Agent Communication

Jane Hsu

Intelligent Interaction

An entity is a software agent if and only if it communicates correctly in an agent communication language. ¡ *Genesereth*

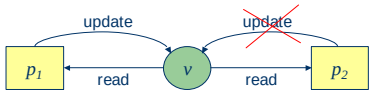
Fundamental components for interoperability

- A common language
- A common understanding of the knowledge exchanged
- The ability to exchange the above

Communicating Concurrent Systems

Main research problem [1970s-1980s]: synchronizing multiple processes

- Two processes (cf. agents) need to be synchronized if there is a possibility that they can interfere with one another in a destructive way.
- Example: lost update



Communication in OOP

Communication as *method invocation*

e.g. Object o_2 executing an instruction like

```
o1.m1(arg)
```

- Which object makes the decision about the execution of method m_1?

Object o_1 has no control over this execution of its method m_1!

The decision lies entirely with o_2.

Agent-Oriented Programming

Given agents *i* and *j*, where *i* has the capability to perform action .

Every autonomous agent has control over both its state and its behavior.

Agents can neither force other agents to perform some action, nor write data onto the internal state of other agents.

Agents can perform communicative actions in an attempt to influence other agents appropriately.

Example: Saying ¡It is raining in London¡ in an attempt to modify your beliefs (or internal state).

Speech Acts

Speech act theory treats communication as action.

- Assumption: speech actions are performed by agents just like other actions, in the furtherance of their intentions.

Communications are modeled as actions that alter the mental state of communication participants.

Theory of Speech Acts

John Austin (*How To Do Things with Words*. 1962)

- A certain class of natural language utterances has the characteristics of actions, e.g.
  - Declaring war
  - ¡I now pronounce you man and wife.¡
- Performative Verbs
  - Request
  - Inform
  - Promise

Aspects of Speech Acts

Locution ¡ the physical utterance by the speaker
e.g. *Please make some tea.*
Illocution ¡ the intended meaning by the speaker
e.g. *He requested me to make some tea.*
Perlocution ¡ the action that results of the locution.
e.g. *He got me to make tea.*

Human communication can be ambiguous.
I am cold.

- An assertion
- A request for a sweater
- A demand for an increase in room temperature

Necessary Conditions [Searle, 1969]

Normal I/O conditions. HEARER is able to hear the *request* performed in *normal* circumstances.

Preparatory conditions, which must be true of the world in order that SPEAKER correctly choose the speech act.

Sincerity conditions. Insincere performance of the act might occur if SPEAKER did not really want ACTION to be performed.

Classes of Speech Acts

Representatives: committing the speaker to the truth of an expressed proposition. E.g. informing.

Directives: attempting on the part of the speaker to get the hearer to do something. E.g. requesting.

Commissives: committing the speaker to a course of action. E.g. promising.

Expressives: expressing psychological state of the speaker. E.g. thanking.

Declaratives: effecting some changes in an institutional state of affairs. E.g. declaring war.

Plan-based Theory of Speech Acts

Question: how the properties of speech acts can be represented such that planning systems can reason about them?

[Cohen & Perrault, 1979] Speech acts are modeled in a planning system as (STRIPS) operators in the same way as physical actions.

Multi-modal logic with operators for describing the *beliefs*, *abilities*, and *wants* of the participants in the speech act.

Example: *Inform(S, H, f)*

Preconditions

Cando.pr (S Believe)
Want.pr (S Believe (S Want *informinstance*))

Effect

(H Believe (S Believe))

Example: *Request(S, H, f)*

Preconditions

Cando.pr (S Believe (H Cando))
(S Believe (H Believe (H Cando)))
Want.pr (S Believe (S Want *requestinstance*))

Effect

(H Believe (S Believe (S Want)))

- No guarantee that the desired action is performed.
- Lead to the development of theory of rational action

Knowledge Sharing Effort

Funded by DARPA in the early 1990s.

Goal: to develop protocols for the exchange of represented knowledge between autonomous information systems.

