# *Agent Interaction: Languages, Dialogues and Protocols*


Peter McBurney

Agents and Intelligent Systems Group

Department of Informatics

King's College London

Strand London WC2R 2LS

www.dcs.kcl.ac.uk/staff/mcburney/

peter.mcburney@kcl.ac.uk


Collaborators: Katie Atkinson, Tim, Miller, Simon Parsons.

# Outline

- 1. Introduction – Human Language
- 2. Agent Language Syntax
- 3. Semantics of Agent Languages
- 4. Pragmatics of Agent Languages – Action Locutions
- 5. Agent Interaction Protocols
- 6. Protocol Libraries
- 7. Conclusions

# Research Problem

- The design of languages and protocols for automated communications between intelligent machines
  - Not human-human communications
  - Not human-machine communications (NLP, NLG).
- Despite some views to the contrary, this problem is not yet solved.
- We will draw on (human) linguistics and the philosophy of language, but we are not claiming to contribute to these domains, nor to Natural Language Processing.

# 1: Introduction - Human Language

# Human language has many functions

- A means of information transfer
  - The weather is sunny today.
  - I am hungry.
  - I prefer chicken to fish.
  - I intend to eat lunch.
- A means of co-ordinating joint actions
  - Would you be free for lunch today?
  - Let us divide the bill equally.
- A means of establishing and maintaining social relationships
  - Let's do lunch!
  - Phatic communications
- A signalling system
  - Let me pay for lunch!

# Aspects of human languages

Linguistic theory distinguishes:

- **Syntax** of a language: its words, phrases, sentences and grammar
- **Semantics** of a language: what meanings are assigned to the words, phrases & sentences
  - ie, the relationship between well-formed expressions in the syntax and objects or concepts in the world.
- **Pragmatics** of a language: non-semantic aspects of meaning, such as the speaker's intentions in making the utterance.

# Speaker Intentions

Alice says to Bob:

"The meeting is tomorrow at 17:00."

What can Bob validly infer?

# Valid potential inferences by Bob

- That the meeting is tomorrow.
- That Alice believes that the meeting is tomorrow.
- That Alice wants Bob to believe that the meeting is tomorrow.
- That Alice wants Bob to believe that the meeting is not tomorrow.
- That Alice wants Bob to believe that Alice believes that the meeting is tomorrow.
- That Alice wants Bob to believe that Alice does not believe that the meeting is tomorrow.
- That Alice wants Bob to believe that Alice wants Bob to believe that the meeting is tomorrow.
- That Alice wants Bob to believe that Alice does not want Bob to believe that the meeting is tomorrow.
- That the meeting is not tomorrow.
- etc (*ad infinitum*).

# Propositions and their semantics

- Propositions are statements of fact, asserting that some property holds in the world.
- These statements may be true or false
- Most of philosophy, for most of its history, has focused on propositions.
- Most of philosophy, for most of its history, has ignored other types of statements and utterances, and ignored actions.

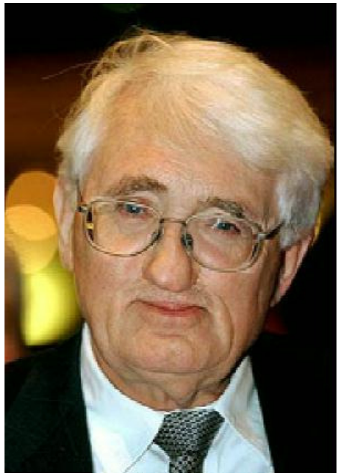
# Semantics of Propositions

- Suppose P is a proposition
  - eg, "The sky is red today."

- Truth-Conditional semantics (due to Frege, Tarski)
  - Speaker says:
    - *"I believe P"*
  - This statement may be challenged with:
    - *"Is P true?"*

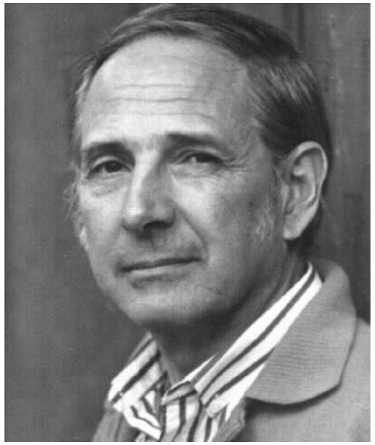
- Verificationist semantics (due to Dummett, Wright)
  - Many factual statements cannot be objectively verified
  - Speaker says:
    - *"I believe P"*
  - This statement is better challenged with:
    - *"Can you justify your belief in P ?"*

# Not all utterances are propositions

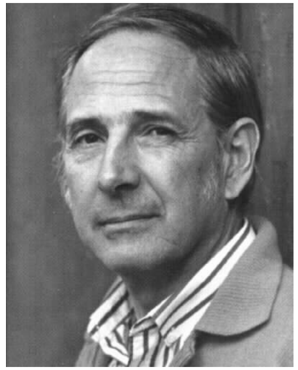


- Speech act theory is an account of these other utterances

- This theory is due to:
  - Thomas Reid (1710 - 1796)
  - Adolph Reinach (1883 - 1917)
  - John Austin (1911 – 1960)
  - John Searle (1932 - )
  - Jurgen Habermas (1929 - )



# Speech acts



- Some statements change the world

by their very utterance, eg.

  - "I name this ship, *The Queen Elizabeth.*"
  - "I declare you man and wife."

- These statements perform some action, but only under certain pre-conditions:
  - eg, For a marriage declaration to be legally binding in the UK, the celebrant must be registered, the location must be a registered location, the individuals must be single, at least two witnesses must be present, etc.

- Speech acts can be defined in terms of their felicity conditions and their rational effects.
  - Modern theory due to: Austin 1955, Searle 1969.

- Applications to linguistics, social sciences, computer science.

# A typology of human utterances (1)

- Factual statements
  - Purport to describe states of the real-world
- Expressive statements
  - Purport to describe internal mental states of speaker
- Social Connection statements
  - Purport to describe social relations between participants or others
- Commissives
  - Speaker desires a particular world state, and proposes actions to establish or maintain this world state.

# A typology of human utterances (2)

- Directives
  - Speaker desires a particular world state, and seeks to direct others to undertake actions to establish or maintain this world state
- Inferences
  - Statements which draw logical inferences from earlier statements
- Argumentation statements
  - Statements which question, challenge or justify earlier statements
- Control statements
  - Statements which refer to the dialog itself, aiming to facilitate or hinder communication.

*McBurney & Parsons 2004, drawing on Austin, Searle, Habermas.*

# Habermas: How may utterances be challenged? (1)

- Factual statements
  - On verifiability
- Expressive statements
  - On speaker sincerity
  - [On consistency]
- Social Connection statements
  - On normative rightness
  - [On speaker sincerity]
  - [On consistency]
- Commissives
  - On feasibility, efficacy, speaker sincerity, etc

# Habermas: How may utterances be challenged? (2)

- Directives
  - On normative rightness, feasibility, efficacy, speaker sincerity, etc
- Inferences
  - On logical validity or appropriateness
- Argumentation statements
  - On appropriateness, timing or dialectical rightness
- Control statements
  - On appropriateness, timing, consistency
  - Not usually challenged.

# Detailing these contestations

We have developed an argumentation theory for rational challenges and responses to challenges for certain types of utterances.

- For proposals for action (examples of commissives):

  K. Atkinson, T. Bench-Capon and P. McBurney [2006]:
  Computational representation of practical argument.
  *Synthese - Knowledge, Rationality and Action,*
  152 (2): 157—206.

- For commands (examples of directives):

  K. Atkinson, R. Girle, P. McBurney and S. Parsons [2008]:
  Command dialogues.
  *ArgMAS 2008.*

# Questioning and challenging proposals for action

For example, questions such as:

- Does the proposed action achieve the intended goals?
- Are there alternative ways of realizing the same goals?
- Does doing the proposed action have undesirable side effects?
- Does the execution of the proposed action preclude doing some other action which would be desirable?
- Can the desired goal be realized?
- Has the proposed action already been performed?

# Questioning and challenging commands

Questions regarding:

- The selection of the action
- The selection of Receiver to perform the action
- The authority of Commander to issue the instruction to the Receiver (normative rightness)
- Performance of the action, including questions regarding its timing and feasibility.

# Typology of Dialogs (Walton and Krabbe)

- Information-seeking dialogs
  - One party seeks some information from another party
- Inquiry dialogs
  - Two or more parties seek an answer to a factual question
- Persuasion dialogs
  - One party seeks to get another party to endorse some statement
- Negotiation dialogs
  - Participants seek to divide a scarce resource
- Deliberation dialogs
  - Participants seek to jointly agree a course of action in some situation
- Eristic dialogs
  - Participants quarrel as a substitute for physical fighting.

# Other types (not listed by W&K)

- Information-giving dialogs
  - One party seeks to provide information to another party
- Examination dialogs
  - One party seeks to assess the extent to which another party knows some factual information, or has some skill
- Discovery dialogs
  - Two or more parties seek to identify future states or actions with pre-defined valuations, eg
    - risks of some plan; business opportunities; etc.
- Command dialogs
  - One party seeks to have another party agree to execute some action.
- Etc.

# The basis of the W&K typology

- The W&K classification is based on:
  - The beliefs of the participants at the start
  - The goals or purposes of the participants at the start
  - The purposes of the dialog.

- Exercise: Examine each type against these criteria.

- D. Walton and E. Krabbe [ 1995]: *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* SUNY Press, Albany NY, USA.

# Issues with this typology

- How can a dialog have purposes?
- What if participant goals change in the course of the dialog?
- What is the relationship of participant collaboration and participant conflict in each type?
- Info-seeking, Inquiries and Persuasions are about Beliefs
- Persuasions, Negotiations and Deliberations are about Actions.

