Agent Communication

Pınar Yolum
Email: p.yolum@uu.nl

Department of Information and Computing Sciences
Utrecht University

Interaction

- Interactions occur when agents exist and act in close proximity
  - Bumping into each other
  - Pulling an object from another's hand
  - Requesting an object
- Communications are the interactions that preserve autonomy of all participants
- Communications can be realized in several ways
  - through shared memory (if agents are collaborative)
  - by shared conventions (e.g., raising a hand)
  - by messaging passing

Message passing

Requires shared representation

- Syntax: A common language to represent information and queries
- Semantics: A structured vocabulary and a shared framework of knowledge (a shared ontology)
- Pragmatics:
  - Knowing whom to communicate with and how to find them
  - Knowing how to initiate and maintain an exchange
  - Knowing the effect of the communication on the recipient

Thinking of language

- Orientation
  - Describe the situation in terms of objects and their properties
  - Derive rules that apply to situations
  - Apply the rule to the current situation
- Literal meaning (not context-dependent)
- Hard to use in many settings (Winograd and Flores)
  - A: Is there any water in the fridge?
  - B: Yes
  - A: Where? I don't see it.
  - B: In the cells of the eggplant.
- Background is necessary

Semantic composition

- Meaning of a sentence is made up of its parts
  - Ex: Chair is a furniture
  - Exceptions exist
- Idioms
  - Different meaning all together
  - Ex: He twisted my arm to get a beer.
- Adjectives
  - Some clauses are learned as they are (e.g., white wine)
  - Some adjectives only make sense for the current noun (e.g., big cars vs. big cell phones)
- Context: Information that exists outside the sentence

Speech act theory

- Developed for natural language and views communication as action (as opposed to describing a given state) (Austin, 1962)
- Considers three aspects of a message
  - Locution, or how it is phrased, e.g., ”It is hot here”
  - Illocution, or how it is meant by the sender or understood by the receiver, e.g., a request to turn on the air conditioner or an assertion about the temperature
  - Perlocution, or how it influences the recipient, e.g., turns on the air conditioner, opens the window, ignores the speaker
- Illocution is the core aspect for agent communication

Speech act types

- Assertives: Describe the state of the world (ex. I inform you that it is raining)
- Directives: Attempt (in varying degrees) to make the other person do something (ex. I request you to talk)
- Commissives: Commit the speaker (in varying degrees) to a course of actions (ex. I commit to give you 5Euros)
- Expressives: Express a psychological state (ex. I apologize for being late).
- Declaratives: Make the content of the act match reality (ex. I declare you husband and wife)
- Permissives: Allow an action to be taken (ex. I authorize you to leave early)
- Prohibitives: Ban an action to be taken (ex. I forbid you to talk)

Speech acts as messages

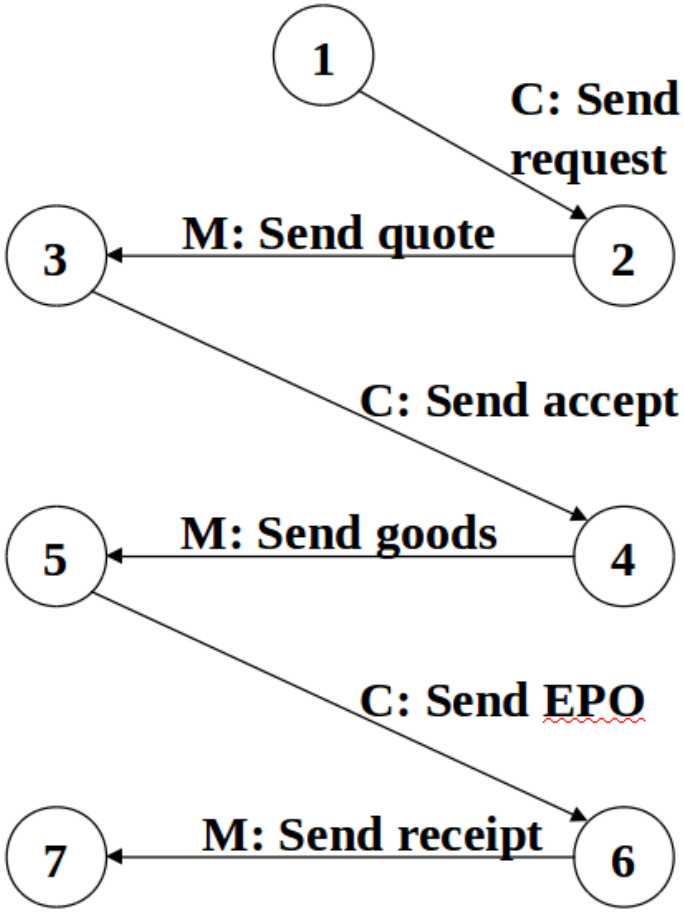
- Each message is a speech act and a propositional content (e.g., inform that it is raining)
- By exchanging speech acts, agents are changing the world state
- Expressive enough to capture real life interactions
- For working together (e.g., e-commerce), commissives are important as they capture promises
- Design protocols based on commitments where the protocol semantics is
  - Meaningful. Captures message content
  - Verifiable. Allows detection of non-compliant agents
  - Declarative. Defines what each action brings about rather than how they are brought about

