# Intelligent Agent Technology

Colin G. Harrison
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
cgh@watson.ibm.com

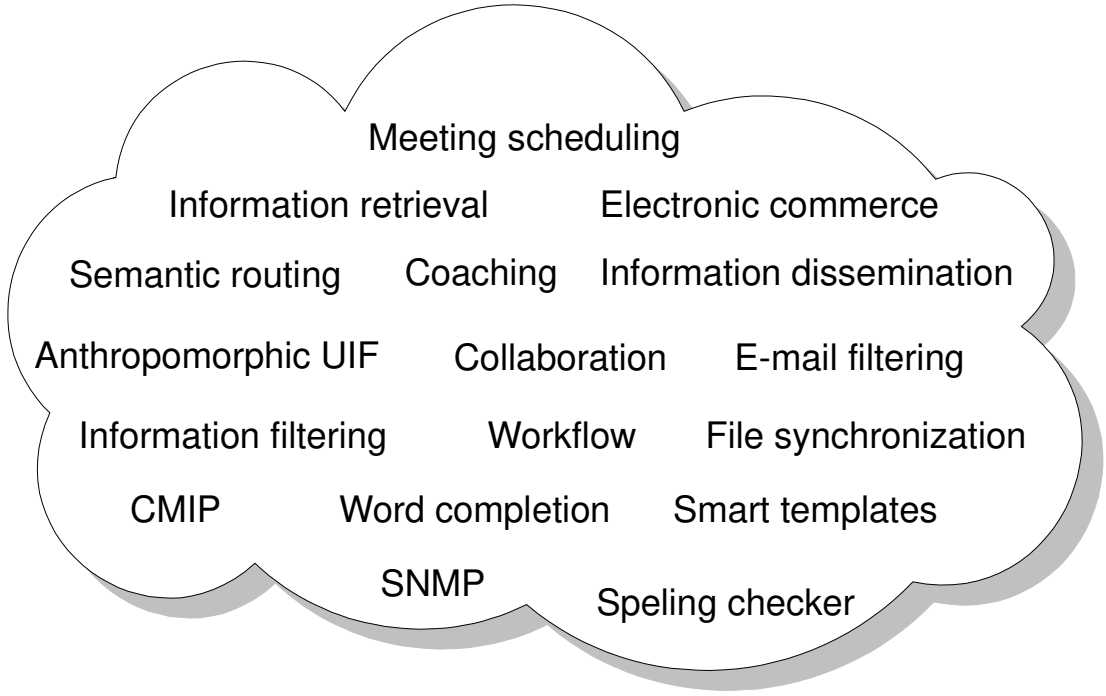
Copyright IBM 1995

IBM Research/Communication Systems & Networked Services

## Outline, Goals & a Principle

- Outline:
  - 1) What is an Intelligent Agent?
  - 2) Intelligence:
    - *Machinery*
    - *Content*
  - 3) Agency:
    - *Access*
    - *Security*
  - 4) Who needs Intelligent Agents anyway?
- Goals:
  - Deconstruction, demystification
  - Practical concepts
  - Practical examples
  - Practical applications
- Principle:
  - KISS