# With MAS, we always need to remember:

- Open systems will have agents designed and implemented by different teams
  - These teams may be unknown
  - These teams and their agents may have unknown beliefs, preferences and objectives.
- Unlike mainstream economists, we cannot assume that agents will be self-interested actors aiming to maximize their (short-term) utility
  - They may be malicious or whimsical, naive or stupid, or seeking to achieve some over-arching, long-term goals.
- In addition, software agents are computer programs:
  - They may contain buggy code
  - Not all observed agent behaviours may be intentional
  - We always have to allow for disruptive and non-self-interested behaviours.

# 2: Agent Language Syntax

# What are Agent Communication Languages?

- As with human languages, a means of communication
  - between independent, autonomous entities

  Functions: information transfer, action co-ordination, social manipulation, signalling.

- Programming languages for agents
  - To enable software entities to achieve their goals
    - eg, to communicate with other agents
  - We desire that agent communication ultimately be automated

- Software engineering methods
  - To enable software engineers to achieve *their* goals
    - eg, for their agents to interact with other agents
  - We desire that the level of abstraction facilitate the s/e task.

- Formal languages
  - Defined syntax
  - Defined semantics
  - We desire that properties be known and understood before deployment.

# So we draw upon:

- Linguistic theory
- The philosophy of language
- The philosophy of communicative action
- The theory of semiotics
- Social anthropology
- The theory of programming languages
- Software engineering theory
- Formal logic.

*Note that the word "semantics" has different meanings in these different fields.*

# Agent Communication Languages (ACLs)

- Two major proposals for ACLs:
  - USA DARPA's *Knowledge Query and Manipulation Language (KQML)*
    - Arose from attempts to merge multiple knowledge bases.
  - *Foundation for Intelligent Physical Agents ACL (FIPA ACL)*
- Both ACLs distinguish between two layers of communicated messages:
  - The topics of conversation (which may be represented in a suitable logical language)
  - The illocutions which refer to these topics
    - For example:
      - *query* (Is it raining?)
      - *inform* (It is raining)

# FIPA Agent Communications Language

- FIPA ACL has 22 illocutions
  - e.g. *inform, query-if, request, agree, refuse.*
  - Each has a defined syntax:

```
(inform
        :sender (agent-identifier:name j)
        :receiver (agent-identifier:name i)
        :content
            "weather (today, raining)"
        :language Prolog)
```

- The origins of FIPA ACL in knowledge-sharing and contract negotiations are apparent:
  - 11 of the 22 locutions concern requests for or transmissions of information
  - 4 involve negotiation (e.g. *cfp, propose, reject-proposal*)
  - 6 involve performance of action (e.g. *refuse, request*)
  - 2 involve error-handling of messages (e.g. *failure*).

# FIPA ACL: Locutions (1)

- **Accept Proposal (accept-proposal)** The action of accepting a previously submitted proposal to perform an action.
- **Agree** The action of agreeing to perform some action, possibly in the future.
- **Cancel** The action of one agent informing another agent that the first agent no longer has the intention that the second agent perform some action.
- **Call for Proposal (cfp)** The action of calling for proposals to perform a given action.
- **Confirm** The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition.
- **Disconfirm** The sender informs the receiver that a given proposition is false, where the receiver is known to believe, or believe it is likely that, the proposition is true.
- **Failure** The action of telling another agent that an action was attempted but the attempt failed.

# FIPA ACL: Locutions (2)

- **Inform** The sender informs the receiver that a given proposition is true.
- **Inform If (inform-if)** A macro action for the agent of the action to inform the recipient whether or not a proposition is true.
- **Inform Ref (inform-ref)** A macro action for send to inform the receiver the object which corresponds to a descriptor, for example, a name.
- **Not Understood (not-understood)** The sender of the act informs the receiver that it perceived that the receiver performed some action, but that did not understand what it just did.
- **Propagate** The sender intends that the receiver treat the embedded message as sent directly to the receiver, and wants the receiver to identify the agents denoted by the given descriptor and send the received *propagate* message to them.
- **Propose** The action of submitting a proposal to perform a certain action, given certain preconditions.
- **Proxy** The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them.

# FIPA ACL Locutions (3)

- **Query If (query-if)** The action of asking another agent whether or not a given proposition is true.
- **Query Ref (query-ref)** The action of asking another agent for the object referred to by an [*sic*] referential expression.
- **Refuse** The action of refusing to perform a given action, and explaining the reasons for the refusal.
- **Reject Proposal (reject-proposal)** The action of rejecting a proposal to perform some action during a negotiation.
- **Request** The sender requests the receiver to perform some action. One important class of uses of the request act is to request the receiver to perform another communicative act.
- **Request When (request-when)** The sender wants the receiver to perform some action when some given proposition becomes true.
- **Request Whenever (request-whenever)** The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again.
- **Subscribe** The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes.

# Classification of the 22 FIPA ACL locutions

| | |
|---|---|
| Factual statements: | 8 locutions (eg, *confirm*) |
| Expressive statements: | 1 locution (*inform)* |
| Social connection statements: | 1 locution (*inform)* |
| Commissives: | 5 locutions (eg, *propose*) |
| Directives: | 5 locutions (eg, *request*) |
| Inferences: | 1 locution (*inform)* |
| Argumentation statements: | 0 locutions |
| Control statements: | 4 locutions (eg, *not-understood*) |

*Conclusions:*

*- An absence of locutions supporting argumentation*

*- An overloading of some locutions (eg, inform).*

# Problems with FIPA ACL (1)

- The language implicitly assumes eternal connections between the agents.
  - Where are the illocutions for:
    - Entering and leaving dialogues?
    - Permitting or contesting participation?
    - Agreeing a topic of conversation?
    - Deciding rules for decision-making?
  - Only *cfp* appears to initiate a new topic of conversation.

# Problems with FIPA ACL (2)

- As befits a language for knowledge-sharing, the semantics impose sincerity:
  - Agents cannot utter beliefs they do not hold
  - But how are conflicting beliefs handled?
  - There are many situations where sincerity may not be appropriate, e.g.:
    - Negotiations (over the division of scarce resources)
    - eg The Cuban Missile Crisis (May & Zelikow 1997).

# Problems with FIPA ACL (3)

- As befits a language for contract negotiations, the underlying (implicit) argumentation theory is simplistic.
  - There are no illocutions for contesting statements, or for requesting or giving reasons for claims, or for structuring dialogue.
  - The participants incur no dialectical obligations
    - e.g. Agents which make claims need not justify them.
  - The language does not support *rational interaction*, in the sense of philosophy.

# Fatio Protocol

We have proposed additional locations for FIPA ACL to enable argumentation interactions

- assert(A, θ) – *Agent A asserts statement θ.*
- question(B, A, θ) – *Agent B asks agent A to justify statement θ*
- challenge(B, A, θ) – *Agent B challenges A to justify statement θ*
- justify(A, Δ |- θ) - *Agent A justified θ with Δ*
- retract(A, θ) – *Agent A withdraws a prior statement or justification of θ.*

# Fatio protocol (2)

These 5 locutions are subject to 6 combination rules:

- CR1: *assert* may be uttered anytime.
- CR2: *question* and *challenge* may be uttered anytime following *assert* , or following *justify*.
- CR3: Immediately following *question* and *challenge* in response to *assert* or *justify,* the agent who uttered *assert* must reply with *justify*.
- CR4: : *question* and *challenge* may be uttered anytime following *challenge*.
- CR5: Immediately following *question* and *challenge,* the agent who uttered *challenge* must reply with *justify*.
- CR6: *retract* may be made at any time following *assert* or *justify* by the same agent.

# Fatio Protocol (3)

- More to follow on this protocol.

- See:

  P. McBurney and S. Parsons [2004]: Locutions for argumentation in agent interaction protocols. pp. 209—225 in: R. M. van Eijk *et al.* (Editors): *Agent Communication. Revised Proceedings of the International Workshop on Agent Communication (AC2004).* (Springer, LNAI 3396).

# Problems with FIPA ACL (4)

- The absence of an explicit argumentation theory causes a state-space explosion:
  - Any illocution may follow any other: Disruptive behavior is not precluded.
  - This will always be a potential problem with *software* agents, since code may always have bugs.

- The language does not readily support *self-transformation*
  - How may an agent express a change of its beliefs?
    - Only by repeated utterances of *inform* locutions.
    - What distinguishes such sequences from whimsical or malicious behaviour, or from bug-ridden code?

# Problems with FIPA ACL (5)

- The private axiomatic semantics is not verifiable.
  - The internal states of agents in open agent systems are not usually available for inspection by the system.
  - In any case, a sufficiently-clever agent can always simulate any required internal state.
  - Indeed, a clever agent may adopt a belief for just so long as is needed to utter an *inform* locution sincerely.

# 3: Semantics of Agent Languages

# Human language semantics

- Recall that for human languages, the semantics of utterances concerns the relationship between

  – well-formed expressions in the syntax

  AND

  - objects and concepts in the world

- Example of a well-formed syntactical expression:
  – "The elephant in the room"

- May refer to an actual animal

  OR

  (as a metaphor) to a concept – some issue or idea which everyone knows but is not discussing.

# Recall: What are Agent Communication Languages?

- A means of communication
  - between independent, autonomous entities

  Functions: information transfer, action co-ordination, social manipulation, signalling.

- Programming languages for agents
  - To enable software entities to achieve their goals
    - eg, to communicate with other agents

- Software engineering methods
  - To enable software engineers to achieve *their* goals
    - eg, for their agents to interact with other agents

- Formal languages
  - Defined syntax
  - Defined semantics
  - We desire that properties be known and understood before deployment.

# What are the purposes of an ACL semantics?

1. To provide a shared understanding of the meaning of utterances, sequences of utterances, and dialogs to software agents engaged in communicative interactions.

2. To provide a shared understanding . . . to the human principals (if any) of such software agents.

3. To provide a shared understanding . . . to human designers of agents and human designers of agent protocols.

4. To provide a shared understanding . . . to other stakeholders (eg, regulators).
   - System owners may need to provide an account of what happens when a system is run.

# Purposes of an ACL semantics (2)

5. To enable rigorous study of the formal properties of agent languages and protocols, seeking to answer questions such as;
   - Will legal dialogs always terminate?
   - Can successful termination be achieved?
   - What are the properties of outcome states?

