Multi-agent and Semantic Web Systems: Agent Communication

Fiona McNeill

School of Informatics

14th March 2013

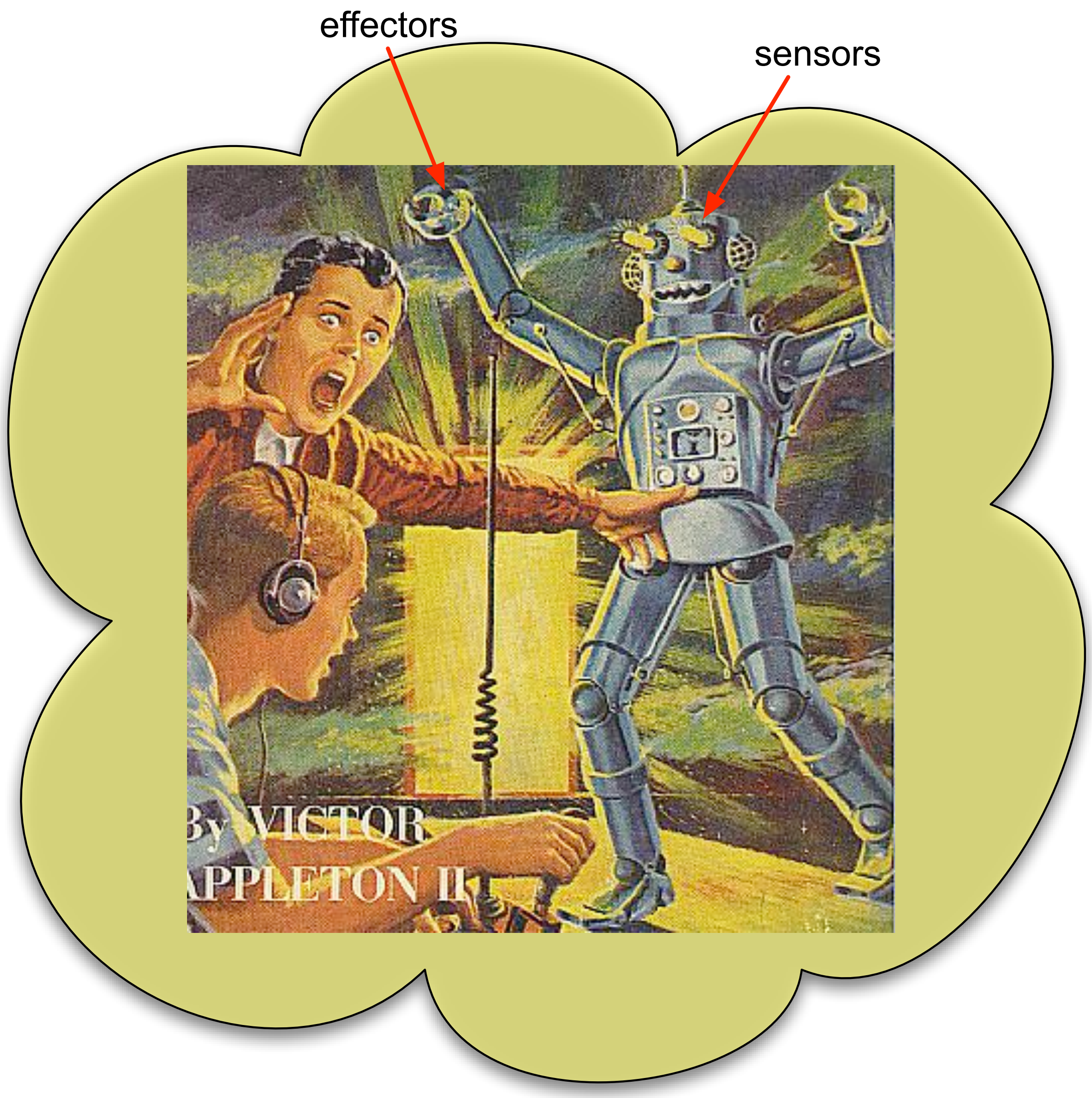
Agents and this course

- Agents can be thought of as representing the ultimate vision of the Semantic Web: fully autonomous systems dynamically interacting in complex ways.