Motivation:

- Knowledge reuse
- Knowledge exchange
- Interoperable software systems

Deliverables

- Knowledge Query & Manipulation Language
- Knowledge Interchange Format

KSE Working Groups

Interlingua Group

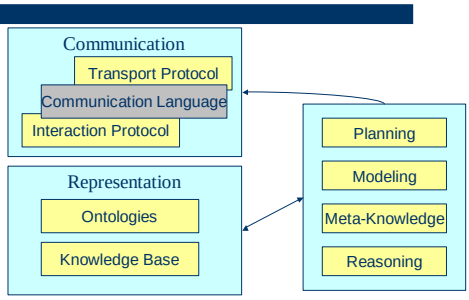
- KIF: Knowledge Interchange Format
- KRSS: Knowledge Representation System Specification

The SRKB Group

- Shared, Reusable Knowledge Bases

External Interfaces Group

Agent Communication Architecture



KSE Basics

Communication

- Interaction protocol
  - Negotiation schemes + game theory protocols
- Communication language, e.g. KQML
- Transport protocol, e.g. TCP, SMTP, http etc.

Representation

- Translation from one language into another
- Sharing semantic content of the represented knowledge among different applications

Communication Protocol

Sender
Receiver(s)
Language in the protocol
Encoding and decoding functions
Actions to be taken by the receiver(s)

KIF [Genesereth & Fikes, 1992]

First-order predicate calculus
- Properties of things in a domain
  `(vegetarian Michael)`
- Relationships between things in a domain
  `(married George Barbara)`
- General properties of a domain
  e.g. Everybody has a mother.

Lisp-like syntax
- Boolean connectives and quantifiers
- Pre-defined basic vocabulary of objects
- Standard functions for numbers, characters, and strings etc.

KIF: Examples

```
(= (temperature m1)
   (scalar 83 Celsius))
(defrelation bachelor (?x) :=
  (and (man ?x)
    (not (married ?x))))
(defrelation person (?x) :=>
  (mammal ?x))
```

KQML

A message-based language for agent communication.
- Defines a common format for messages.
- Not concerned with the message content.

KQML Performatives
http://kmi.open.ac.uk/people/emanuela/JATLiteBean/KQMLperf.html
- Pre-defined set: neither minimal nor closed
- Extensible
- standard

Description of KQML

Content layer: the actual message content
- Representation in application¡s own language
- Opaque to KQML

Message layer: the core of KQML
- Protocol
- Speech acts
- Description of the content

Communication layer: low-level parameters
- Sender identity
- Receiver identity
- Unique identifier

Types of KQML Performatives

Basic query, e.g. `ask-one`
Multi-response, e.g. `stream-all`
Response, e.g. `reply`
Generic informational, e.g. `tell`
Generator, e.g. `next`
Capability-definition, e.g. `advertise`
Networking, e.g. `register`

KQML: Sample Messages

```
(ask-one
  :content (PRICE IBM ?price)
  :receiver stock-server
  :language LPROLOG
  :ontology NYSE-TICKS)
(ask-all
  :content ¡price(IBM, [?price, ?time])¡
  :receiver stock-server
  :language standard_prolog
  :ontology NYSE-TICKS)
```

FIPA [1995]

Foundation for Intelligent Physical Agents

FIPA ACL

- Syntax: similar to KQML
- 20+ performatives
- No mandate for any specific language for message content.
- Comprehensive formal semantics
  - Speech acts [Cohen & Levesque, 1990]
  - Sadek¡s enhancements
  - SL ¡ beliefs, desires, uncertain beliefs of agents, & actions

Performatives in FIPA ACL

Accept-proposal	Propagate
Agree	Propose
Cancel	Proxy
Cfp	Query-if
Confirm	Query-ref
Disconfirm	Refuse
Failure	Reject-proposal
Inform	Request
Inform-if	Request-when
Inform-ref	Request-whenever
Not-understood	subscribe

Semantics: *inform*

<*i*, *inform*(*j*, *f*)>

feasibility precondition:

$B_i f \quad B_i (Bif_j f \quad Uif_j f)$

rational effect:

$B_j f$.

Where $B_i f$ means ¡agent *i* believes *f* ¡.

Ontology

Goal: to build knowledge-based software that can be used as off-the-shelf technology

Ontology: conceptualization of the world

e.g. objects, functions, relations

Common Ontology: an explicit specification of the ontological commitments of a set of programs

- Written in KIF
- Definitional vocabulary of *Ontolingua*, which is a language for specifying ontology

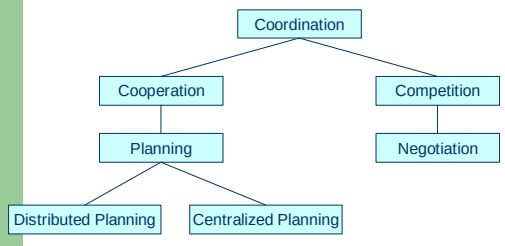
Interaction Protocols

Governing the exchange of a series of messages among agents ¡ a conversation.