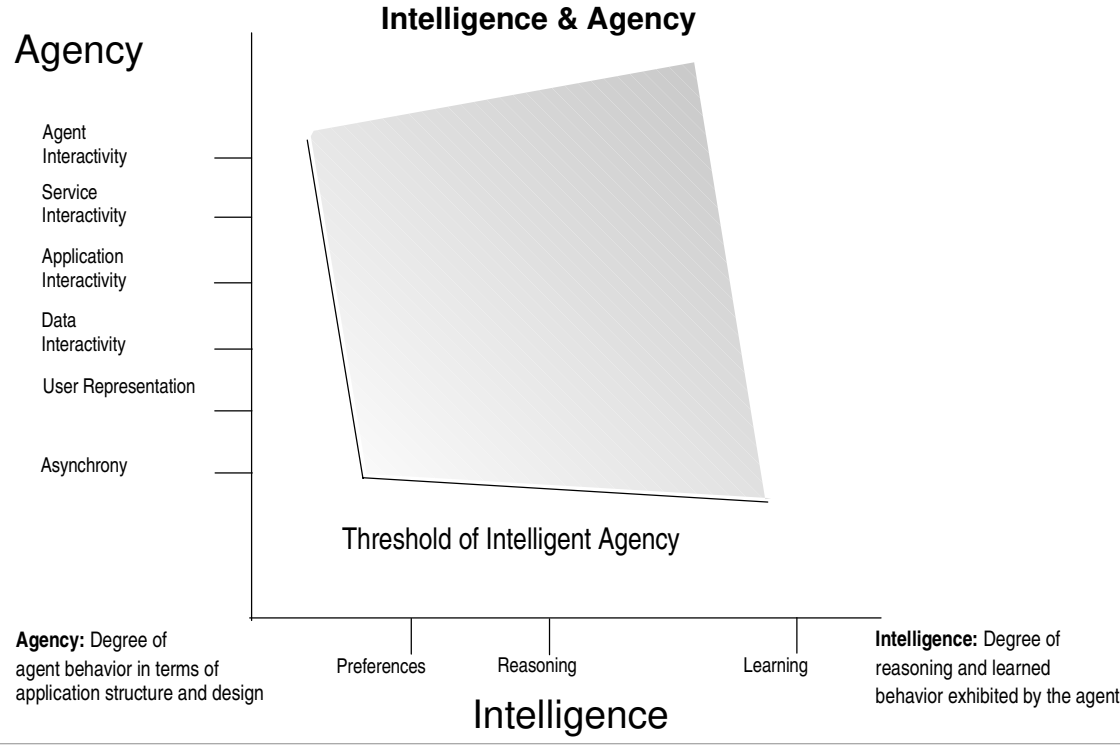
**Intelligent Agents?**




IBM Research/Communication Systems & Networked Services

**Attempted Definitions of Intelligent Agents**

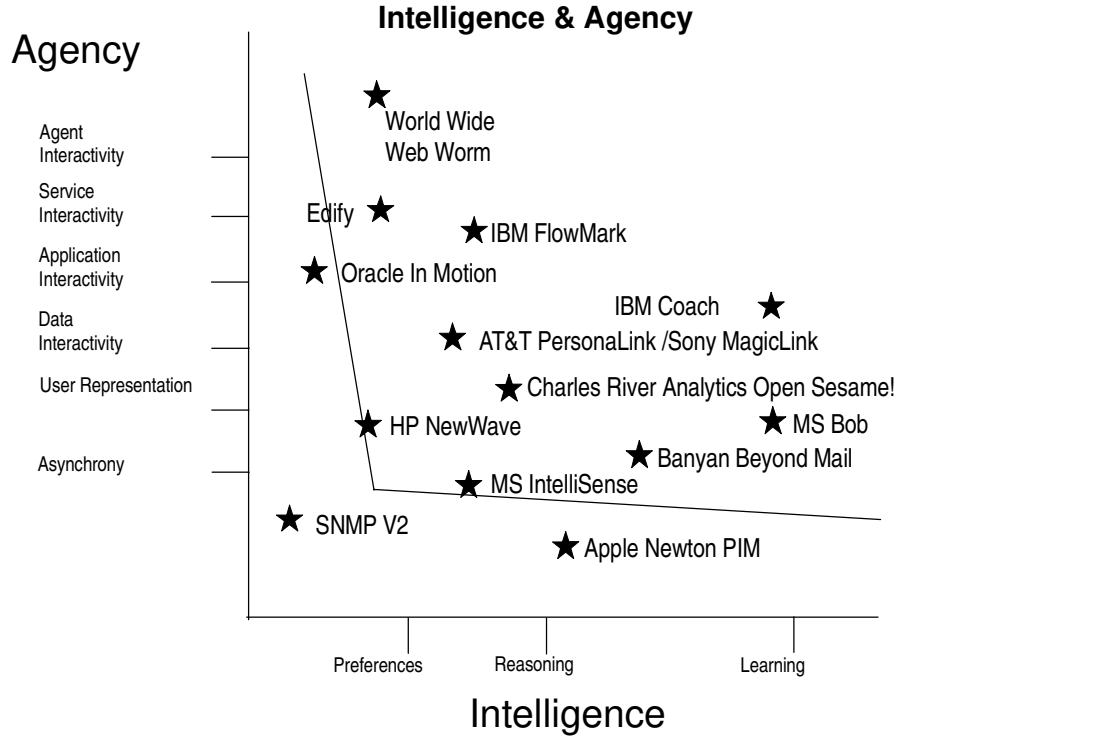
- Wooldridge & Jennings, 1994
  - *Autonomy*: Agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
  - *Social ability*: Agents interact with other agents (and possible humans) via some kind of *agent-communication language;*
  - *Reactivity:* Agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
  - *Pro-activeness:* Agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by *taking the initiative.*
- Selker, 1995
  - *Intelligent Agents:* Things that know how to do things for you that you could probably do yourself if you had the time.
- Gilbert, 1995
  - *Intelligent Agent Technology*: Intelligent agents are any one of a range of software applications, whose goal is to assist a user and to act on his or her behalf. Intelligent agents are more of a design philosophy than a specific technology, and hence appear in several different forms.




**Agency:** Degree of agent behavior in terms of application structure and design

**Intelligence:** Degree of reasoning and learned behavior exhibited by the agent

IBM Research/Communication Systems & Networked Services

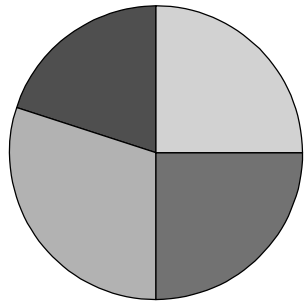


**Intelligence & Agency**

## Intelligence

Machinery:
*inferencing, learning, validation, representation*

Content: *rules, context, application ontologies & grammars*

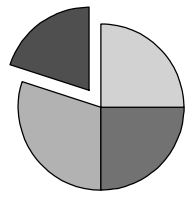


## Agency

Security: *mutual, public authentication, privacy, payment*

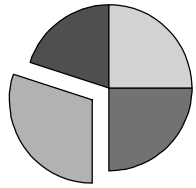
Access: *to applications, data & services, networking, mobility*

**Intelligence: Machinery**



- Inference engines
- Learning
- Rules validation
- Events & Actions
- Agent theory: negotiation, collaboration

