6CCS3AIN: Week 8 Argumentation & Dialog – Machine Languages

Peter McBurney
Department of Informatics
King's College London
London

peter.mcburney@kcl.ac.uk

Outline

- Software agents
- Agent Communication Languages (ACLs)
- Syntax of ACLs
- Semantics of ACLs
- Dialog Game Protocols

Software Agents

We are on the verge of a revolution . . .

- Computational devices and systems will soon be:
  - Everywhere
  - Interconnected
  - Always active
  - Intelligent and autonomous.

- Software systems will thus be:
  - Situated
    - Responsive to and influential upon their environment
  - Open
    - Entities will enter & leave these environments continually
  - Autonomous
    - Entities and systems will be goal-directed and exhibit autonomous behaviours
    - Systems and sub-systems will have multiple threads of control.

Implications for software engineering

A revolution will be needed for software engineering:

- Robust Systems
  - Robust to: changes in their environment, new entities, new relationships between entities, new goals, etc.
  - Designing for system-level properties:
    - Engineers need to generate desired system behaviours without designing all the components, and without knowledge of all the initial states and possible interactions.
  - Living Systems
    - Systems need to be tested, maintained and upgraded without removal, or even without downtime.

Implications for software engineering

- Software engineering starts to resemble:
  - Economic mechanism design (the design of marketplaces)
  - Ecology and ecosystem management
  - Statistical physics.

Reminder: Autonomous intelligent software agents

It helps to conceive of computer systems as consisting of interacting autonomous entities ("agents").

- A software agent is a computational entity with (some degree of):
  - Social awareness
  - Proactive behaviour towards defined goals
  - Reactive behaviour in response to its environment
  - Decision-making autonomy. *(Wooldridge & Jennings 1995)*

- Agents v. Objects (Object-Oriented Programming)
  - Agents are autonomous (they are requested to act, not invoked)
  - A multi-agent system (MAS) has multiple, interacting threads of control, not one.
  - Agents have dynamic relationships with each other, not static.

Some applications

- Air Traffic Control systems
  - aircraft and controllers
- E-commerce
  - buyers, sellers, auctioneers
- Provisioning of complex products and services
  - eg, telecommunications services
- Logistics and management of fleets
  - vehicles, satellites, sensor networks, drones, etc.
- Conceptualization of distributed software systems.

Agents direct the second economy

- Parallel to the physical economy, we now have a second economy
  - Digital, Internet-enabled
  - With machine-to-machine communications

- The second economy increasingly relies on interacting software entities, which are
  - Intelligent
  - Autonomous
  - Dynamic
  - Able to learn
  - Self-organizing.

Two key agendas:

- How best to architect (design and create) agents
  - The most common approach is based on the Philosophy of Intention and Rational Agency (Bratman, Pollock)
    - Eg, In the BDI model, agents are assumed have three types of mental states: Beliefs, Desires, and Intentions.
    - Considerable work has focused on formalizing these models using dialects of modal logic (epistemic, temporal, deontic, etc).
- How best to architect Multi-Agent Systems (MAS)
  - How may agents interact with one another?
  - How may they make collaborate or compete, or make joint decisions?
  - How to ensure desired system properties when other programmers design the entities involved?

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?
  - We can divide the bill equally.
- A means of establishing and maintaining social relationships
  - Let's do lunch!
- 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

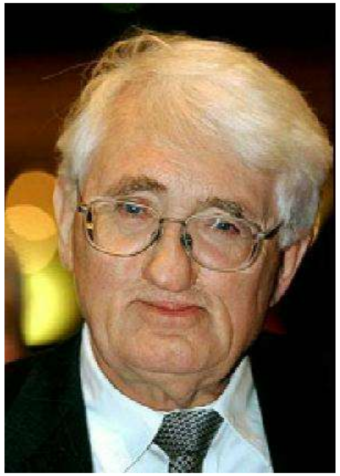
Pragmatics of a language: how the words, phrases and sentences are used in conversation.

A typology of human utterances

- Factual statements (Propositions)
  - 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
- 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 communication.

