CmpE 593
Multiagent Systems

Pınar Yolum
pinar.yolum@boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

Agent Communication

Based largely on
Service-Oriented Computing: Semantics, Processes, Agents
– Munindar P. Singh and Michael N. Huhns, Wiley, 2004

Interaction and Communication

- Interactions occur when agents exist and act in close proximity:
  - resource contention, e.g., bumping into each other
- Communications are the interactions that preserve autonomy of all participants
- Communications can be realized in several ways, e.g.,
  - through shared memory
  - because of shared conventions
  - by messaging passing

3

Rationalistic Tradition

- 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
  - Example of water in the fridge (Winograd and Flores)
  - "John has never failed a student in Linguistics 265" (Winograd and Flores)

4

Speech Act Theory

- Speech act theory, developed for natural language, views communication as action
- It considers three aspects of a message:
  - Locution, or how it is phrased, e.g., "It is hot here" or "Turn on the air conditioner"
  - 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

5

Speech Act Theory

- Assertives: Describe the state of the world
- Directives: Attempt (in varying degrees) to make the other person do something
- Commissives: Commit the speaker (in varying degrees) to a course of actions
- Expressives: Express a psychological state (e.g., apologies).
- Declaratives: Make the content of the act match reality

6

Speech Act Theory Applied

- Classifications of illocutions motivate message types, but are typically designed for natural language
  - rely on NL syntax,
- Most research in speech act theory is about determining the agents' beliefs and intentions, e.g., how locutions map to illocutions
- For agents,
  - determining the message type is trivial, because it is explicitly encoded
  - determining the agents' beliefs and intentions is impossible, because the internal details of the agents are not known

Syntax, Semantics, Pragmatics

For message passing

- *Syntax*: requires a common language to represent information and queries, or languages that are intertranslatable
- *Semantics*: requires 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

ACL Semantics

What is the semantics of queries, requests, promises?

- *Mentalist*: each agent has a knowledge base that its messages refer to. An agent promises something if it intended to make that promise
- *Public*: semantics depends on laws, protocols, and observable behavior

Evaluation: For open systems, public semantics is appropriate, because a semantics without compliance doesn't make sense

9

Informing

How can one agent tell another agent something?

- Send the information in a message (message passing)
- Write the information in a location where the other agent is likely to look (shared memory)
- Show or demonstrate to the other agent (teaching)
- Insert or program the information directly into the other agent (master --> slave; controller --> controllee; "brain surgery")

10

Querying

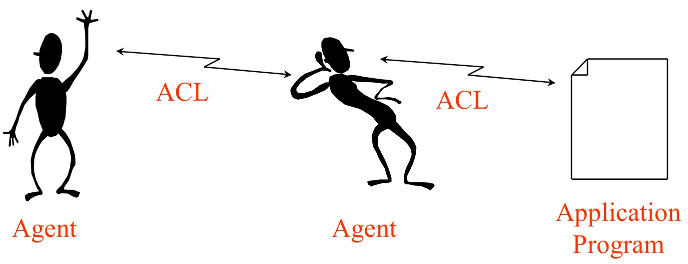
How can one agent get information from another agent?

- Ask the other agent a question (message passing)
- Read a location where the other agent is likely to write something (shared memory)
- Observe the other agent (learning)
- Access the information directly from the other agent ("brain surgery")

11

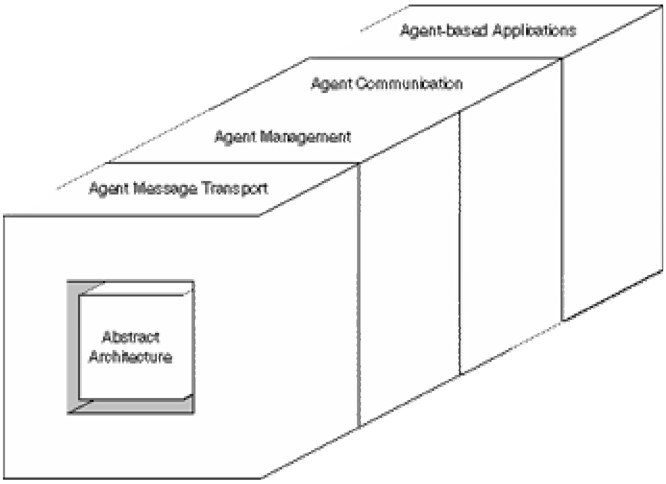
Agent Communication Languages (ACL)

- KQML: Knowledge Query and Manipulation Language
- FIPA ACL



13

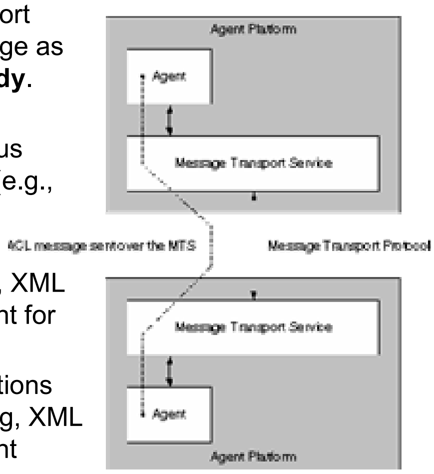
Structure of Specifications



14

Agent Message Transport

- Agent Message Transport (AMT) defines a message as an **envelope** plus a **body**. Together they handle
  - Guidelines for various transport protocols (e.g., IIOP, HTTP, WAP)
  - Message envelope representation (e.g., XML for HTTP, bit-efficient for WAP).
  - FIPA ACL represention s (e.g., string encoding, XML encoding, bit-efficient encoding).



