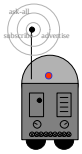
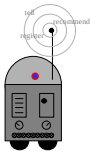
UMBC Research on Software Agents and Multi-Agent Systems

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





http://www.csee.umbc.edu/~finin/at98/
November 1998

1

Some Assumptions and Biases

2

The Agent Paradigm

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

3

Why is communication important?

- Most, but not all, would agree that communication is a requirement for cooperation.
- Societies can do things that no individual (agent) can.
- Diversity introduces heterogeneity.
- Autonomy encourages disregard for other agents' internal structure.
- Communicating agents need only care about understanding a "common language".

4

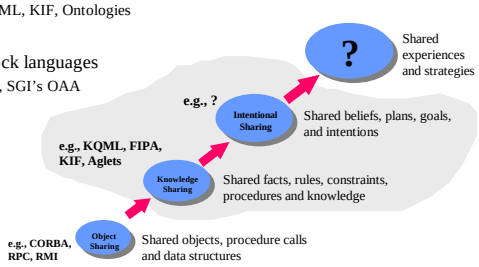
Agent Communication

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

5

Some ACLs

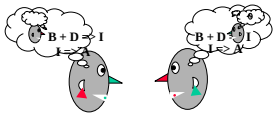
- Is CORBA an ACL?
- Knowledge sharing approach
  - KQML, KIF, Ontologies
- FIPA
- Ad hock languages
  - e.g., SGI's OAA



6

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**, which are understood in terms of an intentional-level description of an agent



- An intentional description makes references to beliefs, desires, intentions and other **mental states**.
- BDI frameworks have the power to describe an agents' behavior, including communicative behavior
- Describing behavior at this level is an important contribution of the agent-based approach.

7

Agents and agencies

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

8

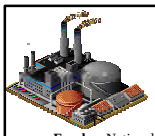
Some Agent Research at UMBC

9

UMBC agent research

- Funding from NIST, DARPA, NSA, IBM, Fujitsu
- Focus:
  - Agent communication languages
  - Scalable Information filtering and retrieval
  - Mobile agent frameworks
  - Data mining
  - Applications to several problem domains
    - enterprise integration
    - distributed information retrieval
    - network management
    - Electronic commerce

10



CIIMPLEX EECOMS

Consortium for Integrated Intelligent Manufacturing Planning and Execution

Extended Enterprise Coalition for Integrated Collaborative Manufacturing Systems

- **Funder:** National Institute of Standards and Technology / Advanced Technology Program
  - Technologies for the Integration of Manufacturing Applications (TIMA)
  - ~ $45M over six years in two ATP projects
- **Goal:** Plug and Play framework of business objectives and integration-enabling tools allowing a suite of solutions that can be implemented "out-of-the-box" at small and midsized manufacturing and process sites including MES, ERP, Finite Scheduling, and Capacity Analysis/Decision Support
- **Objectives:** interoperability, configurability, adaptability, extensibility, plug and play.

Participants

- IBM Corp
- Universities
  - University of Maryland Baltimore County
  - University of North Carolina at Charlotte
  - University of Florida
- Berclain USA Ltd.
- Boeing
- QAD Inc
- GSE Systems
- Lucent Technologies
- Ingersoll-Rand Co.
  - Demand Solutions
  - DLoG Remex Inc.
- Intercim
- EnvisionIt Software
- The Haley Corporation

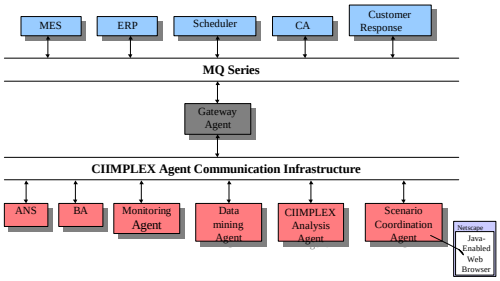
11

Manufacturing Enterprise Integration

- Integration of planning and execution is imperative for agile manufacturing
  - parts delivery is delayed by the part supplier
  - a preferred customer asks to move ahead a delivery
  - machine breaks down on shop floor
- This involves collaboration among business applications and managers
- Business applications are legacy systems
  - not intended to talk to each other (no API, no means of communication)
  - developed over long period of time (expensive to change)
  - many decision steps are not covered (white space between applications)
- Multi-agent system (MAS) approach
  - flexible and dynamic communication among applications
  - plug-and-play
  - interface agents to interact with people
  - other agents to fill the white space between business applications

