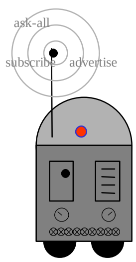
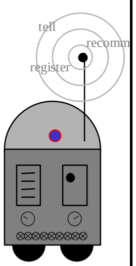
Software Agents and Agent Communication Languages

Tim Finin
University of Maryland
Baltimore County
finin@umbc.edu

March 23, 1999





http://umbc.edu/~finin/talks/unicom99

1

Overview and objectives

- The agent paradigm -- what is it and what's important about it (IMHO)
- The centrality of communication for agents
- Some historical context and future directions
- Basic concepts of agent communication and their realization in FIPA
- Some examples of applications

2

What is a software agent?

- No consensus yet, but several key concepts are important to this emerging paradigm.
- A software agent is a program that assists people and acts on their behalf. People can delegate tasks and work to agents. An agent:
  - is an autonomous, goal-directed process
  - is situated in, is aware of, reacts to and adapts to its environment
  - cooperates with other agents (software or human) to accomplish its tasks

3

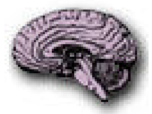
Agent Characteristic: Mobility?

A mobile agent is an executing program that *migrates* from machine to machine in a *heterogeneous network* under its *own control*.

- Examples: programs in Telescript, Agent-Tcl, Voyager, etc. and, to a limited degree, Java Applets.
- Note -- this definition implies some agent attributes, e.g. autonomy, persistence, ...
- Mobile agents offer some very interesting advantages as well as some disadvantages.
- This is an important technology for distributed systems but is largely orthogonal to other “agent” issues.

4

Agent Characteristic: Intelligence?



Q: What makes an agent an “intelligent agent”?

A: The size of the price tag.

More seriously…

- The paradigm covers agents of varying degrees of intelligence
- Intelligent agents will tend to
  - know and apply more sophisticated domain knowledge
  - recognizing underlying goals and intentions
  - react to unexpected situations in a robust manner
  - better NLP skills
  - etc.

Much of what we will be saying applies to agents of little or no intelligence.

5

Some key ideas

- Software agents offer a new paradigm for very large scale distributed heterogeneous applications.
- The paradigm focuses on the **interactions** of autonomous, cooperating processes which can adapt to humans and other agents.
- Mobility is an orthogonal characteristic which many, but not all, consider important.
- Intelligence is always a desirable characteristic but is not strictly required by the paradigm.
- The paradigm is still forming.

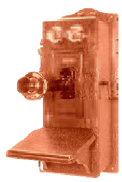
6

Why is communication important?

- Most, but not all, would agree that communication is a requirement for cooperation.
- Societies can do things that no individual (agent) can
- Diversity introduces heterogeneity.
- Autonomy encourages disregard for other agents' internal structure -- communicating agents need only care about understanding a "common language".
- For social agents (and animals), the communicative actions are are among the most important actions to master.

7

Agent Communication

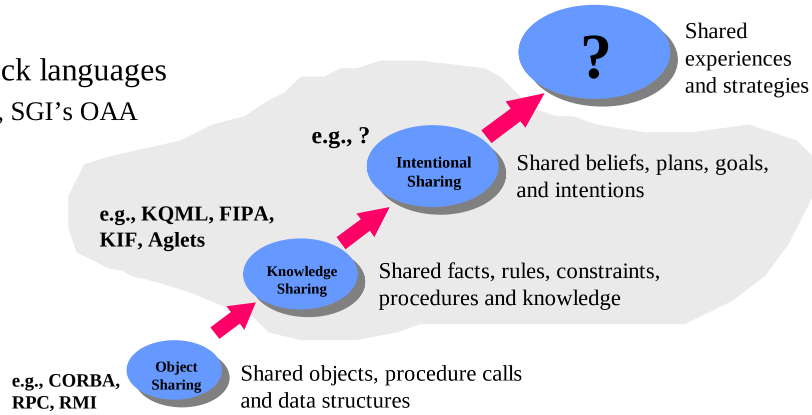


- Agent-to-agent communication is key to realizing the potential of the agent paradigm, just as the development of human language was key to the development of human intelligence and societies.
- Agents use an ***Agent Communication Language*** or ACL to communication information and knowledge.
  - Genesereth (CACM, 1992) defined a software agent as any system which uses an ACL to exchange information.
- Understanding a "common language" means:
  - understanding its vocabulary, i.e., the meaning of its tokens
  - knowing how to effectively use the vocabulary to perform tasks, achieve goals, effect one's environment, etc.
- For ACLs we're primarily concerned with the vocabulary