Dynamic interactions of agents

Agents

- must be minimally constrained in their interactions (Autonomy).
- can be diverse (Heterogeneity).
- must be able to handle any unexpected conditions (Exceptions).
- should be able to take advantage of available opportunities (Opportunism).

FSM representation of the NetBill protocol



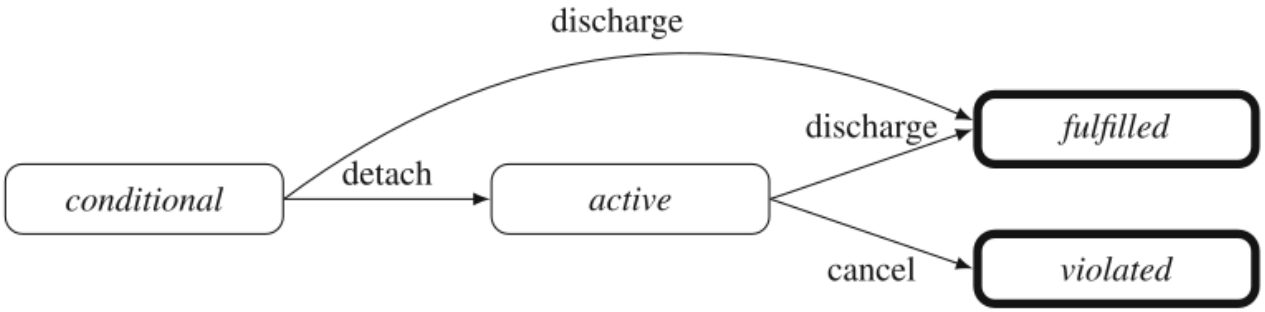
Variations not allowed in the FSM:

- The merchant may start the protocol by sending a quote.
- The customer may send an accept prior to offer.
- The merchant may send the goods prior to accept.

Social commitments

- A social commitment is an obligation from one party (debtor) to another (creditor) to bring about a condition.
- C(x, y, p): x commits to y to bring about p.
- Examples: C(*customer*, *merchant*, *payment*)
- CC(x, y, p, q) is a conditional commitment: x commits to y to bring about q if p is brought out first.
- Examples: CC(*customer*, *merchant*, *delivery*, *payment*)

Reasoning rules



- Detach: CC(x,y,p,q) ceases to exist when the proposition p becomes true, but C(x,y,q) is created.
- Discharge conditional: CC(x,y,p,q) ceases to exist when the proposition q becomes true.
- Discharge active: C(x,y,p) ceases to exist when the proposition p becomes true.
- Cancel active: C(x,y,p) remains open when the proposition p becomes false.

Commitment protocols

- Represent protocol states in terms of *social commitments*.
- Represent protocol executions as a series of operations that create and manipulate commitments.
- Allow agents to create and modify commitments by performing actions, so that they are only constrained to fulfill their commitments.

Definitions for message content

Atomic propositions:

- request(i): the customer has requested a quote.
- goods(i): the merchant has delivered the goods.
- pay(m): the customer has paid the agreed amount.
- receipt(i): the merchant has delivered the receipt.

Conditional commitments:

- accept(i, m): CC(CT, MR, goods(i), pay(m))
- promiseGoods(i, m): CC(MR, CT, accept(i, m), goods(i))
- promiseReceipt(i, m): CC(MR, CT, pay(m), receipt(i))
- offer(i, m): promiseGoods(i, m) $\wedge$ promiseReceipt(i, m)