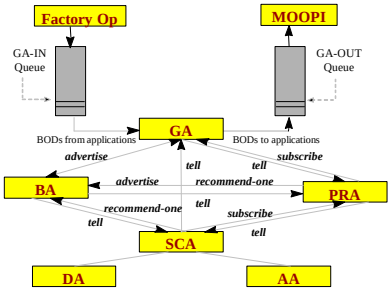
12

CIIMPLEX Architecture



13

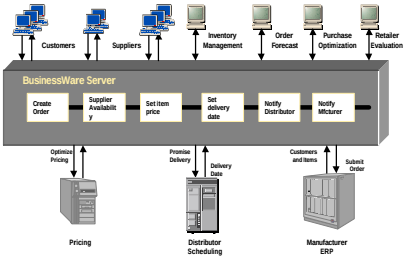
CIIMPLEX MAS Architecture



14

Supply Chain Enterprise Application Integration

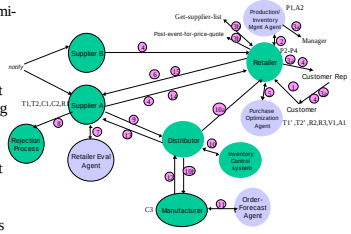
Global Visibility and Control of Business Processes across Apps



Negotiation among agents in the Supply Chain

Goals

- Support automated or semi-automated negotiation among applications in a supply-chain
- Develop an approach that can integrate with existing business practices and procedures
- Develop an approach that uses standards and technologies likely to be acceptable to the business community



16

General Problems

- There is no standard language for applications to express actions in a negotiation interaction
  - everything is vendor-specific, impeding the creation of a market
- Existing languages and standards for EDI are too weak to capture some desirable or required information
  - e.g., Business rules, constraints, contingencies, etc.
- Autonomous negotiation does not fit into existing business practices
  - How can we trust autonomous agents to negotiate on our behalf?
  - How do we integrate them into existing procedures for authorization, monitoring and auditing?
- Getting people and organizations to adopt radical new technology is very difficult

17

Specific Objectives

- Develop a high-level language for negotiation
  - primitives for calls for proposal, proposals, counter-proposals, acceptances, rejections, clarifications, etc.
  - security issues, e.g., authentication, authorization, signatures, etc.
- Developing content languages for negotiation
  - For expressing business documents, business rules, constraints and other knowledge
- Integrating humans and agents in the negotiation process
  - To provide oversight and monitoring
  - To integrate with existing business practices
- Realism
  - Base solutions on emerging standards, both conceptual and technological, e.g., FIPA ACL, XML, PKI, TCP/IP

18

Specific Approach

1 Use FIPA ACL primitives for negotiation
- Important contribution is the set of primitives and their semantics

2 Use XML, extended with KIF, as the content language
- KIF-based extensions allow the use of constraints and business rules

3 Introduce the notion of adjustable autonomy into agent-based supply chain negotiation
- Use of *"decision rules"* to decide how to respond augmented with *"authorization rules"* which decide if the action should be reviewed for authorization and by whom.

19

1 Negotiation primitives

- Based on the FIPA ACL with extensions
- Basic negotiation primitives:
  - **cfp**: call for proposals
  - **propose:** propose (or counter-propose) an action
  - **accept-proposal:** accept a proposal
  - **reject-proposal:** reject a proposal (with optional reason)
- Other ACL primitives useful in negotiation
  - inform, query, request, not_understood, refuse, ...
  - advertise, subscribe, broker, register, …
- Specific negotiation protocols are defined using these primitives
  - e.g., Iterated-contract-net, English-auction, etc.