8

Some ACLs

- •Is CORBA an ACL?
- •Knowledge sharing approach
  - – KQML, KIF, Ontologies
- •FIPA
- •Ad hock languages
  - – e.g., SGI's OAA



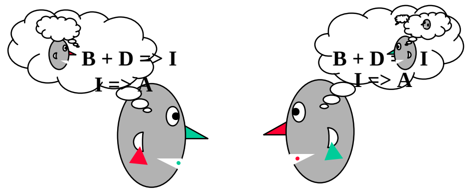
9

Agent Communication, at the technical level

- Messages are transported using some lower-level transport protocol (SMTP,TCP/IP, HTTP, IIOP, etc.)
- An Agent Communication Language (ACL) defines the types of messages (and their meaning) that agents may exchange.
- Over time, agents engage in "conversations." Such interaction protocols (negotiation, auction, etc.), defines task-oriented, shared sequences of messages.
- Some higher-level conceptualization of an agent's goals and strategies drives the agent's communicative (and non-communicative) behavior.

10

Agent Communication, at the theoretical level



- ACL have message types that are usually modeled after **speech acts**, which are understood in terms of an intentional-level description of an agent
- An intentional description makes references to beliefs, desires, intentions (BDI) and other **mental states**.
- BDI frameworks have the power to describe an agents' behavior, including communicative behavior
- Describing behavior at this level is an important contribution of the agent-based approach.

11

The intentional stance

- Agents have "*propositional attitudes*"
- Propositional attitudes are three-part relationship between
  - an agent,
  - a content-bearing proposition (e.g., "*it is raining*"), and
  - a finite set of propositional attitudes an agent might have w.r.t. the proposition (e.g., believing, asserting, fearing, wondering, hoping, etc.)
- <*a*, fear, raining(t_{now}) >

12

On ascribing mental qualities to machines

- The issue is not whether a system is really intentional but whether we can coherently view it as such (Daniel Dennett)
- Ascribing mental qualities to machines (John McCarthy):
  - **legitimacy**: the ascription expresses the same information about a machine that it expresses about a person
  - **usefulness**: the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair it or improve it.
- As MAS get more complex, we will find it useful to ascribe mental qualities to them, just as we do for other animals.

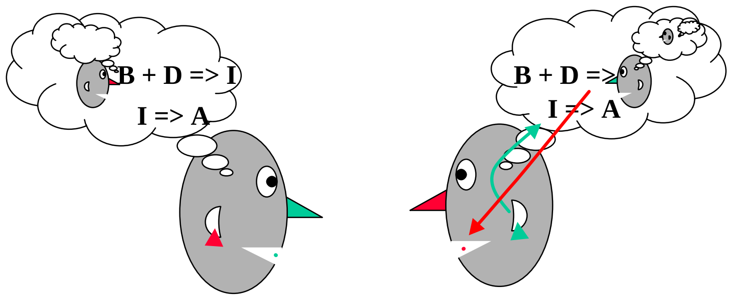
13

BDI Agents, Theories and Architectures

- BDI architectures describe the internal state of an agent by the mental states of beliefs, goals and intentions
- BDI theories provide a conceptual model of the knowledge, goals, and commitments of an agent
- BDI agents have some (implicit or explicit) representations of the corresponding attitudes

14

BDI Model and Communication



- **Communicative actions** attempt to (1) inform others about our BDI state, and (2) effect the BDI state of others.
- Note the recursion: an agent has beliefs about the world, beliefs about other agents, beliefs about the beliefs of other agents, beliefs about the beliefs another agent has about it, ...

15

Criticism of BDI theories

- The necessity of having all three modalities is questioned from both ends:
  - too few
  - too many
- System builders question their relevance in practice:
  - multi-modal BDI logics do not have complete axiomatizations
  - they are not efficiently computable

➡ There is a gap between theory and practice

16

Speech Act Theory

High level framework to account for human communication, *Language as Action* (Austin)

- Speakers do not just utter true or false sentences
- Speakers perform speech acts:
  requests, suggestions, promises, threats, etc.
- Every utterance is a speech act
- The intended underlying speech act may need to be inferred
  - "Pass the salt", "Please pass me the salt", vs. "Can you pass the salt"

17

Agents and agencies

- Groups of agents can form a team to cooperate and act as one super-agent.
- Opening up an agent we may find it useful to describe its internal architecture as a collection of sub-agents.
- What's going on here? Is it *agents all the way down*?
- My take -- a group of agents which can be modeled as having collective "mental states" (e.g., beliefs, desires, intentions) and can take collective actions can be usefully described as an agent.