- This lecture and the next lecture discuss what these systems would be like and would kind of technology this would require.

- The final lecture discusses how they could utilise the Semantic Web as a Multi-agent system to fully realise this goal.

What are Agents?



- Situated in an environment;
- can sense the environment, and
- perform actions.

What are Agents?

'Intelligent' characteristics of agents:

- autonomy
- reasoning ability
- learning ability
- mobility
- sociability
- cooperation
- negotiation

Autonomy

- Can initiate tasks without external intervention.
- Behaviour is based on experience as well as in-built knowledge.

Multi-Agent Systems (MAS)

- Distributed system which incorporates independent agents.
- The collection of agents interact, and
- solve problems that are outside their individual capacities.
- Focus on properties that emerge from cooperation (vs. capabilities of individual agents)
- 'standard' distributed system: coordination must be specified in advance
- Multi-Agent system: (some aspects of) coordination achieved dynamically at run-time
- Possible advantages:
  - fault-tolerance
  - scalability
  - concurrency

Agent Communication, I

Why communicate?

- Information Exchange: share knowledge;
- Collaboration: optimising use of resources and distribution of tasks, coordinating execution;
- Negotiation: reaching agreement in the presence of conflict

Agent Communication, 2

- Communication involves sharing at various levels, including
  - how to describe states of affairs (vocabulary, grammar),
  - how to carry out speech acts (performative verbs),
  - how to engage in dialogue.

- Main goal of communication: influence other agents; e.g., to make them perform actions or to make them believe certain propositions.

- Other agent decides whether to perform action or believe proposition.

Speech Acts

- Most treatments of communication in (multi-)agent systems borrow their inspiration from speech act theory.
- Speech act theories are pragmatic theories of language, i.e., theories of language use;
- attempt to account for how natural language is used to achieve goals and intentions.
- Original theory by J.L.Austin, *How to do things with words*.

Speech Act Theory: Communication as action

Speech Acts: Performatives "change the state of the world like a physical action"; everything we utter is uttered with the intention of satisfying some goal or intention

Kinds of Speech Act

- Locutionary act: the act of generating sounds that are linked together by grammatical conventions so as to say something meaningful.

- Illocutionary act: the speech act of doing something else in the process of uttering meaningful language.
  - *I will gladly pay you £1.00 next week*
  - illocutionary act of making a promise

- Perlocutionary act: the contingent effect on those who hear a meaningful utterance.
  - By telling a ghost story late at night, one may accomplish the perlocutionary act of frightening someone.

Performatives

Speech acts often divided into two components:

- performative verb: *request, inform, promise, ...*
- propositional content: *the door is closed*

ϕ = *the door is closed*

Example Performatives

REQUEST + $\phi \Rightarrow$ *Please close the door!*
INFORM + $\phi \Rightarrow$ *The door is closed.*
INQUIRE + $\phi \Rightarrow$ *Is the door closed?*

More Speech Act Theory

Searle (1972) categorization of performatives:

Assertives: informing, claiming something is true
Directives: requesting, commanding
Commissives: promising, refusing
Declaratives: making change to state of world (e.g. marrying)
Expressives: expressing mental states

Like all taxonomies, this is open to debate.

Conditions on Speech Acts

- Austin & Searle described conditions under which speech acts are successful.

- Austin's felicity conditions:
  - Has to be conventional procedure for carring out speech act;
  - procedure has to executed correctly;
  - act has to be sincere, uptake must be completed as far as possible.

- Searle:
  - Preparatory conditions must hold (e.g., requested action can be performed, speaker must believe this, hearer not already intending to perform action).
  - Sincerity conditions (speaker wants action to be performed).