20

Examples of negotiation primitives

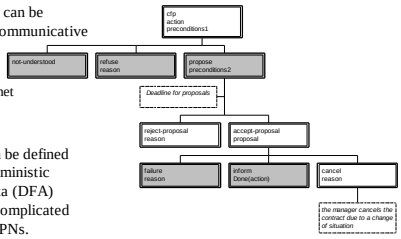
```
(cfp
   :from " http://umbc.edu/~finin/self"
   :language KIF/XML
   :ontology http://ec.ibm.com/ec42
   :content " <proposal>
     <salesContract>
     <buyer> " http://umbc.edu/~finin/self" </buyer>
     <price unit=usd> ?Price </price>
     <goods> … </goods>
     </salesContract>
     <constraints>
       <kif> <rel name=less>
               <arg ?Prince> <arg 5000>
             </rel>
       </kif>
     </constraints>
     </proposal>"
   :protocol bestAndFinal03
   :reply-with cfp3245a.11.06.99 )
```

```
(propose
   :to " http://umbc.edu/~finin/self"
   :from " http://compusa.com/self"
   :language KIF/XML
   :ontology http://ec.ibm.com/ec42
   :content " <proposal>
     <salesContract>
     <buyer> " http://umbc.edu/~finin/self" </buyer>
     <price unit=usd> 4500 </price>
     <goods> … </goods>
     </salesContract>
     </proposal>"
   :in-reply-to cfp3245a.11.06.99
   :protocol bestAndFinal03
   :reply-with offer4762579.11.06.736125409 )
```

21

Defining negotiation protocols

- Different protocols can be defined using the communicative primitives
  - contract-net
  - iterated contract-net
  - English auction
  - etc
- Most protocols can be defined with a simple deterministic finite-state automata (DFA) formalism. More complicated ones will require CPNs.
- Negotiations can be augmented by "side conversations" composed of queries, informs, etc.



22

2 An XML-based content

- We are exploring the use of an XML-based content language
- XML will be the language of the web
  - XML will rapidly become the dominant "content" encoding used on the web for ecommerce and other applications.
  - Businesses (and their agents) will continue to interact by exchanging documents (POs, invoices, catalogues, etc) but encoded in XML.
- XML supports the required extensions
  - We envision extensions to encode rules, constraints and agent-agent negotiation.

23

Laptop description seen "by eye"

Laptop Computer

IBM Thinkpad
560X
233 Mhz
32 Mb
4 Gb
4.1 pounds
$3200

HTML laptop description

```
<TITLE>Laptop
  Computer</TITLE>
<BODY>
<UL>
<LI>IBM Thinkpad 560X
<img src="560.gif">
<LI>233 Mhz
 <LI>32 Mb
 <LI>4 Gb
 <LI>4.1 pounds
 <LI>$3200
</UL>
</BODY>
```

XML Laptop Description

```
<COMPUTER TYPE="LAPTOP">
<MANUFACTURER>IBM</MANUFACTURER>
<image>http://www.vendor.com/images/560.gif</image>
<LINE>Thinkpad</LINE>
<MODEL>560X</MODEL>
<SPEED UNIT="MHZ">233</SPEED>
 <MEMORY UNIT="MB">32</MEMORY>
 <DISK UNIT="GB">4</DISK>
 <WEIGHT UNIT="POUND">4.1</WEIGHT>
 <PRICE CURRENCY="USD">3200</PRICE>
</COMPUTER>
```

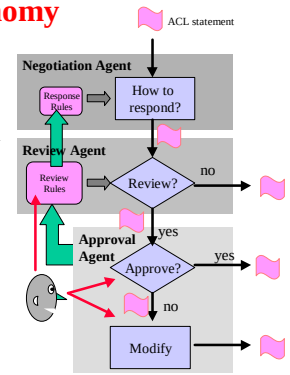
3 Controlling negotiating agents

- The potential for automated negotiation raises many concerns
  - Do we really trust our agent not to be fleeced?
  - How can we monitor what our agent has done and is doing?
  - Can our agent learn our preferences and negotiation strategies?
  - How can we accommodate existing procedures for authorization and review?