18

Dividing up the problem

19

Historical Note: Knowledge Sharing Effort

- Initiated by DARPA circa 1990
- Sponsored by DARPA, NSF, AFOSR, etc.
- Participation by dozens of researchers in academia and industry.
- Developing techniques, methodologies and software tools for *knowledge sharing* and *knowledge reuse*.
- Sharing and reuse can occur at *design*, *implementation* or *execution* time.

20

Knowledge Sharing Effort

- Knowledge sharing requires a communication which requires a common language
- We can divide a language into syntax, semantics, and pragmatics
- Some existing components that can be used independently or together:
  - **KIF** - knowledge interchange format *(syntax)*
  - **Ontolingua** - a language for defining sharable ontologies *(semantics)*
  - **KQML** - a high-level interaction language *(pragmatics)*

Propositional (KIF, Ontolingua)

Propositional attitudes (KQML)

21

Knowledge Sharing Effort => FIPA

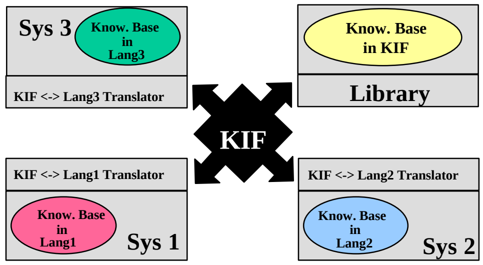
- Knowledge sharing requires a communication which requires a common language
- We can divide a language into syntax, semantics, and pragmatics
- Some existing components that can be used independently or together:
  - **KIF** - knowledge interchange format *(syntax)* **=> SL**
  - **Ontolingua** - a language for defining sharable ontologies *(semantics)*
  - **KQML** - a high-level interaction language *(pragmatics)* **=> ACL**

Propositional

Propositional attitudes

22

Knowledge Interchange Format



- KIF ~ First order logic set theory
- An **interlingua** for encoded declarative knowledge
  - Takes translation among n systems from $O(n^2)$ to O(n)
- Common language for reusable knowledge
  - Implementation independent semantics
  - Highly expressive - can represent knowledge in typical application KBs.
  - Translatable - into and out of typical application languages
  - Human readable - good for publishing reference models and ontologies.
- Current specification at http://logic.stanford.edu/
- FIPA's SL ≈ KIF + modal operators - default reasoning

23

KIF Syntax and Semantics

- Extended version of first order predicate logic
- Model-theoretic semantics
- Simple list-based linear ASCII syntax, e.g.,

  (forall ?X (=> (P ?X) (OR (Q ?X) (R ?X)))

  (exisits ?person (mother mary ?person))

  (=> (apple ?x) (red ?x))

  (<<= (father ?x ?y) (and (child ?x ?y) (male ?x))
- KIF includes an axiomatic specification of large function and relation vocabulary and a vocabulary for numbers, sets, and lists

24

Common Semantics
Shared Ontologies and Ontolingua

- ***Ontology***: A common vocabulary and agreed upon meanings to describe a subject domain.
- Ontolingua is a language for building, publishing, and sharing ontologies.
  - A web-based interface to a browser/editor server.
  - Ontologies can be automatically translated into other content languages, including KIF, SL, LOOM, Prolog, etc.
  - The language includes primitives for combining ontologies.

25

Common Pragmatics
Knowledge Query and Manipulation Language

- KQML is a high-level, message-oriented, communication language and protocol for information exchange independent of content syntax and ontology.
- KQML is also independent of
  - transport mechanism, e.g., tcp/ip, email, corba, IIOP, ...
  - High level protocols, e.g., Contract Net, Auctions, …
- Each KQML message represents a single *speech act* (e.g., ask, tell, achieve, …) with an associated semantics and protocol.
- KQML includes primitive message types of particular interest to building interesting agent architectures (e.g., for mediators, sharing intentions, etc.)