Protocol specification

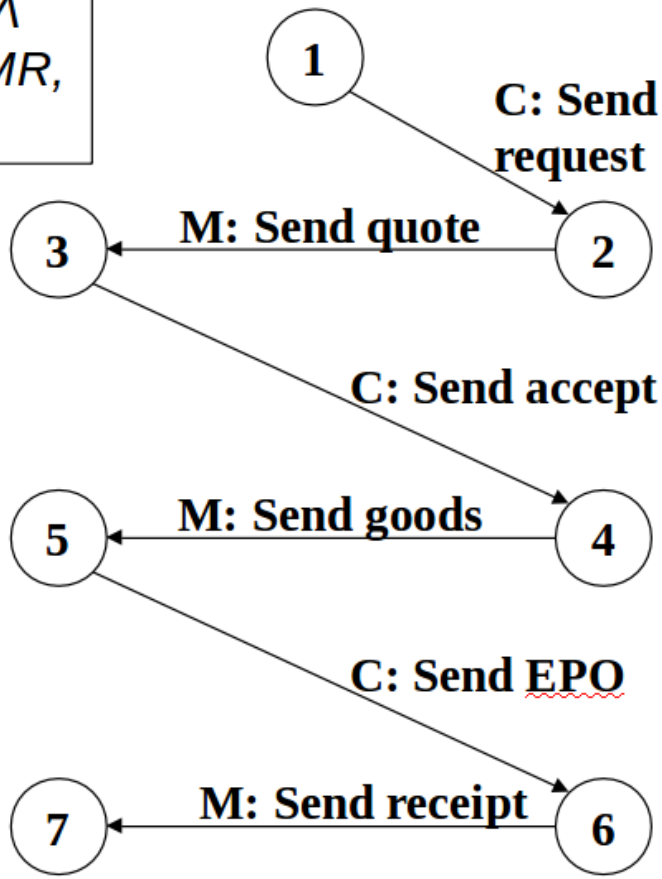
- A protocol specification
  - contains a set of actions (with preconditions).
  - does not specify any final states.
  - does not explicitly state the transitions; transitions follow from operations and reasoning rules on commitments.
- A protocol run
  - specifies the paths between states
  - contains sequence of actions
  - is complete if all the base-level commitments are resolved at the end.

NetBill specification

- $\langle sendRequest: request \rangle$
- $\langle sendQuote: offer \rangle$
- $\langle sendAccept: accept \rangle$
- $\langle sendGoods: goods \wedge promiseReceipt \rangle$
- $\langle sendEpo: pay \rangle$
- $\langle sendReceipt: receipt \rangle$

States in terms of commitments

promiseGoods(i, m): *CC(MR, CT, accept(i, m), goods(i))* ∧ *promiseReceipt(i, m)*: *CC(MR, CT, pay(m), receipt(i))*



request(i): the customer has requested a quote.

accept(i, m): *CC(CT, MR, goods(i), pay(m))*

promiseReceipt(i, m): *CC(MR, CT, pay(m), receipt(i))*

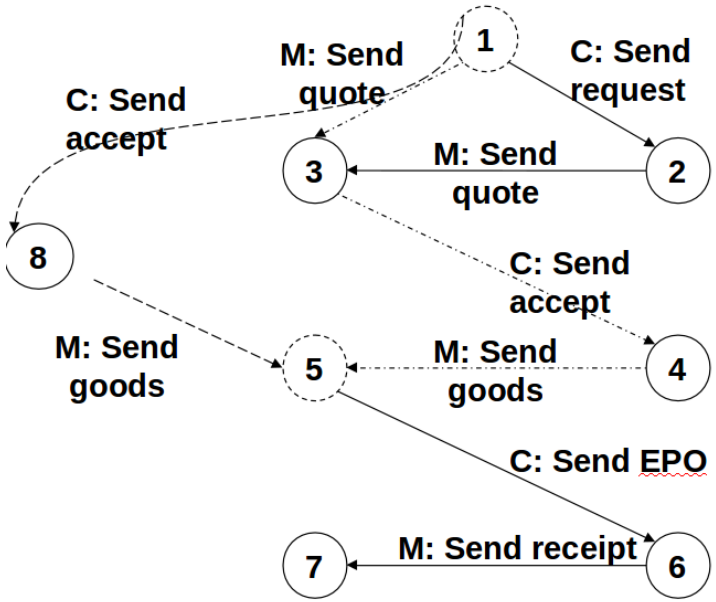
pay(m): the customer has paid the agreed amount.

receipt(i): the merchant has delivered the receipt.

Protocol execution

- Given
  - protocol specification that contains actions
  - initial state that does not contain propositions
  - goal state that consists of propositions that need to hold
- Generate protocol runs
- Interpretation of a protocol run
  - each timepoint maps to (the snapshot of) a state
  - exactly one event occurs between any two consecutive timepoints
  - any two consecutive timepoints along with the intervening event produce a transition
- Usage
  - Run time: Generate protocol runs
  - Compile time: Plan library
  - Agents can choose among the protocol runs

Sample protocol runs



- The merchant can now start the protocol by sending a quote.
- The customer can now send an accept prior to offer.

Run-Time compliance checking

- An agent can keep track of
  - its pending commitments
  - commitments made by others that are not satisfied
- It uses this local model to see if a commitment has been violated
- An agent who benefits from a commitment can always determine if it was violated