con) to not hold
- RELEASE(SBJ, OBJ, ant, con): performed by OBJ; causes C (SBJ, OBJ, ant, con) to not hold
- DELEGATE(SBJ1, OBJ, SBJ2, ant, con): performed by SBJ1; causes C (SBJ2, OBJ, ant, con) to hold
- ASSIGN(SBJ, OBJ1, OBJ2, ant, con): performed by OBJ1; causes C (SBJ, OBJ2, ant, con) to hold

Bring facts into existence:

- DECLARE(x, y, r): performed by x to inform y that the r holds
  - DECLARE is NOT a commitment operation; it only conveys information

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)



Exercise

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