26

KQML

Knowledge Query and Manipulation Language

- KQML is a high-level, message-oriented, communication language and protocol for information exchange independent of content syntax and ontology.
- KQML is independent of
  - the transport mechanism (e.g., tcp/ip, email, corba objects, IIOP, etc.)
  - Independent of content language (e.g., KIF, SQL, STEP, Prolog, etc.)
  - Independent of the ontology assumed by the content.
- KQML includes primitive message types of particular interest to building interesting agent architectures (e.g., for mediators, sharing intentions, etc.)

27

A KQML Message

Represents a single *speech act* or *performative*

ask, tell, reply, subscribe, achieve, monitor, ...

with an associated *semantics and protocol*

$tell(i,j, B_i\phi) = fp[B_i B_i\phi \wedge \neg B_i(Bif_j B_i\phi \vee Uif_j B_i\phi)] \wedge re[B_j B_i\phi]$ ...

and a list of *attribute/value pairs*

:content, :language, :from, :in-reply-to

performative → (***tell***

parameter → :**ontology**

value → “***price(ISBN3294,24.95)***”

```
(tell :sender       bhkAgent
      :receiver     fininBot
      :in-reply-to  id7.24.97.45391
      :ontology     ecbk12
      :language     Prolog
      :content      "price(ISBN3294,24.95)")
```

28

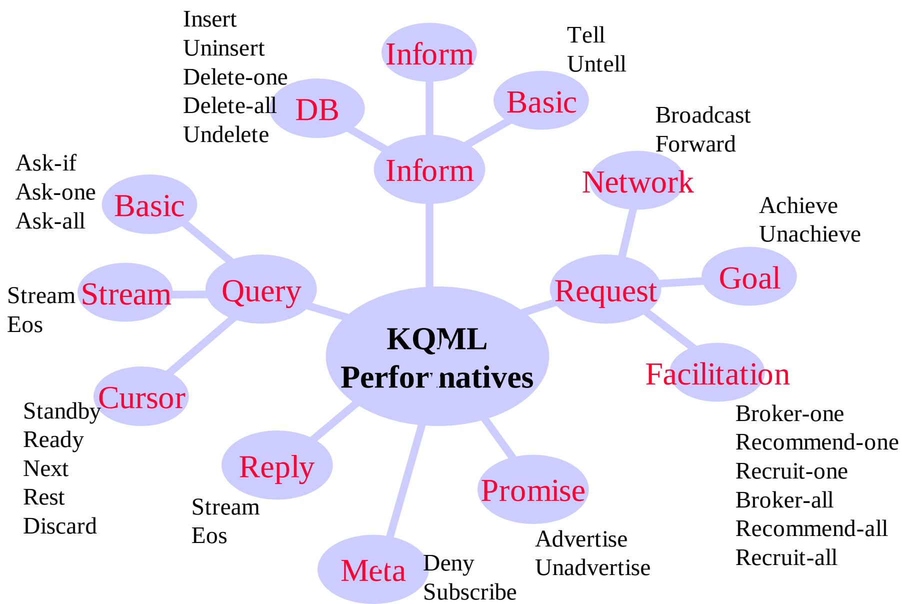
KQML Reserved Parameter Keywords

1997

:sender	the actual sender of the performative
:receiver	the actual receiver of the performative
:from	the origin of the performative in *:content* when forward is used
:to	the final destination of the performative in *:content* when forward is used
:in-reply-to	the expected label in a response to a previous message (same as the *:reply-with* value of the previous message)
:reply-with	the expected label in a response to the current message
:language	the name of the representation language of the :content
:ontology	the name of the ontology (e.g., set of term definitions) assumed in the :content parameter
:content	the information about which the performative expresses an attitude

29

Performatives (1997)



30

Facilitation Services

Facilitators are a class of agents who

- traffic in meta-knowledge about other agents.
- provide communication services such as:
  - message forwarding and broadcasting
  - resource discovery
  - matchmaking
  - content-based routing
  - meta-knowledge queries
- Performatives of special interest to facilitators are
  - advertise, broker, recruit, recommend, forward, broadcast, etc.
- Brokers are generally considered to focus on matchmaking
- Facilitators can be intelligent or not
  - Intelligent facilitators use domain knowledge in matching services needs and offers.

31

If KQML is so great, why do we need the FIPA ACL?