DARPA Knowledge Sharing Effort (KSE)

- http://www-ksl.stanford.edu/knowledge-sharing/

- Project from early '90s aimed at developing techniques, tools and re-usable resources to support building large-scale knowledge-based systems and knowledge bases.

- Results in the area of knowledge representation:

Syntax	KIF (propositional content)
Semantics	Ontolingua (language for defining sharable ontologies)
Pragmatics	KQML (high-level interaction language)

ACLs

- KIF and KQML are both Agent Communication Languages (ACLs)

- KQML (Knowledge Query and Manipulation Language). 'Outer' language; defines message envelope format.

- KIF (Knowledge Interchange Format). 'Inner' language; defines message content.
  - Essentialy a LISP-based notation for First Order Logic.
  - Intended as interlingua for encoding declarative knowledge.
  - Features: semantics independent of implementation; non-decidable, 'human readable'
  - Current specification at http://logic.stanford.edu/kif

KIF

Possible to express:

• properties of things in a domain (*John has a mobile phone*, *Mary has short hair*)

• relationships between things in a domain (*Luke is married to Jane*)

• general properties of a domain (*every student has a matriculation number*)

• Standard vocabulary for data structures (numbers, strings etc., complex objects, lists etc.) and methods on them.

KIF Examples

The temperature of m1 is 83 Celsius

```
(= (temperature m1) (scalar 83 celsius)
```

X is a bachelor iff X is a man and is not married

```
(defrelation bachelor (?x) :=
  (and (man ?x)
  (not (married ?x))))
```

If X is a person then X is a mammal

```
(defrelation person (?x) :=> (mammal ?x))
```

KQML

- Knowledge Query and Manipulation Language (KQML).
- High-level, message-oriented language for information exchange.
- Makes no assumptions about:
  - transport mechanism;
  - content language;
  - ontology used in message content.
- Message types particularly oriented towards multi-agent communication.

KQML Message

Example KQML Message

```
(tell :sender        amazon.com
      :receiver      info-agent0011
      :in-reply-to   msid-7.24.97.45391
      :ontology      http://amazon.com/Books
      :language      prolog
      :content       price(item567, gbp, 24.95)
)
```

KQML Message

Example KQML Message

```
(tell :sender        amazon.com
      :receiver      info-agent0011
      :in-reply-to   msid-7.24.97.45391
      :ontology      http://amazon.com/Books
      :language      prolog
      :content       price(item567, gbp, 24.95)
)
```

- performative

KQML Message

Example KQML Message

```
(tell :sender         amazon.com
      :receiver       info-agent0011
      :in-reply-to    msid-7.24.97.45391
      :ontology       http://amazon.com/Books
      :language       prolog
      :content        price(item567, gbp, 24.95)
)
```

- performative
- parameter (an attribute/value pair)

KQML Message

Example KQML Message

```
(tell :sender        amazon.com
      :receiver      info-agent0011
      :in-reply-to   msid-7.24.97.45391
      :ontology      http://amazon.com/Books
      :language      prolog
      :content       price(item567, gbp, 24.95)
)
```

- performative
- parameter (an attribute/value pair)
- declarative message content

KQML Parameters

Parameter	Interpretation
`:content`	content of message
`:language`	formal language of message content
`:ontology`	vocabulary employed by message
`:force`	will sender ever deny content of message?
`:reply-with <label>`	whether the sender expects a reply, and if so, a label for the reply
`:in-reply-to <label>`	the expected label in a reply
`:sender`	sender
`:receiver`	receiver

KQML Example Performatives

Performative	**Interpretation**
`ask-if`	*Is it true that ...?*
`achieve`	*Make it true that ...*
`tell`	*It is true that ...*
`reply`	*The answer is ...*
`evaluate`	*Simplify the sentence ...*

Basic Query Dialogue

Agent A sends the following performative to agent B:

```
(evaluate :language KIF
          :ontology motors
          :reply-with q1
          :content (val (torque motor1) (sim-time 5)))
```

Agent B replies with:

```
(reply :language KIF
       :ontology motors
       :in-reply-to q1
       :content (scalar 12 kgf))
```

KQML/KIF Evaluation

- KQML/KIF widely adopted, but
- list of performatives (~ 41) not fixed, and lacks clear rationale;
- no formal semantics for performatives;
- lacks commissives (e.g., promises).