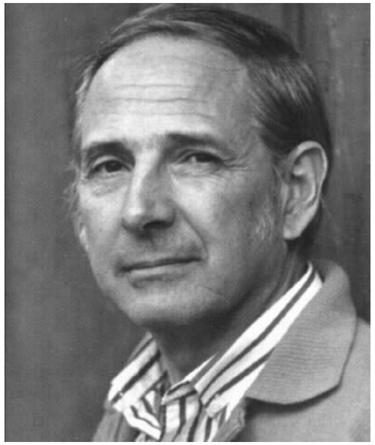
McBurney & Parsons 2004, drawing on Austin, Searle, Habermas.

Not all utterances are propositions

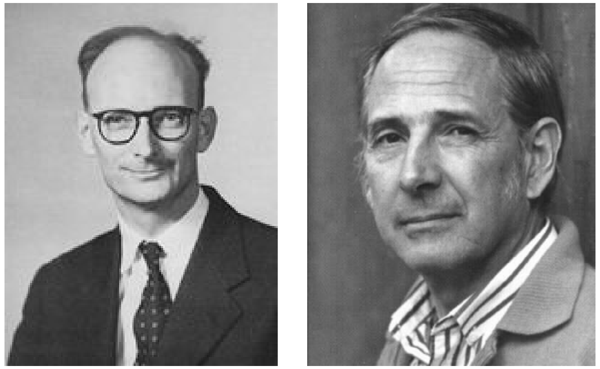


Speech Act Theory in the Philosophy of Language is an account of these other utterances

- 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, both 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.

Aside: Shannon's communication theory

- Claude Shannon developed an influential theory of communication in the 1930s and 1940s
  - Widely used in the design of electronic communications systems
- But: Shannon explicitly ignored the semantics of messages
  - This is inappropriate for agent-to-agent communications
- Example:
  - I promise you to wash the car.

  vs.
  - I command you to wash the car.

Syntactically, there is no difference between these two statements, but they have very different meanings and implications.

Agent Communications Languages

What are Agent Communication Languages?

A means of communications (as with human languages)

- 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 can 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 (signs and signalling)
- 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:
  - *Knowledge Query and Manipulation Language (KQML)*
  - *IEEE 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 utterances which refer to these topics
    - For example:
      - *query* (Is it raining?)
      - *inform* (It is raining)

FIPA Agent Communications Language

- IEEE FIPA ACL has 22 locutions (types of utterances)
  - eg, *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 are in knowledge-sharing and automated contract negotiations.

Classification of 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).

Some flaws with FIPA ACL

- As befits a language for knowledge-sharing, the semantics impose sincerity
  - For many applications, sincerity is not appropriate (eg, negotiations).
- As befits a language for contract negotiations, the underlying (implicit) argumentation theory is simplistic
  - No ability to argue, to challenge or to justify statements.
- The absence of an explicit argumentation theory causes a state-space explosion
- The language does not readily support *self-transformation*
- The private axiomatic semantics is not verifiable.

Semantics of
Agent Communications Languages

What are the purposes of an ACL semantics?

To ensure **shared understanding** of language in communication

- By agents
- By their human principals
- By the software developers of the MAS
- By the software developers of the participating agents
- By other stakeholders (eg, regulators).

To enable study of **formal properties** of the language, eg:

- Can dialogs always terminate?
- Can successful termination be achieved?
- What are the properties of outcome states?
- Can dialogs always continue?

To provide an account of **what is happening** when an ACL is used

- Eg, social semantics

To facilitate successful **software engineering & implementation**.

Semantics of ACLs

- Considerable work on defining semantics of individual utterances
- Less work on semantics of dialogues under a given protocol
- Very little work yet on semantics of protocols
- Types of semantics (from programming language theory):
  - Axiomatic
  - Operational
  - Game-theoretic
  - Denotational.

Axiomatic Semantics

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

  – What needs to be true beforehand

  and

  – What becomes true aftterwards

- This is usually done in a formal logical language, such as First-Order Logic or modal logic.

Semantics of FIPA ACL

- FIPA ACL has been given a formal, axiomatic semantics using speech act theory, called SL (“Semantic Language”)

- 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)

For example: *inform*

- Suppose agent Alice (A) informs agent Bob (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

  and
  - *A* must desire that *B* also comes to believe "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".

- Following the utterance by A, B may or may not adopt the belief "It is raining".

The FIPA semantics for *inform*

- The single most common computer interaction is a request for a password.
  - System S asks agent A to enter a password to login into system S

- Can A use the *inform* utterance to give the password?
  - inform (A, S, password= pword263)

- Three pre-conditions:
  - *A* must believe "password=pword263". YES
  - *A* must not already believe that *S* has a belief regarding whether or not "password=pword263".