6. To enable different agent languages and protocols to be compared.

7. To enable agent languages and protocols to be readily implemented in production systems.

8. To help ensure that implementation of communications in open, distributed agent systems is undertaken uniformly.
   - In open MAS, ACLs function as an API.

# Purposes of research in semantics and pragmatics

- To provide a conceptual framework for the creation of different types of semantics and pragmatics for ACLs
- To analyze and compare the properties of different ACL semantics and pragmatics.
- To use the different types of semantics and pragmatics to study the properties of ACLs and protocols.

Note that:

- Software engineering is about DOING something.
- Research is about the THINKING necessary or behooving to undertake the DOING.

# Semantics of ACLs: status report

- Considerable work on defining semantics of individual utterances
- Some work on semantics of dialogues under a given protocol
- Very little work yet on semantics of protocols as a class

# Classifying Semantics: Typology 1

A classification from programming language theory:

- Axiomatic semantics
- Operational semantics
- Denotational semantics
  - including game-theoretic semantics.

# Axiomatic Semantics

- An axiomatic semantics articulates the pre-conditions and post-conditions of an utterance
  - The semantics define the pre-conditions required for an utterance to be validly made, and the post-conditions which occur upon its utterance.
  - This is usually done in a formal logical language, such as First-Order Logic or Epistemic Modal Logic.
- Example: Good Morning.

# Semantics of FIPA ACL

- FIPA ACL has been given a formal, axiomatic semantics using speech act theory, called SL

- An axiomatic semantics articulates the pre-conditions and post-conditions of each utterance

- The speech act semantics SL for FIPA ACL links utterances to the private mental states of the participants
  - Beliefs
  - Uncertain Beliefs
  - Desires, and
  - Intentions
- This semantics has been formalized using modal epistemic logic
  - Bretier, Cohen, Levesque, Perrault, Sadek (1979, 1990, 1997).

# Example – Fatio axiomatic semantics

- *assert*(A, θ ) – Agent A asserts proposition θ.
- **Pre-conditions:** Agent A desires that all other agents believe that agent A believes proposition θ.
- **Post-conditions:** All other agents believe that A desires that all other agents believe that agent A believes proposition θ.
- **Dialectical Obligations:** The proposition θ is added to the dialectical obligations store of agent A, where it remains unless and until agent A utters *retract*(A, θ). Its presence in this store means that agent A is required to provide a justification for θ when questioned or challenged.

# For example: *inform (1)*

- Suppose agent A informs agent B that "It is raining".
- Required **Pre-conditions**: Before a valid utterance by A:
  - *A* must believe "It is raining",
  - *A* must not already believe that *B* has a belief regarding whether or not it is raining
    - eg, A must believe that B has an uncertain belief about this matter

  and

  - *A* must desire that *B* also comes to believe "It is raining".

# For example: *inform (2)*

- Suppose agent A informs agent B that "It is raining".
- **Post-conditions**: Upon receipt by B of such an utterance by A:
  - B must believe that A believes "It is raining"

    and

  - B must believe that A desires that B believes "It is raining".
- Note that following the utterance by A, B may or may not adopt the belief "It is raining".

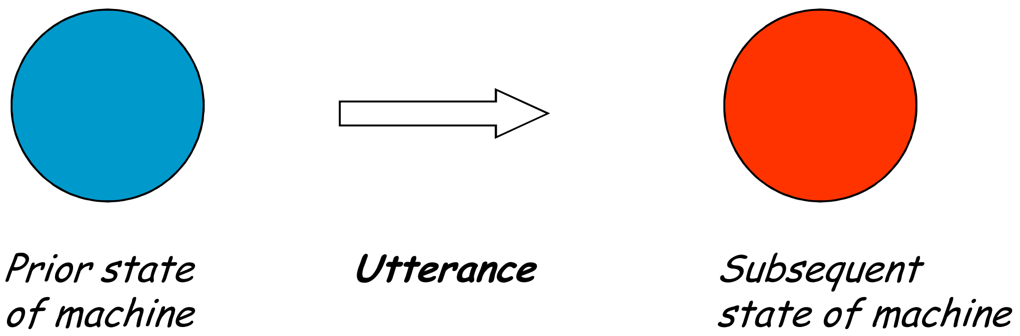
# Problems

Problems with each of pre-cons:

- *A* must believe "It is raining"
  - What about negotiation, inquiry, deliberation, discovery dialogs?
- *A* must not already believe that *B* has a belief regarding whether or not it is raining
  - What about password responses?
- *A* must desire that *B* also comes to believe "It is raining"
  - What about other intentions of *A*?
  - Recall Alice telling Bob: "The meeting is tomorrow at 17:00."

# Operational Semantics

- An operational semantics treats the utterances in an agent interaction as programming commands on some large, virtual machine
  - The commands acts to change the state of this virtual machine.
- We can therefore view the utterances as functions which cause state transitions.
- Does the virtual machine include the mental states of the interacting agents?



# Operational Semantics

# Example – Fatio operational semantics

For the utterance assert(A, θ), the following two transition rules apply:

- TR2: <A, D1, assert(A, θ)> → <A, D5, listen>
- TR3: <A, D1, assert(A,θ)> → <B, D5, D2>

where :

D1 is an internal decision procedure for an agent to decide to utter a claim

D5 is internal decision procedure for an agent to await a new utterance from another agent and then execute decision procedure indicated as the next argument.

D2 is internal decision procedure for an agent to decide to utter a question or a challenge or say nothing in response to a claim by another agent.

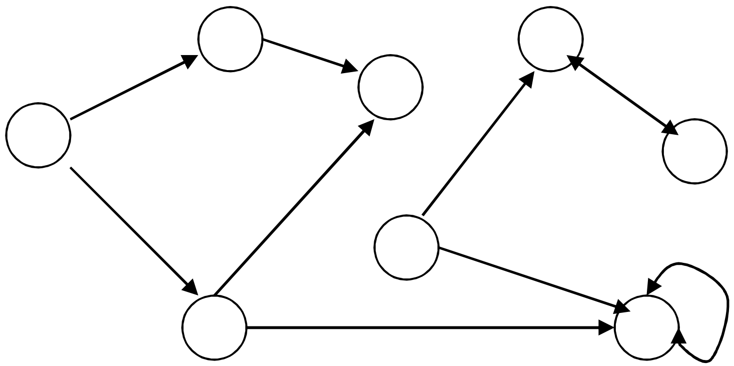
# Denotational Semantics

- Each syntactical expression is mapped to an object or objects (either real objects or objects in a mathematical space)
  - E.g. Mapping logical formulae to the set {True, False} or {0,1}.

- A well-formed proposition in the syntax of the language is evaluated as TRUE iff it describes actual objects and relationships between them. Otherwise it is FALSE.
  - Picture

# Denotational Semantics - Example

- The standard semantics for modal logic languages is the Possible Worlds semantics
  - Due to Leibniz, Kanger (1957), Kripke (1959/1962), Hintikka (1962)
  - This is a collection of states of the world, at each of which some propositions are true and some not.
  - Some worlds are connected by accessibility relationships, indicating (for example) that it is possible to move from one world-state to another.



# Game-Theoretic Semantics

- Recall truth-functional semantics
  - A well-formed proposition in the syntax of the language is TRUE if it maps to real objects in the world, and describes an actual relationship between them.
- Instead, we map each proposition to a game between two players, a Proponent (PRO) and an Opponent (OPP or CON).
  - The proposition is evaluated as TRUE iff a strategy exists for PRO to win the game regardless of the plays used by CON.
- First developed by Peirce (19th century) and then by Lorenzen (1950s) & by Hintikka (1960s).
  - Hintikka; Independence-Friendly (IF) Logic.

