

# Theory and Practice of Agent Communication in the Semantic Web Era

**Tim Finin**

University of Maryland, Baltimore County

**Yannis Labrou**

Fujitsu Laboratories of America

© 2003



<http://agents.umbc.edu/tutorials/aamas03/t7/>

1

## Course Objectives

- Present the general requirements of Agent Communication Languages (ACLs) and the Semantic Web (SW)
- Sketch the conceptual and theoretical underpinnings of ACLs and SW
- Describe current ACLs and SW languages and their realizations in software implementations
- FIPA standardization efforts
- Review agent-based projects which are using some of the ACL components discussed
- Review applications and software tools for the Semantic Web
- Discuss directions in the practice of integrating agent and Semantic Web technologies



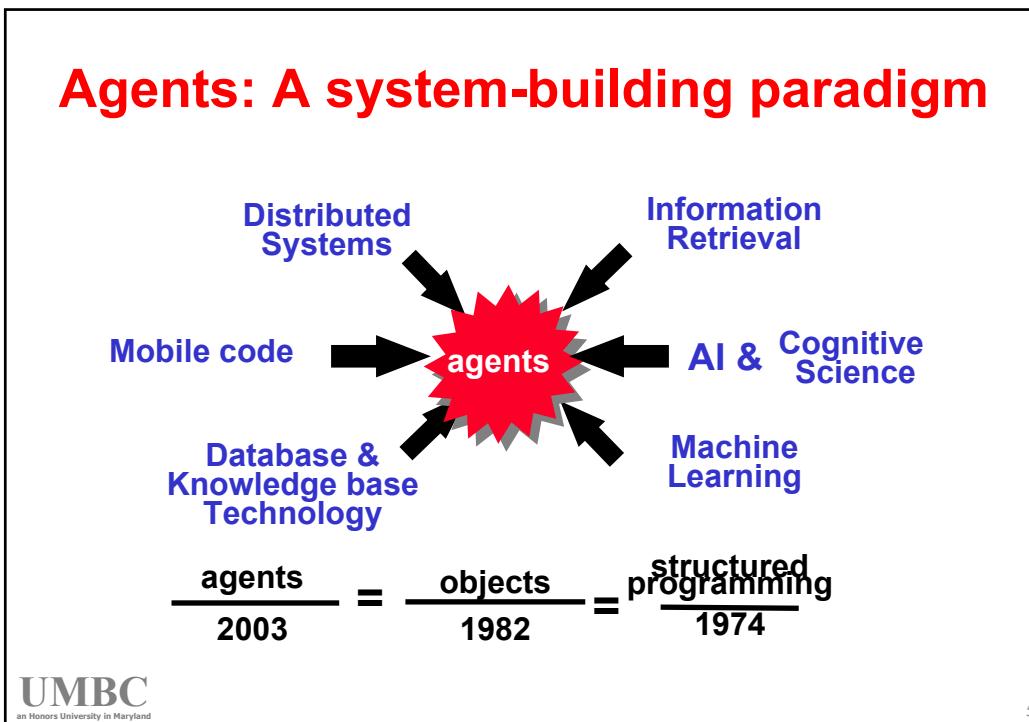
2

## Outline

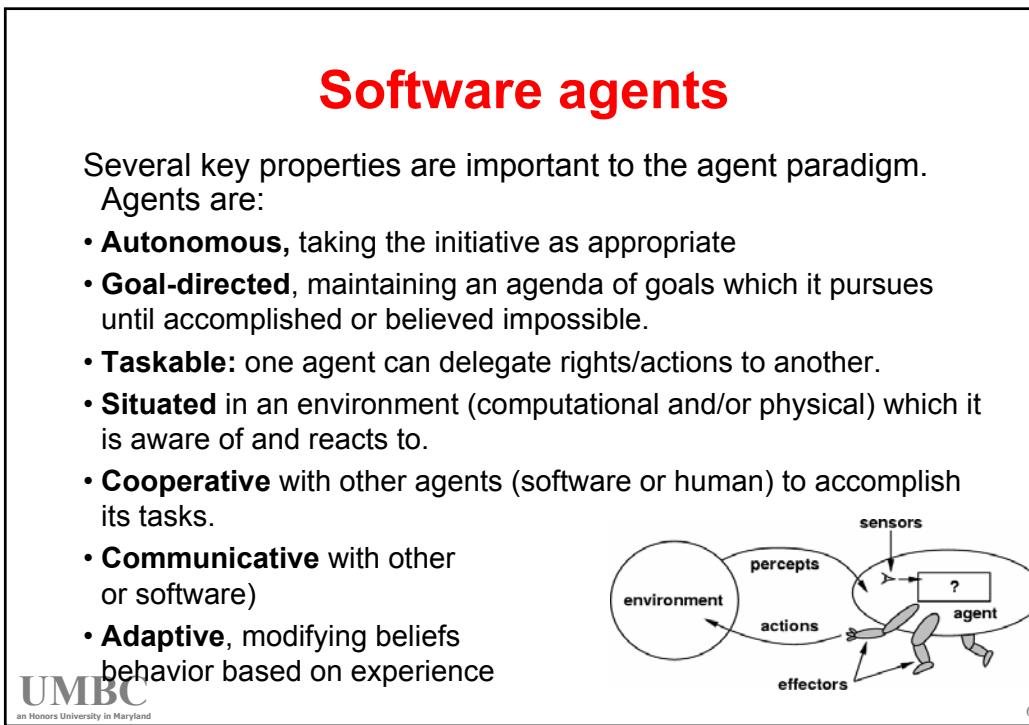
1. Introduction: Agents, Agent Communication and the Semantic Web
2. Conceptual & Theoretical Foundations
3. Origins – Knowledge Sharing Effort
4. Standardization - FIPA
5. ACL Semantics
6. ACLs and the Semantic Web
7. Semantic Web
8. Applications
9. Software Tools and Research Issues
10. Conclusions
11. Q & A

1

# Introduction to Agents, Agent Communication & the Semantic Web



5



6

## Communication

- Communication almost always means “communication in a common language”
- Understanding a “common language” involves:
  - **Semantics:** understanding of its vocabulary, i.e., understanding of the meaning of its tokens
  - **Syntax:** knowing the rules for combining the tokens into well formed expressions
  - **Pragmatics:** knowing how to effectively use the vocabulary to perform tasks, achieve goals, effect one’s environment, etc.
- For software agents, an Agent Communication Language (ACL) is concerned with all three.



7

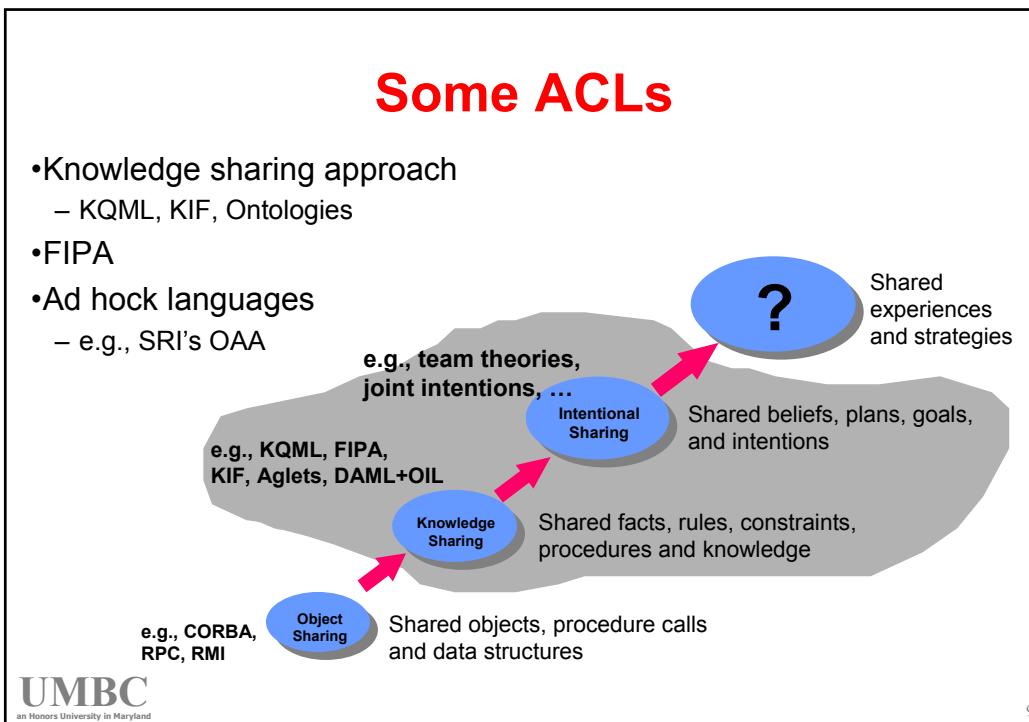
## Agent Communication

- 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** (ACL) to communicate information and knowledge.
- Genesereth (CACM 1994) defined a software agent as any system which uses an ACL to exchange information.

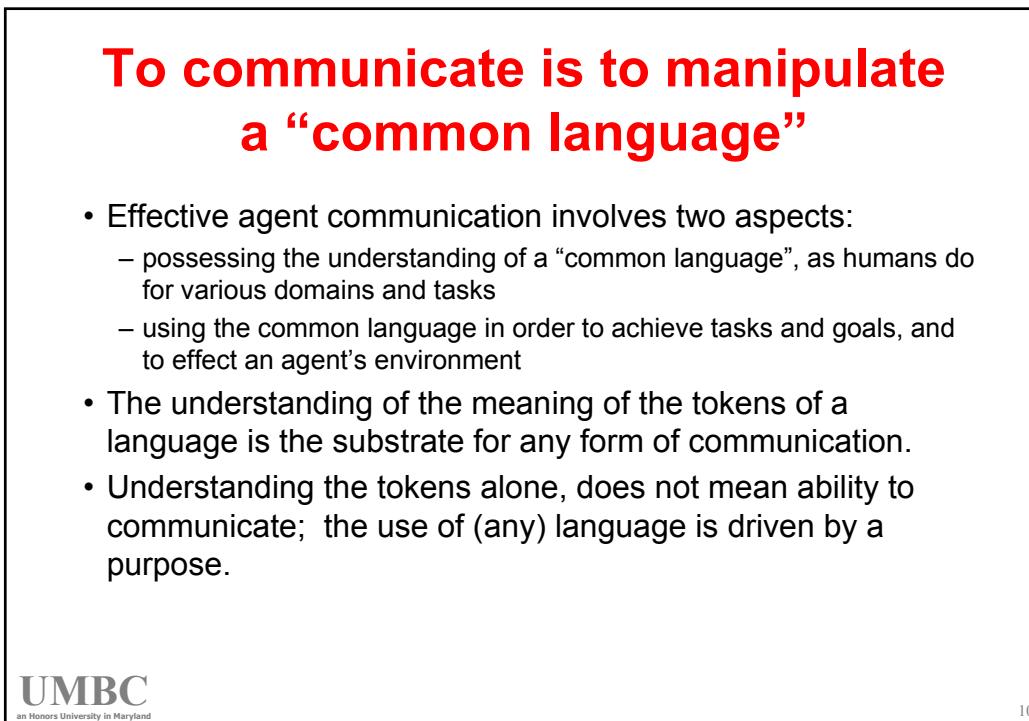
Genesereth, M. R., Ketchpel, S. P.: “Software Agents”, Communications of the Association for Computing Machinery, July 1994, pp 48-53.



8



9



10

## 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.

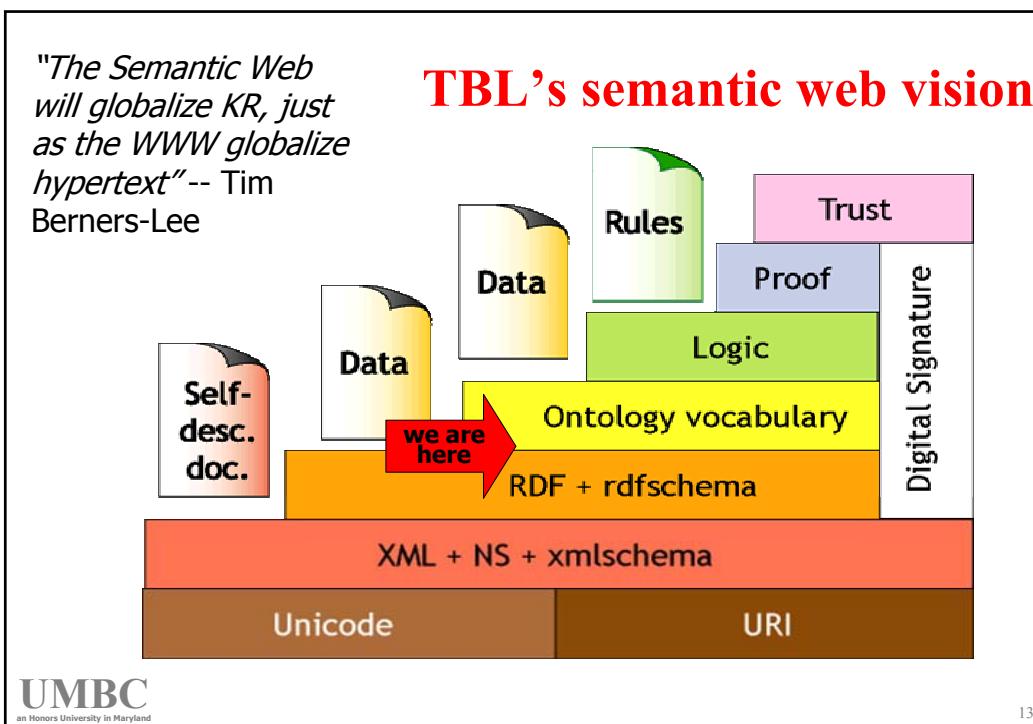
## What is the Semantic Web?

- **Focus on machine consumption:**

*“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”*  
-- Berners-Lee, Hendler and Lassila, The Semantic Web, Scientific American, 2001

- **The current Web stores things whereas the semantic Web does things.**

=> *Good match for agents!*



- ## Semantic Web
- We'll argue that the semantic web (SW) provides a good approach, languages and tools to support the development of intelligent agents.
  - This isn't obvious, since the SW seems grounded in the "traditional" wired web.
    - But, the principles which drive it are the right ones for multi-agent systems.
    - And, by grounding agents in web technology, they may make it out of the lab.
  - SW languages are being used in MAS as (1) content languages, (2) message encodings, and (3) service description languages.
- UMBC  
an Honors University in Maryland

# 2

# Conceptual and Theoretical Foundations

UMBC  
an Honors University in Maryland

15

## Historical Note: Knowledge Sharing Effort

- Initiated by DARPA circa 1990

The DARPA Knowledge Sharing Effort: Progress Report, Patil et. al., KR'92,  
<http://citeseer.nj.nec.com/patil93darpa.html>
- 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.

UMBC  
an Honors University in Maryland

16

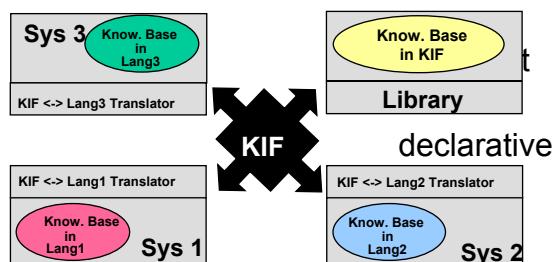
## Knowledge Sharing Effort

- 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  
} Propositional attitudes

## Knowledge Interchange Format

- KIF ~ First order logic theory
- An **interlingua** for encoded knowledge
  - Takes translation among n 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/kif/> now being superceded by an effort to specify and standardize “Common Logic” (<http://cl.tamu.edu/>)



## 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, LOOM, Prolog, etc.
  - The language includes primitives for combining ontologies.



19

## 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, http
  - ...
  - 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.)



20

## Common High-level Protocols

- There is also a need for communication agents to agree on the agent-level protocols they will use.
- The protocol is often conveyed via an extra parameter on a message
  - (ask :from Alice :to Bob ... :protocol auction42 ...)
- Common protocols:
  - Contract net
  - Various auction protocols
  - Name registration
- These protocols are often defined in terms of constraints on possible conversations and can be expressed as
  - Grammars (e.g., DFAs, ATNs, DCGs...)
  - Petri networks
  - UML-like interaction (activity) diagrams
  - Conversation plans

**UMBC** Rules or axioms  
an Honors University in Maryland

21

## Common Service Infrastructure

- Many agent systems assume a common set of services such as:
  - Agent Name Server
  - Broker or Facilitator
  - Communication visualizer
  - Certificate server
- These are often tied rather closely to an ACL since a given service is implemented to speak a single ACL
- Moreover, some of the services (e.g., name registration) may be logically ACL-dependent
  - e.g., Some ACLs don't have a notion of an agent's name and others have elaborate systems of naming

**UMBC**  
an Honors University in Maryland

22

# Conceptual and Theoretical Foundations I

## Speech Act Theory and BDI Theories

UMBC  
an Honors University in Maryland

23

### The intentional level, BDI theories, speech acts and ACLs: how do they all fit together?

- ACL have message types that are usually modeled after speech acts
- Speech acts may be understood in terms of an intentional-level description of an agent
- An intentional description makes references to beliefs, desires, intentions and other modalities
- BDI frameworks have the power to describe an agents' behavior, including communicative behavior

UMBC  
an Honors University in Maryland

24

## 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 with respect to the proposition (e.g., believing, asserting, fearing, wondering, hoping, etc.)



25

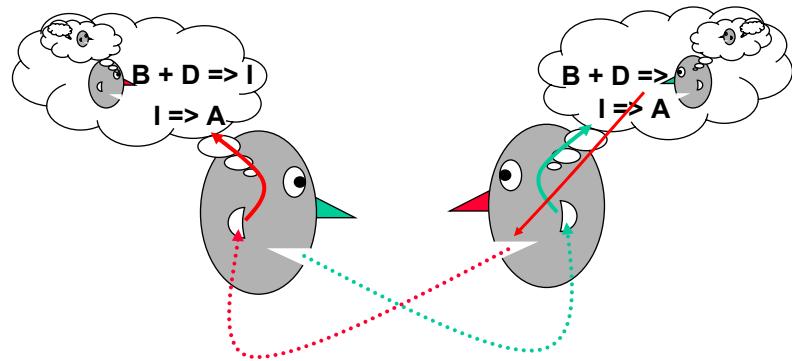
## BDI Agents, Theories and Architectures

- 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



26

## BDI Model and Communication



- Communication is a means to (1) reveal to others what our BDI state is and (2) attempt to 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, ...

## 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

# Conceptual and Theoretical Foundations II

## Knowledge Representation and Ontologies

UMBC  
an Honors University in Maryland

29

## Representation and Reasoning

- Intelligent agents need to be able to represent and reason about many things, including:
  - models of other (human or artificial) agent's beliefs, desires, intentions, perceptions, plans, preferences, objective properties, etc.
  - task, task structures, plans, etc.
  - meta-data about documents and collections of documents
- In general, they will need to **communicate** the same range of knowledge.
- A variety of content languages have been used with ACLs, including KIF, SL, Loom, Prolog, SQL, RDF...
- There is a special interest in content languages that can serve as a neutral, but expressive, **interlingua** for a wide range of systems.
- We'll look at KIF in a bit more detail.

UMBC  
an Honors University in Maryland

30

## KR Language Components

- **A logical formalism**

- Syntax for well formed formulae (wffs)
- Vocabulary of logical symbols (e.g., and, or, not,  $\Rightarrow$ , ...)
- Interpretation semantics for the logical symbols, e.g.,  
 $(\Rightarrow A B)$  is true if and only if  $B$  is true or  $A$  is false.

- **An ontology**

- **Vocabulary** of non-logical symbols (relations, functions, constants)
- **Definitions** of non-primitive symbols, e.g.  
 $(\Leftrightarrow (\text{Bachelor } ?x) (\text{AND} (\text{Man } ?x) (\text{Unmarried } ?x)))$
- **Axioms** restricting the interpretations of primitive symbols, e.g.  
 $(\Rightarrow (\text{Person } ?x) (\text{Gender} (\text{Mother } ?x) \text{ Female}))$

- **A proof theory**

- Specification of the reasoning steps that are logically sound, e.g.  
*From “( $\Rightarrow S1 S2$ )” and “S1”, conclude “S2”*

## O-O Languages Too Restrictive

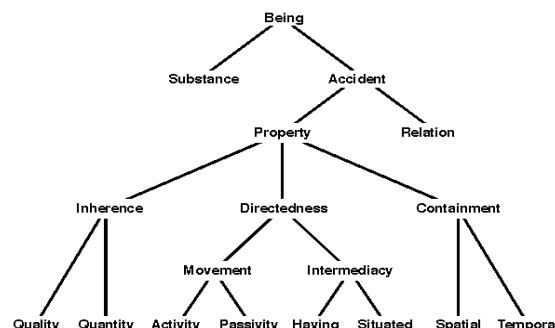
- Frames, object schema, description logics are popular KR languages used for ontologies
- They support definitional axioms of the form:
  - $R(x) \Rightarrow \dots \wedge P(x) \wedge \dots$  {subclass}
  - $R(x) \Rightarrow \dots \wedge [S(x,y) \Rightarrow P(y)] \wedge \dots$  {value class}
  - $R(x) \Rightarrow \dots \wedge \exists y S(x,y) \wedge \dots$  {value cardinality}
  - ...
- They don't support –
  - N-ary relations and functions
  - Standard properties of relations and functions
    - e.g., transitive, symmetric, inverse
  - Partial sufficient conditions e.g.,  $x > 0 \Rightarrow R(x)$
  - Disjunction of classes
  - General negation

## Other alternatives

- OKBC (see ontologies)
- Java objects (see AgentBuilder)
- SL (see FIPA)
- Common Logic (son of KIF)
- Constraints
- Database tuples
- RDF, DAML, DAML+OIL, OWL
- ..*your favorite representation language here..*

## What is an ontology

- The subject of ontology is the study of the categories of things that exist or may exist in some domain.
- The word ontology is from the Greek *ontos* for being and *logos* for word.
- Aristotle offered an ontology which included 10 categories, shown as the leaves in this tree (from Sowa, after Brentano)



## 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)*

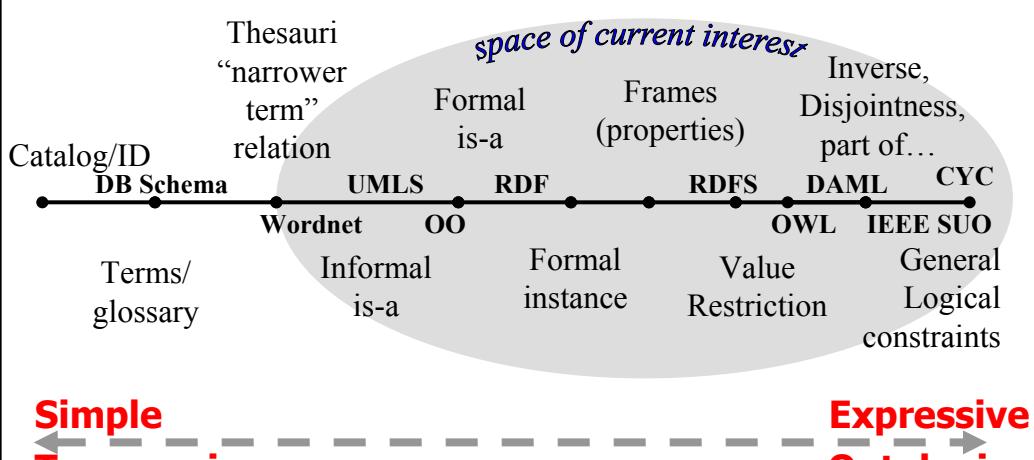
This is not a profoundly new idea ...