- There are two KQML specification documents and many dialects and “extended” versions of KQML plus many important concepts not yet addressed (e.g., security).
- The FIPA ACL has been specified along with other critical aspects of agent systems (e.g., agent management).
- The FIPA ACL has the support of a formal standardization process and organization.
- FIPA provides us with an opportunity to revisit and improve on many of the design decisions made in KQML
  - e.g., primitive CAs are defined in such a way as to allow them to be composed, creating new CAs

32

The FIPA ACL

33

Major Features of FIPA ACL compared to KQML

- Management and facilitation primitives (register, broker, recruit, etc.) are not part of the ACL
- Primitives can be defined compositionally from "core" primitives
- Use of a powerful language to define agents' states (Semantic Language, or SL)
- Semantics based on mental attitudes (belief, intention, etc.)
- The meaning of primitives is given in terms of Feasibility Preconditions (FPs) and Rational Effect (RE)

34

Outline of FIPA ACL Semantics

- A primitive's meaning is defined in terms of FPs and REs
- The Feasibility Preconditions of a CA define the conditions that ought to be true before an agent may plan to execute the CA
- The Rational Effect is the effect that an agent hopes to bring about by performing an action (but with no guarantee that the effect will be achieved)
- The FPs and the REs involve **agents state descriptions** that are given in SL

35

Semantic Language (SL)

- SL is the formal language used to define the semantics of FIPA ACL
- In SL, logical propositions are expressed in a logic of mental attitudes and actions
- The logical framework is a first order modal language with identity (similar to Cohen & Levesque)
- SL provides formalizations for three primitive mental attitudes: Belief, Uncertainty and Choice (or Goal); Intention is defined as a Persistent Goal
- SL can express propositions, objects and actions

36

An example of FIPA ACL semantics (inform)

<*i*, inform(*j*, ϕ)>

FP: $B_i\phi \wedge \neg B_i(Bif_j\phi \vee Uif_j\phi)$
RE: $B_j\phi$

Agent i informs agent j that (it is true that) it is raining today:

```
(inform
    :sender i
    :receiver j
    :content "weather(today,raining)"
    :language Prolog
    :ontology weather42)
```

37

Another example of FIPA ACL semantics (request)

<*i*, request(*j*, a)>

FP: FP(a) [i\j] $\wedge B_i$ Agent(*j*, a) $\wedge \neg B_i I_j$ Done(a)
RE: Done(a)

Agent i requests j to open a file:

```
(request
    :sender i
    :receiver j
    :content "open \"db.txt\" for input"
    :language vb)
```

38

Composing new primitives

<i, query-if(j, ϕ) $\equiv$ <i, request(j, <j, inform-if(i, ϕ)>)>

FP: $\neg \text{Bif}_i \phi \wedge \neg \text{Uif}_i \phi \wedge \neg B_i\, I_j$ Done(<j, inform-if(i, ϕ)>)

RE: Done(<j, inform(i, ϕ)>|<j, inform(i, $\neg\phi$)>)

Agent i asks agent j if j is registered with domain server d1:

```
(query-if
    :sender i
    :receiver j
    :content (registered (server d1) (agent j))
    :reply-with r09
    …)
```

Agent j replies that it is not:

```
(inform
    :sender j
    :receiver i
    :content (not (registered (server d1) (agent j)))
    :in-reply-to r09)
```

39

Ontologies

40

Overview

- What is an ontology?
- Tools for building, using and maintaining ontologies
- Existing ontologies of general interest
- FIPA's view on agents and ontologies

41

Common Semantics
Shared Ontologies and Ontolingua

Ontology : A common vocabulary and agreed upon meanings to describe a subject domain.

On*tol"o*gy (?), n. [Gr. the things which exist (pl.neut. of , , being, p.pr. of to be) + -logy: cf.F. ontologie.]

That department of the science of metaphysics which investigates and explains the nature and essential properties and relations of all beings, as such, or the principles and causes of being.

Webster's Revised Unabridged Dictionary (G & C. Merriam Co., 1913, edited by Noah Porter)

This is not a profoundly new idea …

- Vocabulary specification
- Domain theory
- Conceptual schema (for a data base)
- Class-subclass taxonomy
- Object schema

42

Conceptual Schemas

A conceptual schema specifies the intended meaning of concepts used in a data base

Data Base:

139	74.50
140	77.60
...	...

Data Base Schema:

Table: price
***stockNo: integer; cost: float**

Conceptual Schema:

price(x, y) =>
$ (x', y') [auto_part(x')
& part_no(x') = x
& retail_price(x', y', Value-Inc)
& magnitude(y', US_dollars) = y]

Auto Product Ontology

Product Ontology

Units & Measures Ontology

43

Implicit vs. Explicit Ontologies

- Systems which communicate and work together must share an ontology.
- The shared ontology can be **implicit** or **explicit**.
- Implicit ontology are typically represented only by procedures
- Explicit ontologies are (ideally) given a declarative representation in a well defined knowledge representation language.