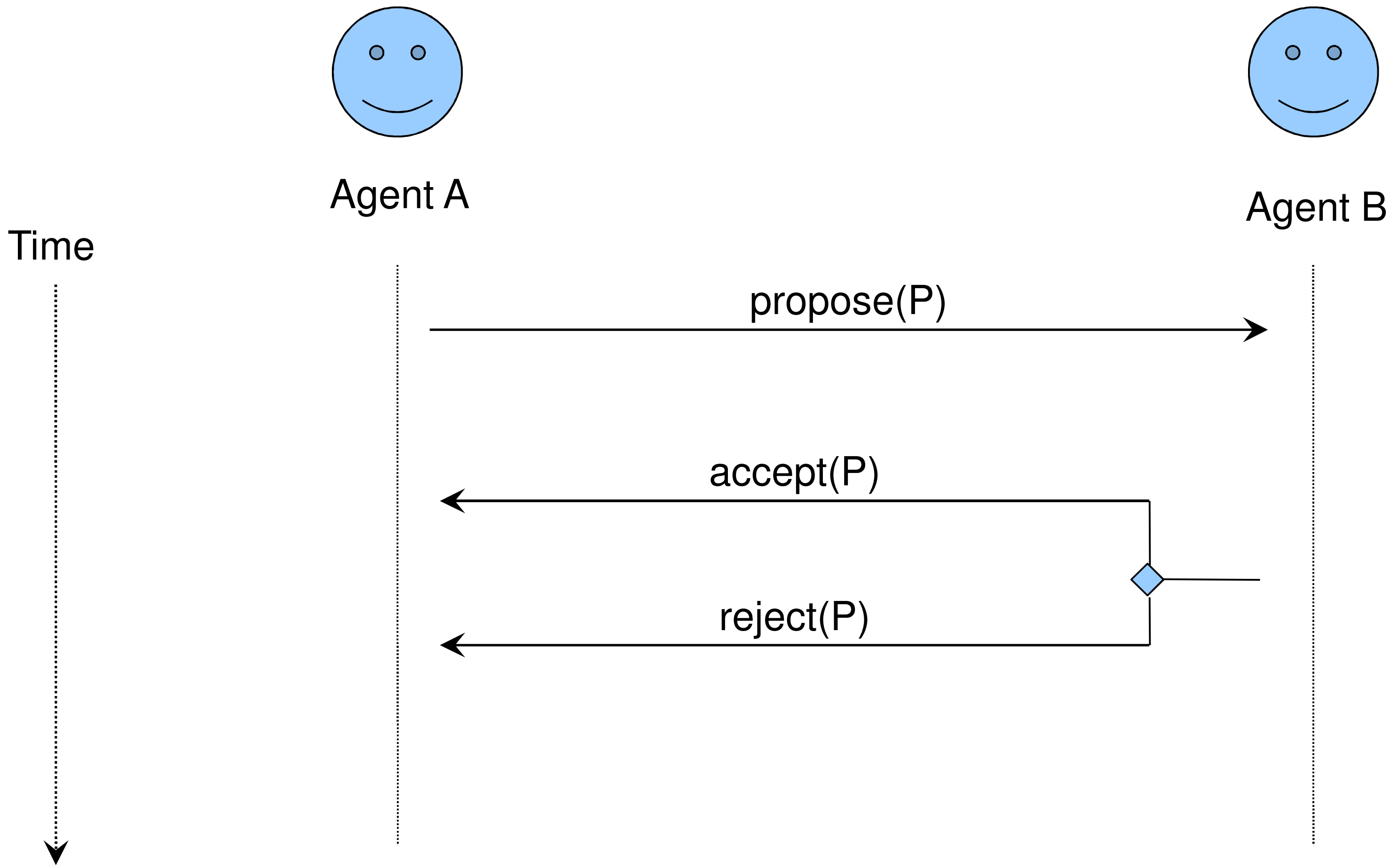
# GTS

# Classifying Semantics: Typology 2

A classification of ACL semantics in terms of the targets of semantic mappings:

- Semantics that refer to internal mental states of agents
  - eg, FIPA ACL SL
- Semantics that refer to dialectical obligations of agents in the interaction
  - eg, Contestability Semantics; game-theoretic pragmatics
- Semantics that refer to entities and relationships outside the interaction
  - eg, Social Semantics.

# Typology 2 Message Sequence Chart



# Social Semantics

- Utterances in agent dialogs potentially influence the social relationships between the agents
  - If an agent A tells lies in one dialog, other agents may no longer believe agent A in other dialogs.
  - Agent utterances may also create, alter or discharge commitments-to-action between the agents.
- This can be viewed as a form of denotational semantics
  - Agent utterances map to situations in the social world of the agents.
- Developed in various influential papers by Singh, Colombetti and their co-authors.

# Contestability Semantics

- Ensure that all statements of beliefs, preferences, and intentions by agents are allowed to be contested in interactions
- Other agents may question or challenge such statements
  - According to defined rules of interaction and dialectical obligation
- An insincere agent requires interlocking arguments to defend itself against such contestation
  - Knowing this makes insincerity more difficult for an agent, and thus should encourage sincerity
- Based on the Deliberative Democracy model in Political Theory and the notion of contestable markets in Economics.

# Semantics of ACLs

- ACLs and protocols usually defined with an axiomatic semantics for the utterances

- Some protocols have been given an operational semantics
  - Does an operational semantics have a sensible meaning in an open distributed system, when any encompassing machine is entirely virtual?

- In a handful of cases, a denotational semantics has been given for utterances in a protocol.
  - Research is still in its early stages.

# Comparing Types of Semantics (1)

- Axiomatic semantics show pre- and post-conditions of utterances, conversations, dialogs, and protocols
  - Can be used to reason about the pre-conditions and consequences of utterances, and how to achieve desired goals.
  - Can facilitate software engineering of protocols.

# Comparing Types of Semantics (2)

- Operational semantics can be used for formal analysis of interactions, eg
  - To identify non-reachable dialog states, states with gridlock, etc
  - To identify internal agent decision-mechanisms needed to participate in a dialog.
  - Can facilitate software engineering of agents.

# Comparing Types of Semantics (3)

- Denotational semantics also used to understand protocol properties
  - eg, computational complexity, convergence to truth, etc.
  - Can facilitate human understanding and software engineering of protocols.

# Comparing Types of Semantics (4)

- Semantics based on internal mental states
  - OK for closed MAS with one design team
  - But not verifiable for open MAS
- Semantics based on dialectical obligations
  - Rules often domain specific (eg, law, science)
  - So no general theory as yet
- Social Semantics based on the relationship between utterances and social relationships
  - True, but how to operationalize?
  - There seems to be large gap here: What mediates it?

# 4: Pragmatics of Agent Languages: Action Locutions

(Joint work with Simon Parsons, Brooklyn College, CUNY)

# Pragmatics

- Pragmatics is about non-semantic aspects of meaning of utterances
  - For example: the intentions of the speaker
- We can see this especially in utterances over actions
- We present a game-theoretic analysis of the pragmatics of action locutions
  - This is mostly unpublished work.

# Beliefs and actions are different

- Propositional statements may be objectively verified or justified
  - At least in principle
  - Maybe not always in practice, eg,
    - Statements about mental states or preferences
    - Statements about the past or the future.
- So beliefs about propositions may be assessed as to their truth value (true or false)
- Thus, rational agents with access to the same information should not disagree over beliefs.

# Actions

- Agents may disagree over their
  - Preferences
  - Desires
  - Goals
  - Values

- There is no objective justification possible for actions
  - Not even in principle
  - The best we can do is to conclude that an action is optimal given a particular desired world-state, and (usually) given certain other constraints.

- It is therefore possible and common for rational agents to disagree on proposed actions.

# Action locutions

- Utterances over actions have a special form
- Action locutions (aka social acts)
  - Require expression (cannot be mental)
  - Have an intended audience other than the speaker (ie, two or more participants)
  - May be accepted or rejected (acceptance called *uptake*)
  - They may be retracted/revoked/withdrawn/annulled, but who can do this differs according to the utterance.
- Theory of social acts due initially to Thomas Reid (1774) and Adolph Reinach (1913).

# Examples

- I promise you to wash the car
- I command you to wash the car
- I request you to wash the car
- I propose to wash the car
- I request you to allow me to wash the car
- I entreat you to wash the car
- Let us together wash the car
- Let us get Fred to wash the car
- etc.

# Problem for analysis

- I promise you to wash the car
- I command you to wash the car
- I XXXXXXX you to wash the car
- Same syntax, but:
  - who does action is different
  - who uptakes is different
  - who revokes is different.

# Uptake of action locutions

- Most action locutions require a response from an intended hearer before they take effect.
  - If Agent A promises to agent B to execute action θ, then this promise only takes effect if and when B accepts the promise.
- If the action locution is accepted by the intended hearer, this response is called an *uptake.*
- Any commitments to action created by an utterance only come into existence upon uptake.
  - This is what *"taking effect"* means.

# Uptake

- Who has the power to uptake (to create a commitment) ?

*Agent A speaks to agent B about action θ, which is then accepted by agent B.*

| *Agent A does θ* | *Agent B does θ* | |
|---|---|---|
| A offers to do θ | A commands B to do θ | A uptakes |
| A promises B to do θ | A entreats B to do θ<br>A requests B to do θ<br>A proposes B to do θ | B uptakes |
| A proposes that A does θ | | A or B can uptake |

# Retraction, revocation, anulment, etc

- Who has the power to revoke or retract an action locution?

*Agent A speaks to agent B about action θ, which is then accepted by agent B.*

| *Agent A does θ* | *Agent B does θ* | |
|---|---|---|
| A offers to do θ | A commands B to do θ | Only A can revoke |
| A promises B to do θ | A entreats B to do θ | Only B can revoke |
| A proposes that A does θ | A requests or A proposes B to do θ | A or B can revoke |

# Implications for semantics & pragmatics

- This view of action locutions has implications for any agent communications framework
- Any action-commitments created by an action locution only come into existence following uptake
  - So the meaning of action utterances must involve all parties, not just the speaker
  - The meaning of an utterance depends upon its use in dialog
- The creation of utterance meaning is incremental and non-monotonic
- Different action locutions have different conditions for creation, updating and removal of action-commitments.

# What is the FIPA semantics of action locutions?

- Locution: *propose (_ )*
  - Agent A proposes to agent B for A to execute some action θ, given certain pre-conditions.

Semantics given as as:

- Agent A informs B that, once B informs A that B has adopted the intention for A to perform action θ, and the preconditions for A performing θ have been established, A will adopt the intention to perform the action θ.

- IEEE FIPA [2002]: *Communicative Act Library Specification.* Document No. SC00037J. Standard, 2002-12-03.

# Comments on FIPA ACL *propose(.)*

- Semantics thus defined in terms of mental states of agent A
  - Including agent A's beliefs about the mental states of agent B
  - Semantics does not include mental states of agent B
- Uptake considered only indirectly
- No dialogical rules
  - eg, agent B is not required to respond to this utterance
- No consideration of retraction or revocation
- As with the SL semantics of FIPA ACL generally, the semantics is not verifiable.

# A Game-Theoretic pragmatics for action locutions

- The utterance *actionlocution* by agent A produces an action-commitment on agent X

iff

In the dialog game between agents A and B,

- The uttered proposal for action is uptaken by agent Y

AND

- Whilesolong as it remains unrevoked by agent Z.

Different types of action-locution instantiate X, Y and Z differently.

X, Y, Z ε {A, B}.

# Some examples of instantiated Game-Theoretic Pragmatics

- Promise by A to do action θ
  - Action commitment on X = A
  - Uptaken by Y = B
  - Revokable only by Z = B.
- Command by A for B to do action θ
  - Action commitment on X = B
  - Uptaken by Y = A
  - Revokable only by Z = A.
- Etc.

# Exercise

- Examine all 8 assignments of X, Y, Z ε {A, B}.