27

Adjustable Autonomy

- Response rules determine how the agent should respond in a negotiation
- Review rules determine if and by whom the proposed response should be reviewed
- Typical review rules:
  - –review if price > $500
  - –review if untrusted vendor
  - –review if critical resource
  - –review if new product
  - –review if offer only partly understood
  - –random review
  - –etc…
- Machine learning techniques can be used to automatically learn a reviewers preferences and strategies



28

Recent Results

- Developed tools for representing ACL messages in XML
  - Defined ACL DTD
  - Defined XSL style sheet for ACL
  - Implemented parser for ACL to XML
- Developed tools for representing KIF in XML
  - Defined KIF DTD
  - Defined XSL style sheet for for KIF
  - Implemented parser for KIF to XML
- A web-based demonstration of negotiation for the purchase of a laptop is available

29

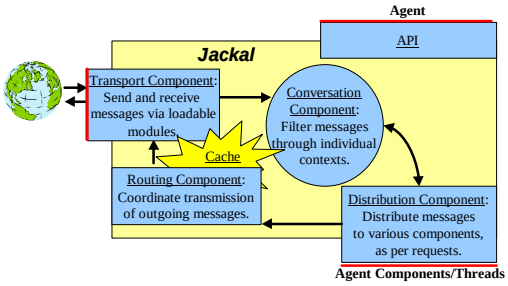
Jackal

A Communications Infrastructure for Java-based Multi-agent Systems

- a Java package facilitating the use of the KQML ACL.
- Presents a simple yet powerful API.
- Situates messages within conversational context.
- Blackboard provides flexible interface to messages traffic.
- Does not require ***any*** modification to existing code.
- Supports multiple agents within the same virtual machine.
- Plug n' Play interface for communication protocols.
- 100% pure Java, using only SunSoft Java libraries.
- Implements many aspects of the proposed KNS specification, including multi-protocols, alias resolution and authentication.
- Provides scalable, reliable messaging infrastructure.

30

Jackal Architecture: Overview



31

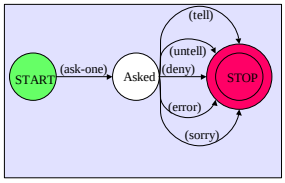
Conversation-based Protocols

- 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 in jackal
  - Currently modeled as DFA
  - Each conversation managed by separate thread.
  - Maintain context through local data store, accessible to agent.
  - Declarative specification.

32

Current Conversation Specification

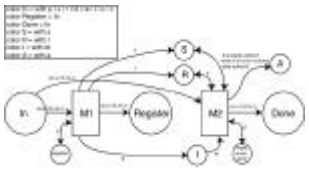
```
// Conversation Template
(conversation
   (name kqml-ask-one)
   (author "R. Scott Cost")
   (date "3/4/98")
   (start-state START)
   (accepting-states STOP)
   (transitions
      (arc (label ask-one) (from START) (to Asked) (match "(ask-one)"))
      (arc (label tell)    (from Asked) (to STOP)  (match "(tell)"))
      (arc (label deny)    (from Asked) (to STOP)  (match "(deny)"))
      (arc (label untell)  (from Asked) (to STOP)  (match "(untell)"))
      (arc (label sorry)   (from Asked) (to STOP)  (match "(sorry)"))
      (arc (label error)   (from Asked) (to STOP)  (match "(error)"))))
```



33

Using Colored Peter Net Specifications

- Colored Petri Nets (CPNs) offer a well defined and expressive formal model for specifying conversations and behavior.
- Formally equivalent to regular Petri nets.
- Supports modeling of concurrency.



CPN for KQML Register Conversation

- Widely used and supported by a large body of literature
- Simple enough for general use and practical implementation.
- Intuitive graphical representation

34

Jackal Message Distributor

- Provides a simple blackboard through which the agent logic access agent services and threads.
- Communication focal point for agent processes.
- Processes can check or wait for messages of a specified type.
- Messages are bundled with context information.
- Match on:
  - Message form/content
  - Priority
  - Others
- Specify:
  - Number of matches
  - Delete or write protect matches
  - Lifetime of a request

35

Typical Components of a Multi-agent System

- Client Agents
- White Pages Service (Agent Name Server)
- Yellow Pages Service (Broker)
- Resource Management (Control Agent)
- Resource Providers (Libraries)
- General Service Providers

36

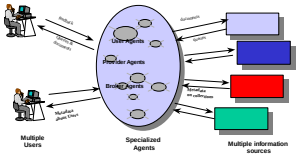
KQML Naming Scheme (KNS)

- A DNS-like scheme for agent naming
- Protocols for dynamic group formation and disbanding.
- Transparent maintenance of a distributed, persistent identity for agents.
- Facilities for 'no-fault' access to agents and basic agent information.