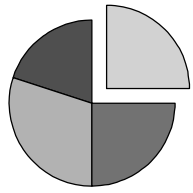
## Intelligence: Content



- Knowledge Representation, languages, translation
- Ontologies, domain-specific vocabularies
- Rules, multiple representations, problem of user input
- Observations: Events, User Actions
- Free-text, natural language understanding
- Dialogues, application grammars

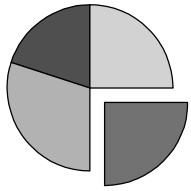
IBM Research/Communication Systems & Networked Services

## Agency: Security



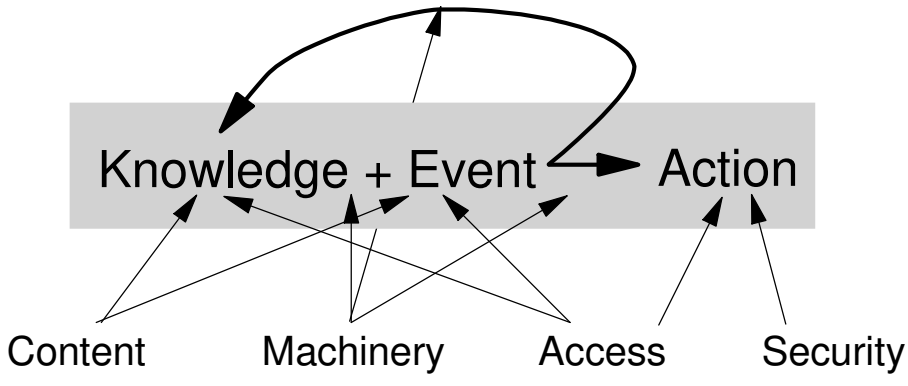
- Authority, authentication, access control
- Privacy, integrity, auditability
- Payment reconciliation
- Padded cells for itinerant agents
- Emergent phenomena
- Trust, reputation

**Agency: Access**

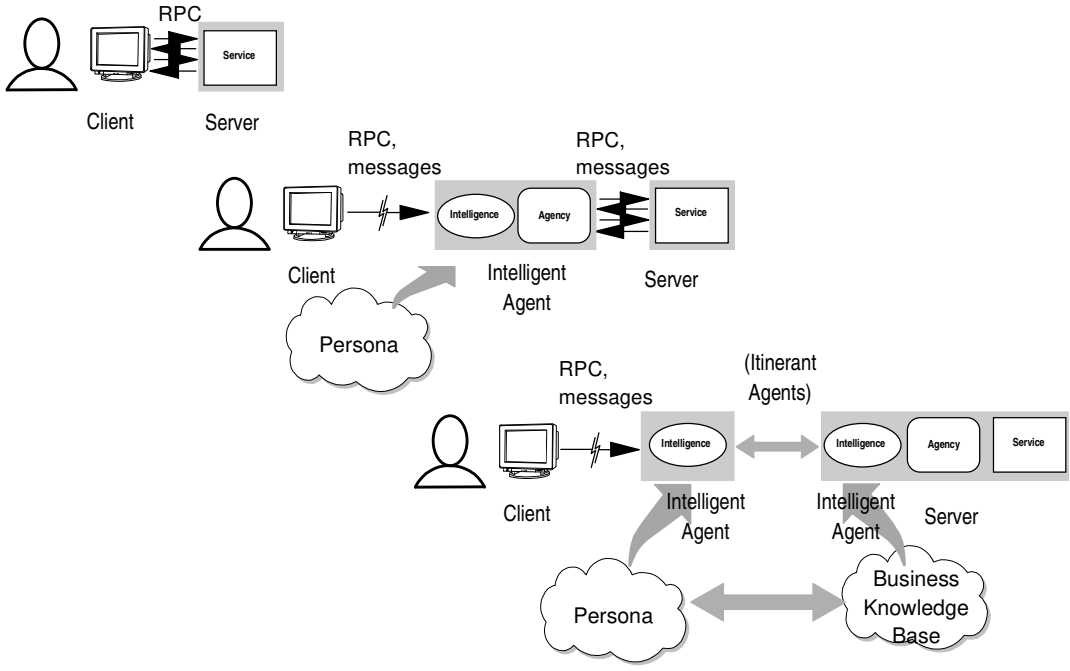


- Action/Event frameworks - What's going on?
- Launching actions from the Machinery
- Hooks into 'legacy' applications
- Distributed systems: distributed object frameworks, communication protocols, itinerant agents