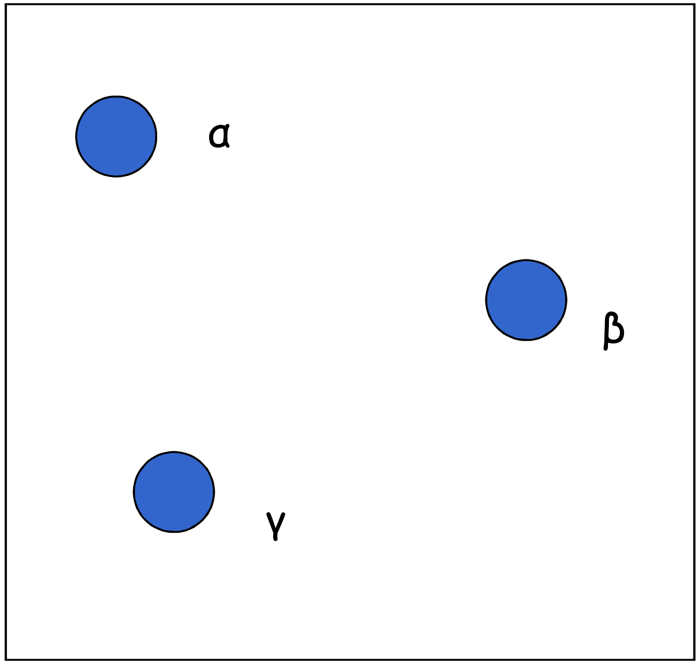
# Why a GTP?

- So, we are mapping action utterances to a game-theoretic pragmatics
- Useful for understanding the implications of utterances
- Not necessarily so useful for software implementation
- We present a denotational sematics using tuple-spaces which is better for implementation.

# A tuple-space semantics

- We can view dialogs over action as joint manipulations of action-intentionality tokens, using a Co-ordination Artefact
  - This is a blackboard-like architecture.
- Each action utterance corresponds to a token in a tuple space
- Utterances concerning the action manipulate the corresponding token.

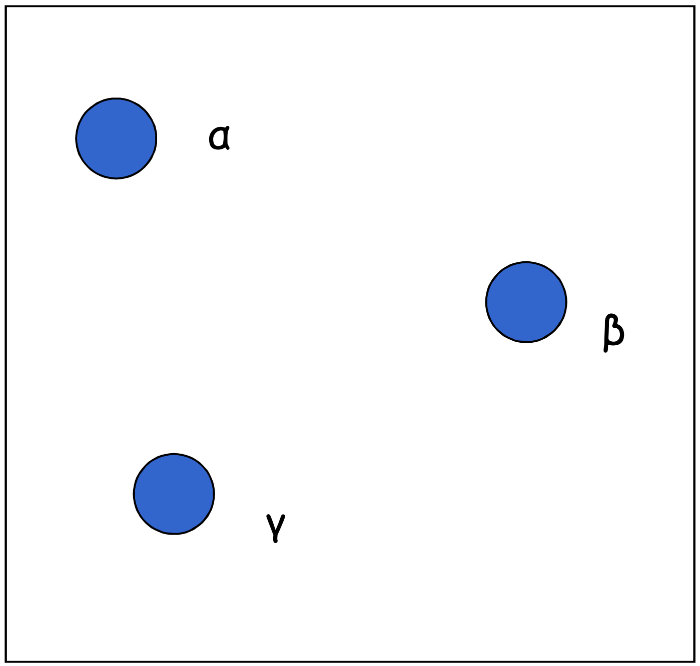
- PROMISE(agent A, agent B, α)
- COMMAND(agent B, agent A, β)
- PROPOSE(agent A, agent B, γ)



# Aside: Tuple Spaces

- *A tuple space* (aka *a Blackboard*) is a shared space to which distinct computational processes may insert, copy or delete objects, called *tuples*.
  - Linda is a language for insertion, copying and deletion of tuples.

- Tuple spaces enable sharing of objects across space and over time.



# Understanding dialogs over action

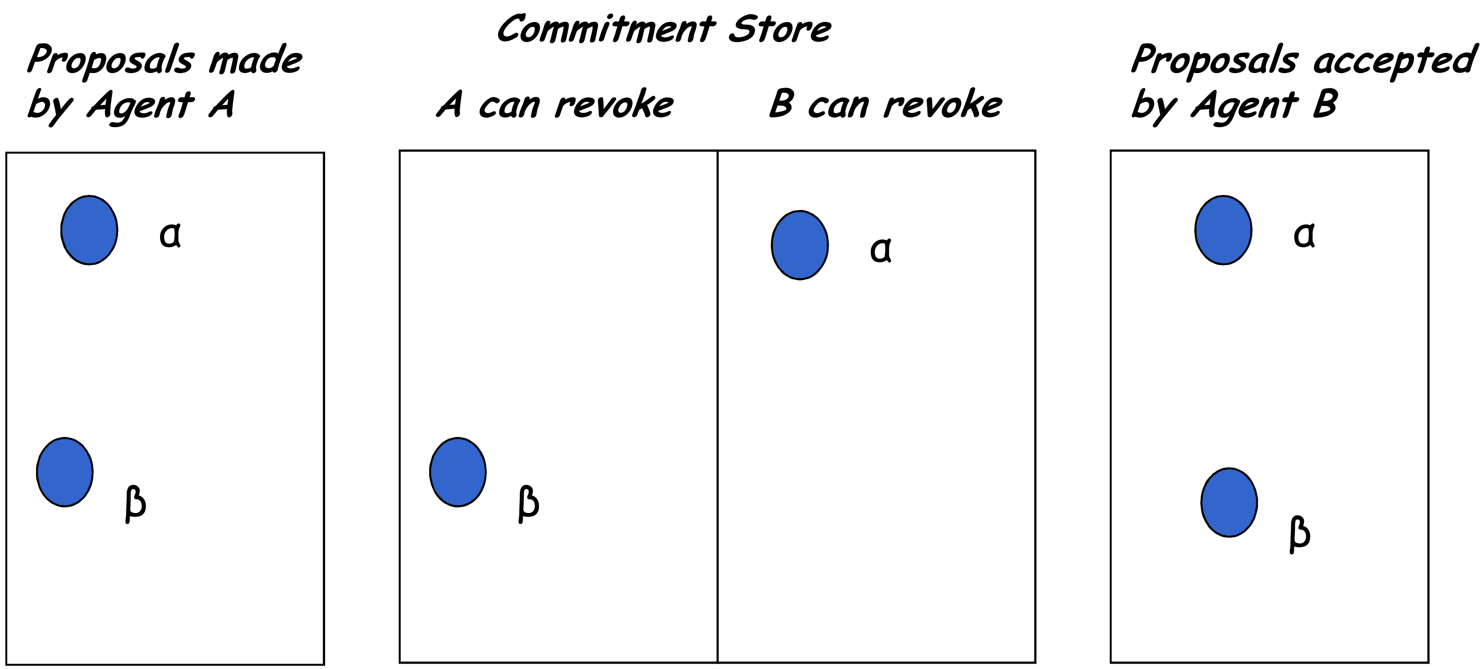
- Tokens are annotated by their dialectical status, eg
  - Which agent uttered the initial locution
  - Has the utterance been uptaken?
  - Which agent has the power to retract or annul the action intention, and thus delete the token.

- We can view the annotations as creating a partition across the tuple space.
  - Not all agents have permission to delete tokens from all sub-spaces of the partition
  - The token annotations create different read/write/copy conditions in the different sub-spaces of the partition.

# A partitioned tuple space

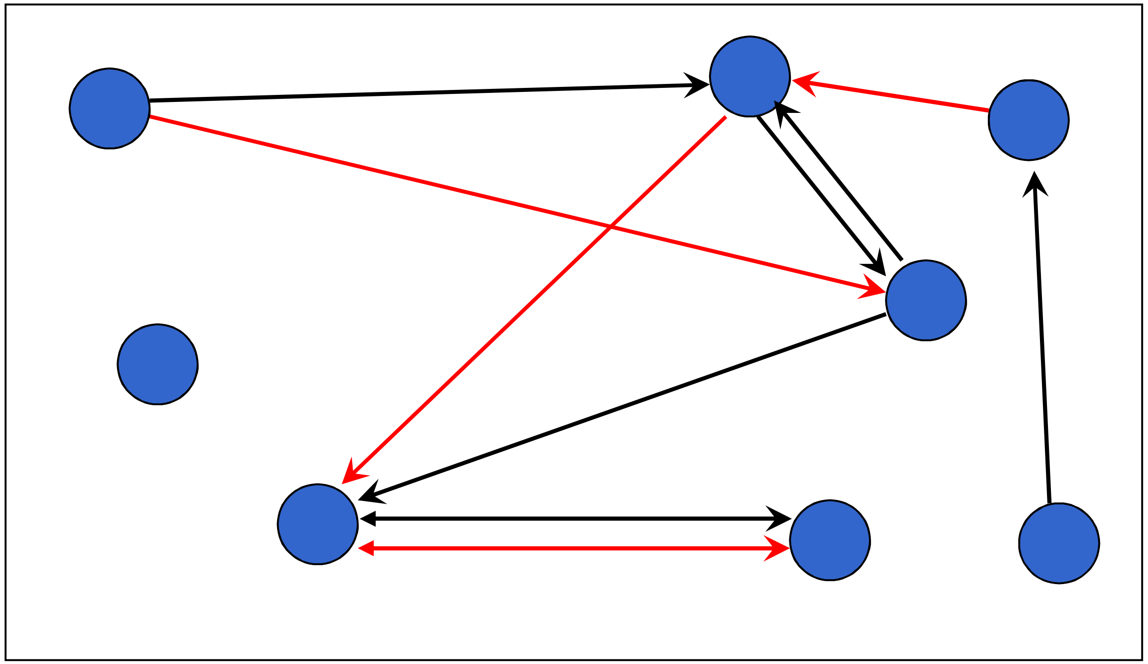
- PROMISE(agent A, agent B, α)
- ACCEPT-PROMISE(agent B, agent A, α)

- COMMAND(agent A, agent B, β)
- ACCEPT-COMMAND(agent A, agent B, β)



# A denotational trace semantics

- With some assumptions, this framework can be formalized as a collection of mathematical categories.
  - The action-tokens are denoted by objects in the categories
  - Agent preferences over the actions are denoted by arrows between the objects
- Utterances in the dialog result (variously) in the creation, copying and deletion of objects from the categories.