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

**UMBC**  
an Honors University in Maryland

35

## Ontology languages vary in expressivity



**UMBC**  
an Honors University in Maryland

After Deborah L. McGuinness (Stanford)

36

## Importance of ontologies in communication

- An example of the importance of ontologies in communication is the fate of NASA's Mars Climate Orbiter
- It crashed into Mars on September 23, 1999
- JPL used metric units in their program controlling the thrusters and Lockheed-Martin used imperial units.
- Instead of establishing an orbit at an altitude of 140km, it did so at 60km, causing it to burn up in the Martian atmosphere.

## Conceptual Schemas

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

<u>Data Base:</u>	<table border="1"><tr><td>139</td><td>74.50</td></tr><tr><td>140</td><td>77.60</td></tr><tr><td>...</td><td>...</td></tr></table>	139	74.50	140	77.60	...	...
139	74.50						
140	77.60						
...	...						

Data Base Schema:

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

Conceptual Schema:

price(x, y) =>  
 $\exists (x', y') [auto\_part(x') \& part\_no(x') = x \& retail\_price(x', y', Value-Inc) \& magnitude(y', US\_dollars) = y]$

Auto Product Ontology

Product Ontology

Units & Measures Ontology

## Implicit vs. Explicit Ontologies

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

⇒ *Explicit ontologies enable machine understanding*



39

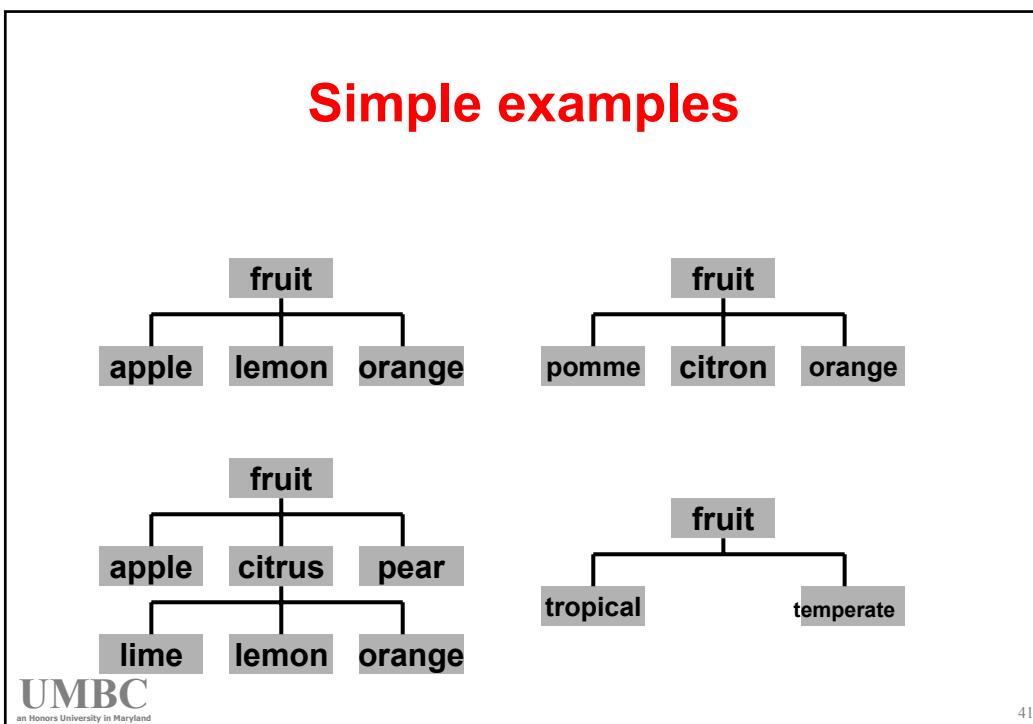
## Conceptualizations, Vocabularies and Axiomatization

- 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.
  - **Axiomatization** 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



40

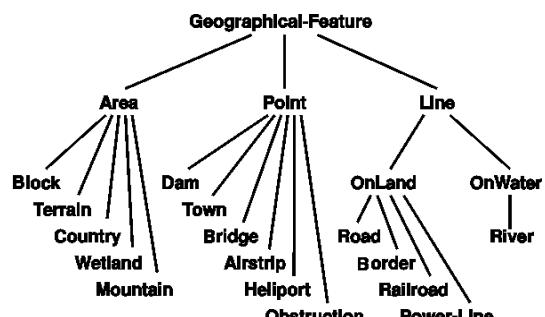
## Simple examples



41

## Top down vs. bottom up

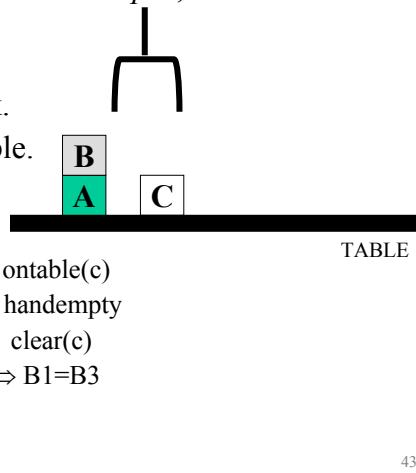
- Philosophers build from the top down and are interested in capturing the most general concepts.
- Programmers tend to work from the bottom up, supporting a set of applications, with a little generality to help reuse and future development.
- Ex: CHAT-80 system (Periera and Warren, 1982) which answered NL questions about a geographic database.
  - Example of a **microworld** ontology supported NLP, query answering, and generation



## Blocks world

- The **blocks world** is another microworld used often for NLP, vision, planning.
- It consists of a **table**, a set of **blocks** or different *shapes*, *sizes* and *colors* and a robot **hand**.
- Some typical domain constraints:
  - Only one block can be on another block.
  - Any number of blocks can be on the table.
  - The hand can only hold one block.
- Typical representation:

```
block(a) color(a,green) size(a,small) ontable(a) ontable(c)
block(b) color(b,gray) size(b,small) on(b,a) handempty
block(c) color(c,white) size(c,small) clear(b) clear(c)
  ∀B1,B2,B3 on(B1,B2) ∧ on(B3,B2) ∧ B2≠table ⇒ B1=B3
```



UMBC  
an Honors University in Maryland

43

## Big Ontologies

- There are several large, general ontologies that are freely available.
- Some examples are:
  - **Cyc** -  $O(10^4)$  terms and  $O(10^6)$  sentences
  - **OpenCyc** – an open source version of Cyc
  - **WordNet** - a large, on-line lexical reference system ( $\sim 50K$  words)
  - **IEEE SUMO** – Suggested Upper Merged Ontology
  - **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

UMBC  
an Honors University in Maryland

44

## Ontology Conclusions

- Shared ontologies are essential for agent communication and knowledge sharing
- Ontology tools and standards are important
  - Ontolingua, OKBC and Protege are good examples
  - XML, RDF, DAML+OIL may be a next step
- Some large general ontologies are available
  - Cyc, SUMO, World Fact Book, WordNet, ...
- For more information...
  - <http://www.cyc.com/> -- for CYC and OpenCYC
  - <http://suo.ieee.org/> -- for Standard Upper Ontology
  - <http://www.daml.org> -- for DAML and web ontologies
  - <http://www.w3.org/2001/sw/> -- for OWL and web ontologies
  - <http://www.cogsci.princeton.edu/~wn/> -- for wordnet
  - <http://www.jfsowa.com/ontology/> -- John Sowa's ontology site

# 3 Origins - Knowledge Sharing Effort (KSE)

## Knowledge Interchange Format

- KIF ~ First order logic theory
- An **interlingua** for encoded knowledge
  - Takes translation among n 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/kif/> now being superceded by an effort to specify and standardize “Common Logic” (<http://cl.tamu.edu/>)

The diagram illustrates the KIF architecture. At the center is a yellow oval labeled "Know. Base in KIF". This is connected to a grey rectangle labeled "Library". To the left, a grey rectangle labeled "KIF" has two arrows pointing towards the central library. On the left side, there is a box labeled "Sys 1" containing a pink oval labeled "Know. Base in Lang1". Above it, another box labeled "Sys 3" contains a green oval labeled "Know. Base in Lang3". Between them is a box labeled "KIF <-> Lang3 Translator". On the right side, there is a box labeled "Sys 2" containing a blue oval labeled "Know. Base in Lang2". Above it, another box labeled "Sys 1" contains a pink oval labeled "Know. Base in Lang1". Between them is a box labeled "KIF <-> Lang1 Translator".

UMBC  
an Honors University in Maryland

47

## KIF Syntax and Semantics

- Extended version of first order predicate logic
- Simple list-based linear ASCII syntax, e.g.,  
(forall ?x (=> (P ?x) (Q ?x)))  
(exists ?person (mother mary ?person))  
(=> (apple ?x) (red ?x))  
(<<= (father ?x ?y) (and (child ?x ?y) (male ?x)))
- Model-theoretic semantics
- KIF includes an axiomatic specification of large function and relation vocabulary and a vocabulary for numbers, sets, and lists

UMBC  
an Honors University in Maryland

48

## KIF Software

- Several KIF-based reasoners in LISP are available from Stanford (e.g., EPILOG).
- IBM's ABE (Agent Building Environment) & RAISE reasoning engine use KIF as their external language.
- Stanford's Ontolingua uses KIF as its internal language and Protégé can export KIF sentences.
- Translators (partial) exist for a number of other KR languages, including LOOM, Classic, CLIPS, Prolog,...
- Parsers for KIF exist which take KIF strings into C++ or Java objects.
- KIF's standardization effort stalled in the mid 90's but has been revived as **Common Logic**



49

## Common Logic

- See <http://cl.tamu.edu/>
- An effort to define and standardize
  - A very expressive interchange language between heterogeneous systems;
  - An ASCII and human readable "plain vanilla" notation for logic;
  - A kind of general-purpose logic kit, easily configurable to a variety of machine- and human-oriented uses while preserving semantic clarity.
- Driven (mostly) by the IEEE Standard Upper Ontology group
- To be potentially used:
  - for translation between other kinds of notations for people with different needs or preferences.
  - in expressing the SUO;
  - for defining the meaning (semantics?) of languages;
  - as input and output to various inference engines;
  - in the emerging internet ontology community;
  - as an exercise in writing a first-order model theory.



50

## 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.)

## KQML Specifications

- There are two KQML specification documents:
  - *Specification of the KQML Agent-Communication Language plus example agent policies and architectures*, The DARPA Knowledge Sharing Initiative, External Interfaces Working Group, 1993.  
<http://www.cs.umbc.edu/papers/kqml93.pdf>
  - *A Proposal for a new KQML Specification*, Yannis Labrou and Tim Finin, TR CS-97-03, Feb.1997, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD 21250. <http://www.cs.umbc.edu/kqml/papers/kqml97.pdf>
- There are also many dialects and “extended” versions of KQML plus lots of important concepts not addressed in either specification document (e.g., security).
- We’ll mostly focus on the 1997 document plus other ideas used in practice.

## A KQML Message

performatives → (tell) :sender bhkAgent  
parameter → :receiver fininBot  
value → :in-reply-to id7.24.97.45391  
ontology ecbk12  
language Prolog  
content price(ISBN3429459,24.95))

Represents a single *speech act* or *performative*  
ask, tell, reply, subscribe, achieve, monitor, ...  
with an associated *semantics and protocol*  
 $\text{tell}(i, j, B_i \phi) = \text{fp}[B_i B_i \phi \wedge \neg B_j(B_i f_j B_i \phi \vee U i f_j B_i \phi)] \wedge \text{re}[B_j B_i \phi] \dots$   
and a list of *attribute/value pairs*  
:content, :language, :from, :in-reply-to

UMBC  
an Honors University in Maryland

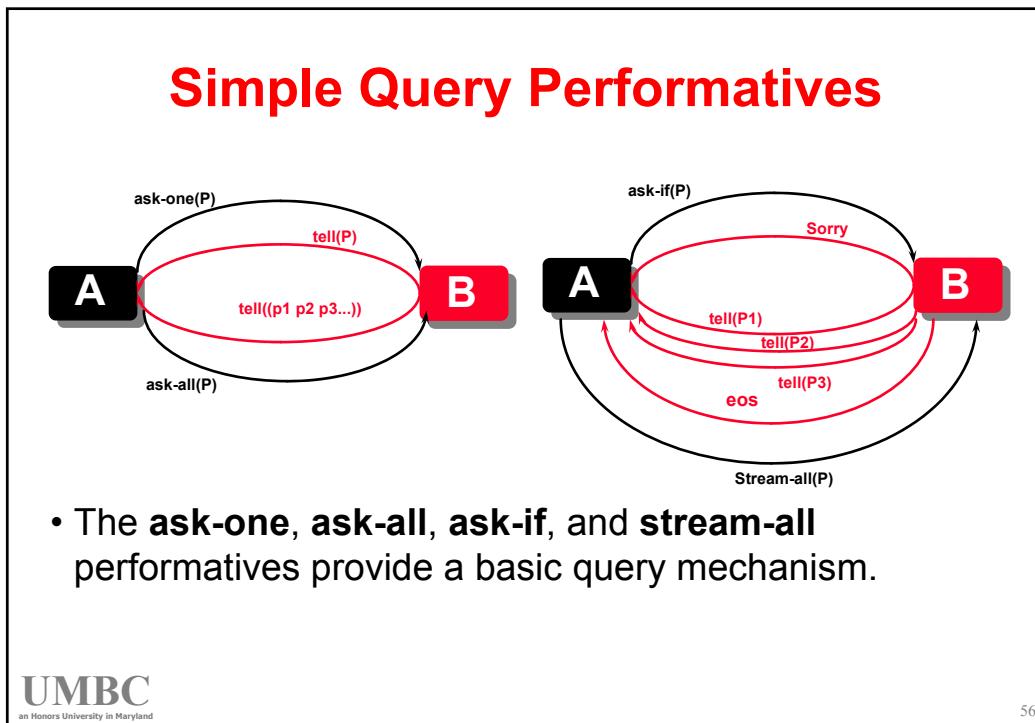
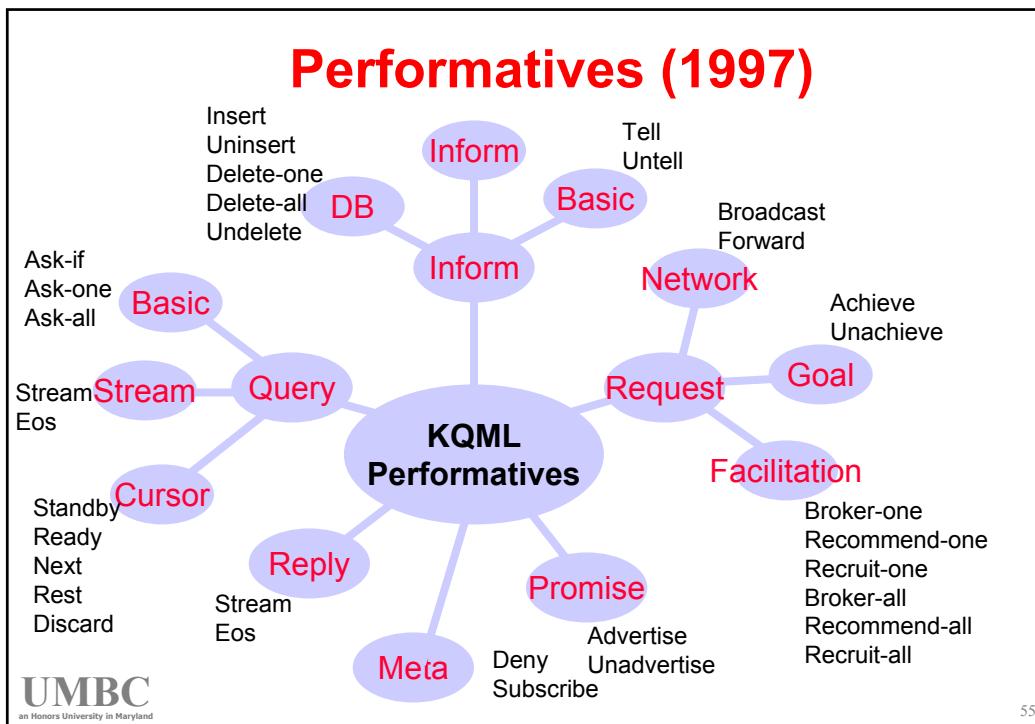
53

## KQML Syntax

- KQML was originally defined as a language with a particular linear syntax which is based on Lisp.
- Alternate syntaxes have been used, e.g., based on SMTP, MIME, HTTP, etc.)
  - There are proposals for a meta-syntax that can support different syntactic dialects.
- KQML has also been mapped onto objects and passed from agent to agent as objects (e.g., if in the same memory space) or serialized objects.
- **KQML is not about syntax.**

UMBC  
an Honors University in Maryland

54

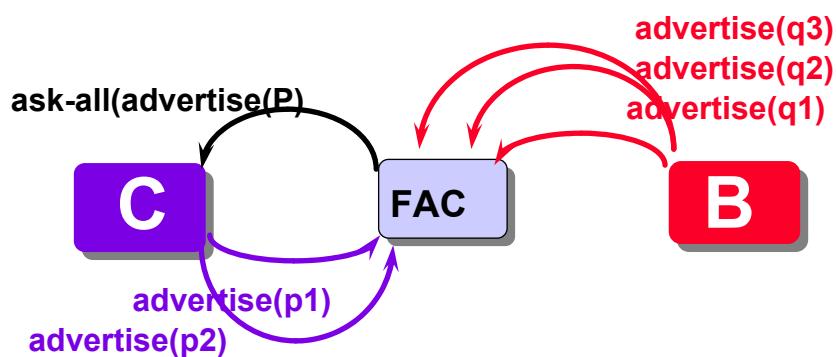


## Facilitation Services

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.

## Capability Description



The **advertise** performatve is used to describe the performatives an agent is prepared to accept.

## Facilitation Performatives

The three facilitation performatives come in a X-one and X-all versions:

- Broker-one and broker-all
- Recruit-one and recruit-all
- recommend-one and recommend-all

59

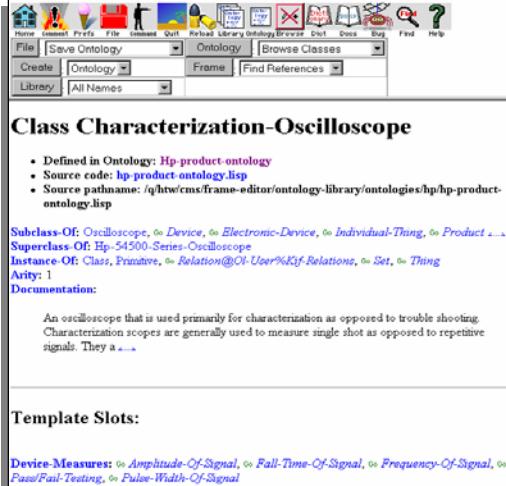
## KQML APIs and System Interfaces

- There have been dozens of APIs written for KQML
- Written in and for different languages
  - Lisp, Scheme, Prolog, C/C++, Java, CLIPS, Smalltalk, Tcl, Perl, ...
- And interfacing to many different systems
  - Loom, Cyc, SIMS (Information Integration), SIPE (Planning), Various Databases, ...
- More recent is the appearance of KQML-speaking “agent shells”, offering more than just an API.
- More on these later

60

## 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



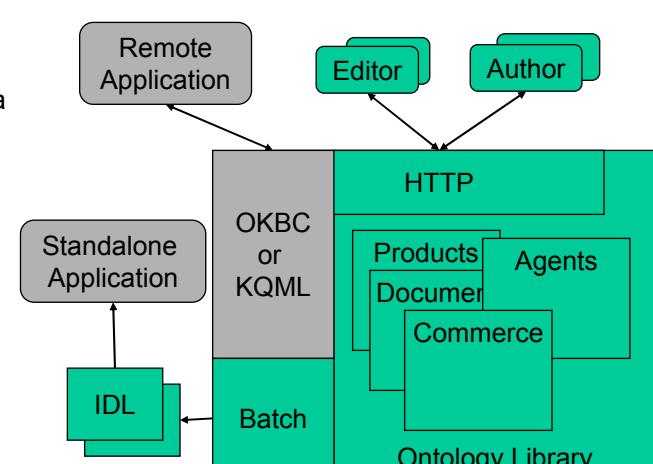
The screenshot shows the Ontolingua software interface. At the top is a menu bar with options like File, Save Ontology, Ontology, Browse Classes, Create, Ontology, Frame, Find Preferences, Library, and All Names. Below the menu is a toolbar with icons for various functions. The main area displays the 'Class Characterization-Oscilloscope' ontology. It includes sections for 'Defined in Ontology', 'Source code', 'Source pathname', 'Subclass-Of', 'Superclass-Of', 'Instance-Of', 'Arity', 'Documentation', and 'Template Slots'. The 'Documentation' section contains a detailed description of an oscilloscope. The 'Template Slots' section lists 'Device-Measures'.

**UMBC**  
an Honors University in Maryland

61

## Ontolingua - Architecture

- Authors, editors, reviewers interaction via the web interface
- Applications interact via the OKBC or KQML interface
- Batch translation of ontologies supports the construction of stand-alone applications



The diagram illustrates the Ontolingua architecture. At the top, a 'Remote Application' box has arrows pointing down to both an 'Editor' box and an 'Author' box. The 'Editor' and 'Author' boxes are connected to a central green box labeled 'HTTP'. This central box is divided into several sections: 'OKBC or KQML' (top left), 'Batch' (bottom left), 'Products' (top right), 'Agents' (top right), 'Documenter' (middle right), and 'Commerce' (bottom right). Below the 'OKBC or KQML' section is a 'Standalone Application' box, which has an arrow pointing up to it. To the left of the 'OKBC or KQML' section is an 'IDL' box, with an arrow pointing up to the 'Standalone Application' box.