**Basic Model for an Intelligent Agent**

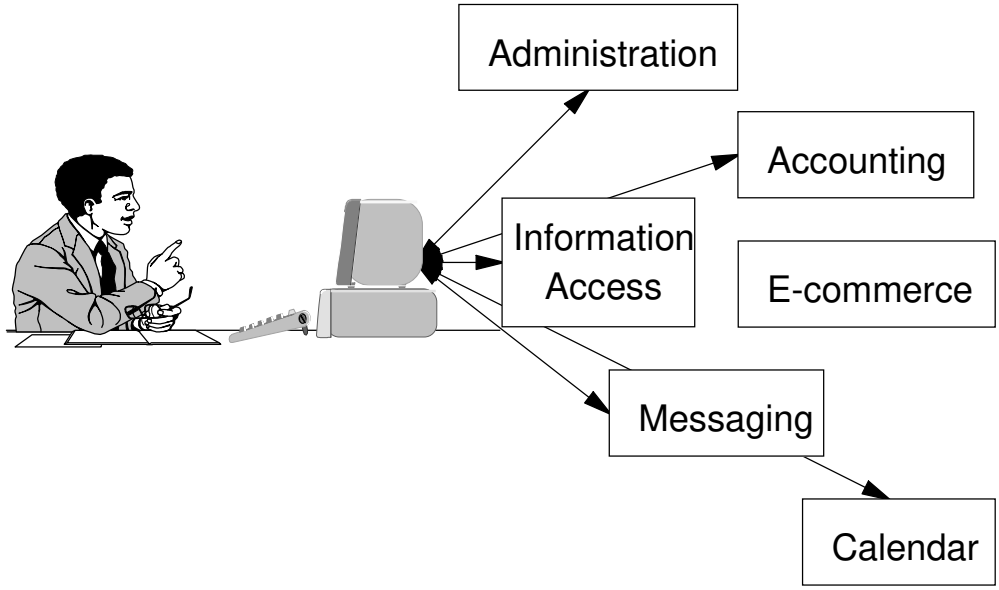


## Evolution towards Intelligent Agent Applications

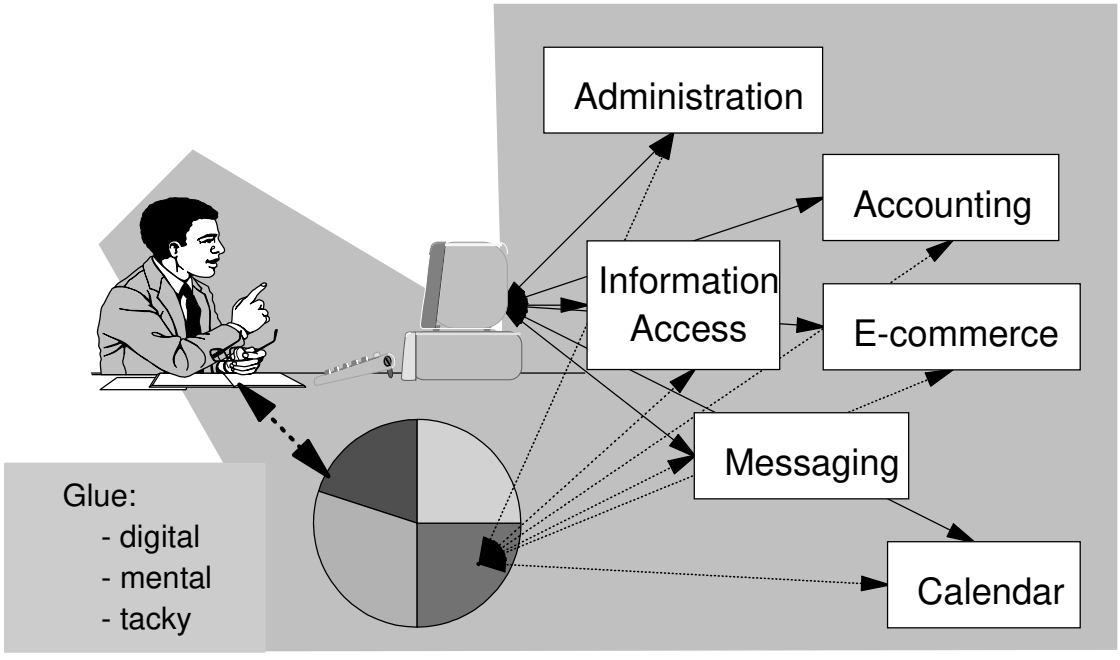





## Intelligent Agent as Glue: Before



**Intelligent Agent as Glue: After**

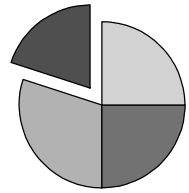



IBM Research/Communication Systems & Networked Services

**Summary: What is an Intelligent Agent?**

- Broad category of application techniques, specific IA technology.
- Intelligence & Agency, degrees of agentry:
  - Machinery
  - Content
  - Access
  - Security
- Intelligent Agents
  - Glue (access & security)
  - Application flow as data (knowledge & events)

## Intelligence: Machinery



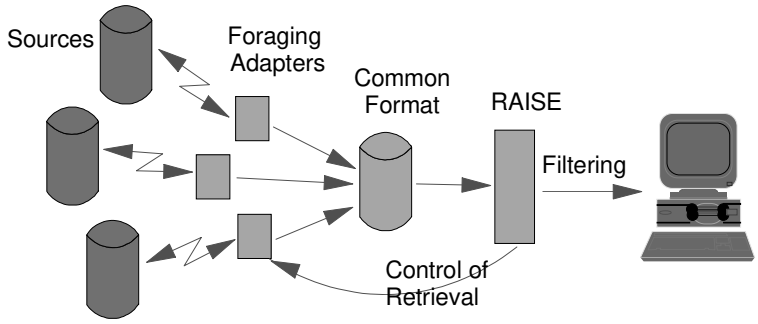
- Inference engines
- Learning
- Rules
- Events & Actions
- Agent theory: negotiation, collaboration