# Benefits of this approach

- The framework incorporates all the relevant participants, not merely the initial speaker
- The framework makes explicit the aspects of uptake and revocation
- We are able to distinguish between different types of action locutions which have identical syntax
  - e.g. proposals v. promises; proposals v. commands.
- The framework is independent of any theory of action
  - The layered modular structure means any theory of action could be readily incorporated.

# 5: Agent Interaction Protocols

# Dialogue Game Protocols

- Games between 2+ players where each "moves" by uttering locutions, subject to some rules.
  - The rules typically constrain the combination of locutions, and articulate the conditions for termination of the dialogue.
  - First developed by Aristotle and studied in the philosophy of argumentation (Hamblin, Lorenzen).
- Applied in AI and CS for the last two decades, e.g.
  - Modeling legal reasoning
  - Explaining natural language discourse
  - For the design of intelligent tutoring systems
  - Agent interactions.

# Dialogue Game Protocols for MAS

- Applied to design of protocols for interaction between autonomous software agents, engaged in:
  - Persuasion dialogs
  - Inquiry dialogs
  - Information-seeking dialogs
  - Negotiation dialogs
  - Deliberation dialogs
  - Command dialogs
  - Discovery dialogs.

# Why proposed for agent interactions?

- DG protocols are more expressive than auction mechanisms.
- Participants can typically:
  - Transmit more information, including the reasons for assertions
  - Question, contest and rebut the assertions of others.
  - Make complex commitments to each other.
- We would expect that requesting and giving of reasons for proposals should increase the possibility of successful termination, and may hasten it along.

# Why proposed for agent interactions? (2)

- DG protocols are more constrained than generic Agent Communications Languages (such as FIPA ACL)

- Under DG protocols:
  - Rules govern combinations of locutions
  - Agents cannot say just anything at any time
  - Disruptive behaviour is typically precluded.

# Relationship of different types of mechanisms

***Generic ACLs*** ***DG Protocols*** ***Auction Mechanisms***

→ *Increasing constraints on utterances*

← *Increasing expressiveness*

# Reasons for formal specification

- To model human dialogs
  - eg, Prakken KER 2006
- For the design and engineering of agent interactions
  - eg, McBurney and Parsons JOLLI 2002
- To enable formal study of agent interactions

# DG Protocols: Components of Specification

- Commencement Rules
- Legal locutions
- Rules for combinations of locutions
- Rules for creating commitments
- Rules for combining commitments
- Rules for turn-taking
- Rules for dialog termination.

# Commitment Stores

- Stores to track commitments incurred by participants in a dialog
  - Introduced by Hamblin 1970
- Dialectical commitments
  - eg, commitments to defend an assertion if questioned or challenged
- Semantic commitments
  - eg, commitments to execute some action or to maintain some state in the world outside the dialog
- Not persistent intentions
  - Another meaning of commitments in MAS

# Example

- Protocol from:
  - Amgoud, Parsons, Maudet [2000]: *Arguments, dialogue, and negotiation.* ECAI 2000.

- Intended purpose: Multi-agent negotiation protocol
- Agents have internal argumentation mechanisms

- Topics: p, q are formulas, S is a set of formulas (the argument for some claim), arrow indicates one resource is exchanged for another.
- Some utterances result in entries to the Commitment Store
  - eg, *assert(p), assert(S)*

- Termination: Winner is that agent who makes last argument.

# APM 2000: Locutions

- Legal Locutions
  - assert(p), assert(S)
  - question(p), challenge(p)
  - request(p)
  - promise(p -> q)
  - accept(p), accept(S), accept(p->q)
  - refuse(p), refuse(p->q)

# APM 2000: Combination Rules

- Not defined explicitly as rules
  - only as responses to legal locutions
- For example: responses to *assert(p)*:
  - *accept(p)*
  - *assert(-p)*
  - *challenge(p)*
- Agents must satisfy internal rationality rules to make responses
- If an agent utters *challenge(p)*, then the other agent can only respond with
  - *assert(S)*, where S is the support of the argument for p.

# Questions facing a DG protocol designer

- What locutions to include?
  - Proposals vs. Assertions
  - Retractions
  - Questions, contestations, rebuttals
  - Arguments for statements
  - Statements of preference
- What combination rules?
  - e.g. Must assertions be defended when questioned
  - e.g. Can utterances be repeated?
- When should utterances incur commitments?
  - e.g. Must speakers repeat a purchase commitment?
- What termination rules?
  - When does a dialogue terminate normally? When abnormally?

# Desiderata for DG protocols (1)

1: Define a stated purpose for the dialogue

2: Allow for a diversity of participant purposes

3: Allow for inclusiveness of participation

4: Ensure transparency of rules and structure

5: Seek fairness of rules and structure

- If not fair, any asymmetries should be known to all at the outset.

6: Encode a specific argumentation theory

- That is, a theory relating utterances to one another.

# Desiderata (2)

7: Define syntax separately from semantics & pragmatics

- Conformance with protocol syntax should be verifiable on the basis of observable linguistic behaviour alone.

8: Seek rules which are internally consistent

9: Seek rules which facilitate resolution of the dialogue

10: Seek rules which discourage disruptive behaviors

11: Ensure the protocol enables self-transformation

- Participants should be able to change their beliefs, preferences or goals in the course of a dialogue, and should be able to express these changes readily in the dialogue.

12: Seek system simplicity, consistent with the 11 criteria above

13: Minimize computational demands, consistent with the 12 criteria above.

# Evaluate protocol: Amgoud, Parsons & Maudet 2000

- Negotiation Protocol
  - L. Amgoud, S. Parsons, and N. Maudet [2000]: Arguments, dialogue, and negotiation. *Proc. ECAI-2000*, pp. 338–342. Berlin, Germany.

- Legal Locutions
  - assert(p), assert(S)
  - question(p), challenge(p)
  - request(p)
  - promise(p -> q)
  - accept(p), accept(S), accept(p->q)
  - refuse(p), refuse(p->q)

- p, q are formulas, S is a set of formulas, arrow indicates one resource is exchanged for another.

# Evaluate FIPA ACL

# Evaluation of FIPA ACL (2)

- FIPA ACL seems to be based on a *"rational choice"* model of joint decision-making.
  - In this model, participants express pre-defined beliefs and preferences in furtherance of pre-defined goals.
- There is limited capability to question, contest or rebut the beliefs, preferences or goals of other participants.
  - And, there are no obligations on agents to defend their statements against challenges.
- There is limited capability to express changes in beliefs, preferences or goals.
  - Self-transformation is limited.
- Rules do not encourage resolution, nor discourage disruption.

# DG protocol designer checklist

- Embody a formal and explicit theory of argument
- Ensure that objectives of a dialogue are stated at the outset
- Include locutions for:
  - Formally enter & withdraw from the dialogue
  - Request & provide information
  - Request & provide reasons for assertions
  - Challenge & defend assertions and arguments
  - Retract prior assertions
  - Make tentative proposals
  - Express degrees of belief, of acceptability or of preference.
- Ensure the syntax is defined in terms of observable linguistic behaviour
- Seek rules to preclude disruptive behaviour
- Define explicit termination conditions
- Define explicitly any differences in participant roles.

# Representing proposals for action

- Representing proposals for action

  $\pi$: R → S ➔ G 🠅 v

  In current circumstances R, doing action $\pi$ will lead to new circumstances S, in which state goal G is true, and being true, this will promote desirable value v.

- Similarly for undesirable values w:

  $\pi$: R → S ➔ G 🠇 w

- Atkinson, Bench-Capon and McBurney [2006]: Computational representation of practical argument. *Synthese*, **152 (2):** 157-206.

# Classification of dialogs over action

- Given R, find π, S, G and v: Deliberation dialog
- Given R and G, find π: AI Planning
- Given R, G and v, find π: Value-based planning
- Given R, S, G and v, find π: Discovery dialogs
  - Given R, S, G, ↑ and v, find π: Search for opportunities
  - Given R, S, G, ↓ and v, find π: Search for risks
- Given π, S, G and v, find R: Feasibility assessment
- Given R, S, find π: Negotiation or Deliberation dialog
- Given R and π, find S, G and v: Action evaluation
- Given R, π, G and v, find S: Assessment of outcomes

# When are two protocols the same?

- We now examine several notions of protocol equivalence.

- Syntactic equivalence
  - Two protocols are syntactically equivalent if they have identical locutions and rules.
  - Two protocols may have identical properties but not be syntactically equivalent
    - e.g. a French version of a protocol defined in English.

# Bisimulation equivalence

- Some locutions have referents in the world external to the dialogue
  - e.g. Factual statements, Requests and Promises
  - Assume these external referents are represented by formulae in the set of discussion topics.
  - Then, for each such locution, we can identify the subset of the discussion topics invoked by it.
  - This is the commitment set of the locution.
  - We can consider a dialogue as a sequence of transitions of commitment sets.
- (Strong) Bisimulation equivalence
  - Two protocols are bisimulation equivalent if the transition in commitment sets effected by a locution under one protocol is the same as by some locution under the other protocol.
  - i.e. Any transition under one protocol can be simulated under the other.

# Final-state equivalence

Perhaps we are only concerned with the state of commitments at the end of a dialogue.