37

CARROT: Cooperating Agent-based Routing and Retrieval of Text

- An example of a mediated agent-based information retrieval architecture developed at UMBC.
- Agents provide access to different corpora, using existing IR engines



- Agents share metadata with Broker agents, which route queries and new documents to the "right" place(s).
- Two enabling technologies:
  - KQML agent communication language
  - N-gram processing

38

Conclusions

39

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.

40

Prospects

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

41

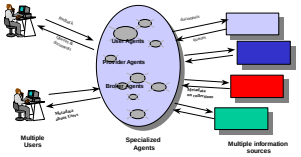
For More Information

- **General information on software agents**
  - http://www.cs.umbc.edu/agents
- KQML
  - http://www.cs.umbc.edu/kqml
- KIF
  - http://www.cs.umbc.edu/kif
- Ontologies
  - http://www.cs.umbc.edu/ontology/
- Agent Communication Languages
  - http://www.cs.umbc.edu/acl/
- Jackal
  - http://jackal.cs.umbc.edu/J3

42

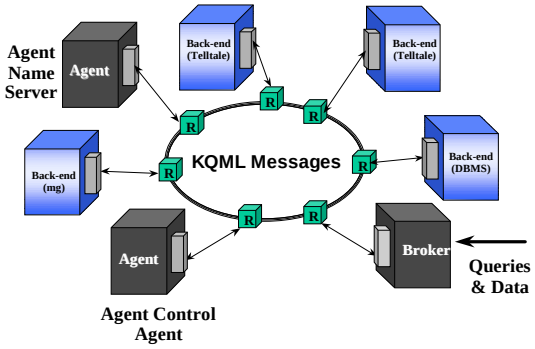
43

CARROT: Cooperating Agent-based Routing and Retrieval of Text



- An example of a mediated agent-based information retrieval architecture developed at UMBC.
- Agents provide access to different corpora, using existing IR engines
- Agents share metadata with Broker agents, which route queries and new documents to the "right" place(s).
- Two enabling technologies:
  - KQML agent communication language
  - N-gram processing

44



45

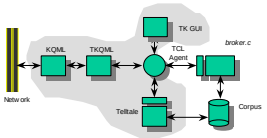
Broker and back-end agents

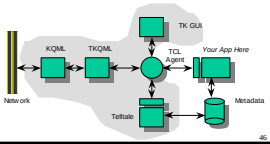
Carrot Broker

- Interacts with local IR/DB engines to access data
- No interference with existing applications
- Generates metadata, to be shared with one or more brokers
- Metadata for a set of documents is a [compressed] *n*-gram profile

Back-end agent

- Collects metadata from back-end agents
- Uses Telltale to manage these metadata corpora
- Otherwise similar to back-end agents
- Can be organized in hierarchies





46

Telltale

- Telltale is an information retrieval engine designed for
  - scalability
  - use with a wide variety of document types and languages
  - embedding in larger systems
  - generating corpus metadata
- Some key features include
  - vector space model for representing documents and queries
  - use of character n-grams
  - use of corpus "centroids" as metadata
  - Tcl/Tk user interface
  - Agent API using KQML

47

n-grams vs. words

- An IR system can use n-grams or words as terms
- Advantages of n-grams
  - Don't need a morphological model (e.g., for stemming) so good for multi-linguistic environment or non-language corpora (e.g., Java code).
  - Robust with respect to letter errors (typos, OCR errors, etc)
  - Provides some context since they span adjacent words
    "computer science" --> compu + … + ter_ s + er_sc + r_sci + _scie + ...
- Advantages of words
  - Can be more precise ("computation" but not "computer")
  - Amenable to boolean combinations
- Bottom line?
  - Depends on specifics of application