  But S already knows the password, and A knows that S knows the password.

  and

  - *A* must desire that *S* also comes to believe "password=pword263".

  But what does A care about the future beliefs of S?

- So an agent compliant with the semantics of FIPA ACL could not validly use the *inform* utterance to provide a password.

Moreover . . .

Suppose Alice tells you that some proposition P is true.

What can you conclude from this fact?

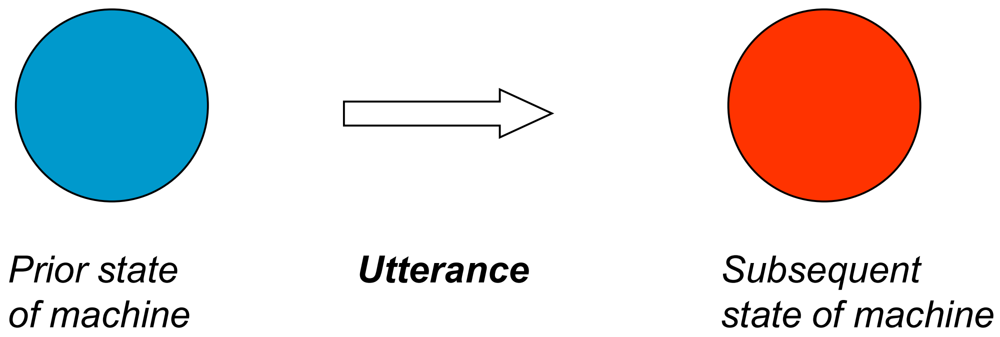
Some possible conclusions

- *That P is true.*
- *That Alice believes that P is true.*
- *That Alice wants you to believe that P is true.*
- *That Alice wants you to believe that she wants you to believe that P is true.*
- *That Alice wants you to NOT believe that P is true.*
- *That Alice wants you believe that she wants you to not believe that P is true.*
- *That Alice wants you to believe that P is not true.*
- *That Alice wants you to believe that she wants you to believe that P is not true.*
- *. . .*

Operational Semantics

- An operational semantics treats the utterances in an agent interaction as programming commands working on some large, virtual machine
  - The commands acts to change the state of this virtual machine.

- We can therefore view utterances as functions which cause state transitions.
- Does the virtual machine include the mental states of the interacting agents?

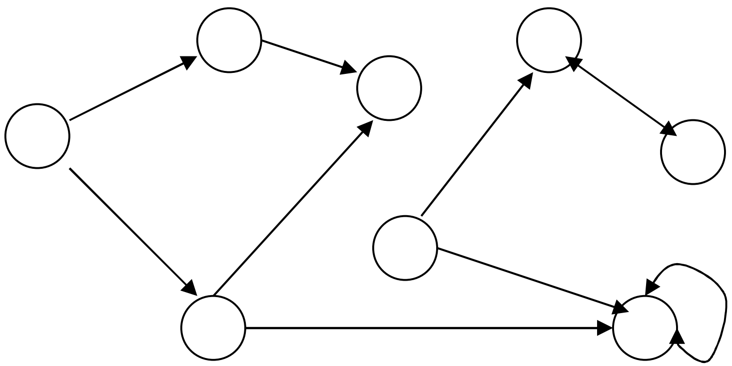


Denotational Semantics

- Each formula is mapped to some object in a mathematical space
- Having a denotational mapping means we can reason about the language by reasoning about the mathematical objects.
- Examples:
  - Possible Worlds Semantics
  - Game Semantics

Possible Worlds Semantics

- The standard semantics for modal logic languages is the Possible Worlds semantics
  - 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

- Each well-formed statement in the language corresponds to a game G, usually between two fictional players, Protagonist P vs. Antagonist A.

- Usually, we say that the goals of the players are:
  - P aims to win the game G, and
  - A aims is to prevent P from winning the game G.

- The statement is true (or valid) if and only if P always has a winning strategy for the game.

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 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.
  - For example: utterances conceived as actions on a tuple space.
  - Research is still immature.

Dialog Game Protocols

FIPA ACL's lack of structure

Problem: Lack of structure, state-space explosion.