Objectives:

- To maximize the payoffs (utilities) of the agents
- To maintain globally coherent performance of the agents without violating autonomy.
  - Determine shared goals
  - Determine common tasks
  - Avoid unnecessary conflicts
  - Pool knowledge and evidence

Taxonomy of Agent Coordination



Task Decomposition

Divide-and-conquer
- ¡ Spatial decomposition by information source or decision point
- ¡ Functional decomposition by expertise

Decomposition can be
- ¡ Done by the system designer
- ¡ Inherent in the problem representation

  e.g. AND-OR graph

Task Distribution

Avoid overloading critical resources

Assign tasks to agents with matching capabilities

Make an agent with a wide view assign tasks to other agents

Assign overlapping responsibilities to agents to achieve coherence

Assign highly interdependent tasks to agents in spatial or semantic proximity to minimize communication costs.

Reassign tasks if necessary for completing urgent tasks.

Distribution Mechanisms

Market mechanisms: tasks => agents
- ¡ Generalized agreement
- ¡ Mutual selection, e.g. commodity pricing

Contract net
- ¡ Announce, bid, and award cycles

Multi-agent planning
- ¡ Planning agents are responsible for task assignment

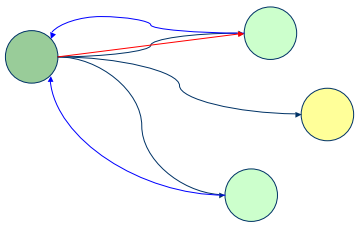
Organizational structure
- ¡ Agents have fixed responsibilities for particular tasks

Contract Net [Smith, 1980]

Manager¡s perspective
- ¡ Announce a task that needs to be performed
- ¡ Receive and evaluate bids from potential contractors
- ¡ Award a contract to a suitable contractor
- ¡ Receive and synthesize results

Contractor¡s perspective
- ¡ Receive task announcements
- ¡ Evaluate my capability to respond
- ¡ Respond (decline, bid)
- ¡ Perform the task if my bid is accepted
- ¡ Report my results

Contract Net Protocol

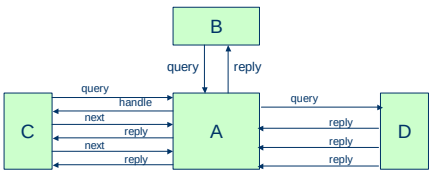


Communication Protocols in KQML

Simple client-server, e.g. A B

Synchronous communication, e.g. A C

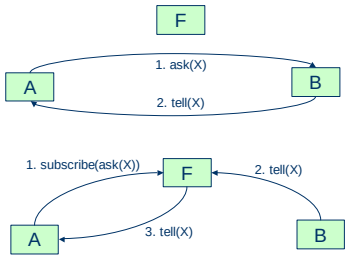
Asynchronous communication, e.g. A D



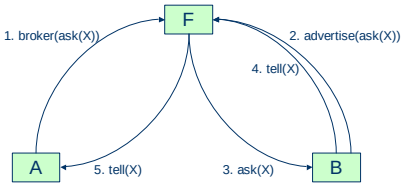
Communication Facilitators

- Maintain a registry of service names.
- Forward messages to named services.
- Route messages based on content.
- Provide matchmaking between information providers and clients.
- Provide mediation and translation.

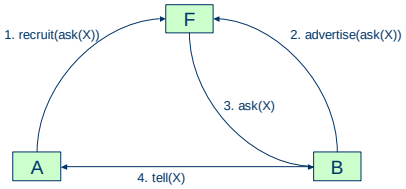
Facilitators in KQML



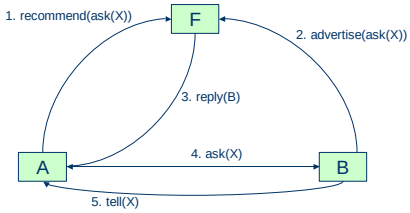
Facilitator: Broker



Facilitator: Recruit



Facilitator: Recommend



KQML Implementation

See the list at
http://www.cs.umbc.edu/kqml/software/

KQML Software

- JATLite (Stanford)
- KQML API (Lockheed)
- Jackal (UMBC)
- TKQML/Tackal