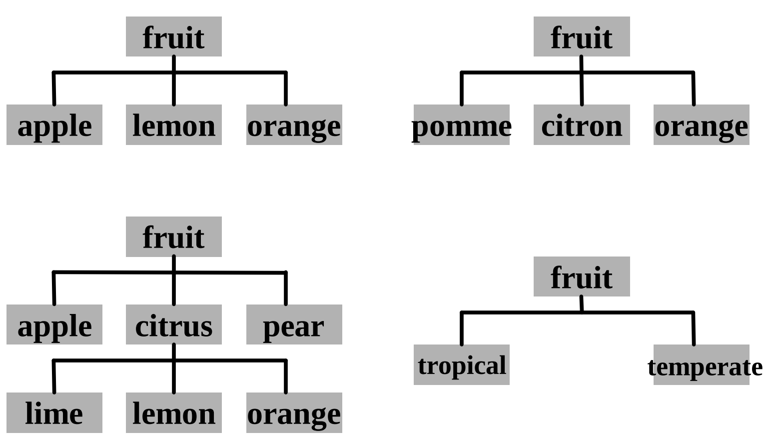
44

Conceptualizations, Vocabularies and Axiomitization

- Three important aspects to explicit ontologies
  - **Conceptualization** involves the underlying model of the domain in terms of objects, attributes and relations.
  - **Vocabulary** involves assigning symbols or terms to refer to those objects, attributes and relations.
  - **Axiomitization** involves encoding rules and constraints which capture significant aspects of the domain model.
- Two ontologies may
  - be based on different conceptualizations
  - be based on the same conceptualization but use different vocabularies
  - differ in how much they attempt to axiomitize the ontologies

45

Simple examples



46

Ontologies vs. KBs

Ontologies are distinguished from KBs not by their form, but by the role they play in representing knowledge

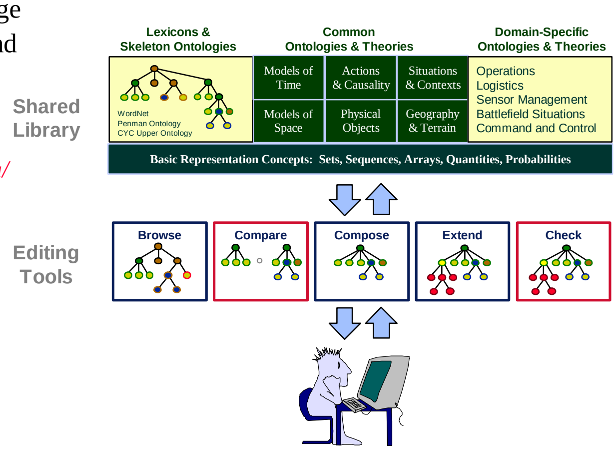
- Consensus models for a domain
- Emphasis on properties that hold in all situations
- Emphasis on classes rather than instances
- Intended to support multiple tasks and methods
- Don't change during problem solving and are suited for "compiling" into tools
- Need to satisfy a community of use
  - Emphasis on collaborative development
  - Emphasis on translation to multiple logical formalisms
- Useful for education

47

Ontology Library and Editing Tools

Ontolingua is a language for building, publishing, and sharing ontologies.

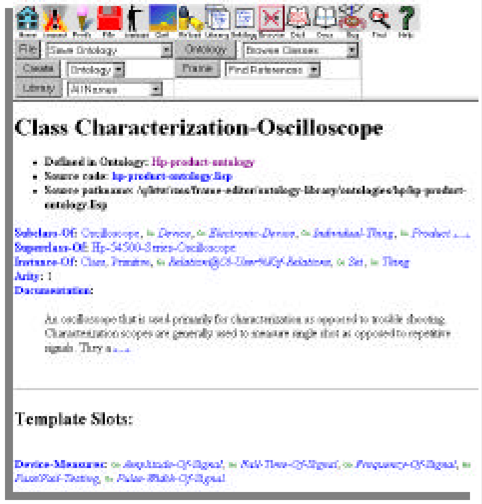
- A web-based interface to a browser/editor server at *http://ontolingua.stanford.edu/* and mirror sites.
- Ontologies can be translated into a number of content languages, including KIF, LOOM, Prolog, CLIPS, etc.