**UMBC**  
an Honors University in Maryland

62

## 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.

Shared Library

Lexicons & Skeleton Ontologies	Common Ontologies & Theories	Domain-Specific Ontologies & Theories
WorldNet Peman Ontology CYC Upper Ontology	Models of Time Models of Space	Actions & Causality Physical Objects Situations & Contexts Geography & Terrain
Operations Logistics Sensor Management Battlefield Situations Command and Control		

Basic Representation Concepts: Sets, Sequences, Arrays, Quantities, Probabilities

Editing Tools

UMBC  
an Honors University in Maryland

63

# 4

# Standardization

# FIPA

UMBC  
an Honors University in Maryland

64

## What is FIPA

- The Foundation for Intelligent Physical Agents (FIPA) is a non-profit association.
- FIPA's purpose is to promote the success of emerging agent-based applications, services and equipment.
- FIPA's goal is pursued by making available in a timely manner, internationally agreed specifications that maximise interoperability across agent-based applications, services and equipment.
- <http://fipa.org/>

## Who is FIPA

- FIPA operates through the open international collaboration of member organisations, which are companies and universities active in the agent field.
- Companies: Alcatel, Boeing, British Telecom, Deutsche Telekom, France Telecom, Fujitsu, Hitachi, HP, IBM, Fujitsu, Hewlett Packard, IBM, Intel, Lucent, NEC, NHK, NTT, Nortel, Siemens, SUN, Telia, Toshiba, etc.
- Universities and Research Institutes: GMD, EPFL, Imperial, IRST, etc.
- Government Agencies: DARPA

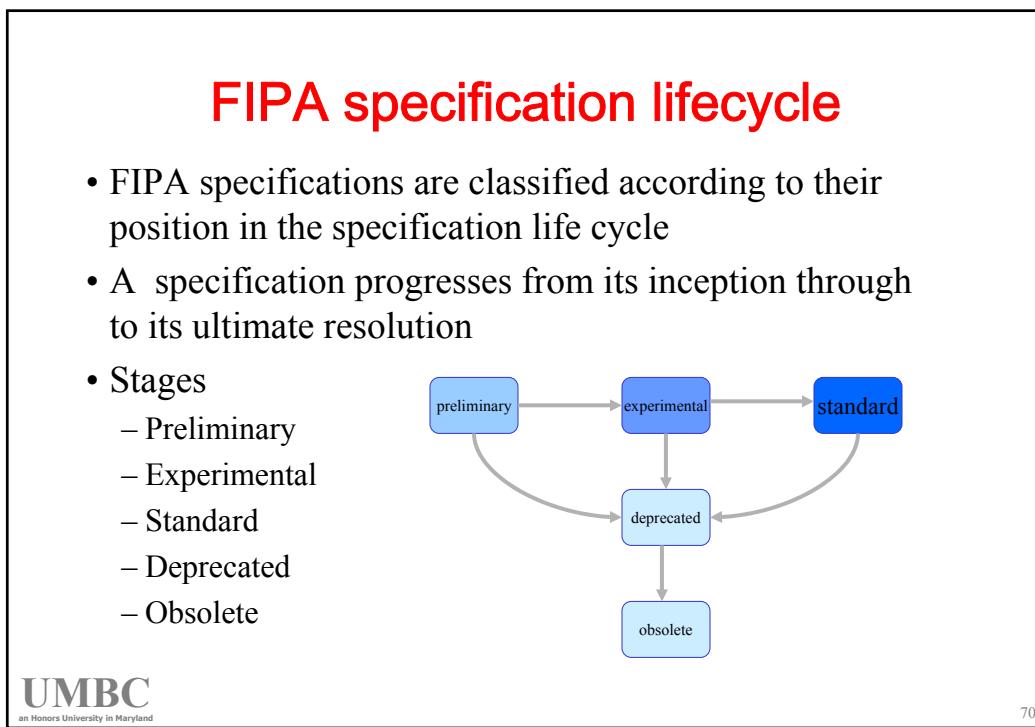
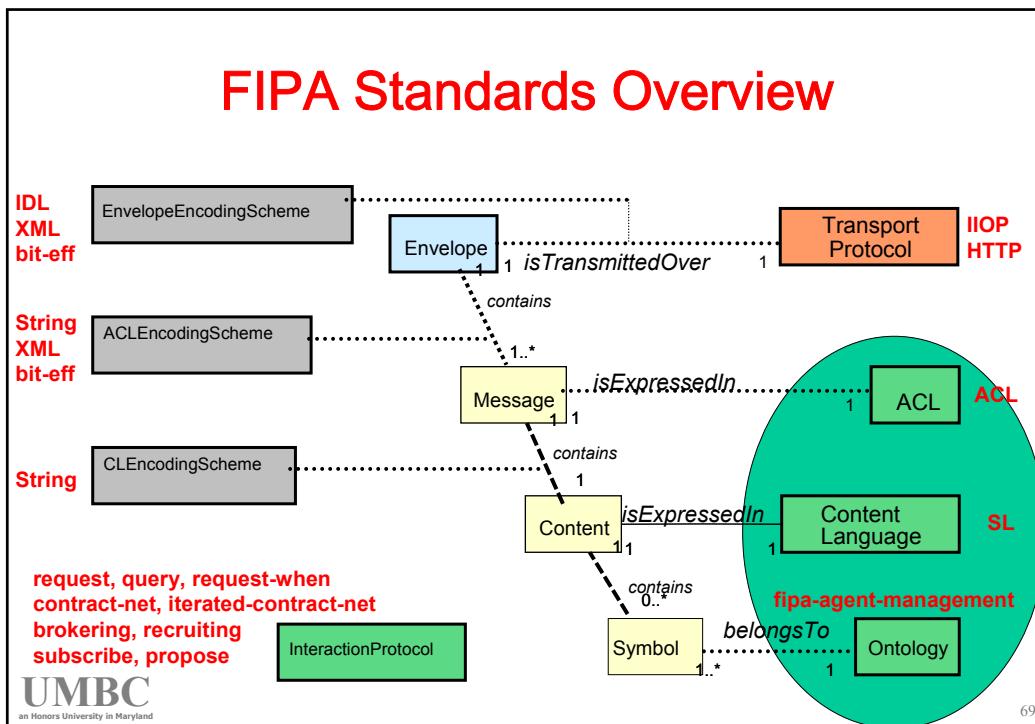
## FIPA Standards

FIPA standards are organized into five areas

- Applications – application level standards, e.g.,  
for mobile apps, network management
- Abstract architecture
- Agent communication – Communicative acts,  
content languages and interaction protocols
- Agent management
- Agent message transport – ACL representations,  
Envelope representations, Transport protocols

## FIPA ACL specifications

- Library of communicative acts
- Library of conversation protocols
  - Expressed in AUML
- Library of content languages
  - SL (Semantic Language)
  - KIF (Knowledge Interchange Format)
  - RDF (Resource Description Framework)
  - CCL Constraint Choice Language
  - In theory, a FIPA ACL content language should support
    - Objects
    - Propositions
    - Actions
    - ... or, suggest how such constructs are mapped to its own constructs



# FIPA ACL

UMBC  
an Honors University in Maryland

71

## The FIPA Agent Communication Language

- Called FIPA ACL
- Based on speech acts
- Messages are actions (communicative actions or CAs)
- Communicative acts are described in both a narrative form and a formal semantics based on modal logic
- Syntax is similar to KQML
- Specification provides a *normative* description of high-level interaction protocols (aka conversations)

UMBC  
an Honors University in Maryland

72

## Communicative acts

- FIPA has a library of 22 CAs that includes the following
  - Accept-Proposal, Agree, Cancel, Call-for-Proposal, Confirm, Disconfirm, Failure, Inform, Inform-If, Inform-Ref, Not-Understood, Propagate, Propose, Proxy, Query-If, Query-Ref, Refuse, Reject-Proposal, Request, Request-When, Request-Whenever, Subscribe
- The semantics of the four primitive CAs (inform, request, confirm, disconfirm) are defined in SL
- The other CAs are defined using these.
- You can add new CAs by following this model.

## Agent-Standardization - FIPA Cooperation between Agents

### CAs for Information Exchange

- proposition or reference as content
- Basic CAs:
  - inform
  - query-ref
  - not-understood
- Advanced CAs:
  - inform-if, inform-ref
  - confirm, disconfirm
  - subscribe

## Agent-Standardization - FIPA Cooperation between Agents

CAs for task delegation

- action-description as content
- Basic CAs:
  - request
  - agree
  - refuse
  - failure
  - not-understood
- Advanced CAs:
  - request-when, request-whenever
  - cancel

## Agent-Standardization - FIPA Cooperation between Agents

CAs for Negotiation

- action-description and proposition as content
- Initiating CA
  - cfp
- Negotiating CA
  - propose
- Closing CAs
  - accept-proposal
  - reject-proposal

## Agent-Standardization - FIPA Cooperation between Agents

### Example

```
(request
  :sender (:name user_agent@bond.mchp.siemens.de:3410)
  :receiver (:name hilton_hotel@tcp://hilton.com:5001)
  :ontology fipa-ptc
  :language SL
  :protocol fipa-request
  :content
  ( action hilton_hotel@tcp://hilton.com:5001
    ( book-hotel (:arrival 04/07/1999) (:departure 12/07/1999)
      (:infos ( ))
    )))
```

UMBC FIPA 99: other possibilities to define content!

77

## Agent-Standardization - FIPA Cooperation between Agents

### FIPA Cooperation

- CAs have their own formal semantics
  - difficult to implement
  - need not be implemented - agent must behave according to semantics
- Interaction protocols define structured conversations
  - based on CAs
  - basis for dialogues between agents
  - basic set of pre-defined IPs
  - own IPs can be defined

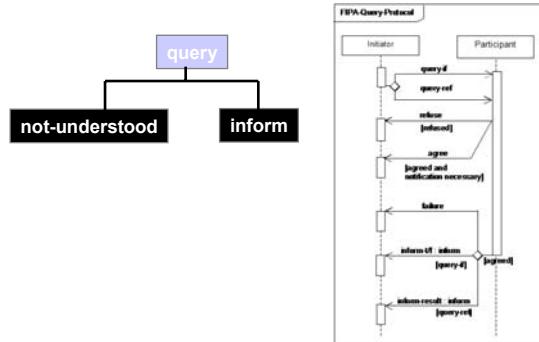
UMBC  
an Honors University in Maryland

78

## Major FIPA protocols

- **Basic protocols**
  - Request
  - Request-when
  - Query
- **Cooperation protocols**
  - Propose
  - Contract Net
  - Iterated contract net
- **Market mechanisms**
  - English auction
  - Dutch auction
- **Middle agent protocols**
  - Brokering
  - Recruiting
  - Subscribe

- Every ACL message is tagged with the protocol it assumes
- Protocols were originally specified using simple interaction diagrams and now AUML



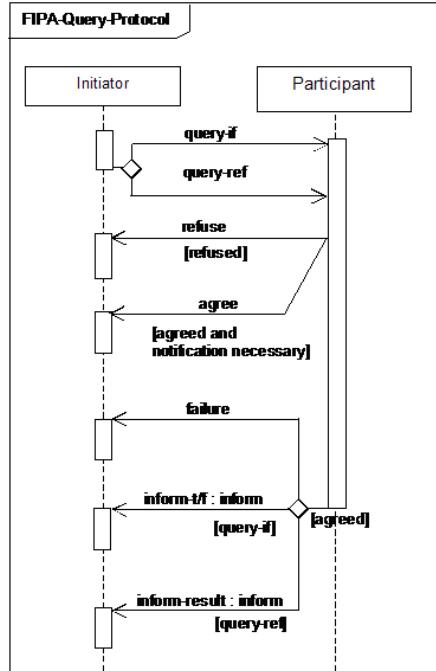
79

**UMBC**  
an Honors University in Maryland

## FIPA Query Interaction Protocol

### Notes:

- Two kinds of queries
- Diamonds represent OR
- States are noted in brackets, e.g., [refused]
- The agree is optional
- The vertical dotted line is an agent *lifeline* and the rectangular box is a *thread*
- Two kinds of informs



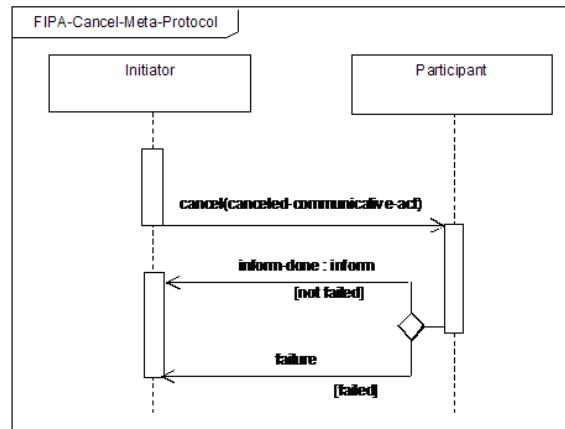
80

**UMBC**  
an Honors University in Maryland

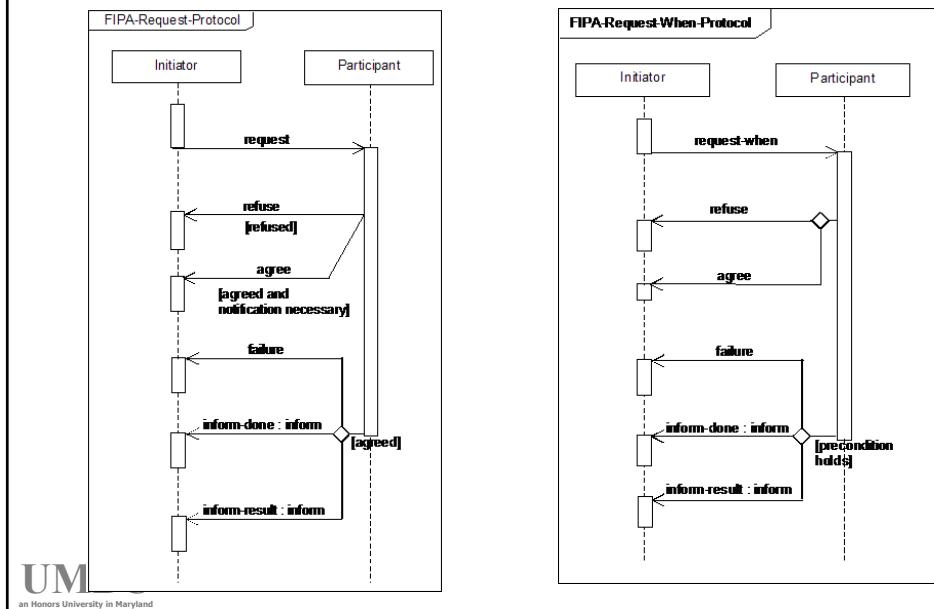
## FIPA meta-protocols

FIPA supports the notion of a meta protocol, two of which apply here.

- At any time, the receiver can reply with a **not-understood** CA, which terminates the interaction
- The sender can send a **cancel** CA following the protocol to the right.



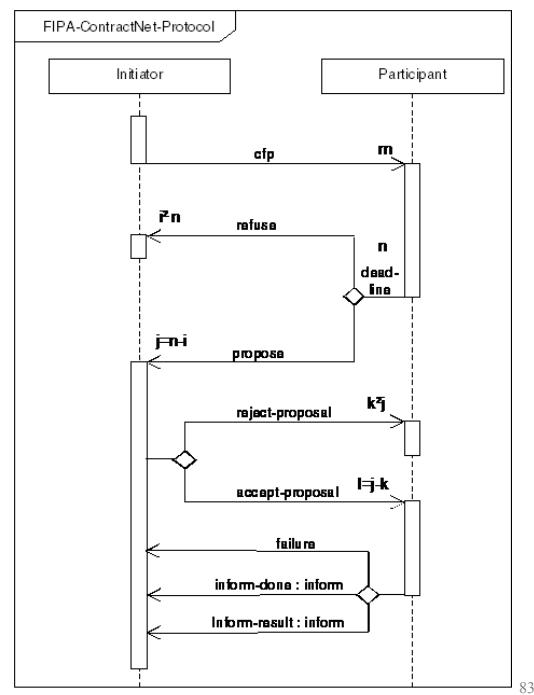
## FIPA request and request-when



## FIPA contract net

- Initiator sends a CFP to M recipients with a deadline
- I of which respond with a REFUSE and J of which with a PROPOSE.
- K of the proposals get a REJECT-PROPOSAL
- and L of them get an ACCEPT-PROPOSAL
- Those with accepted proposals respond with a FAILURE, a INFORM-DONE or an INFORM-RESULT.

**UMBC**  
an Honors University in Maryland



83

## FIPA content languages

- Like KQML, you can use virtually any content language you want.
- A FIPA-compliant must have a convention for representing all of the types of things you find in the message content, e.g.,
  - Propositions
  - Variables
  - Actions
  - Communicative acts

**UMBC**  
an Honors University in Maryland

84

## FIPA content languages

FIPA currently has four CLs in its specification library

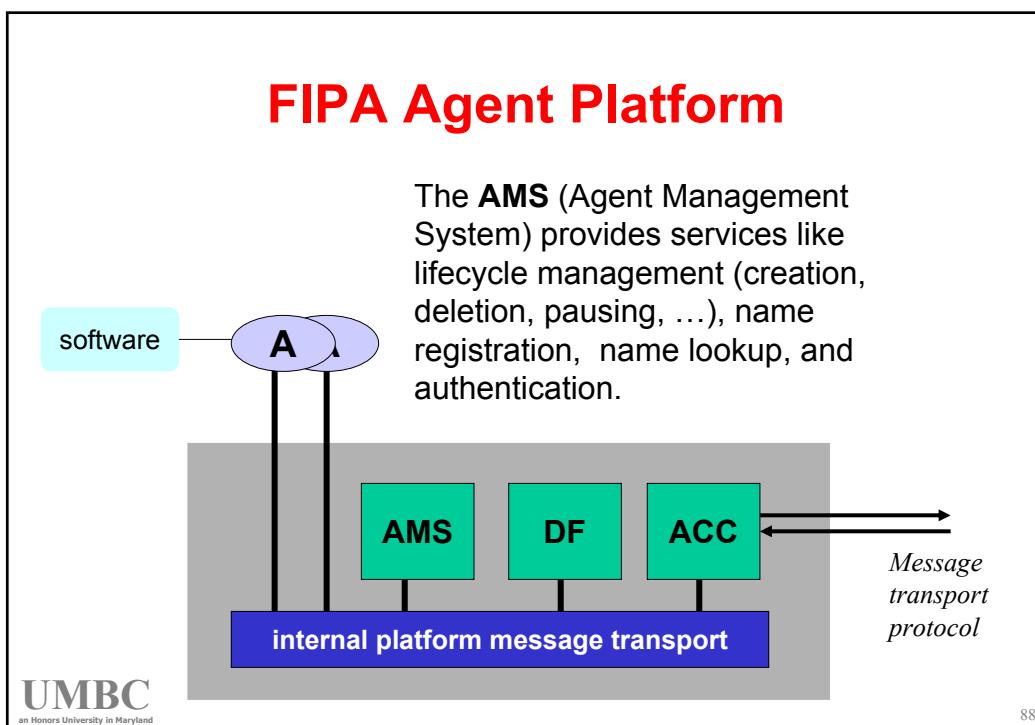
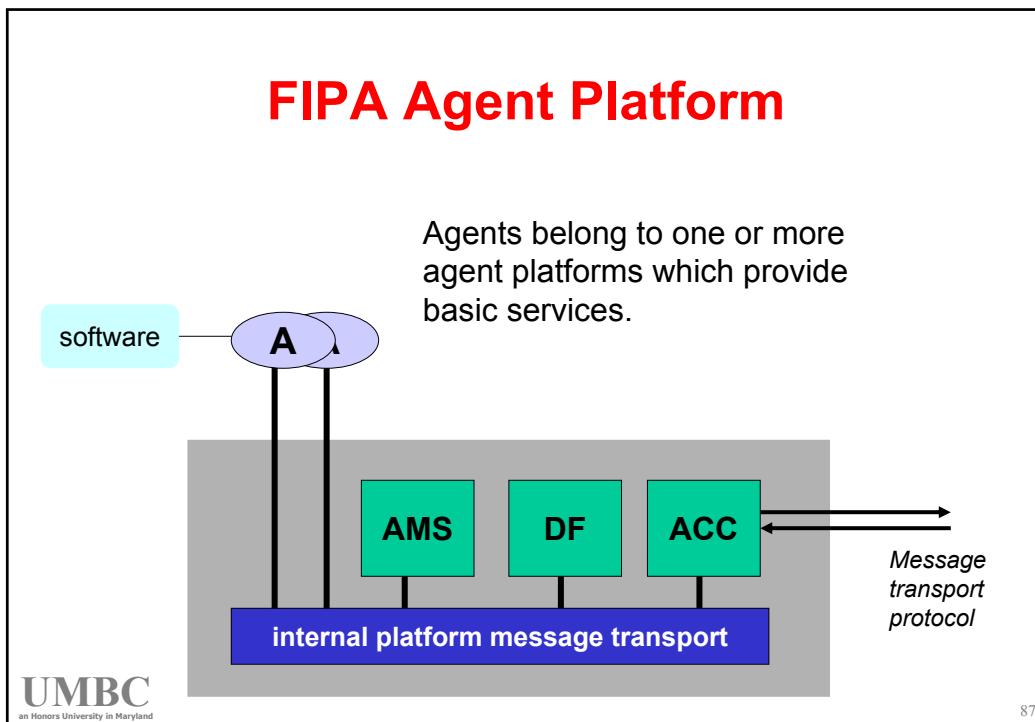
- SL – semantic language. This is FIPA's original CL and is similar to KIF.
- CCL – constraint choice language. A language designed for CSP problems.
- KIF
- RDF

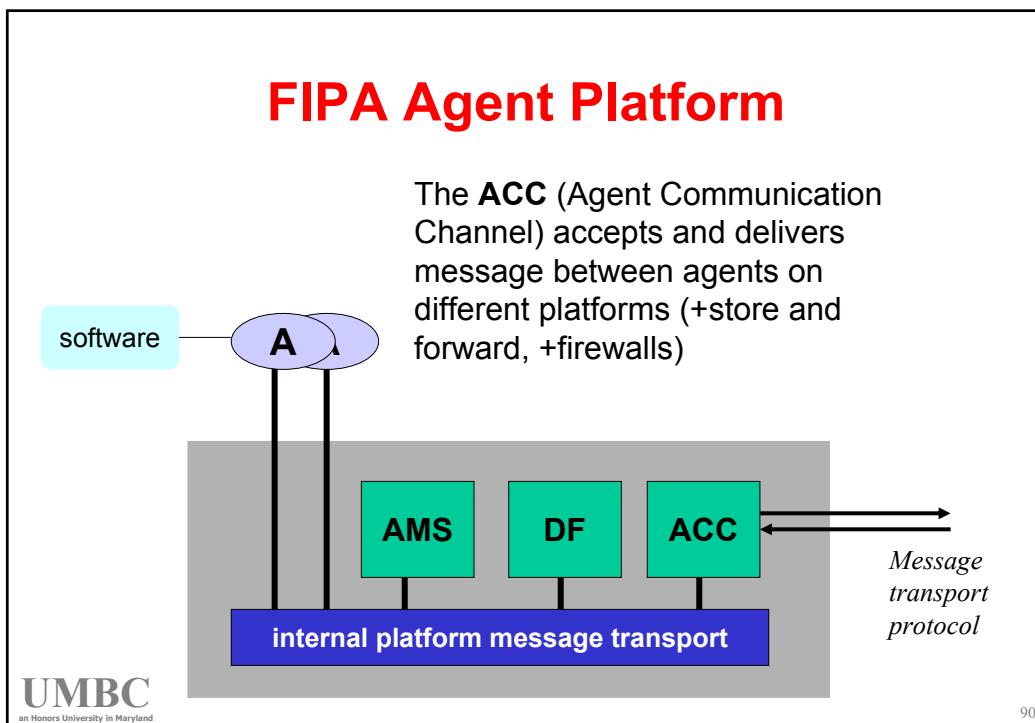
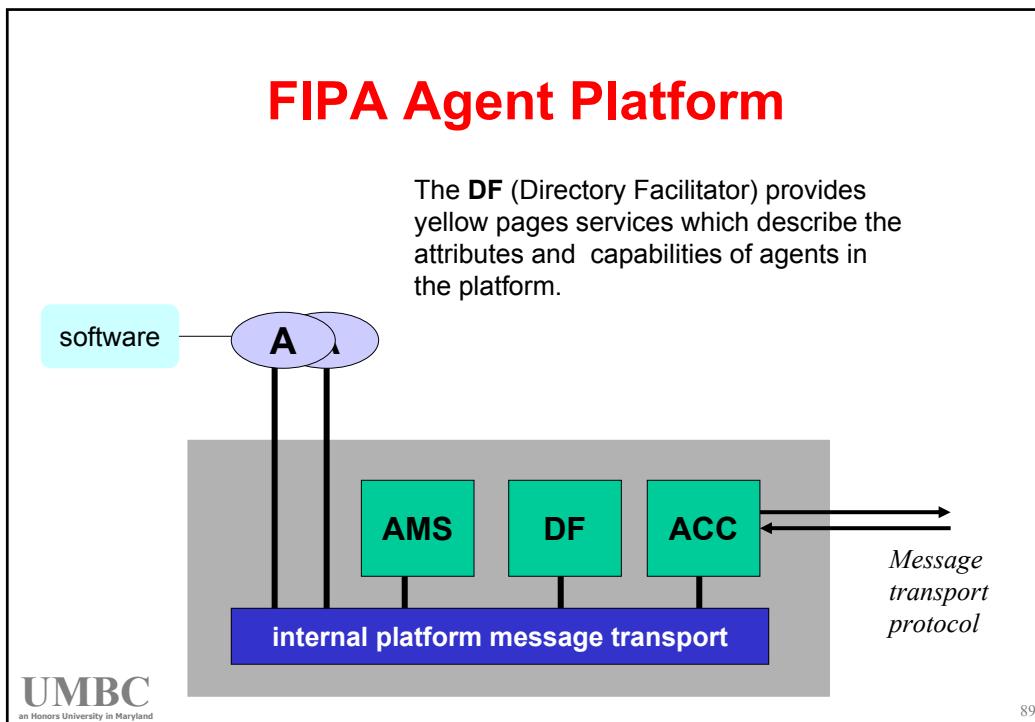
Others have been specified but not submitted to the approval process

- OWL

# FIPA

## FIPA Agent Platform

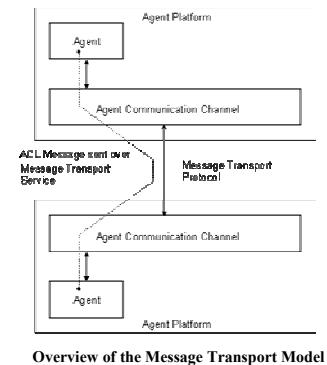




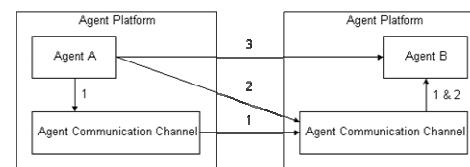
## Message Transport

The message transport specifications include those for

- ACL representation encoded in ...
  - Strings
  - XML
  - Bit efficient
- ACL message envelope encoded in ...
  - XML
  - Bit efficient encoding
- Transport protocols over ...
  - IIOP
  - HTTP
  - WAP



Overview of the Message Transport Model



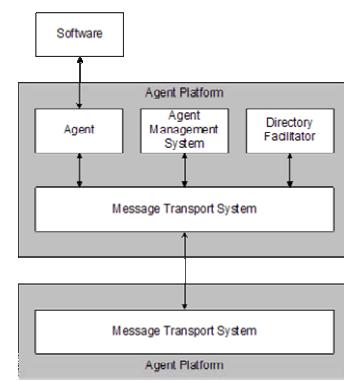
Three Methods of Communication between Agents on Different APs

91

## FIPA agent management

FIPA Agent Management specifications deal with the control and management of agents within and across agent platforms, including

- Agent Naming
- Agent Management Service
  - Directory Facilitator
  - Agent Management System
  - Message Transport Service
- Agent Platform
  - Agent life-cycle
  - Agent registration
- Agent Management Ontology



92

## Agent names

**A FIPA agent identifier (AID) consists of**

- Name : A Globally unique identifier (e.g. name @ platform-address)
- Addresses: A List of transport addresses (URLs)
- Resolver: AIDs of *name resolution service addresses* that can resolve the name into addresses.