IBM Research/Communication Systems & Networked Services

## Inferencing Engines for Intelligent Agents

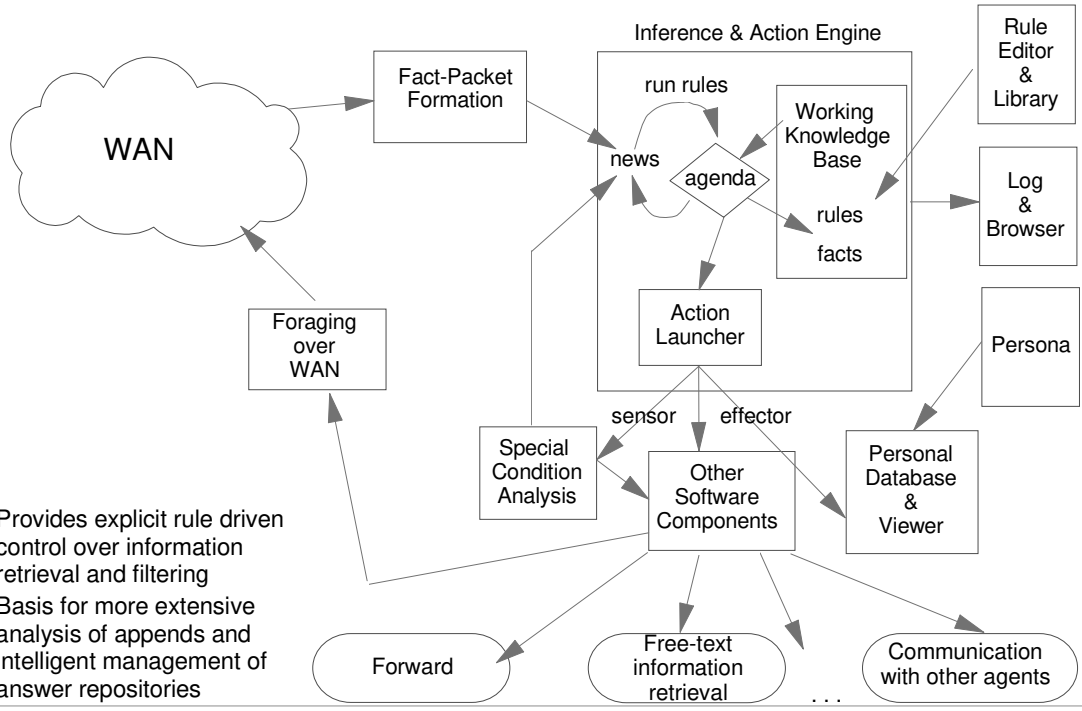
- What is an inferencing engine?
  - Core is an expression evaluator for rules & knowledge
  - May be able to incorporate derived knowledge (forward chaining)
  - May be able to create persistent derived knowledge
  - May be able to retrieve remote knowledge
  - Receives 'Events' (as new knowledge) & may produce Actions
  - May be able to explain its conclusions (reverse chaining)
  - May have absolute beliefs or may assign probabilities of truth (Bayesian or probabilistic reasoning)
  - May produce multiple possible Actions (with differing probabilities of correctness)
- How is an inferencing engine for an Intelligent Agent different from an Expert System?
  - Lighter weight
  - Fewer rules
  - More Events, multiple threads/users
  - Action oriented

## IBM GLOBENET (TM) - An agent-based dissemination service






## IBM RAISE - An object-oriented Inferencing System




- Provides explicit rule driven control over information retrieval and filtering
- Basis for more extensive analysis of appends and intelligent management of answer repositories

## Example of RAISE rules

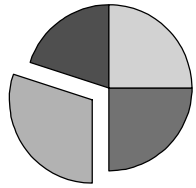
```
BEGIN ERULES
(RepoDisk, IBMPC)>(Importance, High);
(AND(OR(Subject, Windows)(ForumName, OS2WRAP.FOR))(Keyword, BIOS)>(Topic, HomeBios);
(Topic, HomeBios)>(KeepInFolder, HBiosItems);
(Subject, Egghead)>(Topic, RetailSoftware);
(AND(AND(Topic, HomeBios)(Importance, High))(Superior, $X))>(AnnotatedSend, $X, GreatBiosStuff!);
(Importance, High)>(DisplayColor, Red);
(AND(Topic, RetailSoftware)(Importance, High))>(KeepInFolder, Retailers);
END ERULES
BEGIN LFACTS
(Superior, John);
(Superior, Ann);
END LFACTS
BEGIN REFLEXES
SENSOR KeyWord KeyWordProcedure
SENSOR Phrase KeyPhraseProcedure
SENSOR Subject SubjectProcedure
EFFECTOR KeepInFolder SendToFolderProcedure
EFFECTOR PrintMessage PrintMessageProcedure
END REFLEXES
```

IBM Research/Communication Systems & Networked Services

## Learning techniques for Intelligent Agents

- Basis for learning (Content):
  - Event/Actions
  - User actions
  - User-IA dialogs
- Methods:
  - Neural Network pattern extraction
  - Probabilistic analysis
- Outcomes (Content):
  - New knowledge
  - New rules
  - Modification to rules
  - Modification to Bayesian coefficients
- Mostly still a research topics with a few exceptions:
  - Apple Newton (TM)
  - General Magic Magic CAP (TM)
  - Charles River Analytics Open Sesame! (TM)