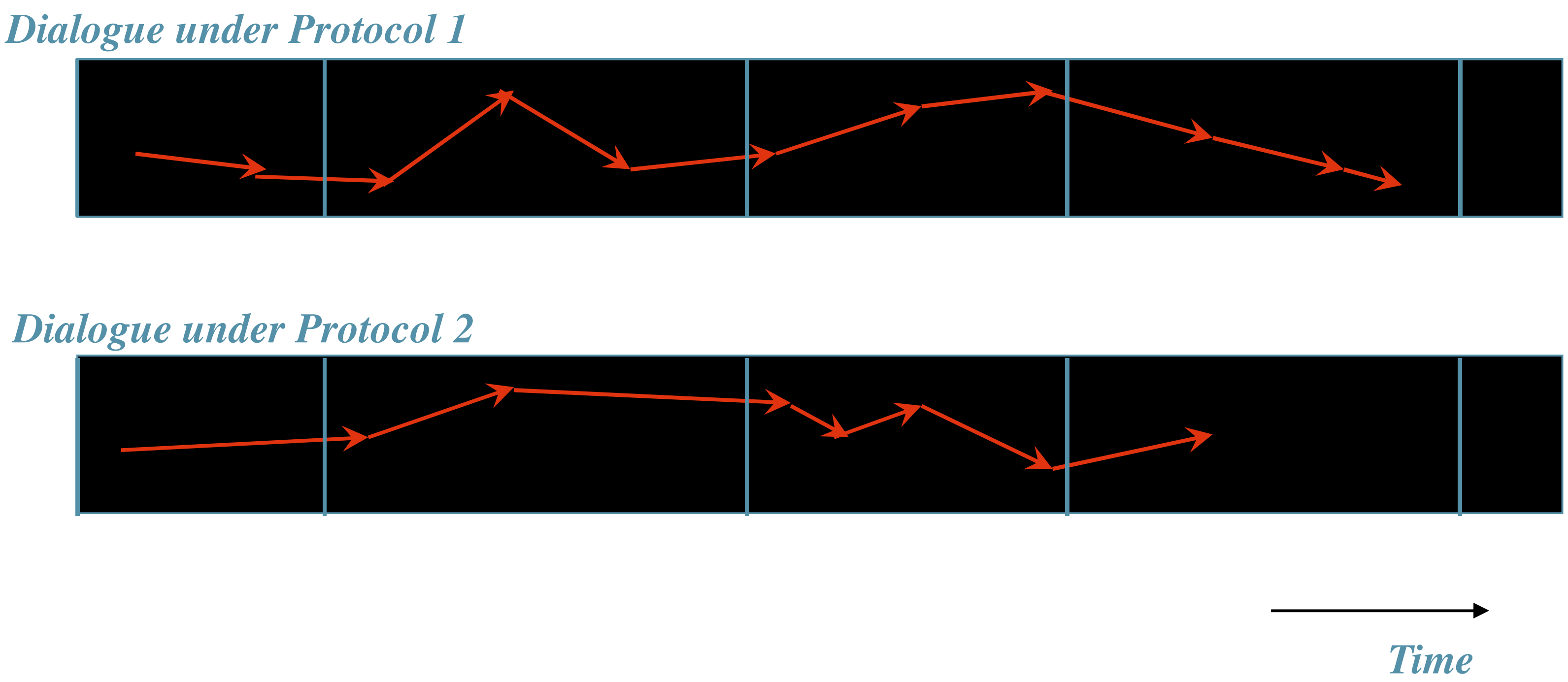
- Final-state equivalence
  - Two protocols are final-state equivalent if the final outcome achieved by a terminating dialogue under one protocol can also be achieved by a terminating dialogue under the other protocol.
- Equal-length final-state equivalence
  - Two protocols are equal-length final-state equivalent if for every terminating dialogue under one protocol there is also a terminating dialogue under the other protocol with the same final outcome and having the same number of utterances.

# Similar length dialogues

Perhaps equal-length is too strict: it may be enough for protocols to be *similar* in length.

- We assume time is represented by the positive real line: [ 0, 1 ).
  - We then divide this interval into a finite or countable partition T of sub-intervals:
  - [0, x) [x, y) [y, z) . . . . .
- T-similar final-state equivalence
  - Two protocols are T-similar final-state equivalent if for every terminating dialogue under one protocol there is also a terminating dialogue under the other protocol with the same final outcome and ending in the same sub-interval of partition T.

# Dialogues under T-similar protocols

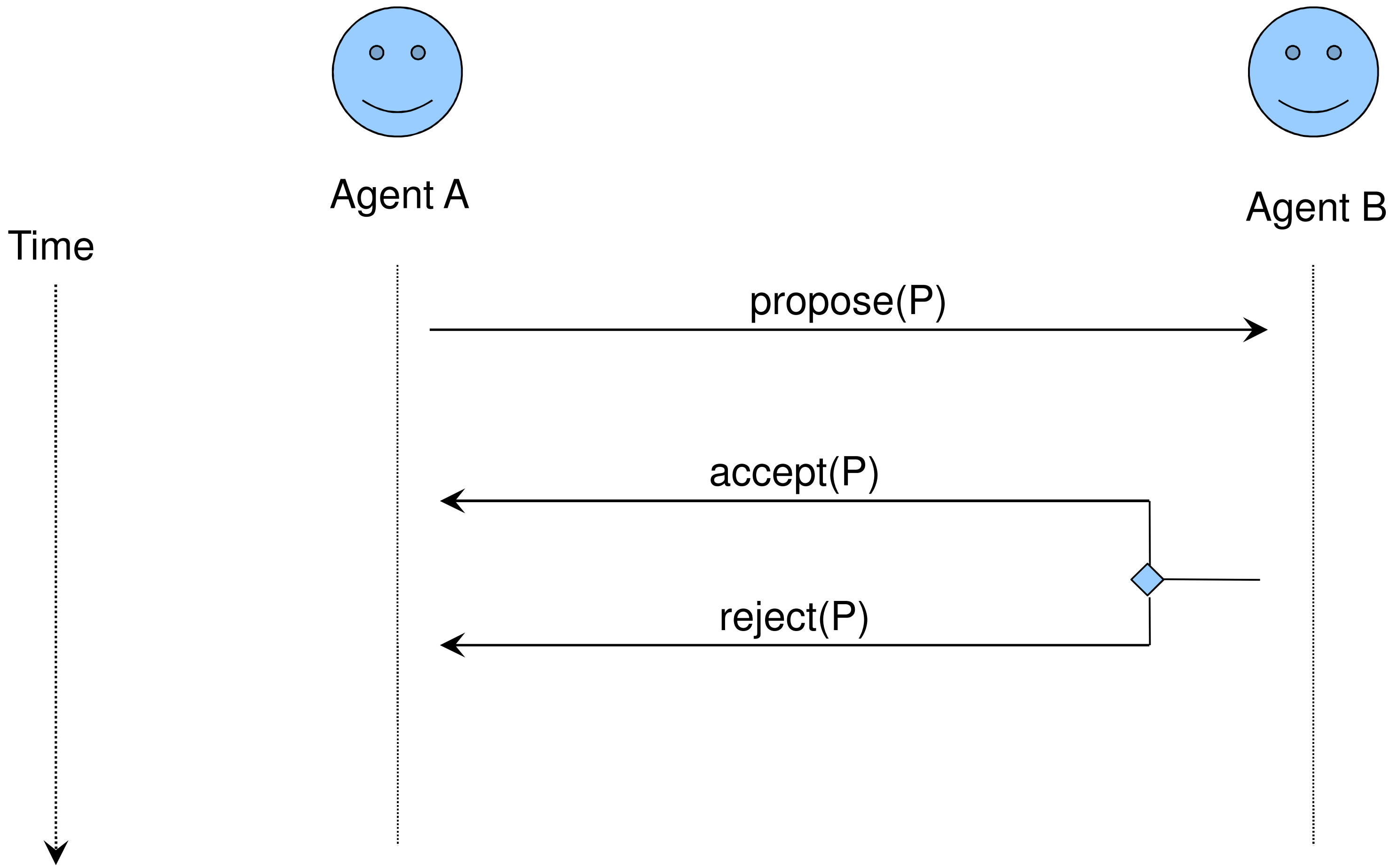



*Arrows represent locutions.*
*Vertical lines represent partition of time.*

# 6: Protocol Libraries

(Joint work with Tim Miller, University of Melbourne)

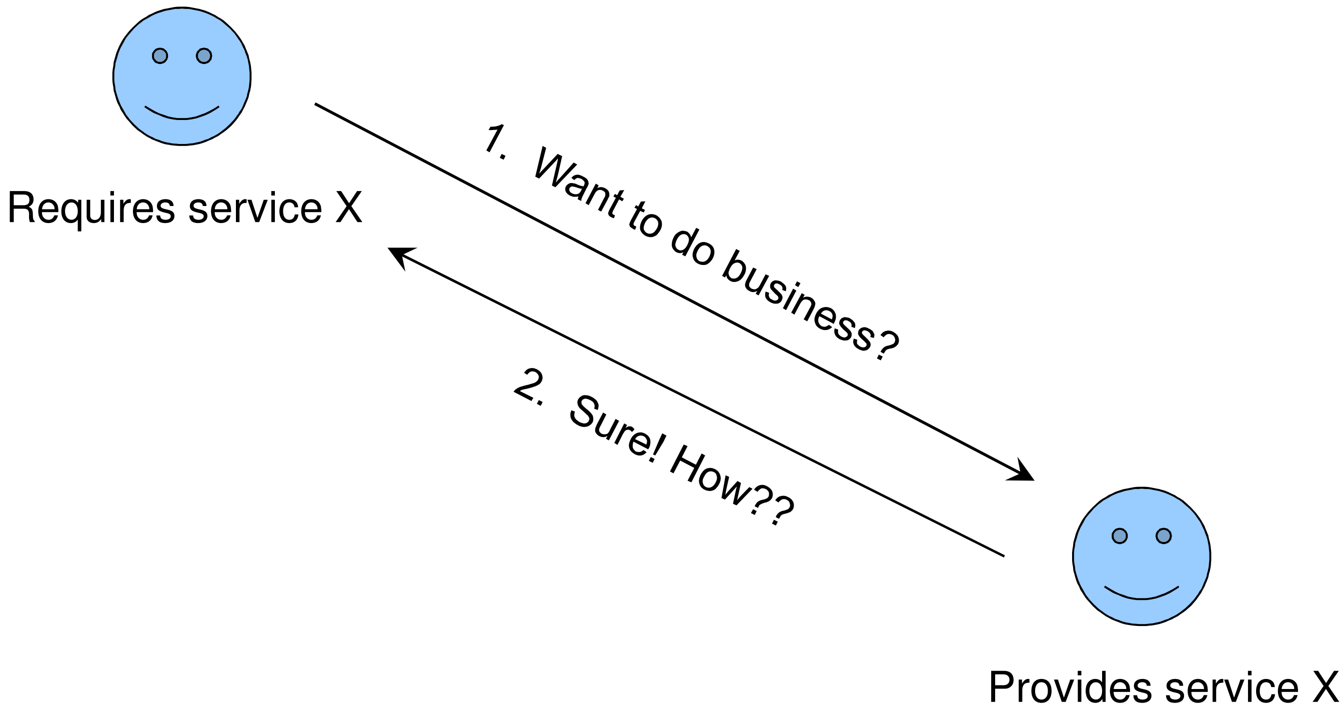
# Agent interaction protocols



# Hard-coded interaction

```
if receive(A, B, propose(P)) then
        process(P, Q);
        if Q is ... then
                send(B, A, accept(P))
        else
                send(B, A, reject(P))
        fi
fi
```