FIPA ACL, I

- FIPA ACL (Agent Communication Language): a specification for inter-agent communication via message passing.
  - Assumes that agents have Beliefs, Desires and Intentions (BDI model)
  - FIPA ACL is similar in design to KQML, but gives an explicit BDI-based semantics to the performatives, using a Semantic language called SL.

Example FIPA ACL Message

```
(inform :sender    agent1
        :receiver  agent2
        :ontology  hpl-auction
        :language  fipa-sl
        :content   (price item567 24.95)
)
```

FIPA ACL, 2

- Basic performatives: `inform`, `request`
- All others (~ 20) defined in terms of these two.
- Semantics divided up into
  - feasibility precondition, and
  - rational effect
- Semantics is formalized using following modal operators:

Statement	**Interpretation**
$B_i\phi$	*i* believes that ϕ
$Bif_i\phi$	$B_i\phi \vee B_i\neg\phi$, i.e., *i* has a definite opinion about the truth of ϕ
$I_i\phi$	*i* has the intention of bringing about ϕ
$Agent(\alpha, i)$	*i* is capable of carrying out action α
$Done(\alpha)$	action α has been carried out
$Uif_i\phi$	*i* is uncertain whether or not ϕ [ignored here]

Inform

$\langle i, inform(j, \phi) \rangle$

feasibility precondition	$B_i\phi \wedge \neg B_i Bif_j \phi$
rational effect:	$B_j\phi$

inform that ϕ indicates that the sending agent:

- holds that ϕ is true,
- does not already believe that the receiver j has any knowledge of the truth of the ϕ, and
- intends that j also comes to believe that ϕ is true.

Request

$\langle i, request(j, \alpha) \rangle$

feasibility precondition	$B_i Agent(\alpha, j) \wedge \neg B_i I_j Done(\alpha)$
rational effect:	$Done(\alpha)$

request that ϕ indicates that the sending agent:

- holds that *j* is capable of carrying out action α, and
- does not believe that *j* is intending to bring it about that α is carried out.

For more details, and information about other performatives, see
`http://www.fipa.org/specs/fipa00037/SC00037J.pdf`