15

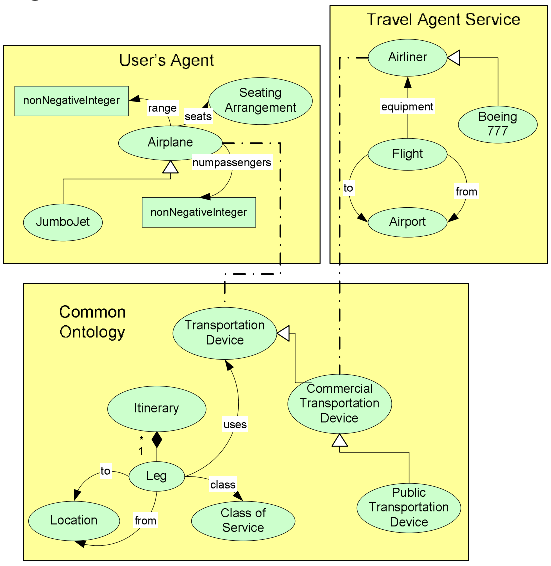
Ontology

- A specification of a conceptualization or a set of knowledge terms for a particular domain, including
  - The vocabulary
  - The semantic interconnections
  - Some simple rules of inference and logic
- Some representation languages for ontologies:
  - Unified Modeling Language (UML)
  - Resource Description Framework Language Schema (RDFS)
  - Web Ontology Language (OWL)
- Some ontology editors: Protégé, Webonto, OilEd

Common Ontologies

- A shared representation is essential to successful communication and coordination
  - For humans: physical, biological, and social world
  - For computational agents: common ontology (terms used in communication)
- Representative efforts are
  - Cyc (and Opencyc)
  - WordNet (Princeton)
  - Several *upper-level* ontologies

Ontologies and Articulation Axioms



Knowledge Representation

- Interoperability levels
  - Syntactic: parse
  - Semantic: understand
- Expressive power
- Procedural versus declarative
  - Declarative pros: enables standardization, optimization, improved productivity
  - Declarative cons: nontrivial to achieve and causes short-term loss of performance
  - Trade-offs shifted by Web to favor declarative modeling

Frames versus Descriptions

- Frame-based approaches are intuitive but rely on names of classes and properties to indicate meaning
- Description logics provide a computationally rigorous means to represent meaning; difficult for people

Managing this trade-off is a major challenge for KR

20

Mappings among Ontologies

- Term-to-term (one-to-one), e.g.,
  hookupWire$_{O1}$ ´ wire$_{O2}$
- Many-to-one, e.g.,
  solidWire$_{O1}$(x, size, color) Æ strandedWire$_{O1}$(x, size, color)
  ´ wire$_{O2}$(x, size, color, (Stranded|Solid))
- Many-to-many, e.g.,
  solidBlueWire$_{O1}$(x, size) Æ
  solidRedWire$_{O1}$(x, size) Æ
  strandedBlueWire$_{O1}$(x, size) Æ
  strandedRedWire$_{O1}$(x, size)
  ´
  solidWire$_{O2}$(x, size, (Red|Blue)) Æ
  strandedWire$_{O2}$(x, size, (Red|Blue))

21

Relations

- Hierarchies in knowledge representation
  - Inheritance (isA) relation
  - Part-whole (isPartOf) relation
- Binary relation R between S and T relates zero or more members in S to zero or more members in T
- Partial order between objects
  - Antisymmetry: If $x \prec y$ and $y \prec x$, then $x=y$
  - Transitivity: If $x \prec y$ and $y \prec z$, then $x \prec z$

22

Hierarchies

- Partially-ordered binary relations
- Taxonomy:
  - *isA* relation denotes subclasses
  - Ex: A human is a mammal
  - Antisymmetric and transitive
- Meronomy:
  - *isPartOf* relation denotes one object is a part of another object
  - Ex: A wheel is part of a car
  - Asymmetric and irreflexive

23

Modeling

- A universe of discourse (set of entities)
- Concepts that identify the entities
- Relationships among entities
  - Cardinality Constraints
  - Temporal Constraints
  - Rule Constraints
- Functions that map entities to other entities

Exercise: Which Conceptualization Has More Expressive Power?

- awg22SolidBlueWire(ID5)
- blueWire(ID5, AWG22, Solid)
- solidWire(ID5, AWG22, Blue)
- wire(ID5, AWG22, Solid, Blue)
- wire(ID5)^size(ID5, AWG22)^type(ID5, solid)^color(ID5, Blue)

Conceptualization

- Guidelines
  - Concepts must have instances
  - Inference of properties based on membership
  - Nonredundancy: Subconcepts must have one different property
- Modularity
  - Don't rewrite predicates when adding properties
  - Ex: wire(ID5, AWG22, Solid, Blue)
- Extensibility
  - Model values as objects
  - Ex: permanent (Blue) ^color(ID5, Blue)

Comparison of Modeling Languages

Feature	RDF	UML	SHOE	OEM	OIL	OWL
Object ID	+	+	+	+	+	+
Subclass	+	+	+	+	+	+
Not Subclass	−	−	−	−	+	+
Basic Typing	+	i	i	+	+	+
Multiple inheritance	+	+	+	+	+	+
Reification	+	+	−	−	−	+
Partitions	−	−	−	−	+	+
Instance Attributes	+	+	+	+	+	+
Class Attributes	−	−	−	−	+	+
Constraints: Cardinality	−	+	−	−	+	+
Constraints: Ordering	−	i	−	−	−	−
Relationships: Binary	+	+	+	+	+	+
Relationships: N-ary	−	+	i	−	−	−