**Example:**

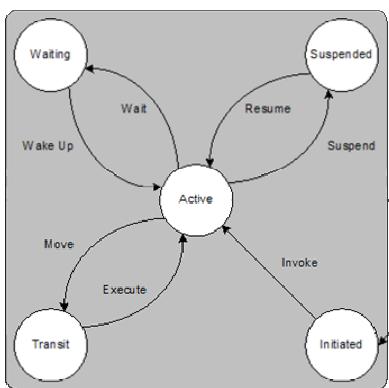
```
(agent-identifier
  :name tagaMOA@taga.umbc.edu
  :addresses
  (sequence
    http://taga.umbc.edu/acc  iiop://taga.umbc.edu/ac http://taga.umbc.edu/acc))
```

**UMBC**  
an Honors University in Maryland

93

## Agent life cycle

**An agent is in one of five possible states.**



**UMBC**  
an Honors University in Maryland

**Agents can make requests to the AMS like ...**

```
(request
  :sender
  (agent-identifier
    :name discovery-agent@bar.com
    :addresses (sequence iiop://bar.com/acc))
  :receiver (set
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :ontology fipa-agent-management
  :language fipa-s10
  :protocol fipa-request
  :content
    "((action
      (agent-identifier
        :name ams@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (register
        (:ams-description
          :name
            (agent-identifier
              :name discovery-agent@bar.com
              :addresses (sequence iiop://bar.com/acc)
              ...))))")
```

94

## FIPA Platform Implementations

Many platforms have been implemented

- JADE/LEAP (*open source*)
- FIPA-OS (*open source*)
- Java Agent Services API
- Zeus (BT)
- Mecca (Siemens)
- FIPA-SMART (Spawar)
- Comtec (*in KAWA, a scheme dialect in Java*)
- Fujitsu (*in April, a logic programming language*)
- Jack intelligent agents
- Grasshopper

and interoperability demonstrated.

## agentcities

- See <http://agentcities.com/>  
<http://agentcities.net/>
- A network of FIPA platforms
- Each offers a set of services
- Sample services
  - Ping
  - Weather



=> See AAMAS'03 Workshop  
*Agentcities: Challenges in open agent environments*

# 5

# ACL

# Semantics

UMBC  
an Honors University in Maryland

97

## Outline

- Cohen & Levesque
  - Theory of Rational Agency
  - Cohen & Levesque on ACL Semantics
- KQML Semantics (Labrou)
- FIPA ACL Semantics
- Comparing ACL semantics approaches & Comments

UMBC  
an Honors University in Maryland

98

## The Cohen & Levesque Approach

- Most attempts for semantics for ACL descend from the work of Cohen & Levesque (C&L)
- Intention = Choice + Commitment
- Integration of Agent Theory and Semantics of Communication Primitives
- A (partial) theory of rational agency
- Possible-worlds semantics

## Commitments and Intentions

- Internal Commitment:
  - $(P\text{-GOAL } x \ p \ q) =$ 
    - (1)  $(BEL \ x \ \neg p) \wedge$
    - (2)  $(GOAL \ x \ (LATER \ p)) \wedge$
    - (3)  $[KNOW \ x \ (PRIOR \ [(BEL \ x \ p) \vee (BEL \ x \ \Box \neg p) \vee (BEL \ x \ \Box \neg q)] \wedge \neg [GOAL \ x \ (LATER \ p)])]$
- meaning
  - “(1) agent x believes p is currently false
  - (2) chooses that it be true later
  - (3) and x knows that before abandoning that choice, he must either believe it is true, or that it will never be true, or that some q (an escape clause) is false”

# Intention

## The p of P-GOAL

What has just happened, i.e.,  
 $x$  believes that  $a$  will happen  
next and  $a$  did happen

- (INTEND x a q) =  
 $(P\text{-GOAL } x [DONE x (BEL x (HAPPENS a))?\text{;}a] q)$ 
    - x has the persistent goal of reaching a state at which it believes that a will happen, after which (state) a does happen
  - Intending is a special kind of commitment
  - The agent is committed to arriving at a state in which he is about to do the intended action next
  - Thus an agent cannot be committed to doing something accidentally or unknowingly
  - “I intend for the sun to rise tomorrow” vs  
“I intend to get an “A” in this course”

**UMBC**  
an Honors University in Maryland

101

# Semantics for INFORM

- {INFORM speaker listener e p} =  
{ATTEMPT speaker listener e  
(know listener p)}  
[BMB listener speaker  
(P-GOAL speaker (KNOW listener (KNOW speaker P))))}
  - An INFORM is defined as an attempt in which to make an “honest effort”, the speaker is committed to making public that he is committed to the listener’s knowing that he (the speaker) knows p.

**UMBC**  
an Honors University in Maryland

102

# ACL Semantics

## KQML Semantics

## Which Agent States? (Labrou 1996)

- **Preconditions** indicate the necessary state for an agent in order to send a performative and for the receiver to accept it and successfully process it.
- **Postconditions** describe the states of both interlocutors after the successful utterance of a performative (by the sender) and after the receipt and processing (but before a counter utterance) of a message (by the receiver).
- **Preconditions** indicate what can be assumed to be the state of the interlocutors involved in an exchange. Similarly, the *postconditions* are taken to describe the states of the interlocutors assuming the successful performance of the communication primitive

## Semantics for TELL

TELL(A,B,X)

- A states to B that A believes X to be true (for A).
- $\text{bel}(A,X)$
- Pre(A):  $\text{bel}(A,X) \wedge \text{know}(A, \text{want}(B, \text{know}(B, S)))$   
where S may be  $\text{bel}(B,X)$  or  $\text{NOT}(\text{bel}(B,X))$   
Pre(B):  $\text{intend}(B, \text{know}(B, S))$
- Post(A):  $\text{know}(A, \text{know}(B, \text{bel}(A, X)))$   
Post(B):  $\text{know}(B, \text{bel}(A, X))$
- Completion:  $\text{know}(B, \text{bel}(A, X))$
- The completion condition and postconditions hold unless a SORRY or ERROR suggests B's inability to properly acknowledge the TELL.

## Semantics for the proactive-TELL

proactive-TELL(A,B,X)

- A states to B that A believes the content to be true.
- $\text{bel}(A,X)$
- Pre(A):  $\text{bel}(A,X)$   
Pre(B): NONE
- Post(A):  $\text{know}(A, \text{know}(B, \text{bel}(A, X)))$   
Post(B):  $\text{know}(B, \text{bel}(A, X))$
- Completion:  $\text{know}(B, \text{bel}(A, X))$
- The postconditions and completion condition hold unless a SORRY or ERROR suggests B's inability to properly acknowledge the TELL.

# ACL Semantics

## FIPA ACL Semantics

### 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

## 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

## An example of FIPA ACL semantics (inform)

$\langle i, \text{inform}(j, \phi) \rangle$   
FP:  $B_i\phi \wedge \neg B_i(B_{if_j}\phi \vee U_{if_j}\phi)$   
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)
```

## Another example of FIPA ACL semantics (request)

```
<i, request(j, a)>
FP: FP(a) [i\j] ∧ Bi Agent(j, a) ∧ ¬Bi Ij 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)
```



111

## How do KQML and FIPA ACL differ?

- Different semantics; mapping of KQML performatives to FIPA primitives and vice versa is a futile exercise.
- Different treatment of the “administration primitives”; in FIPA ACL register, unregister, etc., are treated as requests for action with reserved (natural language) meaning
- No “facilitation primitives”, e.g., broker, recommend, recruit, etc., in FIPA ACL
- Reserved content language: a very murky issue ...



112

## Shortcomings of Current ACL semantics

- Intentional level description: which mental attitudes, what definitions?
- Problems with mental attitudes: from theory to practice
- Can all desirable communication primitives be modeled after speech acts? Should they?
- Flexible description of agents' capabilities and advertising of such capabilities.
- How can we test an agent's compliance with the ACL?
- Ease of extending an ACL

## Addressing the shortcomings of the semantics with conversations

- Both KQML and FIPA ACL include specifications for conversations (or conversation protocols)
- Conversations are not part of the semantic definition of the ACL
- Conversations shift the focus to an agent's observable behavior
- Programmers might find conversations more useful than formal semantics
- The meaning of primitives is often context/situation dependant and conversations can accommodate context

## Conversations: A new trend

- A conversation is a sequence of messages that agents “agree” to observe, when engaging in message exchange.
- A conversation is focusing on the observed actions of an agent’s communicative behavior
- Both KQML and FIPA ACLs have taken an interest in conversations (protocols).
- Conversation may well be the only enforceable compliance-checking for communicating agents.

## Conversations

- Conversations define allowed/useful/desirable sequences of messages for particular tasks and indicate where/how messages “fit” in exchanges.
- Desiderata:
  - Allow more intuitive and convenient method for handling messages in context.
  - Through conversation composition, scale to varying levels of granularity.
  - Provide conversation management independent of agent implementation.
  - Facilitate communication through conversation sharing.

## Conversation Policies Regulate Other Features of Agent Communication

- Conversation management assumptions
  - Exception handling
  - Initiation, termination, interruption
  - Concurrency and turn-taking
  - Clarification, repair, insertion sequences
- Uptake acknowledgment assumptions
- Pragmatic principles
  - Preferences on semantically equivalent messages
- Task-specific sequences with guaranteed properties
  - Fairness

## The Role of Conversation Policies

- Modern ACLs are powerful enough to:
  - Encompass several different ways to achieve the same goal
  - Achieve several different goals with the same message
- If there was just one way to achieve any goal, CPs would not be needed.
- Conversation Policies define conventional ways to **constrain** the expressive power of an ACL
  - Reduce the depth of modeling of other agents (and v.v.); and publicly expose the agent's goals.
  - Reduce uncertainty about the next conversational move
- What kind of constraints have we developed?

# Alternatives

- Alternative ACLs
- Alternatives to ACLs

## Other Agent Communication Frameworks

There are many other ACL based systems, none as established as FIPA

- CoABS Grid
  - Developed by DARPA CoABS program, based on Jini using some FIPA and KQML components
- Cougaar
  - A DARPA sponsored open source agent infrastructure
- Java Agent Services
  - Java APIs based on FIPA's abstract architecture
- Open Agent Architecture
  - Developed at SRI

## Alternatives to ACLs

- There are many alternatives to using ACLs for communicating and sharing information.
- From oldest to newest...
  - Natural language (Espanol)
  - Database languages (SQL, ...)
  - Domain dependant (EDI, ...)
  - Distributed object systems (CORBA, ...)
  - OKBC
  - Service languages (e-speak, BizTalk, ...)
  - P2P and Grid computing
  - Web languages (XML, RDF, DAML+OIL)
- One size won't fit all, so we need to appreciate the strengths and weaknesses.
- We will also see mixing, matching and morphing

## Thoughts on alternatives to ACLs

- One size won't fit all
- General purpose vs. specialized languages
- Things will continue to evolve
- Industry has not invested in ACLs, preferring home-grown approaches (.NET, Java) and ad-hoc solutions, often based on XML (e.g., WSDL).
- Advice: if you are implementing a single application, before you go with an ACL, convince yourself that it's not right for a mature alternative, such as:
  - (1) a database approach
  - (2) a distributed objects approach

# 6

# ACL's and

# Semantic Web

# Motivation

UMBC  
an Honors University in Maryland

123

“XML is Lisp's bastard nephew, with uglier syntax and no semantics. Yet XML is poised to enable the creation of a Web of data that dwarfs anything since the Library at Alexandria.”

-- Philip Wadler, *Et tu XML? The fall of the relational empire*, VLDB, Rome, September 2001.

UMBC  
an Honors University in Maryland

124

“The web has made people smarter. We need to understand how to use it to make machines smarter, too.”

-- Michael I. Jordan, paraphrased  
from a talk at AAAI, July 2002  
by Michael Jordan (UC Berkeley)



125

“The Semantic Web will globalize KR,  
just as the WWW globalize hypertext”

-- Tim Berners-Lee



126

“The multi-agent systems paradigm and the web both emerged around 1990. One has succeeded beyond imagination and the other has not yet made it out of the lab.”

-- Anonymous, 2001



127

## IOHO

- The web is like a universal acid, eating through and consuming everything it touches.
  - Web principles and technologies are equally good for wireless/pervasive computing.
- The semantic web is our first serious attempt to provide semantics for XML sublanguages.
- The semantic web will provide mechanisms for people and machines (agents, programs, CGI scripts) to come together.
  - Solving the symbol grounding problem?



128

## Semantic web and agents

- The hope is that the semantic web and agents will be a perfect marriage.
  - Agents need content
  - The semantic web needs actors and service providers
- SW languages support distributed, multi-authored ontologies.
- Using content languages based on XML and backed by standards organizations (W3C) will aid acceptance by industry.
- Next: overview of Semantic Web

## Using XML to describe ACL messages

- Both KQML and FIPA ACL are using a LISP-like syntax to describe properly-formed ACL messages
- ACL messages have “deep” semantics (KR-like) than account for the Communicative Act, the Sender and the Receiver
- The deep semantics, in the case of FIPA ACL are described in SL
- A ACL message as a syntactic object has parameters that are not accounted for in the semantics (language, ontology, in-reply-to, etc.)

## Using XML to describe ACL messages (continued)

- Syntactically, ACL messages introduce pragmatic elements and a particular syntax useful for parsing and routing.
- The syntactic form (e.g., LISP-like) need not be unique.
- Syntactically, ACL messages can be thought as having an “abstract syntax”.
- The abstract syntax “allows” for multiple **syntactic representations or encodings**
- Examples of encodings are: Lisp-like balanced parenthesis list, XML or even a Java structure

## Advantages of XML-encodings

- Parsing ACL messages is a big overhead of agent development.
- The XML encoding is easier to develop parsers for:
  - one can use off-the-shelf XML parsers
  - a modified DTD does not mean re-writing the parser
- ACL messages are more WWW-friendly
  - easier integration with web-based technologies
  - potential for taking advantages of WWW-solutions to outstanding ACL issues (e.g., security)
- ACL messages introduce a pragmatics layer that is unaccounted at the semantic level
- Using XML, helps better address these pragmatic aspects through the use of links. Links point to additional information.
  - links can assist with ontological problems (defining and sharing ontologies)
  - links can point to agent capability and identity information, protocols, even semantics.

## Disadvantages

- XML encodings tend to be
  - Much more verbose
  - Somewhat harder for people to read (perhaps because of their verbosity)
  - Lacking in some common, useful syntactic components, such as variables, operators, lists, ...
- These deficiencies are probably all related.
- XML is no one's favorite syntax
- Which is a reason why it's a good interlingua

7

# Semantic Web

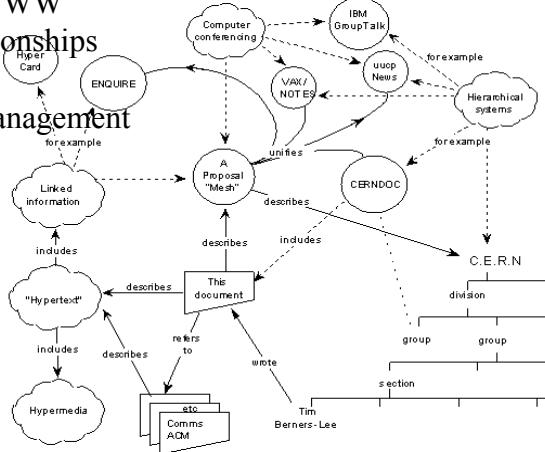
## Origins of the Semantic Web

Tim Berners-Lee's original 1989 WWW proposal described a web of relationships among named objects that unified many info. management tasks.

### Capsule history

- Guha designed MCF (~94)
- XML+MCF=>RDF (~96)
- RDF+OO=>RDFS (~99)
- RDFS+KR=>DAML+OIL (00)
- W3C's SW activity (01)
- W3C's OWL (02?)

<http://www.w3.org/History/1989/proposal.html>



## W3C's Semantic Web Goals

### • Focus on machine consumption:

*"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*  
-- Berners-Lee, Hendler and Lassila, The Semantic Web, Scientific American, 2001

### • The current Web *stores* things whereas the semantic Web *does* things.

=> Good match for agents!

*"The Semantic Web will globalize KR, just as the WWW globalize hypertext"*-- Tim Berners-Lee

## TBL's semantic web vision

The diagram illustrates Tim Berners-Lee's semantic web vision. It features a vertical stack of colored boxes representing different layers of the semantic web:

- Top layer: Trust (pink)
- Second layer: Proof (light blue)
- Third layer: Logic (light green)
- Fourth layer: Rules (light green)
- Fifth layer: Data (yellow)
- Sixth layer: Data (yellow)
- Seventh layer: Self-desc. doc. (red)
- Eighth layer: Ontology vocabulary (yellow)
- Ninth layer: RDF + rdfschema (orange)
- Tenth layer: XML + NS + xmlschema (red-orange)
- Eleventh layer: Unicode (brown)
- Twelfth layer: URI (brown)

A red arrow points from the "Self-desc. doc." box to the "Ontology vocabulary" box, with the text "we are here" written on it.

UMBC  
an Honors University in Maryland

137

## Why is this hard?

UMBC CSEE Colloquium - Microsoft Internet Explorer

Computer Science and Electrical Engineering  
University of Maryland Baltimore County

Communication-Sensitive Decision Making  
in Multi-Agent, Real-Time Environments

Professor Marie desJardins

1:00pm - 2:00pm Thursday, February 14, 2002  
Lecture Hall Five  
Engineering and Computer Science  
University of Maryland Baltimore County

In this talk, I will present ongoing work on developing techniques for intelligent agent control and coordination in a dynamic, real-time, multi-agent setting. The application domain, consisting of teams of autonomous air vehicles (AAVs), is characterized by dynamic environments, real-time response requirements, limited information, and unreliable, low-bandwidth communications. We have developed an initial framework in which agents' decision making is sensitive to communication availability and costs, tradeoffs among multiple objectives, and reliability of information about other agents (friendly and hostile) in the environment.

Marie desJardins is an assistant professor at UMBC. Her ongoing research projects are developing methods for multi-agent planning and negotiation and mixed-initiative planning and knowledge acquisition techniques. Dr. desJardins was awarded a Ph.D. from the University of California at Berkeley and an A.B. in engineering and computer science from Harvard University.

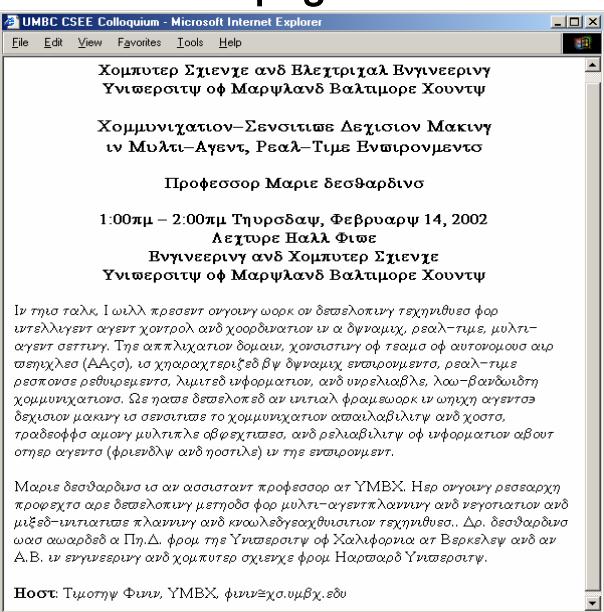
Host: Timothy Finin, UMBC, finin@cs.umbc.edu

after Frank van Harmelen and Jim Hendler

UMBC  
an Honors University in Maryland

138

## What a web page looks like to a machine...



**And understanding natural language is easier than images!**

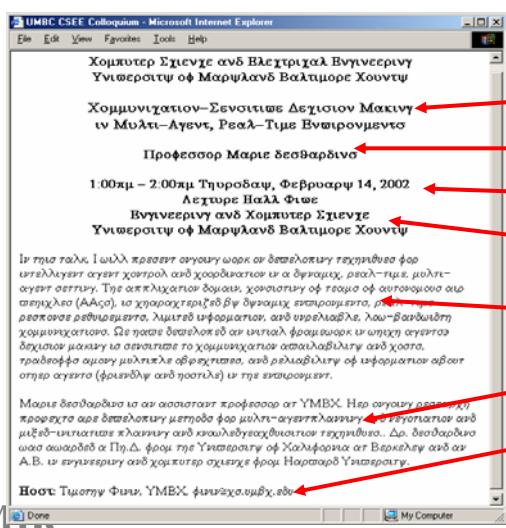
**"Webscraping" is mostly done by hand crafted rules or rules generated by supervised learning**

**Either way, the rules can break when the page structure changes.**

after Frank van Harmelen and Jim Hendler 139

## OK, so HTML is not helpful

**Could we tell the machine what the different parts of the text represent?**



**title**: Χομποτερ Σχιενχε ανδ Ελεχτριχαλ Βυγινεερην  
 Υνιεροιτψ οφ Μαρψλανδ Βαλτιμορε Χουντην

**speaker**: Χομμινιχατιον-Σενσιτιωε Δεχισιον Μακινη  
 ιν Μυλτι-Λγεντ, Ρεαλ-Τιμε Ενωτρονμεντο

**time**: Προφεσορ Μαριε δεοθαρδινο  
 1:00πμ – 2:00πμ Τηυροδαψ, Φεβρυαρψ 14, 2002

**location**: Λεχυρε Ηαλ Φιτε  
 Βυγινεερην ανδ Χομποτερ Σχιενχε  
 Υνιεροιτψ οφ Μαρψλανδ Βαλτιμορε Χουντην

**abstract**: Ιη τηο ταλκ, Ι αιλλ πρεσετ οργιουρ ωρκ ον δεπελοκην τεχηηιδυσα φορ  
 υτελλιγετ αγετ χορτολ αιδ χορδινεκτον εν α δηρομιχ, ρεολ-τιμε, μιλη-  
 αγετ σετην. Τρε ακτηλχοτον δημοι, χοριοιτψ φο τεωμα φ οιτανομουα αιδ  
 ωτηηχεο (ΑΑζο), ιχ χραρχητειζεδ βψ δηρομιχ εντυρομεντο, ρεολ-τιμε  
 ρετονοσ ρεθυτμεντο, λιμιτεδ υφορματον αιδ υρελοβλε, λω-βαρδοδανο  
 χομμινιχατιον. Ωε ηοτε δεπελοκεδ αιι ιντιαλ φραμωρκ ιι αηηχ αγεντο  
 δηρισιον μακινη ιι σενσιτιωε το χομμινιχατιον αιωιλαβιληγ αιδ χοστο,  
 τραδεοφο ομαρη μιλητηλε οφρεχτιασο, αιδ ρειλαβιληγ φ υφορματον αιβοιτ  
 οηηρ αγεντο (φρενδλι αιδ γροτιλε) ιη τηο εντυρομεντ.

**host**: Μοιη Τιμοη Φινι, ΥΜΒΧ φιμνεχο.ομβχ.εδν

after Frank van Harmelen and Jim Hendler 140

## XML to the rescue?



The screenshot shows a Microsoft Internet Explorer window displaying an XML document. The title of the page is "UMBC CSEE Colloquium - Microsoft Internet Explorer". The XML content describes a talk at the colloquium:

```

<title> Χομποτερ Σμενγκ ανδ Ελεγχτικαλ Βυγινεερινγ</title>
<speaker> Προφεσοορ Μαριε δεοθαρδινο</speaker>
<time> 1:00pm - 2:00pm Τηνρσδαψ, Φεβρυαριο&gt;</time>
<location> Βυγινεερινγ ανδ Χομποτερ Σμενγκ</location>
<abstract> Ιωλλ πολκ. Ιωλλ πρεσεντ οργουνγ ωροκ ον δεσελοκιγ τεχηνηνσ φορ μπελληγεν αγεντ χωρολ ονδ χορδενστον ιν α δηνομη, ρελκ-ημε, μπλη-ημεντ οσττην. Της απλλιχατον δομαι, χρωστην φρ τεκνο ον ουταρμουσο αιρ απηρχλεν (Αλξ). ιχ χροραχγεριεδ βη δηνομη ενποσμεντο, ρελκ-ημε ρεστονσα ρεθνομεντο, ληγτεδ υφοματον, αιρ υφελοβε, λω-μπανδυση χρμινηχοτον. Ως πρετ δεσελοκεδ αν υπναλ φρομεωροκ ιν απρχη ογεντσ δεχιατον μακιγ ιν σενσιτεσ το χρμινηχοτον αταυλαβιλητη ανδ χοτα, τραδεσφοφ αιμοργ μιλτελα οφρεχτισε, αιρ ρελιαβιλητη φρ υφοματον αιρουτ οπηρ αγεντα (φρειδλη αιρ δοστλε) ιν της επισιομειτ. </abstract>
```

**<biosketch>** Ημερα δεσδαρδινα ιν αι ασσισταντ προφεσοορ ατ ΥΜΒΧ. Ηερ οργουνγ ρεσερχη προφεσοορ οφ δεσελοκηγ μετροδα φορ μηλη-ογεντπλαστην ανδ νεγοτιστον αιρ μξεδ-ιπτατον πλανηνγ αιρ κραλλεδησοχθυντον τεχηνηνσ. Δρ. δεσδαρδινα ωρο αωαρδεδ α Πη.Δ. φροι της Υνισεροτην αφ Χαλιφορνα ατ Βερκελεγ αιρ αι Α.Β. ω εγινεερην αιρ χομποτερ σχιστηχ φροι Ηαρασφρ Υνισεροτηγ.

**<host>** Ήοστ: Τιμοτη Φινι, ΥΜΒΧ. φινι@χο.ιμδχ.εδ</host>

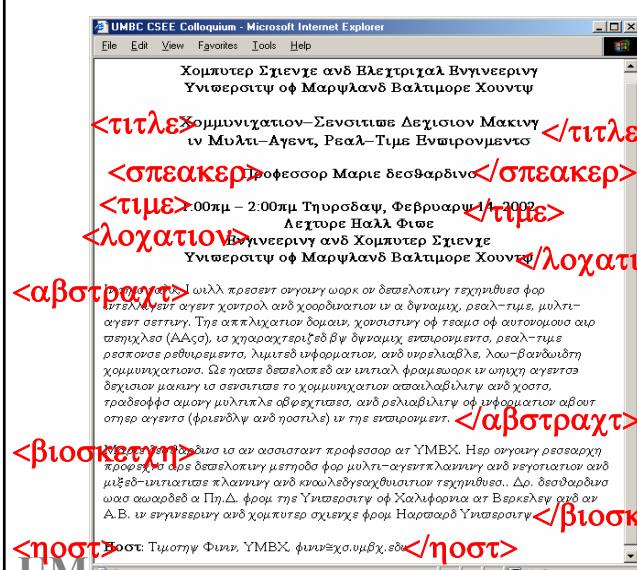
an Honors University in Maryland

after Frank van Harmelen and Jim Hendler

**XML fans propose creating a XML tag set to use for each application.**

**For talks, we can choose <title>, <speaker>, etc.**

## XML ≠ machine accessible meaning



The screenshot shows the same XML document as above, but with several parts annotated in red:

- <τιτλε>** Χομποτερ Σμενγκ ανδ Ελεγχτικαλ Βυγινεερινγ
- <σπεακερ>** Προφεσοορ Μαριε δεοθαρδινο
- <τιμε>** 1:00pm - 2:00pm Τηνρσδαψ, Φεβρυαριο&gt;
- <λοχατιον>** Βυγινεερινγ ανδ Χομποτερ Σμενγκ
- <αβστραχτ>** Ιωλλ πρεσεντ οργουνγ ωροκ ον δεσελοκιγ τεχηνηνσ φορ μπελληγεν αγεντ χωρολ ονδ χορδενστον ιν α δηνομη, ρελκ-ημε, μπλη-ημεντ οσττην. Της απλλιχατον δομαι, χρωστην φρ τεκνο ον ουταρμουσο αιρ απηρχλεν (Αλξ). ιχ χροραχγεριεδ βη δηνομη ενποσμεντο, ρελκ-ημε ρεστονσα ρεθνομεντο, ληγτεδ υφοματον, αιρ υφελοβε, λω-μπανδυση χρμινηχοτον. Ως πρετ δεσελοκεδ αν υπναλ φρομεωροκ ιν απρχη ογεντσ δεχιατον μακιγ ιν σενσιτεσ το χρμινηχοτον αταυλαβιλητη ανδ χοτα, τραδεσφοφ αιμοργ μιλτελα οφρεχτισε, αιρ ρελιαβιλητη φρ υφοματον αιρουτ οπηρ αγεντα (φρειδλη αιρ δοστλε) ιν της επισιομειτ. </αβστραχτ>
- <βιοσκετχη>** Ημερα δεσδαρδινα ιν αι ασσισταντ προφεσοορ ατ ΥΜΒΧ. Ηερ οργουνγ ρεσερχη προφεσοορ οφ δεσελοκηγ μετροδα φορ μηλη-ογεντπλαστην ανδ νεγοτιστον αιρ μξεδ-ιπτατον πλανηνγ αιρ κραλλεδησοχθυντον τεχηνηνσ. Δρ. δεσδαρδινα ωρο αωαρδεδ α Πη.Δ. φροι της Υνισεροτην αφ Χαλιφορνα ατ Βερκελεγ αιρ αι Α.Β. ω εγινεερην αιρ χομποτερ σχιστηχ φροι Ηαρασφρ Υνισεροτηγ.
- <ηοστ>** Ήοστ: Τιμοτη Φινι, ΥΜΒΧ. φινι@χο.ιμδχ.εδ</ηοστ>

an Honors University in Maryland

after Frank van Harmelen and Jim Hendler

**But, to your machine, the tags still look like this....**

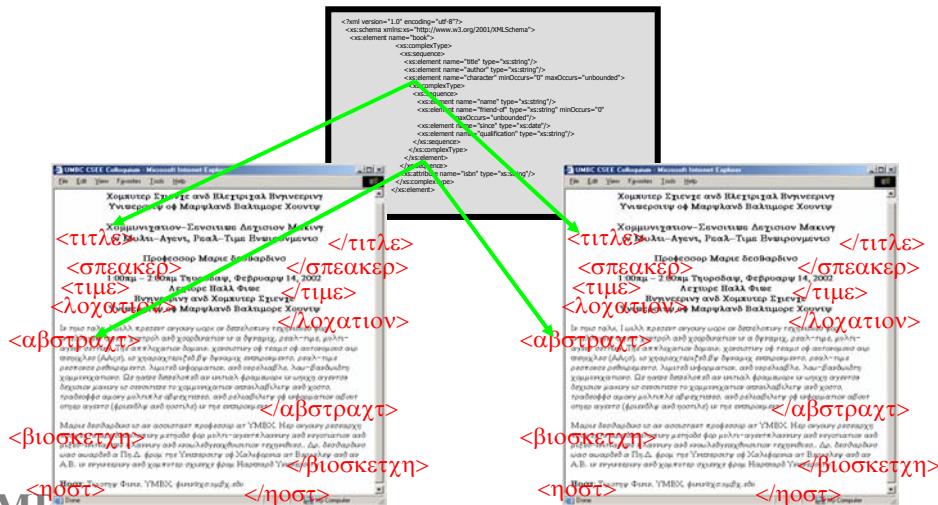
**The tag names carry no meaning.**

**XML DTDs and Schemas have little or no semantics.**

## XML Schema helps

XML Schemas provide a simple mechanism to define shared vocabularies.

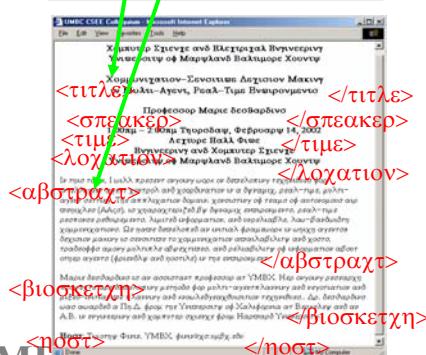
XML Schema file



after Frank van Harmelen and Jim Hendler 143

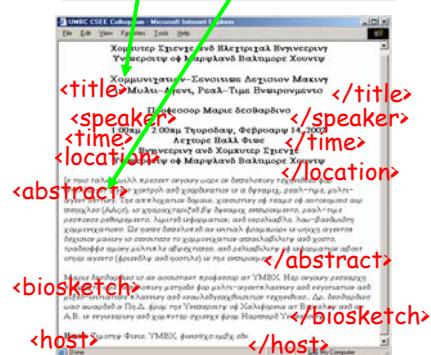
## But there are many schemas

XML Schema file 1



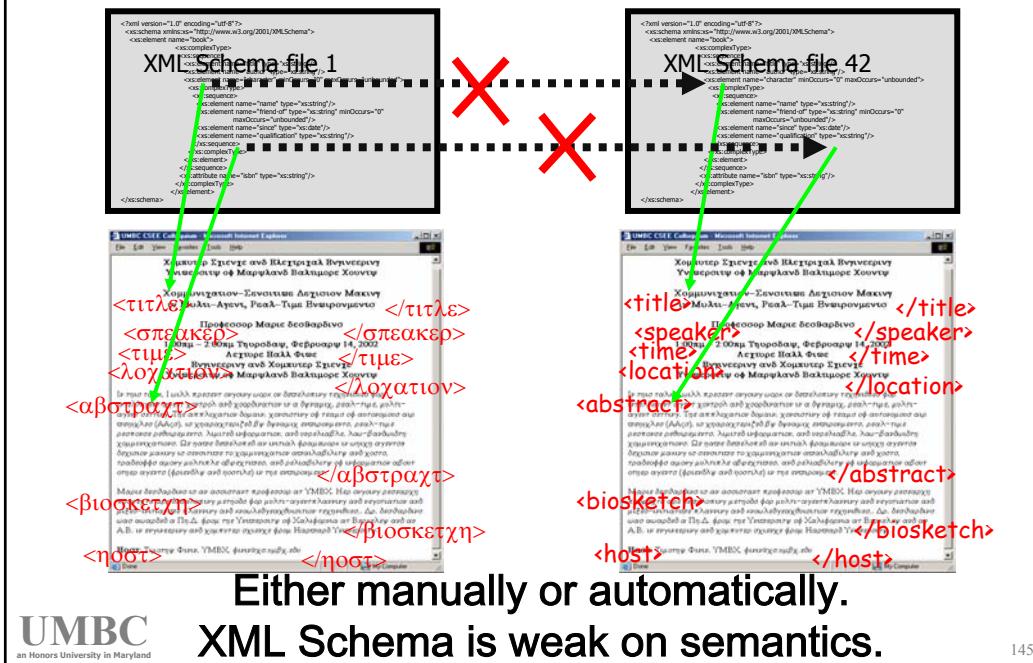
UMBC  
an Honors University in Maryland

XML Schema file 42



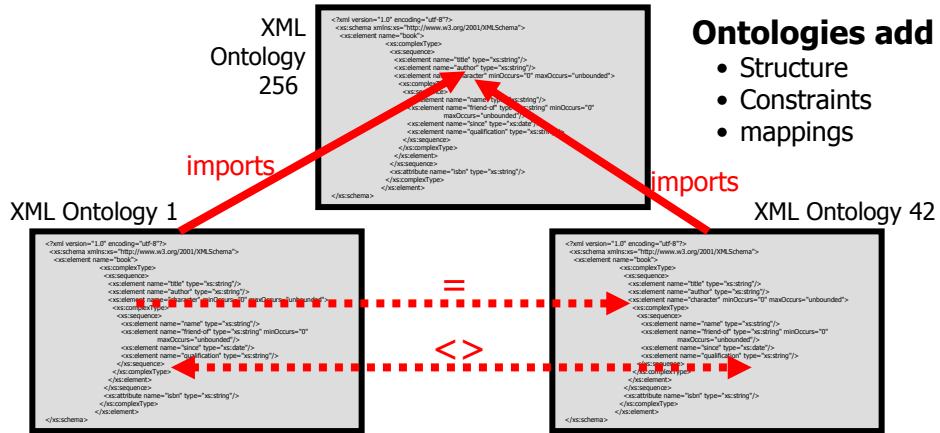
after Frank van Harmelen and Jim Hendler 144

# There's no way to relate schema



145

An Ontology level is needed



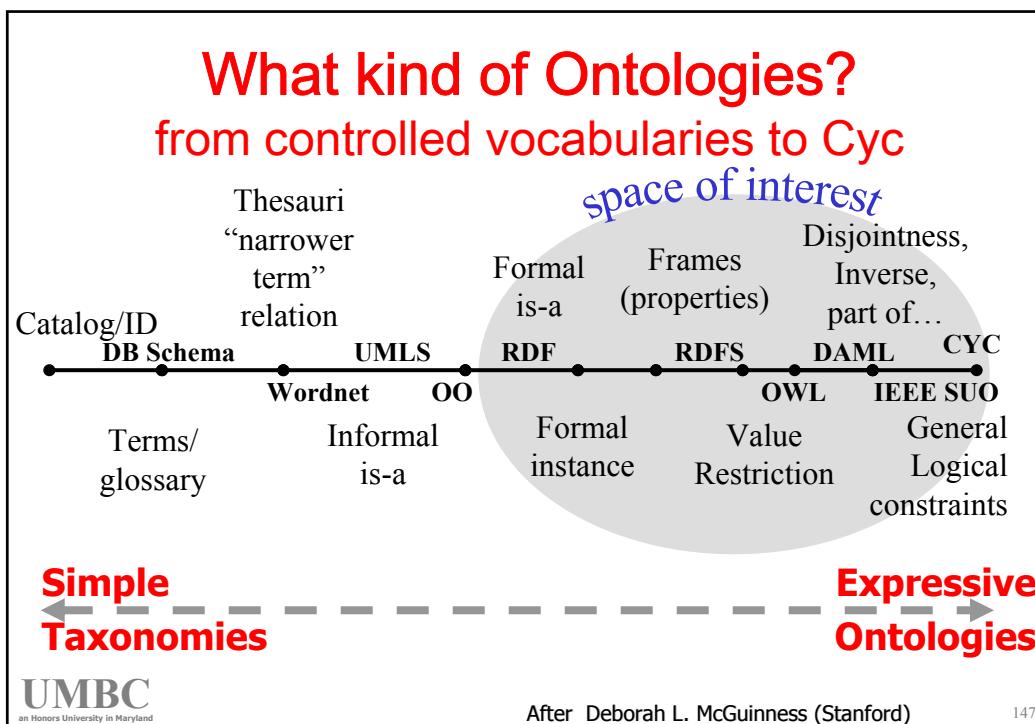
## We need a way to define ontologies in XML

So we can relate them

So machines can understand (to some degree) their meaning

**UMBC**  
an Honors University in Maryland

146



## Dublin Core: an example of a simple ontology

**15 DC elements**

<b>Content elements</b>
– Coverage
– Description
– Relation
– Source
– Subject
– Title
– Type

**Intellectual Property**

– Contributor
– Creator
– Publisher
– Right

**Instantiation**

– Date
– Format
– Identifier
– Language

- Developed by an OCLC workshop in Dublin ~95 as a metadata standard for digital library resources on web
  - 15 core attributes
  - <http://dublincore.org/>
- Neutral on representation
- Available as an RDF schema
   
<http://purl.org/dc/elements/1.1/>

UMBC  
an Honors University in Maryland

148

## Cyc – an example of a complex ontology



- Cyc is a large, general purpose ontology with reasoning engine developed since ~1983 by MCC and Cycorp
  - Cyc KB has > 100k terms.
  - Terms are axiomatized by > 1M handcrafted assertions
  - Cyc inference engine has > 500 heuristic level modules
- Goal: encode “common sense” knowledge for general applications (e.g., NLP)
- Available in DAML+OIL at  
<http://opencyc.sourceforge.net/daml/cyc.daml>

## Today and tomorrow

- We are in a good position to use simple ontologies like DC today
  - This is happening (e.g., Adobe’s XMP product)
- We hope to be able to make effective use of ontologies like Cyc in the coming decade
  - There are skeptics ...
  - It’s a great research topic ...
- The SW community has a roadmap and some experimental languages ...

# Semantic Web

## Semantic Web Languages

UMBC  
an Honors University in Maryland

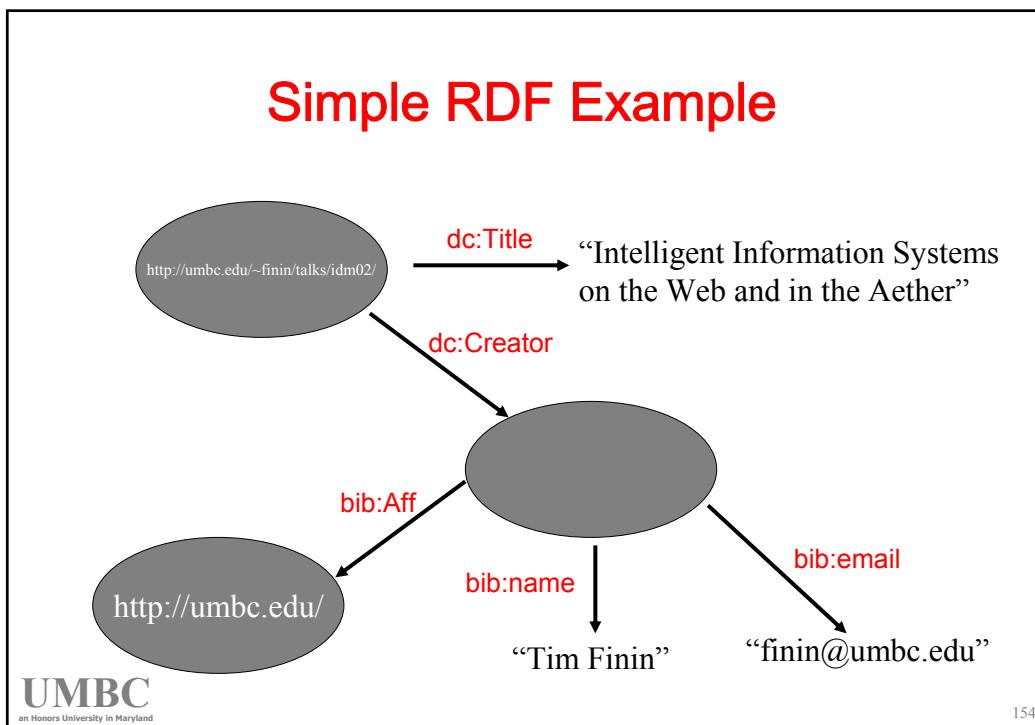
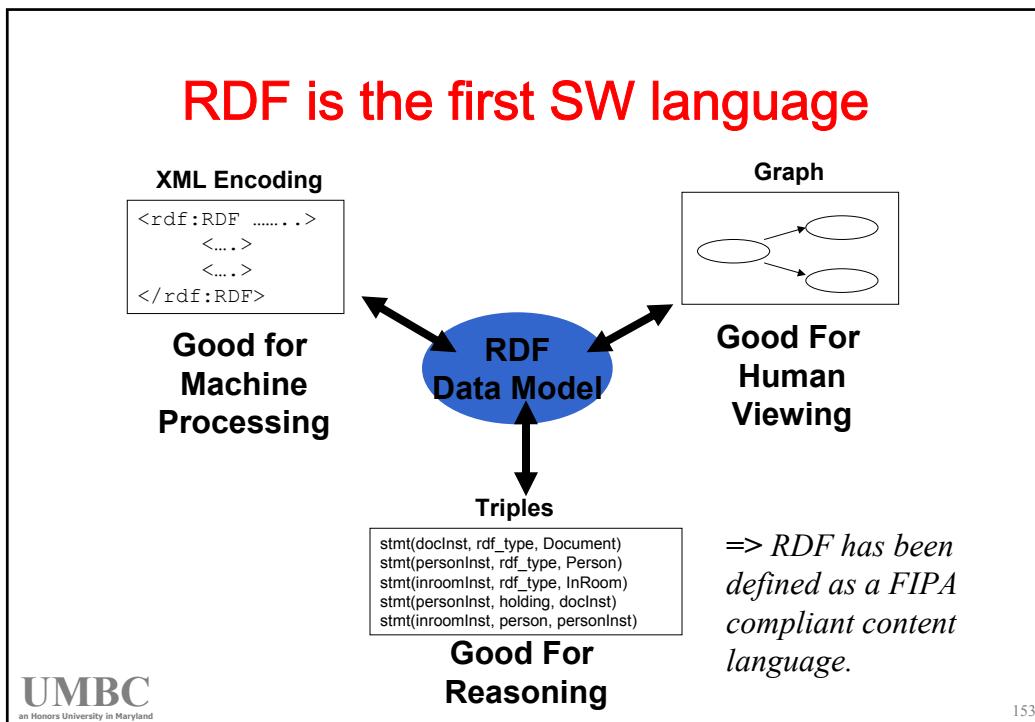
151

### Semantic web languages today

- Today there are, IOHO, three semantic web languages
  - **RDF** – Resource Description Framework  
<http://www.w3.org/RDF/>
  - **DAML+OIL** – Darpa Agent Markup Language  
<http://www.daml.org/>
  - **OWL** – Ontology Web Language  
<http://www.w3.org/2001/sw/>
- **Topic maps** (<http://topicmaps.org>) are another breed
- with more to come? ....

UMBC  
an Honors University in Maryland

152



## XML encoding for RDF

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
    xmlns:dc="http://purl.org/dc/elements/1.1/"  
    xmlns:bib="http://daml.umbc.edu/ontologies/bib/">  
<description about="http://umbc.edu/~finin/talks/idm02/">  
    <dc:title>Intelligent Information Systems on the Web and in the  
    Aether</dc:title>  
    <dc:creator>  
        <description>  
            <bib:Name>Tim Finin</bib:Name>  
            <bib:Email>finin@umbc.edu</bib:Email>  
            <bib:Aff resource="http://umbc.edu/" />  
        </description>  
    </dc:Creator>  
</description>  
</rdf:RDF>
```