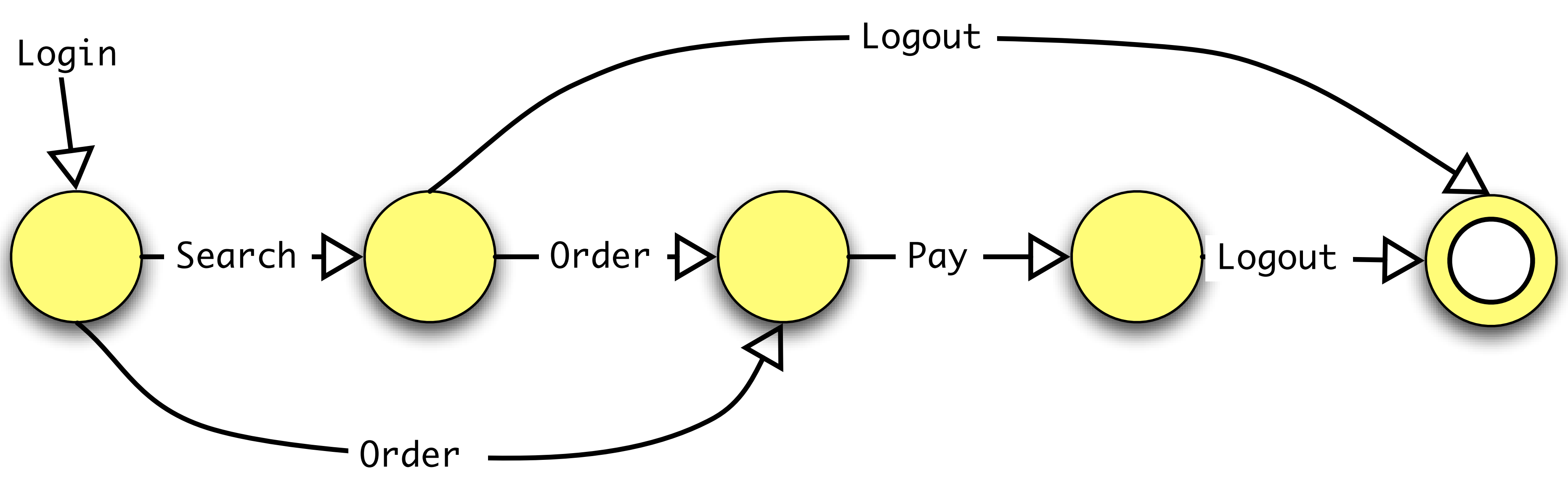
Agent Protocols

- Agents can have divergent interests (e.g., seller and buyer) but still cooperate
- Use protocols:
  - defines 'rules of engagement' between agents
  - Example: an QUERY message is followed by an INFORM message
  - often modelled as a finite-state machine
- Protocols will also contain more detailed specifications depending on the type of interaction
- Example: fixed price sale vs. an auction

Protocols as FSMs

- Protocols are often implemented as Finite State Machines.
- Each state represents a stage in the conversation — sometimes thought of as an information state.
- Arcs represent the exchanged messages that allow transition to a new state.
- Valid messages depend on the current state of the conversation — represented by outgoing arcs.
- Final states represent completed conversations.

Example FSM for a Conversation



Protocols in FIPA ACL

- Notion of agent 'protocol' often refers to stereotyped pattern of conversation between agents.