48

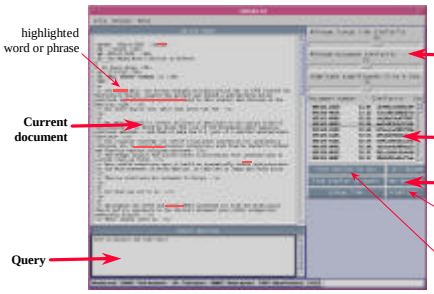
Desiderata for Corpus Metadata

- **Effective** - metadata accurately predicts corpus content
- **Concise** - metadata can be quickly transmitted to brokers
- **Abstractable** - to support hierarchies of brokers
- **Interchangeable** - applies to queries, documents, and corpora in the same manner
- **Generatable** - automatically
- **Versatile** - applicable to a wide variety of documents

➡ Corpus centroids based on n-grams satisfy these criteria

49

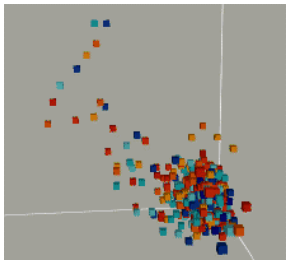
Telltale user interface



50

VR based visualization of retrieval

- The only way to comprehend a large corpus or result set is through visualization.
- SFA, for example, provides
  - Real-time, interactive stereo viewing of results of information retrieval engine
  - Each document returned is rendered as a glyph (icon)
  - Document properties mapped to 3D location, shape, color, transparency, and texture.
  - Spatialization of complex relationships and comprehensible display of multiple variables



51

VR Approach

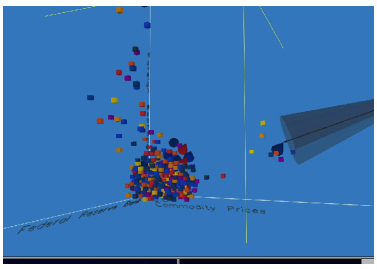
- Immersive
  - Isolates the user from environment
  - Expensive
- Minimally-immersive
  - Access to environment
  - Collaboration possible
  - Low cost
  - Two hands give proprioception
- Uses Two 3D Trackers with Buttons
  - User manipulates 3D scene with trackers
  - Each hand has a distinct role -- left sets up context and right performs fine manipulation

52

Visualizing a document space

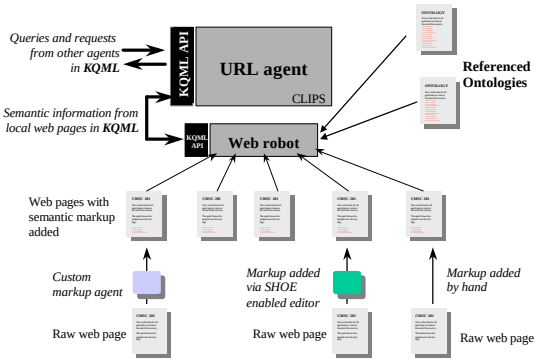
Mappings

X: similarity to "*federal reserve bank*"
Y: similarity to "*commodity prices*"
Z: similarity to "*foreign exchange rate of the dollar*"
Shape: similarity to "*coup attempt against Noreiga*" with cube as lowest and cone as highest
Color: age of document with blue as the oldest and yellow as the newest
Transparency:
Texture:



53

urlAgent Prototype Architecture

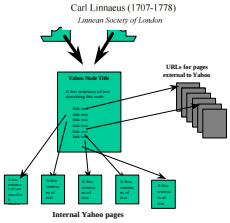


54

Linneaus

- **Problem:** What can we use as an ontology to characterize what a document is about?
- **Solution:** use a pre-existing, "naturally occurring" ontology, e.g.,
  - Yahoo hierarchy -- 150K nodes
  - Newsgroup hierarchy -- 5K newsgroups
  - Encyclopedia articles -- ~10K articles
- **Approach:** automatically classify the target document with respect to the ontology corpus with telltale.

Carl Linnaeus (1707-1778)
Linnean Society of London



55