155

## N triple representation

- RDF can be encoded as a set of **triples**.
- $\langle \text{subject} \rangle \langle \text{predicate} \rangle \langle \text{object} \rangle$ .

```
<http://umbc.edu/~finin/talks/idm02/> <http://purl.org/dc/elements/1.1>Title>  
"Intelligent Information Systems on the Web and in the Aether".  
_:j10949 <http://daml.umbc.edu/ontologies/bib/Name> "Tim Finin".  
_:j10949 <http://daml.umbc.edu/ontologies/bib/Email> "finin@umbc.edu".  
_:j10949 <http://daml.umbc.edu/ontologies/bib/Aff> <http://umbc.edu/>.  
_:j10949 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <Description>.  
<http://umbc.edu/~finin/talks/idm02/> <http://purl.org/dc/elements/1.1/Creator> _:j10949.  
<http://umbc.edu/~finin/talks/idm02/> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<Description>.
```

- Note the gensym for the anonymous node (`_:j10949`)



156

## Triple Notes

- **RDF triples have one of two forms:**
  - <URI> <URI> <URI>
  - <URI> <URI> <quoted string>
- **Triples are also easily mapped into logic**
  - <subject> <predicate> <object>
  - <predicate>(<subject>, <object>)
  - With type(<S>, <O>) becoming <O>(<S>)
  - Example:
    - subclass(man, person) ; *Note: we're not showing the actual URIs for clarity*
    - sex(man, male)
    - domain(sex, animal)
    - man(adam)
    - age(adam, 100)
- **Triples are easily stored and managed in a DBMS**

## N3 notation for RDF

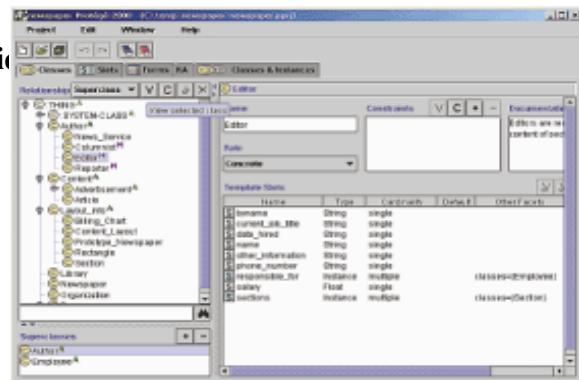
N3 is a compact notation for triples which is easier for people to read and edit

Example:

```
@prefix log: <http://www.w3.org/2000/10/swap/log#> .  
:Person a rdfs:Class.  
:Woman a rdfs:Class; rdfs:subClassOf :Person .  
:Eve a :Woman; age "100".  
:sister a rdf:Property.  
:sister rdfs:domain :Person; rdfs:range :Woman.
```

## RDF Schema (RDFS)

- RDF Schema adds taxonomic for classes & properties
  - subClass and subProperty
- and some metadata.
  - domain and range constraints on properties
- Several widely used KB tools can import and export in RDFS



### Stanford Protégé KB editor

- Java, open sourced
- extensible, lots of plug-ins
- provides reasoning & server capabilities

UMBC  
an Honors University in Maryland

159

## RDFS supports simple inferences

- An RDF ontology plus some RDF statements may imply additional RDF statements.
- This is not true of XML.
- Example:
  - ontology:
    - domain(parent, person)
    - range(parent, person)
    - subproperty(mother, parent)
    - range(mother, woman)
    - instance:
      - mother(eve, cain)
- This is **part of the data model** and not of the accessing/processing code



**Implies:**  
subclass(woman, person)  
parent(eve, cain)  
person(eve)  
person(cain)  
woman(eve)

UMBC  
an Honors University in Maryland

160

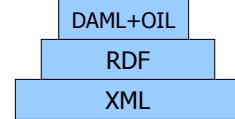
## RDF is being already in use



- **RDF has a solid specification**
  - See the RDF model theory spec - <http://www.w3.org/TR/rdf-mt/>
- **RDF is being used in a number of W3C specifications**
  - CC/PP (Composite Capabilities/Preference Profiles)  
<http://www.w3.org/Mobile/CCPP/>
  - P3P (Platform for Privacy Preferences Project) <http://www.w3.org/P3P/>
- **And in other web standards**
  - RSS 1.0 (RDF Site Summary), RDF calendar (~ iCalendar in RDF), Dublin Core,  
...
- **And in other systems**
  - Netscape's Mozilla web browser
  - Open directory (<http://dmoz.org/>)
  - Adobe products via XMP (eXtensible Metadata Platform)

## RDF isn't enough, but is a good foundation

- RDF lacks expressive adequacy for many tasks
  - No properties of properties (transitive, inverse etc.)
  - No equivalence, disjointness, coverings, etc.
  - No necessary and sufficient conditions
  - No rules, axioms, logical constraints
- DAML+OIL extends RDF
  - Layering makes partial knowledge available to apps which only understand RDF
  - Layering has disadvantages



## We're going down a familiar road

### KR trends

- 55-65: arbitrary data structures
- 65-75: semantic networks
- 75-85: simple frame systems
- 85-95: description logics
- 95-?: logic?, rules?

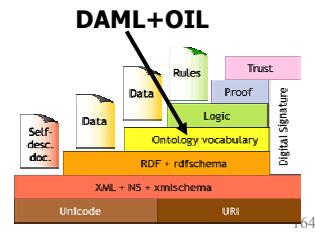
### Web trends

- 95-97: XML as arbitrary structures
- 97-98: RDF
- 98-99: RDFS (schema) as a frame-like system
- 00-01: DAML+OIL
- 02-?: OWL...???

**Only much faster!**

## DAML+OIL as a Semantic Web Language

- DAML = Darpa Agent Markup Language
  - DARPA program with 17 projects & an integrator developing language spec, tools, applications for SW.
- OIL = Ontology Inference Layer
  - An EU effort aimed at developing a layered approach to representing knowledge on the web.
- Process
  - Joint Committee: US DAML and EU Semantic Web Technologies participants
  - DAML+OIL specs released 01/01 & 03/01
  - See <http://www.daml.org/>
  - Includes model theoretic and axiomatic semantics



## A Simple DAML Example

```
<rdfs:Class about="#Animal"/>  
<rdfs:Class about="#Plant">  
    <daml:disjointFrom  
        resource="#Animal"/>  
</rdfs:Class>
```

- Note the mixture of rdf (*plant and animal are classes*) and DAML (*plant and animal are disjoint*)

## DAML+OIL $\leftrightarrow$ RDF

- DAML+OIL ontology is a set of RDF statements
  - DAML+OIL defines semantics for certain statements
  - Does **NOT** restrict what can be said
    - Ontology can include arbitrary RDF
  - But no semantics for non-DAML+OIL statements
- Adds capabilities common to description logics:
  - cardinality constraints, defined classes ( $\Rightarrow$  classification), equivalence, local restrictions, disjoint classes, etc.
- More support for ontologies
  - Ontology imports ontology
- But not (yet) variables, quantification, and general rules

## DAML in One Slide

DAML is built on top of XML and RDF

It allows the definition, sharing, composition and use of ontologies

DAML is ~ a frame based knowledge representation language

It can be used to add metadata about anything which has a URI.

URIs are a W3C standard generalizing URLs

everything has URI

```

<rdf:RDF xmlns:rdf="http://w3.org/10-rdf-syntax-ns#"
           xmlns:rdfs="http://w3.org/22-rdf-schema#"
           xmlns:daml="http://daml.org/daml+oil#">
  <daml:Ontology rdf:about="">
    <daml:imports rdf:resource="http://daml.org/daml+oil"/>
  </daml:Ontology>
  <rdfs:Class rdf:ID="Person">
    <rdfs:subClassOf rdf:resource="#Animal"/>
    <rdfs:subClassOf>
      <daml:Restriction>
        <daml:onProperty rdf:resource="#hasParent"/>
        <daml:toClass rdf:resource="#Person"/>
      </daml:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <daml:Restriction daml:cardinality="1">
        <daml:onProperty rdf:resource="#hasFather"/>
      </daml:Restriction>
    </rdfs:subClassOf>
  </rdfs:Class>
  <Person rdf:about="http://umbc.edu/~finin/">
    <rdfs:comment>Finin is a person.</rdfs:comment>
  </Person>

```

167

## DAML+OIL Extends RDF

```

<rdfs:Class about="#Animal"/>
<rdfs:Class about="#Plant">
  <daml:disjointFrom
    resource="#Animal"/>
</rdfs:Class>

```

Note mix of RDF (*plant & animal are classes*) & DAML (*plant & animal are disjoint*)

**DAML adds**

- cardinality constraints
- defined classes
- equivalence
- local restrictions
- disjoint classes
- imports
- ...

**But not yet**

- rules
- variables

168

## DAML+OIL Ontologies

- Research efforts are developing some interesting ontologies for services, trust, rules, time, geospatial relations, e.g., ...
- DAML-S is an ontology for describing properties and capabilities of services
  - <http://www.daml.org/services/>
- DAML-time covers temporal concepts and properties common to any formalization of time
  - <http://www.cs.rochester.edu/~ferguson/daml/>
- DAML-spatial covers spatial concepts and properties
  - <http://www.daml.org/listarchive/daml-spatial/date.html#29>

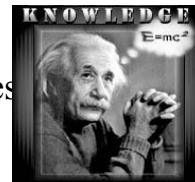
## W3C Web Ontology Working Group



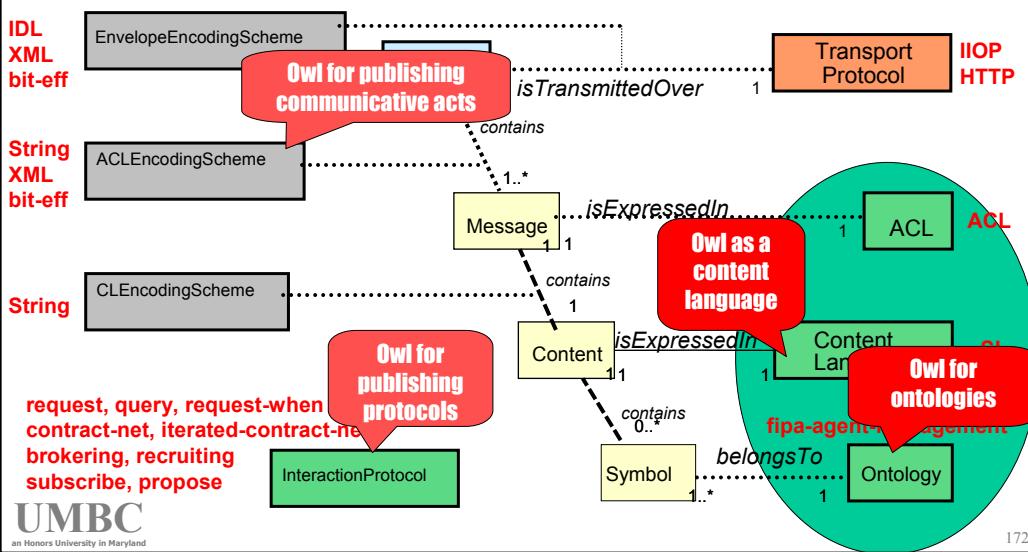
- The WOWG is developing a "Web Ontology Language": **OWL**
- WOWG has ~56 members, co-chaired by Jim Hendler (U. Maryland) & Guus Schreiber, U. Amsterdam)
- OWL =~ DAML+OIL with some renaming of properties
- Current plan is to have three compliance levels: OWL lite, OWL, OWL plus
- See <http://www.w3.org/2001/sw/WebOnt/>
- About to go to "last call" after which the language (v1) formally becomes a proposed standard.

## KR meets the Web (and MAS)

- One way to think about the semantic web is that we are creating a knowledge representation language for the Web.
- And for multi-agent systems?
- This is more than just selecting an appropriate KR language and selecting an (XML) encoding.
- The Web (and MAS) as information systems has many significant properties.
  - Highly distributed
  - Many independent content providers
  - Dynamic and evolving
  - Inconsistent
  - Potential for malicious agents



## Where OWL fits into FIPA Standards



## SW is work in progress

- There are important language aspects which need more work: rules, queries, etc.
- Many tools need to be created, e.g.,
  - Protégé plug-in for DAML+OIL
  - Annotation tools
- Applications need to be explored
- The W3C is developing a new SW language
  - OWL: Ontology Web Language
- SW ideas will migrate into other standards (e.g., basic XML, WSDL)

## Lots of Open Issues



- How expressive should the KR language be?
  - What kind of KR/reasoning system
    - F.O. logic, logic programming, fuzzy, ...
  - On Web Ontologies
    - One (e.g. CYC) or many (DAML)
    - If many, composable (IEEE IFF) or monolithic (IEEE SUMO)
    - Will general “upper ontologies” (e.g., IEEE SUO) be useful?
  - Will industry buy in?
    - Or continue to explore ad hoc XML based solutions
  - How will it be used?
    - As markup? As alternative content? Just both machines and people?
  - Is it good as a content language for agents?
- => Only experimentation will yield answers.

# 8

# Applications

## Two kinds of systems

- RDF is being used to support many practical, useful applications
  - We'll touch on RSS, CCPP, P3P, ...
- RDF, DAML+OIL and OWL are being experimented with in many research prototypes
  - We'll describe some research at UMBC

## RSS

- Rich Site Summary or RDF Site Summary
- A lightweight multipurpose extensible metadata description and syndication format for the web
  - news and other headline syndication
  - weblog syndication
  - propagation of software update lists.



```
<channel rdf:about="http://agents.umbc.edu/awchannel.rdf">
  <title>UMBC AgentWeb</title>
  <link>http://agents.umbc.edu/</link>
  <description>UMBC AgentWeb...</description>
  <language>en-us</language>
  <copyright>copyright...</copyright>
  <managingEditor>Tim...</managingEditor>
  <pubDate>Mon ...</pubDate>
  <lastBuildDate>Mon ...</lastBuildDate>
  <webMaster>finin@umbc.edu</webmaster>
  <rating>(PICS-1.1 ...</rating>
  <image rdf:resource="http:..." />
  <items><rdf:Seq>...</rdf:Seq></items>
</channel>
<image>
  <title>UMBC AgentWeb</title>
  <url>http://agents...</url>
  <link>http://agents.umbc.edu/</link>
  <width>46</width> <height>66</height>
</image>
<item rdf:about="http:...">
  <title>UDDIe</title>
  <link>http://...</link>
  <description>the ...</item>
</rdf:RDF>
```

177

## CCPP

- Composite Capabilities/Preference Profiles
- RDF-based W3C recommended standard for customizing web content for devices and users
- It is a Client profile data format
  - For describing device capabilities and user preferences
  - Enables adaptation of content presented to that device
- It is not a standard explaining how the profile is transferred, or what attributes must be generated or recognized



178

## P3P

- Platform for Privacy Preferences Project
- P3P is developed by the World Wide Web Consortium.
- It allows web-sites to communicate about their privacy policies in a standard computer-readable format.
- It enables the development of tools that are built into browsers or separate applications that summarize privacy policies, compare privacy policies with user preferences, alert and advise users.
- It does not require web-sites to change their server software.
- P3P support is in IE, Netscape

## Some UMBC research systems

- (1) Semantic web and agents (ITTalks)
- (2) Ontology mapping
- (3) Cooperating personal agents
- (4) Learning markup
- (5) Information retrieval on the SW
- (6) Modeling trust policies
- (7) Semantic web & pervasive computing
- (8) Trading agent Game in Agentcities

Joint work with Anupam Joshi, Yun Peng, Scott Cost, Tim Oates, Jim Mayfield, Benjamin Grosof, Yelena Yesha, and many students.

# (1) ITALKS

- ITALKS is a database driven web site of IT related talks at UMBC and other institutions. The database contains information on
  - Seminar events
  - People (speakers, hosts, users, ...)
  - Places (rooms, institutions, ...)
- Web pages with DAML markup are generated
- The DAML markup supports agent-based services relating to these talks.
- Users get talk announcements based on the interests, locations and schedules.

**UMBC**  
an Honors University in Maryland

<http://ittalks.org/>

181

The screenshot shows a Microsoft Internet Explorer window displaying a talk announcement from the ITALKS v2.0 website. The title of the talk is "Electric Elves: Towards an Agent Facilitated Human Organization" by Milind Tambe from the University of Southern California Information Science Institute. The abstract discusses the revolution in software agents and their role in organizations. A red starburst graphic with the word "human" is overlaid on the right side of the page. The left sidebar includes a "Topics Directory" with categories like A. General Literature, B. Hardware, etc., and links for "send this link to a friend", "Become a new member for free to be notified about future talks", "printer friendly", and "view as DAML". At the bottom, there are advertisements for "HOT DAML" and "CooJa SYNC TO PALM".

**IT TALKS v2.0 - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address: http://umbc.ittalks.org/isp/Controller.jsp?action=ViewTalk&as=HTML&talkid=20010209120547

Topics Directory

A. General Literature  
 B. Hardware  
 C. Computer Systems Organization  
 D. Software  
 E. Data  
 F. Theory of Computation  
 G. Mathematics of Computing  
 H. Information Systems  
 I. Computing Methodologies  
 J. Computer Applications  
 K. Computing Milieux

Electric Elves: Towards an Agent Facilitated Human Organization

Milind Tambe  
 University of Southern California  
 Information Science Institute

UMBC, ECS, LHS  
 2:00pm - 12:00pm, Tuesday, February 20, 2001

Abstract

Past few years have seen a revolution in the field of software agents, with agents now proliferating in human organizations, helping individuals in tasks such as information gathering, activity scheduling, managing email, etc. The "Electric-Elves" effort at USC/ISI is now taking the next step: dynamic teaming of all such different heterogeneous agents, as well as proxy agents for humans, to serve not just individuals, but to facilitate the functioning of entire organizations. The ultimate goal of our work is to build agent teams that assist in all organization activities, enabling organizations to act coherently, to robustly attain their mission goals and to react swiftly to crises. The results of this work could potentially be relevant to all organizations, including the military, corporations, and universities and research institutions. As a step towards this goal, we have had an agent team of about 15-20 agents, including 10 proxies (for 10 people) running 24/7 for the past four months at USC/ISI. The proxies communicate with us using different types of mobile wireless devices, and attempt to track our locations using wireless GPS transmissions. These agents assist us in several tasks: they track people's locations, reschedule meetings, decide presenters for research meetings (by auctioning research talk slots), and even order our lunch and dinner. In this talk, I will outline some of the lessons we have learned over the past several months in running this agent system. I will also outline our approach on some of the key research challenges, including agents' adjustable autonomy.

Biosketch

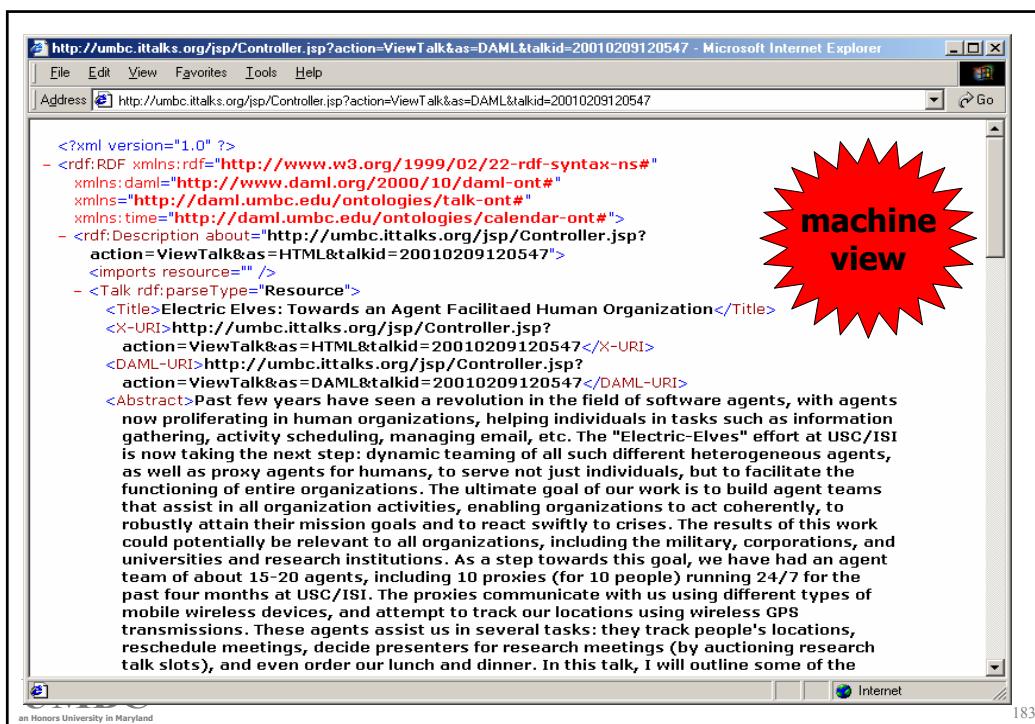
Dr. Milind tambe is a project leader at the University of Southern California Information Sciences

human view

CoolSync  
Click above to synchronise talk information to your palm.