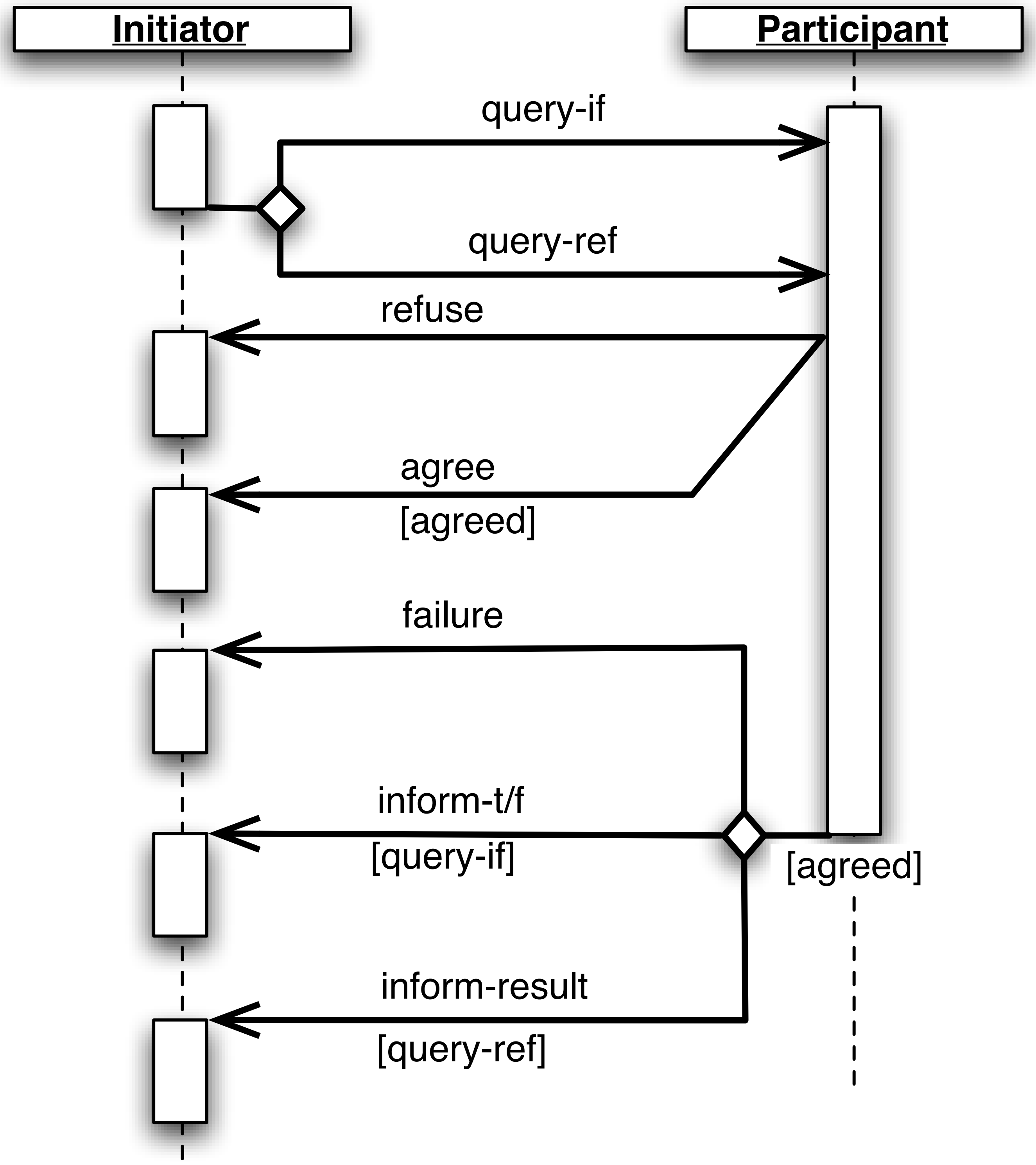
- Available protocols are usually pre-specified by the agent designer.

- Agents somehow need to discover which protocol to follow.

- Choice of protocol to follow could be negotiated,

- but in FIPA ACL, convention is to place name of the protocol in the `:protocol` parameter of the message.

FIPA-Query-Protocol



Explanation of FIPA-Query-Protocol

UML Sequence Diagram. The diamond symbol indicates a decision that can result in zero or more communications being sent.

- Initiator requests Participant to perform a `inform` action:
  - `query-if`: whether proposition `P` is true or false
  - `query-ref`: query about specified objects
- Participant processes request and decides whether to accept or refuse.
- If decides to accept, "agreed" becomes true.
- Participant uses `inform-t/f` to assert whether `P` is true or false;
- uses `inform-result` to refer to object that was queried about.

Problems with Mentalistic Approach

- How can the sender rely on the receiver to adopt certain beliefs?

- More generally, the behaviour of an agent does not give unambiguous information about its mental state.

- Alternative approach in terms of social commitments:
  - agent commits / promises to carry out certain actions;
  - e.g., buyer in an auction commits to paying for goods;
  - fulfilment of commitments can be verified.

Summary

- Focus on agent-agent communication;
- Speech act theory provides theoretical framework for ACLs;
- Two examples:
  - KQML/KIF
  - FIPA-ACL

Reading

- Reading: Wooldridge *An Introduction to MultiAgent Systems*, Chapter 8

- See also Wooldridge's web site:
`http://www.csc.liv.ac.uk/~mjw/pubs/imas`

- Odell, James, Van Dyke Parunak, H. and Bauer, B., Representing Agent Interaction Protocols in UML. In: *Agent-Oriented Software Engineering*, Ciancarini, P. and Wooldridge, M., Eds., Springer, pp. 121–140 Berlin, 2001. `http://www.jamesodell.com/Rep_Agent_Protocols.pdf`