48

Ontolingua - Language

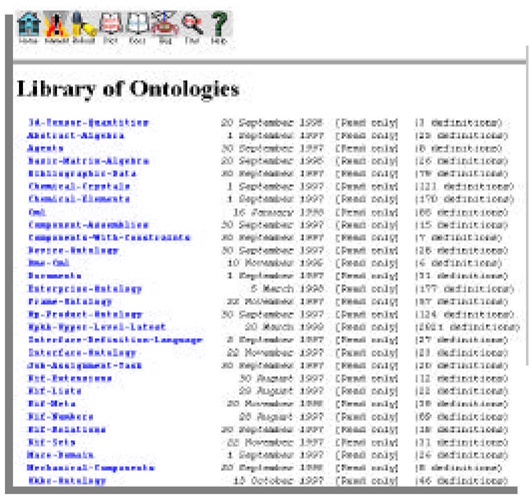
- Ontolingua allows full KIF
  - 1st order logic with relation constants in domain of discourse
  - Extremely expressive
  - Too much for most users
  - Too much for most systems!
- Ontolingua provides an object-oriented projection
- Statements within the o-o sublanguage easy to make
  - But any statement is allowed
- Ontolingua separates representation from presentation



49

Ontolingua - Library

- Library of modules supports reuse
- Authors assemble a new ontology
  - Assembly defines a general graph
  - Cycles are allowed (sports and medicine)
- Authors may augment definitions
  - But you can never say less!
  - Different authors may make incompatible extensions



50

Ontolingua - Usage

- Ontolingua is (one of) the most widely used knowledge development environments
  - Available since 1/94 at http://ontolingua.stanford.edu
  - Over 4500 total users, 1200 current users, 300 active users
  - Over 4,200,000 user commands executed
  - Recently averaging over 7000 commands per day
  - Over 800 ontologies stored on the KSL server
  - Mirror sites in Spain, Netherlands, UMBC, and corporate sites
- Applications include
  - Enterprise modeling, electronic commerce, engineering, ribosomal structure modeling, workflow modeling, molecular biology, cross-disciplinary design and simulation, drug interactions, medical vocabularies, software design reuse, standards development

51

Open Knowledge Base Connectivity

- OKBC is another software tool for building and accessing ontologies
  - OKBC is to KBs what ODBC is to Databases
  - A standard API to Knowledge Representation Systems (KRS)
- OKBC
  - Specifies a protocol for KRS interoperation
  - Supports a client-server model for interaction
  - Provides an object-oriented view of a KRS
  - Supports wide variation in underlying KRS
- Adopted as KRS interoperation protocol within DARPA High Performance Knowledge Base (HPKB) program
- OKBC drivers available for Loom, Ontolingua, Ocelot, ATP, JavaKB, TupleKB, ...
- OKBC applications include GKB (SRI), Jot (KSL), Ontolingua (KSL), Riboweb (SMI), Protégé (SMI), Hike (SAIC), ...
- http://ai.sri.com/~okbc/

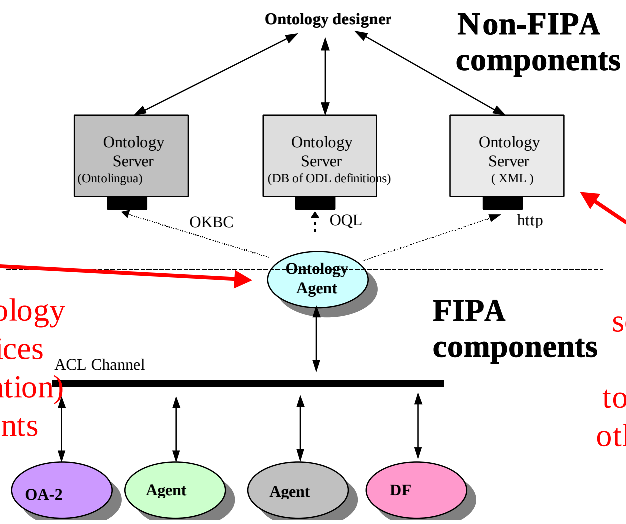
52

Big Ontologies

- There are several large, general ontologies that are freely available.
- Some examples are:
  - **Cyc -** Original general purpose ontology
  - **WordNet -** a large, on-line lexical reference system
  - **World Fact Book --** 5Meg of KIF sentences!
  - **UMLS** - NLM's Unified Medical Language System
  - See http://www.cs.utexas.edu/users/mfkb/related.html for more
- We anticipate the development of ontologies to support ecommerce
  - see www.ontology.org
  - probably in XML