## Intelligence: Content



- Knowledge Representation, languages, translation
- Ontologies, domain-specific vocabularies
- Rules, multiple representations, problem of user input
- Observations: Events, User Actions
- Free-text, natural language understanding
- Dialogues, application grammars

## Sources of Content

- Reasoning deals with both structured & unstructured content
  - structured, explicit parameters, attributes
  - unstructured, properties of items, textual bodies
- Databases
  - structured
  - unstructured, free-text
- Application-domain specific:
  - Themes
  - Grammars
- Statistical 'knowledge (social computing):
  - MIT Media Lab HOMR, WebHound, Doppelgaenger
- Content from observation:
  - Event/Action logs
  - User/Action logs
    - server
    - client(s)
  - Basis for learning
    - new knowledge
    - new rules
    - modified rules
- User-provided data (Persona):
  - rules
  - administrative (calendar, directories, hierarchies,...)

## Knowledge Representation

- How is 'knowledge' different from 'data'?
  - Self-describing
  - Structured
  - Communicable, translatable
  - Expressiveness: intent, goals, capabilities
- Ontology - How to describe a class of 'things'
- Databases
  - structured: schema, deep description of data structure
  - unstructured: free-text, video
- Rules

IBM Research/Communication Systems & Networked Services

## Agent Communication Languages

- Non-AI languages:
  - EDI
  - Medical
  - Airline reservations

## Agent Communication Languages

- KSE
  - ARPA-sponsored effort to deal with the task of developing conventions to facilitate sharing of knowledge bases and knowledge systems. The goal is to define, develop and test infrastructure and supporting technology to enable participants to build much larger and more broadly functional systems than could be achieved by working alone.
- KIF
  - To provide a lingua franca for the development of intelligent applications with an emphasis on interoperation and a common interchange format among knowledge representation languages. It features a prefix version of First Order Predicate Calculus with extensions, declarative semantics and the ability to express meta-knowledge.

IBM Research/Communication Systems & Networked Services

## Agent Communication Languages

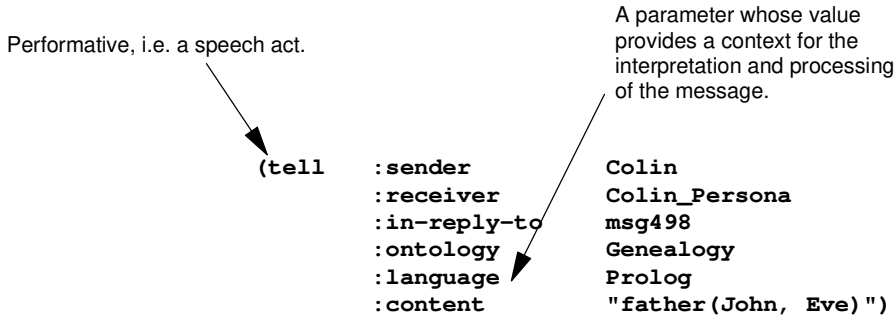
- Ontologies
  - Every knowledge based system relies on some conceptualization of the world (objects, distinctions, and relationships that matter for performing some task). These are embodied in a formal representation scheme in the ontological system.
  - A common ontology refers to the an explicit specification of the ontological commitments of a set of programs.
  - Ontologies defines a set of classes, functions, and object constants for some domain of discourse, for example: shirts.
  - Within the KSE, the Shared Re-usable Knowledge Base group has worked on the construction of ontologies for various domains. Ontologies are written in KIF using the definitional vocabulary of Ontolingua.

## Agent Communication Languages

- KQML
  - A knowledge representation language specifically for agent communication, based on the concept of *'Performatives'*, which are modeled on human speech acts.
  - It facilitates agents that act like secretaries, that is intermediaries between agents making requests for Performatives, and underlying applications.
  - It provides support for distributed applications, such as a directory of services able to execute Performatives and of how these services like to maintain a conversation (application grammar).
  - It supports asynchronous communication between a (client) application which emits a KQML agent and a service point which can engage in the dialogue with the agent.
  - The set of Performatives accommodates a variety of interaction models, architectures and transport protocols. They include:
    - *ask-all, stream-all, ask-if*
    - *reply*
    - *deny*
    - *forward*
    - *tell*

IBM Research/Communication Systems & Networked Services

## Example: KQML




Note: KSE & KQML material drawn from LaBrou, Mayfield, Finin, AAAI Spring Symposium, March 1995