# Hard-coded interaction

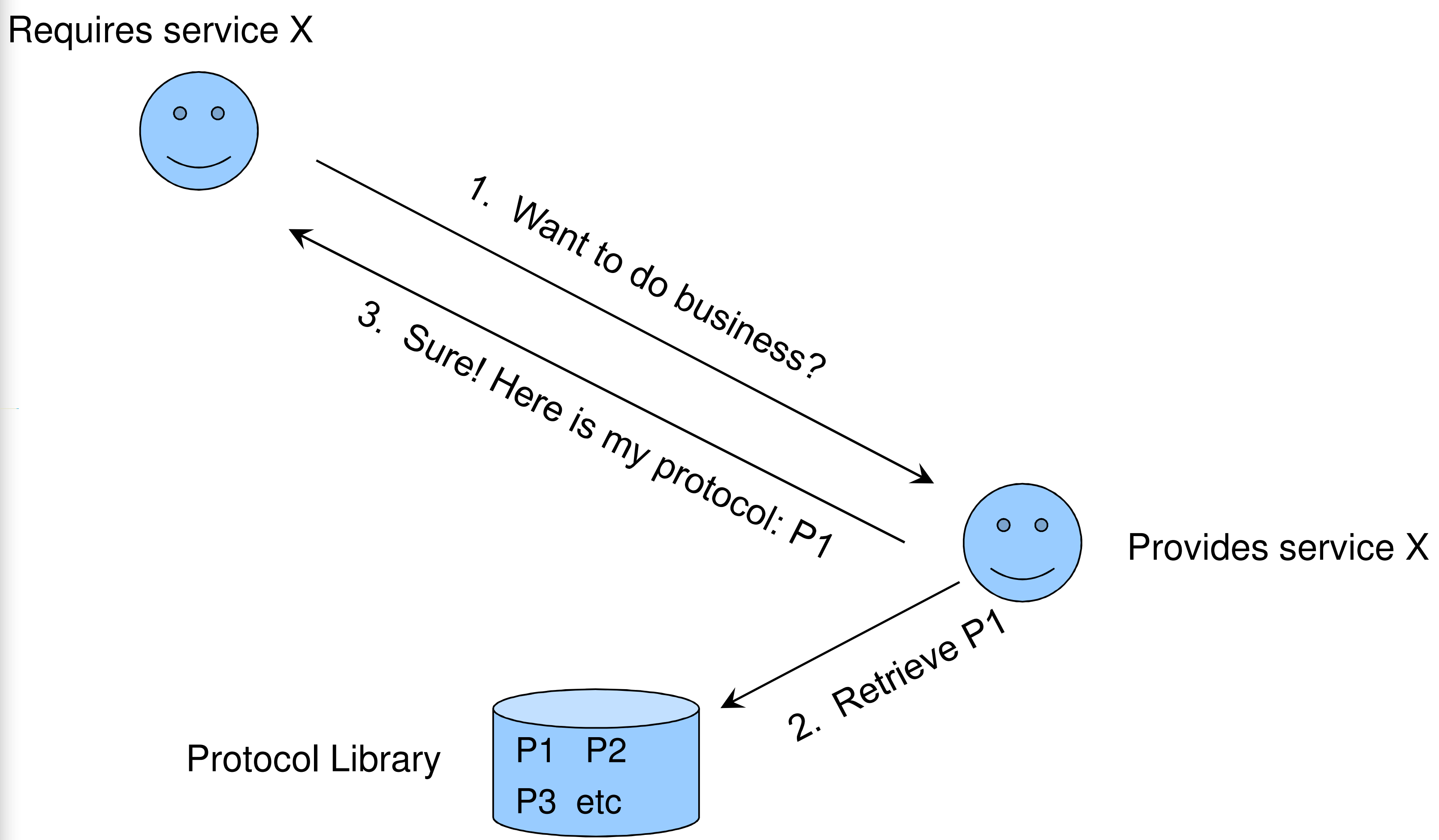


# Hard-coded interaction

Negatives
- Strong coupling between agents and protocols
- Reuse and composition not encouraged
- Inflexible

Positives
- Initial implementation overhead is small
- Developers understand readily
- Allows bottom-up prototyping/development

# First-Class interaction



# First-class protocols

## Positives

- Promotes decoupling of agents from protocols
- Protocols selected and composed dynamically (at run-time)
- Protocols can be shared

## Negatives

- Initial implementation overhead quite large
- Agents must have some reasoning capability
- Proliferation of protocols

## So, we desire protocols which can be

- Referenced
- Inspected and reasoned-over
- Composed
- Shared
- Invoked

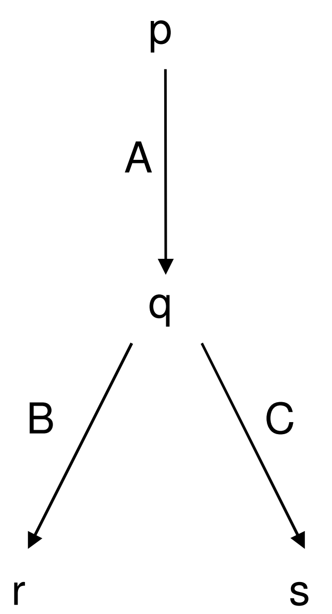
*all at run-time.*

# Protocol libraries

- Goal-directed agents could search a library of protocols to select that protocol which best achieves a desired goal
  - But: calculating protocol outcomes is costly in terms of time and resources
- *Annotation* of protocols provides an alternative
  - Annotate each protocol with possible protocol outcomes
- Could also annotate with other tags, for example:
  - Efficiency of protocol
  - Protocol provenance
  - Correctness proofs, etc.

# Annotation

Protocol X




Sub-protocols: A, B and C

Pre-cons & outcomes: p, q, r, s

Annotations

p : [A] q

[A; B] r

[A; C] s

<A; (B U C)> r

<A; (B U C)> s

q : [B] r

[C] s

<(B U C)> r

<(B U C)> s

# Rules for annotation

- For example:

  For protocols A, B, and A U B (A choose B)

  If we can annotate: [A] p

  Then we can also annotate: <A U B> p

See:


- T. Miller and P. McBurney [2008]: Annotation and matching of first-class agent interaction protocols. *AAMAS 2008.*
- T. Miller and P. McBurney [2010]: Characterising and matching iterative and recursive agent interaction protocols. *AAMAS 2010.*

# 7: Conclusions

# Agent communications

Factors to consider in designing ACLs

- Locutions and Utterances
- Conversations
- Dialogs
- Protocols
- Roles
- Norms
- Institutions

# Research Challenges: ACLs in general

- We don't have a good framework for all the above aspects yet
- Semantics and pragmatics
  - Relationships between different semantic/pragmatic frameworks
  - When different frameworks are appropriate
- A mathematical theory of protocols
- Implementation issues
  - eg, Shifting decisions from design-time to run-time.

# Research challenges: Protocol Libraries

- Other possible annotation methods for protocols
  - Efficiency of protocol
  - Provenance of protocol (eg, IPR)
  - Correctness proofs.

- How do these various annotation types interact?

- How best to annotate, store, and retrieve protocols from protocol libraries efficiently and effectively?

# Further reading

- www.fipa.org

- L. Amgoud, S. Parsons, and N. Maudet [2000]: Arguments, dialogue, and negotiation. *ECAI 2000*, pp. 338–342. Berlin, Germany.

- K. Atkinson, T. Bench-Capon and P. McBurney [2006]: Computational representation of practical argument. *Synthese*, **152 (2):** 157-206.

- M-P. Huget (Editor) [2003]:
  *Communication in Multi-Agent Systems: Agent Communication Languages and Conversation Policies.* Springer, LNAI 2650.

- P. McBurney and S. Parsons [2004]: Locutions for argumentation in agent interaction protocols. pp. 209—225 in: R. M. van Eijk *et al.* (Eds): *Agent Communication (AC2004).* Springer, LNAI 3396.

# Reading (2)


- P. McBurney & S. Parsons [2007]: Retraction and revocation in agent deliberation dialogs. *Argumentation*, 21 (3): 269—289.

- P. McBurney and S. Parsons [2009]: Dialogue games for agent argumentation. Chapter 13 in: I. Rahwan and G. Simari (Editors): *Argumentation in Artificial Intelligence*. Berlin, Germany: Springer, pp. 261-280.

- T. Miller and P. McBurney [2008]: Annotation and matching of first-class agent interaction protocols. *AAMAS 2008*.

- T. Miller and P. McBurney [2010]: Characterising and matching iterative and recursive agent interaction protocols. *AAMAS 2010*, pp. 1207-1214.

# Reading (3)

- H. Prakken [2006]: Formal systems for persuasion dialogue. *The Knowledge Engineering Review,* 21: 163-188.

- D. Walton and E. Krabbe [1995]: *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* SUNY Press, Albany NY, USA.

- M. Wooldridge [2000]: Semantic issues in the verification of agent communication languages. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(1): 9-31.

Thank you!