Possible Solutions:

- Conversation Policies (CPs)
  - Sequences of utterance-patterns for a small number of utterances
    - Eg, a question must be followed by a response
    - Eg, a request for proposals must only be followed by a proposal or proposals.

- Formal Dialog Games (DGs).

Dialogue Game Protocols

- Games between two or more participants where each "moves" by making utterances, subject to some rules.

- Origins in Philosophy
  - Aristotle and medieval philosophers
  - Revived for the study of the logical fallacies in 1960s
  - Applied to quantum physics (Mittelstaedt 1979).

- Within computer science, applied to protocols for automated agent dialogues.

A DG Protocol is defined in terms of:

- A language of statements (the topics of the dialog)
- A set of utterance types instantiated with the statements
  - e.g. *assert(p)*, *accept(p)*, *contest(p)*.
- Combination rules, defining the circumstances in which each instantiated utterance may be uttered
- Termination Rules, defining the circumstances in which dialogs terminate.
- Rules for creating and combining commitments
  - Commitment Stores: publicly-accessible sets of statements, holding the commitments incurred by participants.

Relationship between types of interaction protocols

Generic ACLs

Dialogue Game Protocols

Auction Mechanisms

←

→

Increasing expressiveness

Increasing constraints on utterances

An influential typology of dialogues

Classification on the basis of what each participant knows at outset, and what they each aim to achieve in the dialog.

- Information-seeking dialogues
- Inquiry dialogues
- Persuasion dialogues
- Negotiation dialogues
- Deliberation dialogues
- Eristic dialogues.

Walton and Krabbe (1995)

Aside: HTTP

- What types of dialogs does Hyper-Text Transfer Protocol enable?

- GET
  - One agent (the client) requests another agent (the server) to transfer from server to client a copy of the object specified by URL.
  - Server responds by doing so, or by sending a message explaining why server cannot do so.
    - eg, 404 Error: File Not Found
    - eg, 503 Error: Service Unavailable

- PUT, POST
  - One agent (the client) transfers some data to another agent (the server).

- HTTP combines dialogs for:
  - Action-requests
  - Information-provision

Formal Dialogue-Game Protocols

- Agent DG protocols have been designed for all the Walton and Krabbe types, as well as:
  - Information-provision dialogs
  - Argumentation dialogs
  - Command dialogs
  - Discovery dialogs
  - Design dialogs

- These protocols are more constrained than are generic Agent Communications Languages
  - Rules govern combinations of locutions: agents usually cannot just say anything at any time.
  - Usually, the protocol is designed with a specific purpose in mind, and design may be informed by an explicit theory of argument.

A simple DG protocol for info-seeking

- Information-Seeking dialog
  - Parsons, Wooldridge & Amgoud (2002).

- For p a proposition, and S a set of propositions:

- Valid utterances:
  - question(p)
  - assert(p) assert(S)
  - accept(p) accept(S)
  - challenge(p)

- Plus: special argument *U* for assert(.) to indicate that speaker is unable to give a response.

Information-seeking dialogue (2)

- Just two agents, A and B. Valid combination rules are given by:

1. A: question(p)
2. B: Responds with one of:
   - assert(p)
   - assert(~p)
   - assert(U)
3. A: Responds with one of (respectively):
   - accept(p) v challenge(p)
   - accept(~p) v challenge(~p)
   - DIALOGUE ENDS
4. If accept(.), then DIALOGUE ENDS.
5. B: Replies to challenge with:
   - assert(S) (S an argument for p or ~p)
6. Go to 3.

Example Application: BGP

- The **Border Gateway Protocol (BGP)** is an Internet Protocol at Layer 3 (the Network Layer)
  - Allows for negotiation between neighbouring autonomous domains over routing and reachability of internet addresses.
- The **Fatio Argumentation Protocol** is an extension to FIPA ACL to allow intelligent agents to argue with one another.
  - Allows for claims to be questioned & challenged, and for claims to be justified.
- Fatio has been used to support automated identification and resolution of conflicts between hosts using the BGP.

McBurney & Parsons (2004)

Kodeswaran, Perich, Li, Joshi & Finin (2010)

Research Challenges

- Dialogs over action
- Semantics of protocols
- Properties of protocols
- Better understanding of the relationships between syntax rules and protocol properties
- How best to organize libraries of protocols (for efficient storage and search)
- Automated negotiation over protocols.

Thank you!