## Example: KQML

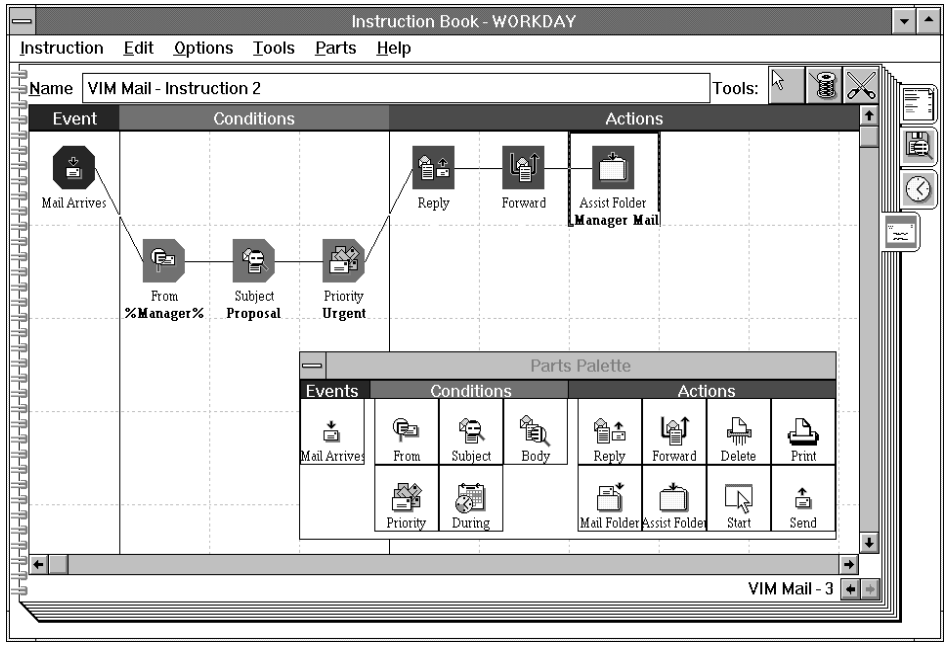
```
(ask-one     :content (PRICE IBM ?price)
             :receive     stock-server
             :language    LPROLOG
             :ontology    NYSE-TICKS)
```

IBM Research/Communication Systems & Networked Services

## Rule Management Problems

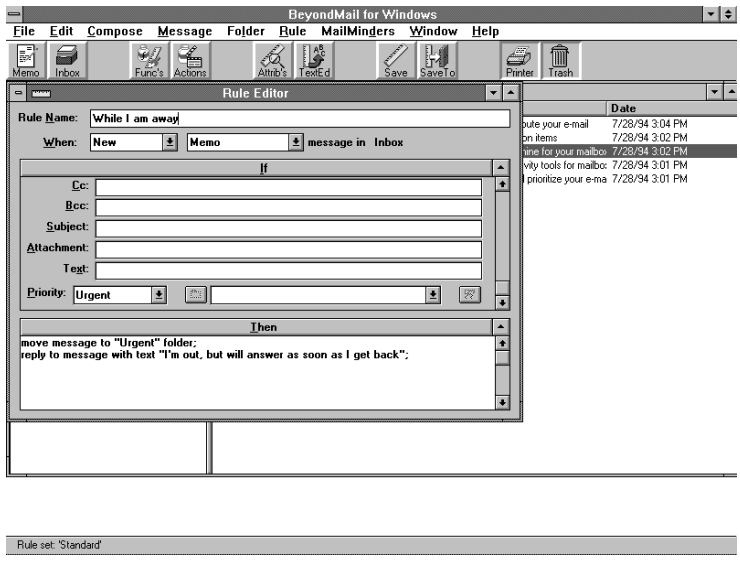
- Rule authoring problems
  - Getting user input, user classification, default class behaviours
  - Avoiding inconsistencies/conflicts in rule sets, detection tools
  - What are the real discriminants? (RAT)
- UIFs for rule authoring
- Rule-base combination & sharing
- Trust & verify