53

Ontologies in the FIPA Reference Model



Ontology Agents provide ontology related services (e.g., translation) to FIPA agents

Ontology servers define and serve shared ontologies to agents and other systems

54

Ontology agent services

- Help a FIPA agent in selecting a shared (sub) ontology for communication,
- Create and update an ontology, or only some terms of an ontology.
- Translate expressions between different ontologies (different names with same meanings),
- Respond to query for relationships between terms or between ontologies,
- Discovery of public ontologies in order to access them.

55

The FIPA knowledge model

- FIPA specifies **fipa-meta-ontology** as the ontology used to talk about ontologies.
- This is largely based on the OKBC model developed by Stanford and SRI
  - See http://ai.sri.com/~okbc
- As its name suggests:
  - OKBC is to KBs what ODBC is to Databases
  - A standard API to Knowledge Representation Systems (KRS)
- Generic OKBC clients (e.g., browsers, editors) exist as well as OKBC drivers for a number of knowledge representation systems.

56

Relationships between ontologies

- fipa-meta-ontology also includes:

  `(ontol-relationship ?O1 ?O2 ?level)`

- to describe translation services, where ?level can be:
  - **Extension -** O1 extends the ontology O2
  - **Identical -** O1 and O2 are identical
  - **Equivalent -** O1 and O2 are equivalent
  - **Strongly-translatable -**every term in O1 is translatable to a term in O2 without loss of information
  - **Weakly-translatable -** some terms in O1 are translatable to terms in O2 with some loss of information
  - **Approx-translatable -** translating terms from O1 to O2 may introduce some inconsistencies
-

57

Ontology Conclusions

- Shared ontologies are essential for agent communication and knowledge sharing
- Ontology tools and standards are important
  - Ontolingua and OKBC are good examples
  - XML and RDF may be a next step
- Some large general ontologies are available
  - Cyc, WFB, Wordnet, …
- For more information…
  - http://www.ontology.org/
  - Ontology mailing list: send mail to majordomo@cs.umbc.edu with "*info ontology*" in message body for information.
  - ANSI Ad Hoc Group on Ontology Standards: http://WWW-KSL.Stanford.EDU/onto-std/

58

Conclusions and Prospects

59

Some key ideas

- Software agents offer a new paradigm for very large scale distributed heterogeneous applications.
- The paradigm focuses on the interactions of autonomous, cooperating processes which can adapt to humans and other agents.
- Agent Communication Languages are a key enabling technology
  - Mobility is an orthogonal characteristic which many, but not all, consider central.
  - Intelligence is always a desirable characteristic but is not strictly required by the paradigm.
- The paradigm is still forming and ACLs will continue to evolve.

60

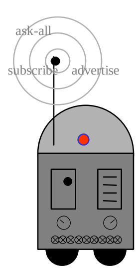
Prospects

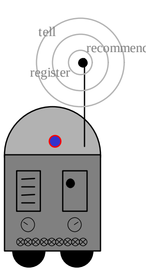
- FIPA's ACL is likely to be the next iteration of a widely used standard ACL.
- Its not clear how ACLs will participate in the rapidly evolving world of Internet languages and protocols
  - The ACL "territory" may be overtaken by efforts from a programming language (e.g., Java, Jini), another interoperability language (e.g., CORBA) or Web technology (e.g., XML).
  - The Agent community is a small fish compared to, e.g., the Java community. What will Microsoft do?
- Many are experimenting with XML for agent communication
  - XML is a good way to represent structured information (e.g., ACL messages, KIF-like content) that is easy to use and understand by all agents, both human and software
  - We've developed DTDs and style sheets for FIPA ACL and KIF
  - XML is not a silver bullet.

61

For More Information

- **General information on software agents**
  - http://www.cs.umbc.edu/agents
- FIPA
  - http://www.fipa.org/
- KQML
  - http://www.cs.umbc.edu/kqml
- KIF
  - http://www.cs.umbc.edu/kif
- Ontologies
  - http://www.cs.umbc.edu/ontology/
- Agent Communication Languages
  - http://www.cs.umbc.edu/acl/





62