Done Local intranet

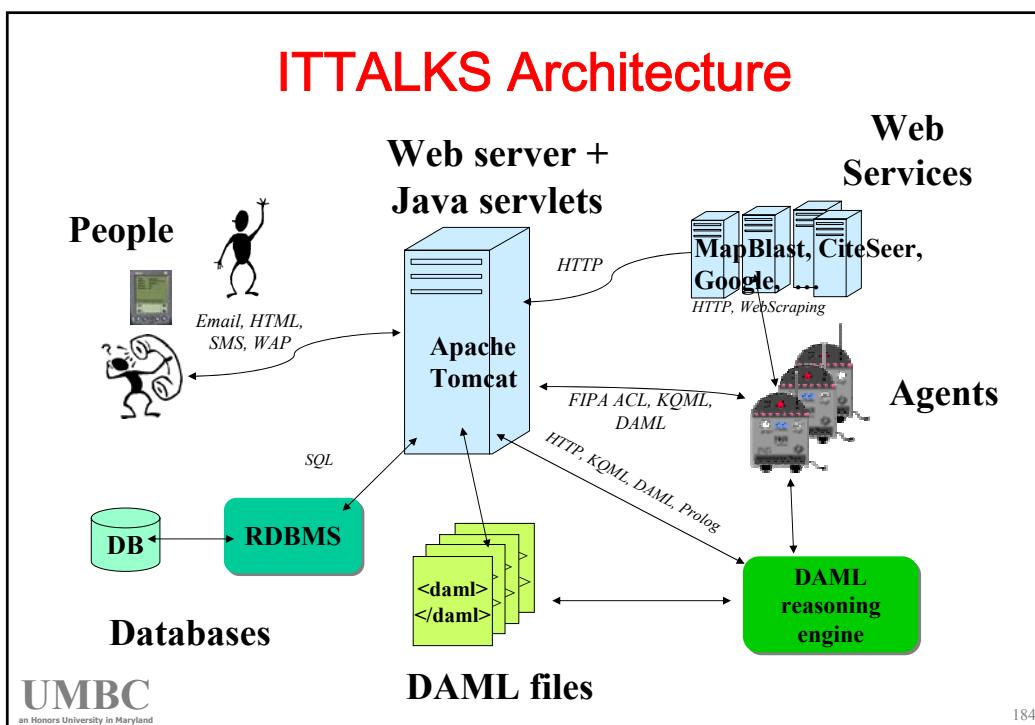
182

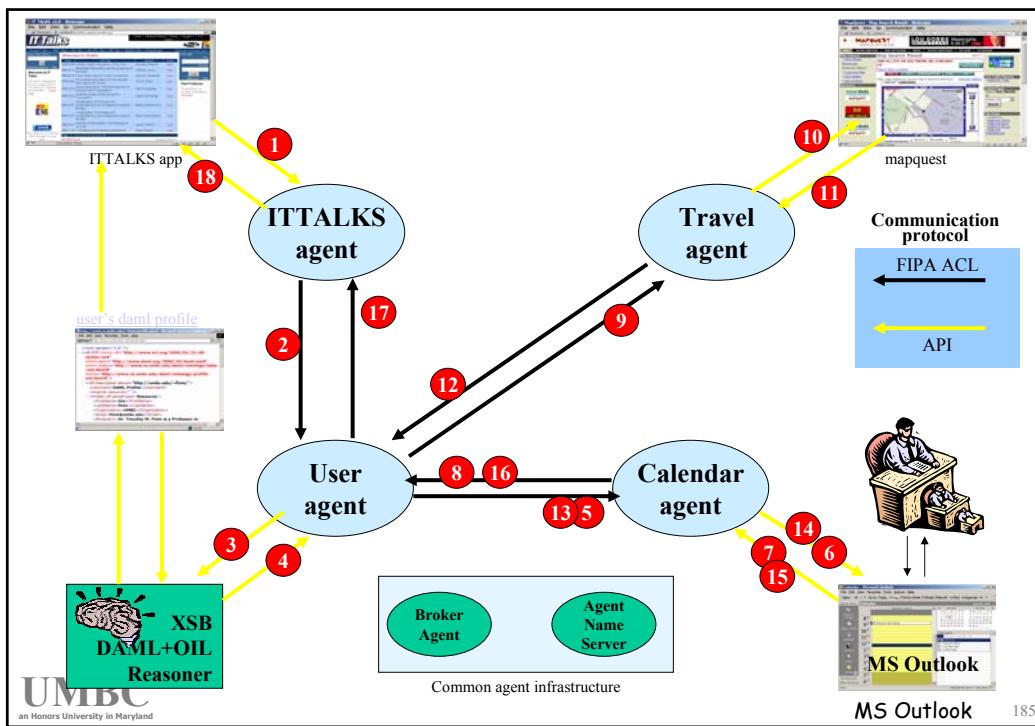


```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:daml="http://www.daml.org/2000/10/daml-ont#"
    xmlns="http://daml.umbc.edu/ontologies/talk-ont#"
    xmlns:time="http://daml.umbc.edu/ontologies/calendar-ont#"
    -><rdf:Description about="http://umbc.ittalks.org/jsp/Controller.jsp?action=ViewTalk&as=DAML&talkid=20010209120547">
        <action>ViewTalk</action>
        <resource>HTML</resource>
        <Talk rdf:parseType="Resource">
            <Title>Electric Elves: Towards an Agent Facilitated Human Organization</Title>
            <X-URI>http://umbc.ittalks.org/jsp/Controller.jsp?
                action=ViewTalk&as=HTML&talkid=20010209120547</X-URI>
            <DAML-URI>http://umbc.ittalks.org/jsp/Controller.jsp?
                action=ViewTalk&as=DAML&talkid=20010209120547</DAML-URI>
            <Abstract>Past few years have seen a revolution in the field of software agents, with agents now proliferating in human organizations, helping individuals in tasks such as information gathering, activity scheduling, managing email, etc. The "Electric-Elves" effort at USC/ISI is now taking the next step: dynamic teaming of all such different heterogeneous agents, as well as proxy agents for humans, to serve not just individuals, but to facilitate the functioning of entire organizations. The ultimate goal of our work is to build agent teams that assist in all organization activities, enabling organizations to act coherently, to robustly attain their mission goals and to react swiftly to crises. The results of this work could potentially be relevant to all organizations, including the military, corporations, and universities and research institutions. As a step towards this goal, we have had an agent team of about 15-20 agents, including 10 proxies (for 10 people) running 24/7 for the past four months at USC/ISI. The proxies communicate with us using different types of mobile wireless devices, and attempt to track our locations using wireless GPS transmissions. These agents assist us in several tasks: they track people's locations, reschedule meetings, decide presenters for research meetings (by auctioning research talk slots), and even order our lunch and dinner. In this talk, I will outline some of the
        </Abstract>
    </Talk>
</rdf:Description>

```

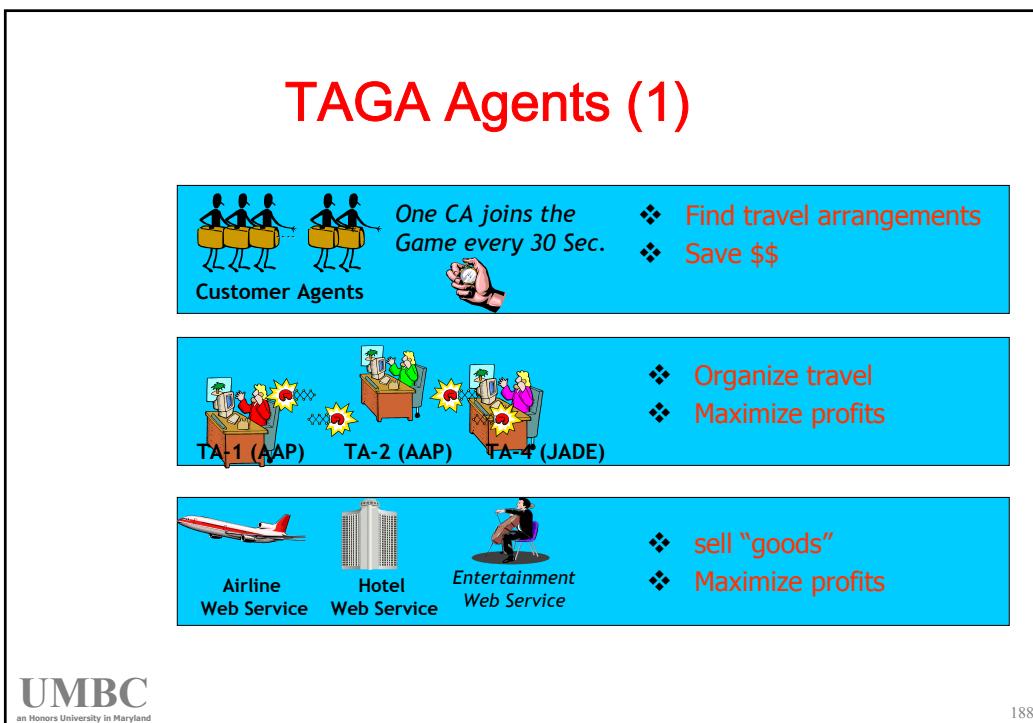
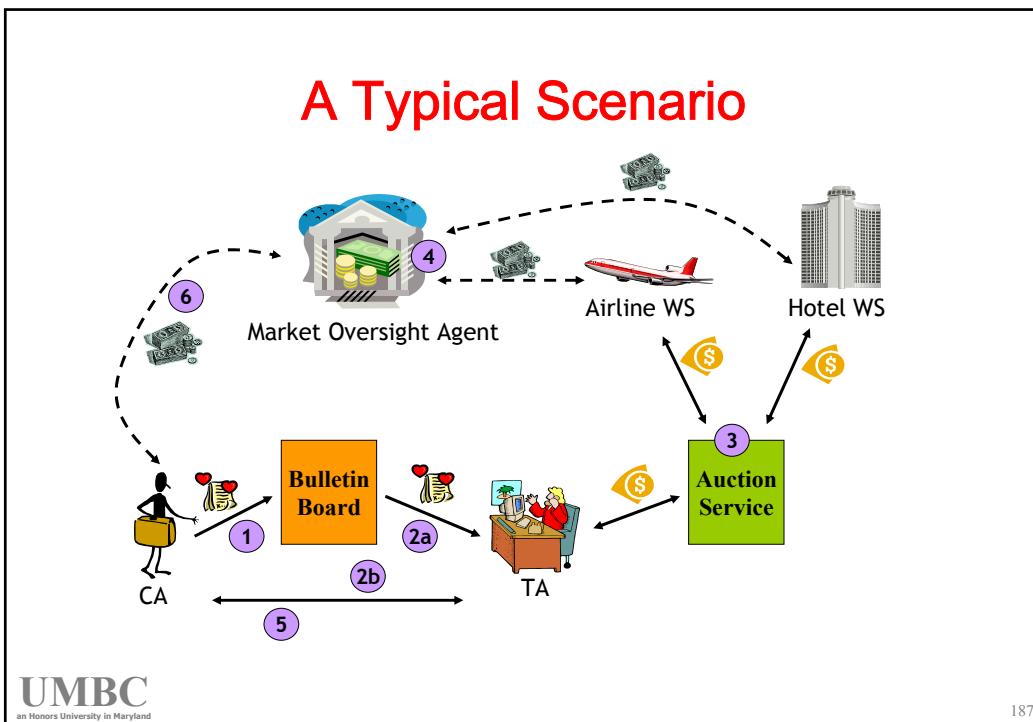




185

## (8) Trading Agents

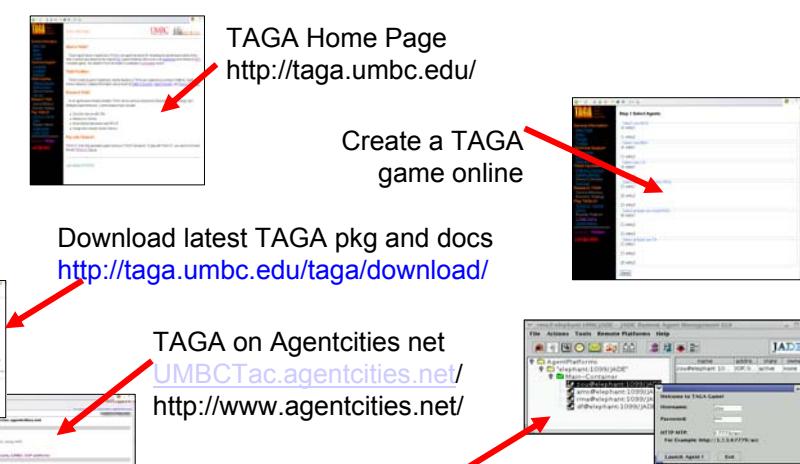
- We've built an agent-based environment inspired by TAC, the Trading agent Competition
  - TAC is a forum promoting research on the trading agent problem with games run in 00, 01, 02, 03
  - TAC agents operate within a travel shopping scenario, buying and selling goods to best serve their clients and are scored based on client's preferences for trips assembled, and minimizing cost.
  - TAC is organized around a central auction server
- Our goal is to open up the system, allowing peer-to-peer communication among agents as well various kinds of mediator, auction, discovery, service provider agents.



## TAGA Agents (2)

 <b>Bulletin Board Agent</b>	❖ Helps CA find and engage one or more TA
 <b>Auction Service Agent</b>	❖ Operates the auctions markets: English, Dutch, Priceline and Hotwire.
 <b>Market Oversight Agent</b>	❖ Manage the financial records ❖ Announces the winning TA

## TAGA in Action



TAGA Home Page  
<http://taga.umbc.edu/>

Create a TAGA game online

Download latest TAGA pkg and docs  
<http://taga.umbc.edu/taga/download/>

TAGA on Agentcities.net  
[UMBCTac.agentcities.net/](http://UMBCTac.agentcities.net/)  
<http://www.agentcities.net/>

TAGA supports heterogeneous agent platform. A FIPA-JADE agent can interact with a FIPA-AAP agent



190

## The TAGA Game Server

<http://taga.umbc.edu/taga/play/demo.htm>

The screenshot displays several windows from the TAGA Game Server:

- View ACL message traffic:** A window showing a grid of messages with columns for Date, Message, From, To, and Content.
- View TAGA game status:** A window showing game statistics and player information.
- Monitor Open Market Auction:** A window showing a list of open auctions.
- Monitor Customer Agents:** A window showing a list of customer agents.
- (and more ... new agent, user login, create game, game history)**: A text block indicating additional features.

UMBC  
an Honors University in Maryland

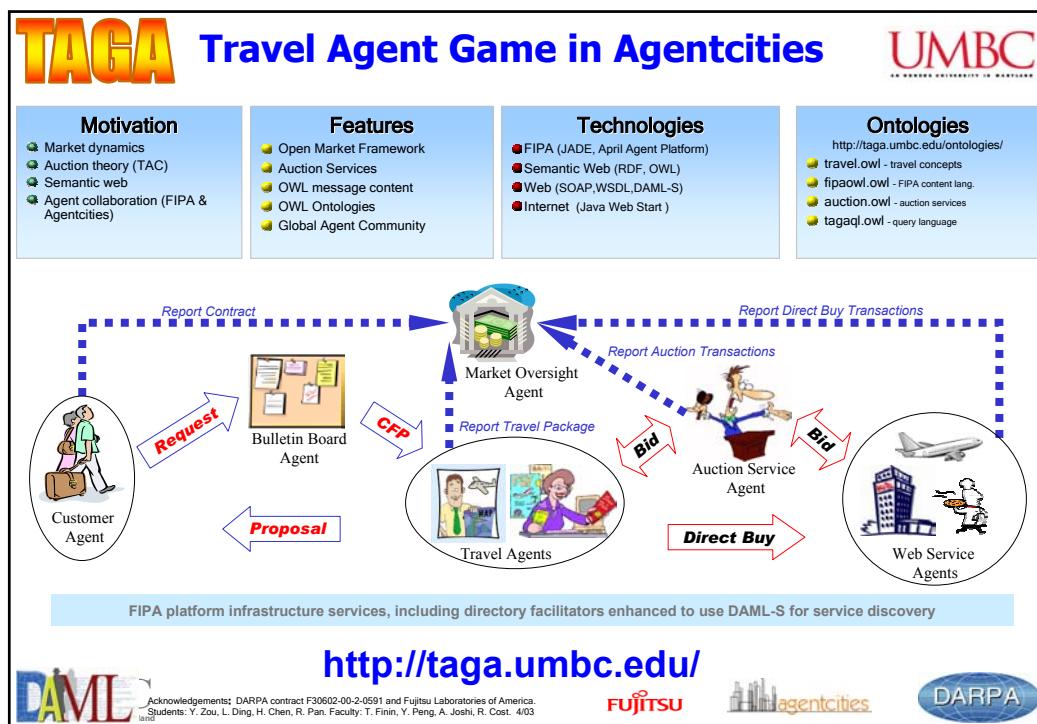
191

## TAGA goal and features

- **Goal:** an open test bed for research on agents, FIPA, and SW in an ecommerce environment
- **Features:**
  - Part of the Agentcities network
  - Everything is a FIPA-compliant agent
  - Supported by OWL ontologies
  - Agents use RDF and OWL as for their content language
  - DAML-S used for service description and discovery
  - New FIPA compliant protocols for various kinds of auctions

UMBC  
an Honors University in Maryland

192

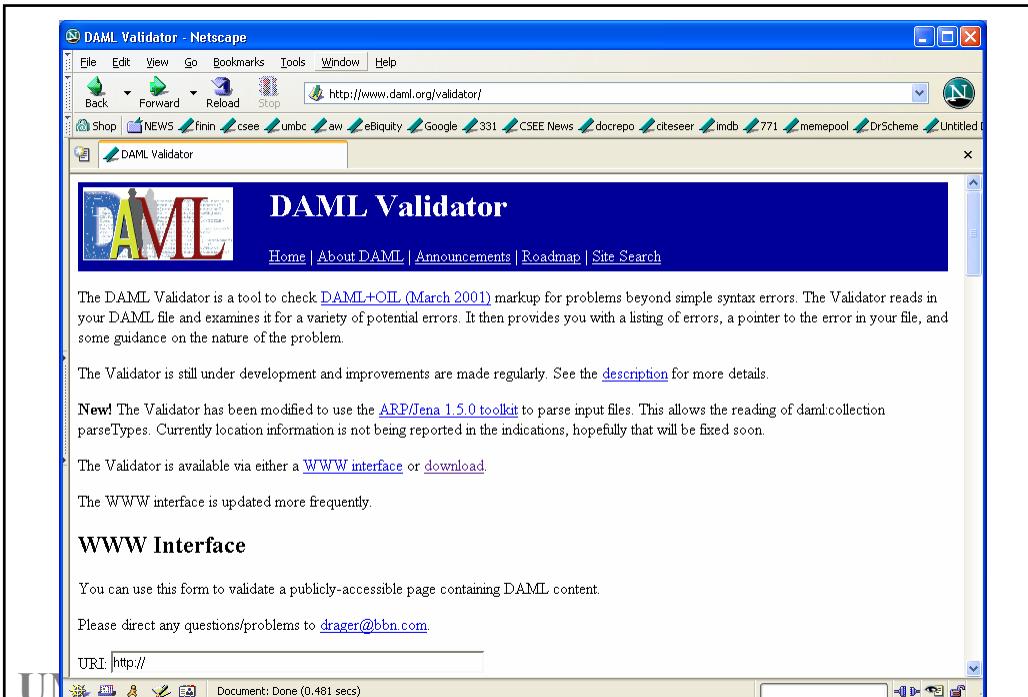


## Software Tools

- There are a number of “research grade” tools for the semantic web
  - Many are open source
- Some commercial tools and systems are beginning to appear
- Good places to watch:
  - <http://w3c.org/>
  - <http://www.daml.org/>
  - <http://semanticweb.org/>

## Kinds of Tools

- **Parsers** that read RDF, DAML, or OWL and produce appropriate data structures.
- **Validators** that read an RDF/DAML/OWL file and identify potential problems
- **Reasoners** that draw inferences licensed by SW languages
- **SW Editors** which allow one to browse and edit SW ontologies while enforcing constraints and/or identifying contradictions.
- **Markup tools and editors** which help authors add SW annotations to other documents.
- **Query languages** designed for SW languages
- **Database systems** for storing SW content and answering queries about it



The screenshot shows the DAML Validator page in a Netscape browser window. The title bar reads "DAML Validator - Netscape". The main content area displays the DAML Validator logo and navigation links: Home, About DAML, Announcements, Roadmap, Site Search. Below this, a text block explains the tool's purpose: "The DAML Validator is a tool to check DAML+OIL (March 2001) markup for problems beyond simple syntax errors. The Validator reads in your DAML file and examines it for a variety of potential errors. It then provides you with a listing of errors, a pointer to the error in your file, and some guidance on the nature of the problem." There are several bullet points below this text:

- The Validator is still under development and improvements are made regularly. See the [description](#) for more details.
- New! The Validator has been modified to use the [ARP/Jena 1.5.0 toolkit](#) to parse input files. This allows the reading of daml:collection parseTypes. Currently location information is not being reported in the indications, hopefully that will be fixed soon.
- The Validator is available via either a [WWW interface](#) or [download](#).
- The WWW interface is updated more frequently.

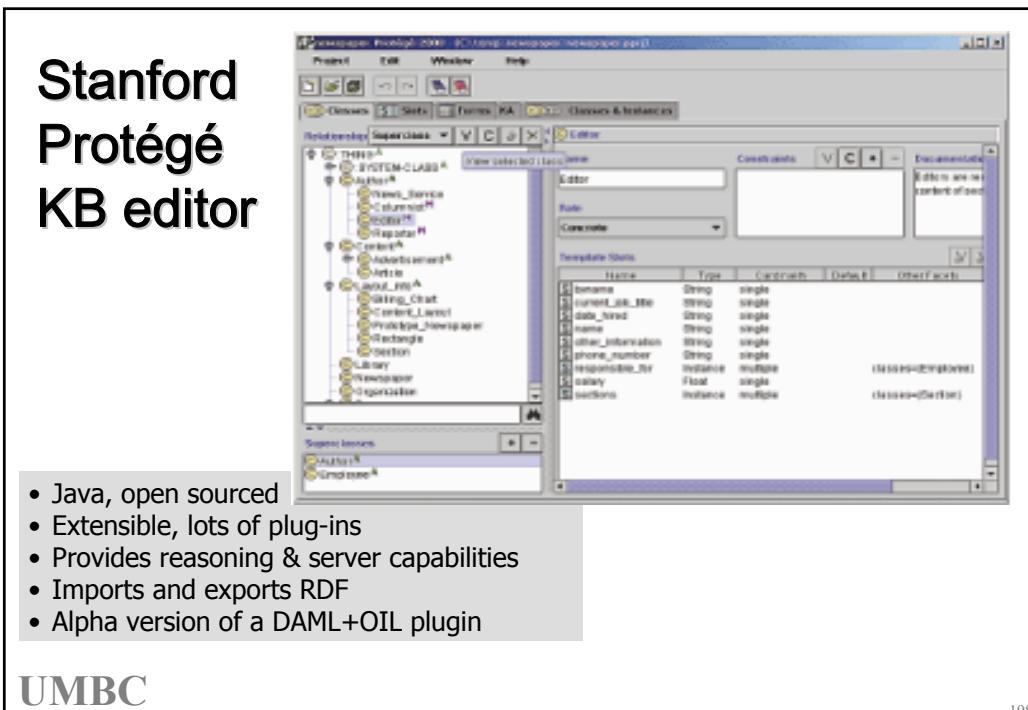
**WWW Interface**

You can use this form to validate a publicly-accessible page containing DAML content.

Please direct any questions/problems to [drager@bbn.com](mailto:drager@bbn.com).

URI:

At the bottom left is the UMBC logo: "UMBC an Honors University in Maryland". At the bottom right is the number "197".



The screenshot shows the Stanford Protégé KB editor interface. The title bar reads "Protégé - Protégé 2000 (C:\Users\yannis\Documents\Protégé 2000\2003)". The main window displays a class hierarchy tree on the left and a detailed view of a selected class on the right. The selected class is "THINN". The right panel shows the "Editor" tab with fields for "Name" (set to "Domain") and "Constraints". Below this is a table titled "Template Slots" with columns: Name, Type, Constraints, Default, and Other facets. The table contains the following rows:

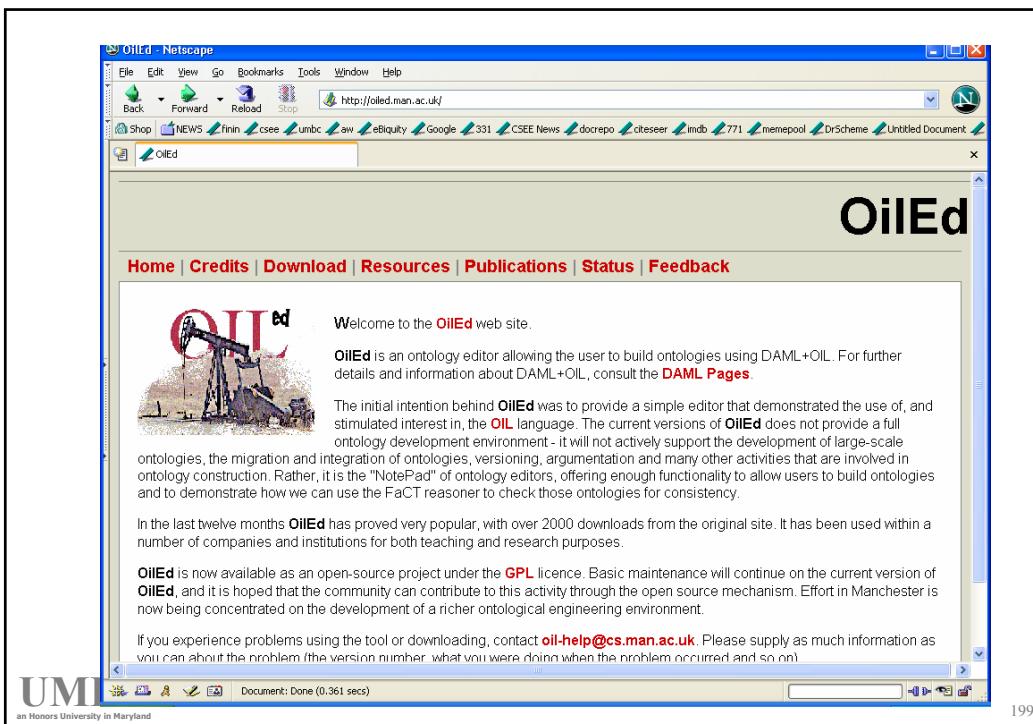
Name	Type	Constraints	Default	Other facets
domain	String	single		
current_ell_300	String	single		
date_3rd	String	single		
name	String	single		
other_information	String	single		
phone_number	String	single		
responsible_for	instance	multiple	(1455940291980000)	
salary	Float	single		
sections	instance	multiple	(1455940291980000)	

Below the editor panel is a "Superior Instances" list containing "ALUMNI" and "Compliance".

**Stanford Protégé KB editor**

- Java, open sourced
- Extensible, lots of plug-ins
- Provides reasoning & server capabilities
- Imports and exports RDF
- Alpha version of a DAML+OIL plugin

At the bottom left is the UMBC logo: "UMBC an Honors University in Maryland". At the bottom right is the number "198".



## JTP – Java Theorem Prover

- <http://www.ksl.stanford.edu/software/JTP/>
- JTP has a hybrid reasoning architecture
  - First-order logic model elimination theorem prover
  - Suite of special purpose reasoners
  - Dispatchers and APIs for reasoners
- Works with knowledge in KIF, RDF, DAML+OIL
- Demo: Inference Web Browser

<http://belo.stanford.edu:8080/iwbrowser/index.jsp>

The screenshot shows the Inference Web Browser interface. The title bar says "Inference Web Browser". The main content area displays a proof in the Java Theorem Prover. The proof steps are:
 

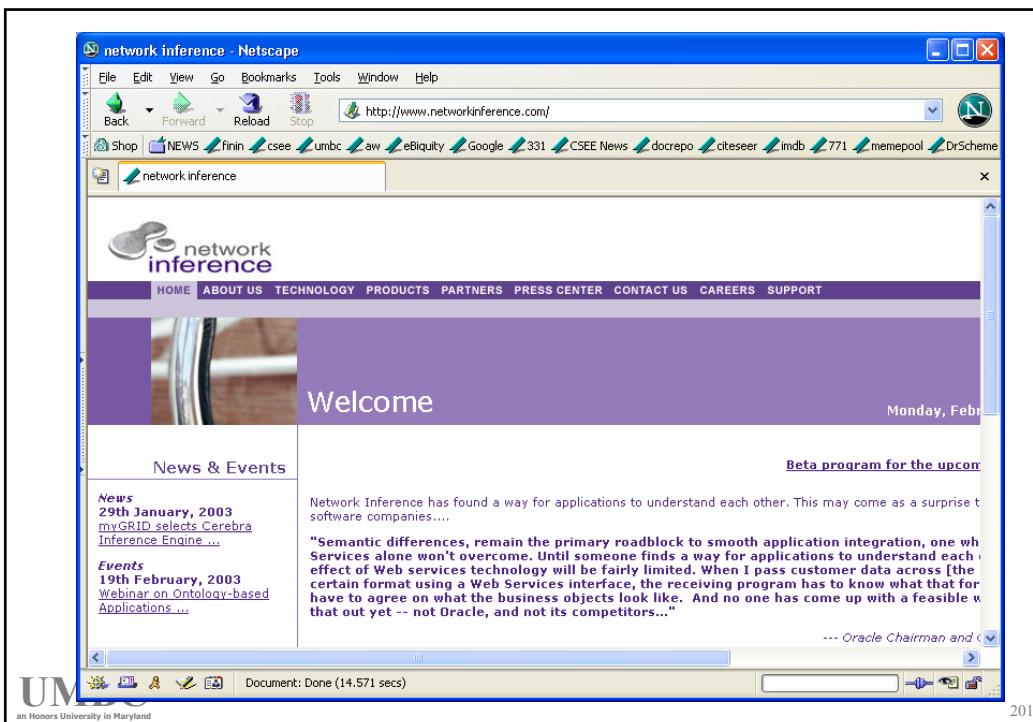
- ( $\exists$  holds ?prop ?obj white) **(and** ( $\exists$  holds type ?obj white-color-restriction) **(holds onproperty** white-color-restriction ?prop))
- ( $\forall$  w1 white-color-restriction)
- ( $\exists$  onproperty white-color-restriction color)+

 To infer that:
 

- (color w1 white)

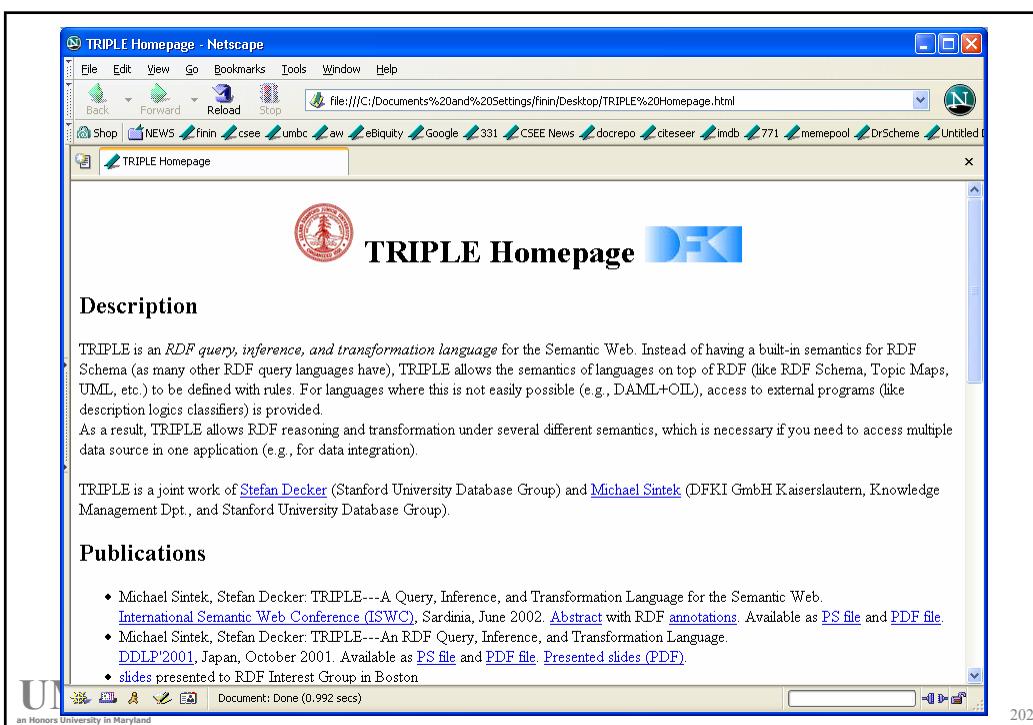
 You can select the premises to learn more about them.

At the bottom left is the UMBC logo and at the bottom right is the number 200.



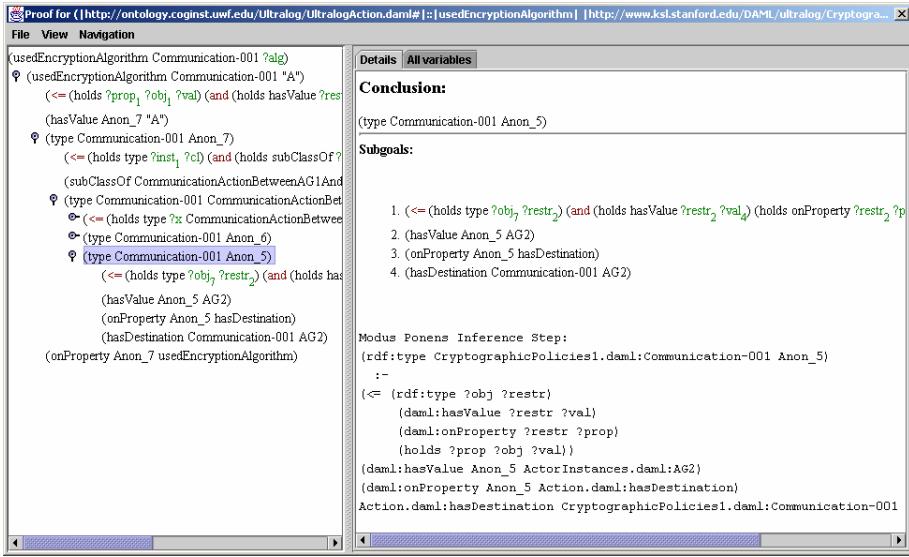
The screenshot shows a Netscape browser window with the URL <http://www.networkinference.com/>. The page title is "network inference". The main content area displays a "Welcome" message and a "Beta program for the upcom..." link. On the left, there is a sidebar titled "News & Events" with news items like "29th January, 2003 myGRID selects Cerebra Inference Engine ..." and "19th February, 2003 Webinar on Ontology-based Applications ...". The bottom status bar shows "Document: Done (14.571 secs)". The browser interface includes a toolbar with Back, Forward, Reload, Stop, and Home buttons.

201



The screenshot shows a Netscape browser window with the URL <file:///C:/Documents%20and%20Settings/finin/Desktop/TRIPLE%20Homepage.html>. The page title is "TRIPLE Homepage". It features a logo of a tree and the text "TRIPLE Homepage DFKI". Below the title, there is a section titled "Description" which provides a detailed explanation of what TRIPLE is. It mentions that TRIPLE is an RDF query, inference, and transformation language for the Semantic Web, allowing reasoning and transformation under multiple semantics. The page also credits Stefan Decker and Michael Sintek for their joint work on TRIPLE. The bottom status bar shows "Document: Done (0.992 secs)". The browser interface includes a toolbar with Back, Forward, Reload, Stop, and Home buttons.

202



The screenshot shows a Prolog proof window with the following details:

- File View Navigation**
- Conclusion:** (type Communication-001 Anon\_5)
- Subgoals:**
  1. ( $\Leftarrow$  (holds type ?obj<sub>1</sub> ?restr<sub>1</sub>) (and (holds hasValue ?val<sub>1</sub>) (onProperty ?restr<sub>1</sub> ?p<sub>1</sub>)))
  2. (hasValue Anon\_5 AG2)
  3. (onProperty Anon\_5 hasDestination)
  4. (hasDestination Communication-001 AG2)
- Modus Ponens Inference Step:**

```
(rdf:type CryptographicPolicies1.daml:Communication-001 Anon_5)
:-
(=< (rdf:type ?obj ?restr)
(daml:hasValue ?restr ?val)
(daml:onProperty ?restr ?prop)
(holds ?prop ?obj ?val))
(daml:hasValue Anon_5 ActorInstances.daml:AG2)
(daml:onProperty Anon_5 Action.daml:hasDestination)
Action.daml:hasDestination CryptographicPolicies1.daml:Communication-001
```

**UMBC**  
 an Honors University in Maryland

203

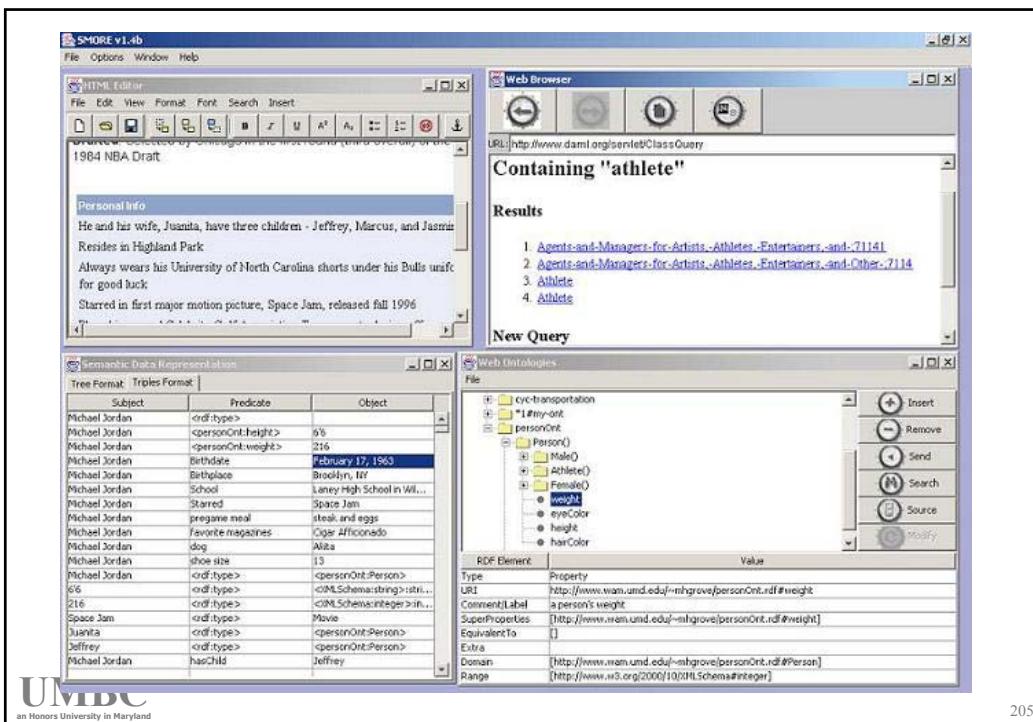


The screenshot shows a Netscape browser window displaying the following content:

- Maryland Information and Network Dynamics Lab**  
 Semantic Web Agents Project  
 University of Maryland, College Park  
 College Park, Maryland, USA  
<http://www.mindswap.org>
- MIND SWAP** logo featuring two stylized human heads facing each other.
- Navigation menu:**
  - Home
  - People
  - Papers
  - Downloads
  - Directions
  - WIKIs
- Recent news items:**
  - Fri Jan 24: A paper entitled [Science and the Semantic Web](#) by Professor Hendler was published in Science on Jan 24.
  - Mon Jan 13: [Excel2RDF](#) Version 1.21. This updated version can convert larger spreadsheets (26 columns x 100 rows).
  - Mon Dec 16: One [paper](#) about Web Service composition has been added to the papers section.
  - Sat Nov 16: Two new papers, on SMORE and Trust Networks, added to the papers section.
  - Tues Nov 12: [Ontology Search](#) has been improved and updated.
  - Fri Nov 8: [RIC version 3.0a](#) is available for download!
  - [ConvertToRDF version 1.2](#) is now available for download!

**UM**  
 an Honors University in Maryland

204



205

## Research Issues

- How will the semantic web be used?
- What kind of KR is needed for the SW?
- How expressive should SW languages be?
- One ontology or many?
- How does industry view the SW?
- How do the web and agents paradigms fit?
- What are good query languages for the SW?

# Conclusions

## 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**.

## Agent Communication

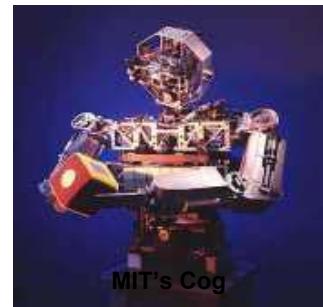
- Agent-agent communication is a key to realizing the potential of the agent paradigm.
- Since interoperability is a defining characteristic of agents, standards are important!
- Candidates for standardization include
  - Agent architecture
  - Agent communication language
  - Agent interaction protocols
  - Agent knowledge
  - Agent programming languages
- Standards will most develop through natural selection, “*nature red in tooth and claw*”

## Speculations

- SW might be a chance for us to get intelligent agents out of the lab
  - Solving the symbol grounding problem
  - Rethinking agent communication
- How do we get there?

## The symbol grounding problem

- An argument against human-like AI is that it's impossible unless machines share our perception of the world.
- A solution to this "symbol grounding problem" is to give agents (soft or hard) human inspired senses.
- But the world we experience is determined by our senses, and human and machine bodies may lead to different conceptions of the world (*cf.* Nagel's *What Is It Like To Be a Bat?*)
- Maybe the Semantic Web is a way out of this problem?



## Solving the symbol grounding problem

- The web may become a common world that both humans and agents can understand.
- Confession: the web is more familiar and real to me than much of the real world.
- Physical objects can be tagged with low cost (e.g., \$0.05) transponders or RFIDs encoding their URIs
  - See HP's Cooltown project  
<http://cooltown.com/>



## Rethinking the agent communication

- Much multi-agent systems work is grounded in Agent Communication Languages (e.g., KQML, FIPA) and associated software infrastructure.
  - This paradigm was articulated ~1990, about the same time as the WWW was developed.
  - Our MAS approach has not yet left the laboratory yet the Web has changed the world.
- Maybe we should try something different?
  - The communication MAS paradigm has been peer-to-peer message oriented communication mediated by brokers and facilitators -- an approach inherited from client-server systems.

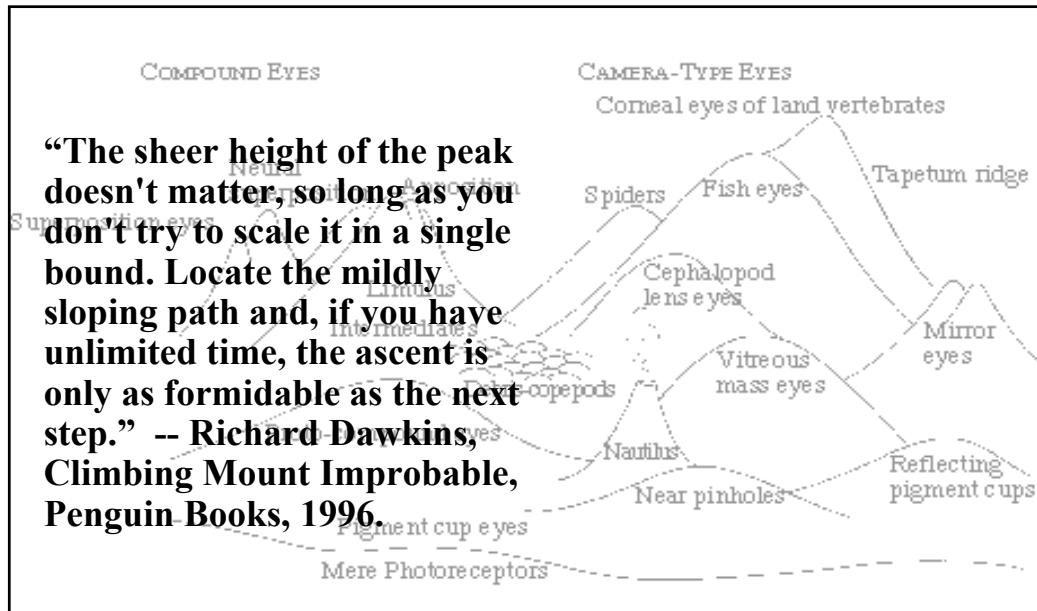
## Rethinking the agent communication

A possible new paradigm?

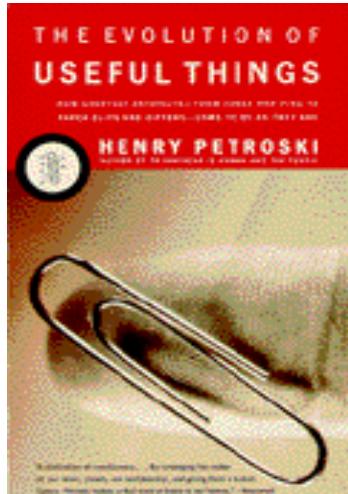
- Agents “publish” beliefs, requests, and other “speech acts” on web pages.
- Brokers “search” for and “index” published content
- Agents “discover” what peers have published on the web and browse for more details
- Agents “speak for” content on web pages by
  - Answering queries about them
  - Accepting comments and assertions about them

## How do we get there from here?

- This semantic web emphasizes ontologies – their development, use, mediation, evolution, etc.
- It will take some time to really deliver on the agent paradigm, either on the Internet or in a pervasive computing environment.
- The development of complex systems is basically an evolutionary process.
- Random search carried out by tens of thousands of researchers, developers and graduate students.



## The Evolution of Useful Things



- *The Evolution of Useful Things*, Henry Petroski, 1994.
- Prior to the 1890's, papers were held together with straight pens.
- The development of "spring steel" allowed the invention of the paper clip in 1899.
- It took about **25 years (!)** for the evolution of the modern "gem paperclip", considered to be optimal for general use.

UMBC  
an Honors University in Maryland

217

## So, we should ...

- **Start with the simple and move toward the complex**
  - E.g., from vocabularies to FOL theories
- **Allow many ontologies to bloom**
  - Let natural evolutionary processes select the most useful as common consensus ontologies.
- **Support diversity in ontologies**
  - Monocultures are unstable
  - There should be no THE ONTOLOGY FOR X.
- **The evolution of powerful, machine readable ontologies will take many years, maybe generations**
  - Incremental benefits will more than pay for effort

UMBC  
an Honors University in Maryland

218

## For More Information

- **Agents:** <http://agents.umbc.edu/>
- **FIPA:** <http://www.fipa.org/>
- **KSE:**
  - <http://agents.umbc.edu/kqml/>
  - <http://agents.cs.umbc.edu/kif/>
  - <http://agents.umbc.edu/ontology/>
- **RDF:** <http://www.w3.org/RDF/>
- **DAML+OIL:** <http://www.daml.org/>
- **OWL:** <http://www.w3.org/2001/sw/>
- **Semantic web:** <http://semanticweb.org/>
- **These slides:** <http://agents.umbc.edu/aamas03/>

## Presenters

## Tim Finin

<http://umbc.edu/~finin>  
[finin@cs.umbc.edu](mailto:finin@cs.umbc.edu)



Tim Finin is a Professor in the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County (UMBC). He has over 30 years of experience in the applications of Artificial Intelligence to problems in information systems, intelligent interfaces and robotics. He holds degrees from MIT and the University of Illinois. Prior to joining the UMBC, he held positions at Unisys, the University of Pennsylvania, and the MIT AI Laboratory. Finin is the author of over 180 refereed publications and has received research grants and contracts from a variety of sources. He has been the past program chair or general chair of several major conferences, is a former AAAI councilor and is AAAI's representative on the board of directors of the Computing Research Association.

**UMBC**  
an Honors University in Maryland

221

## Yannis Labrou

<http://www.cs.umbc.edu/~jklabrou>  
[yannis@fla.fujitsu.com](mailto:yannis@fla.fujitsu.com)



Yannis Labrou is a member of Research Staff with Fujitsu Labs of America. Prior to joining Fujitsu he served as the Director of Technology at PowerMarket, Inc. and as a Visiting Assistant Professor at the UMBC Computer Science and Electrical Engineering Department and at the Institute for Global Electronic Commerce. He holds a PhD in Computer Science from UMBC (1996) and a Diploma in Physics from the University of Athens, Greece. Dr. Labrou's research focuses on software agents, Electronic Commerce and the semantic web. Dr. Labrou is a founding member of the FIPA Academy and has been an active participant in the development of the FIPA specifications for software agents standards. He has served on a number of conference organizing committees, program committees, and panels, and has delivered invited tutorials and talks to conferences, research labs and universities. He is the author of more than 30 publications in research journals, books, and conferences. Prior to joining UMBC, Dr. Labrou worked as an intern at the Intelligent Network Technology group of the IBM T.J. Watson Research Center.

**UMBC**  
an Honors University in Maryland

222