## Example: Rule Authoring UIF in IBM IntelliAgent (TM)

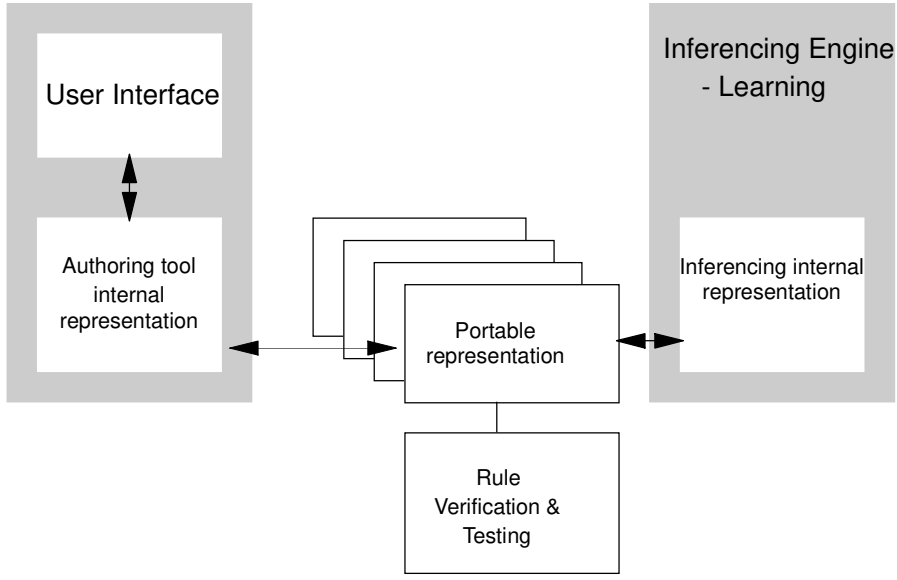





## Rule Authoring: Banyan BeyondMail (TM)

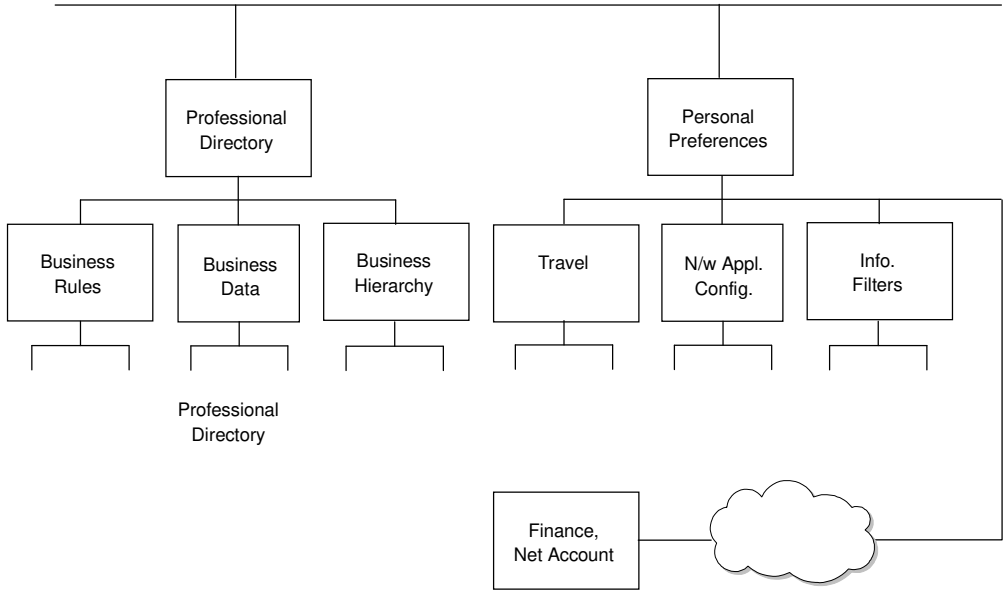


## Multiple Rule Representations






## Persona



## Persona Example: Microsoft Bob (TM)

- "Bob" (formerly Utopia): "Social interface" for unsophisticated Windows users
  - Suite of applets (e.g. word processor, checkbook manager, calendar) with common interface
    - *Apps launched by touching objects in rooms of displayed house*
  - On-screen cartoon-figure guide walks user through tasks
  - Available 3/95, $99
    - *Preloaded on Gateway 2000 and other OEMs' machines*
- AGENCY: Inline; not asynchronous
- INTELLIGENCE: Learning and adaptation to user's behavior and habits
- BUSINESS MODEL: Low-price base package, then add-ons (e.g. recipe book for "kitchen", educational matter for "study")

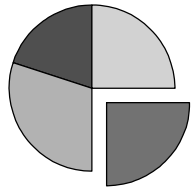
IBM Research/Communication Systems & Networked Services

## Microsoft Bob (TM)

## Agents in the Internet: World Wide Web Worm

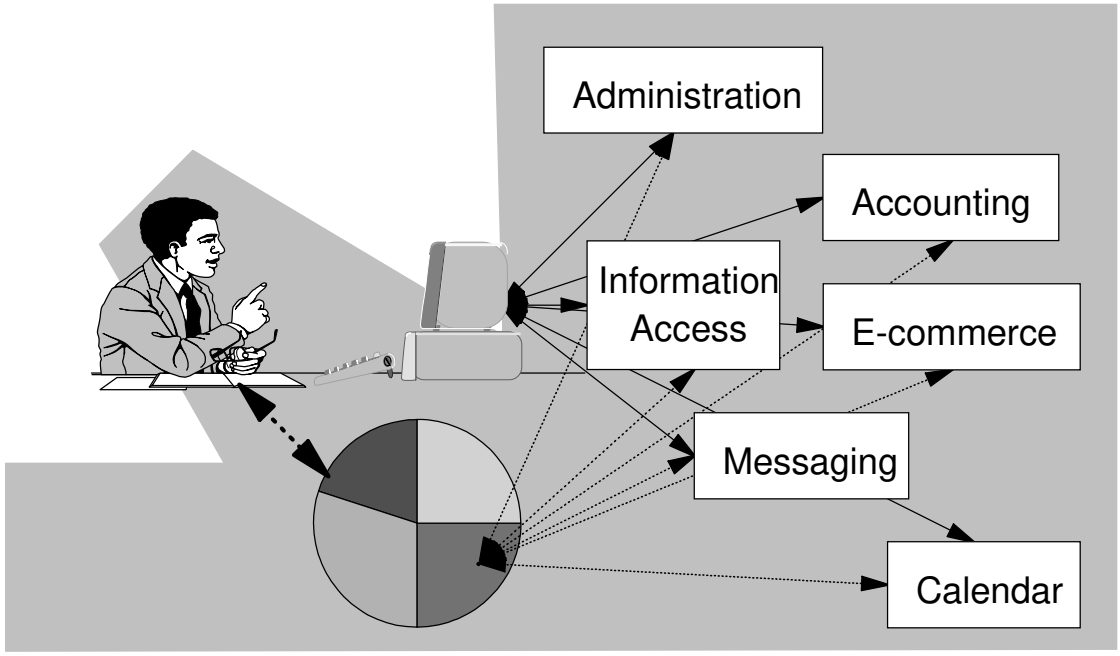
- Description:
- HTTP based tool which queries Web servers to collect new HTML resources and build HTML indices for users.
- AGENCY:
  - Off-line traversal and interaction with servers
  - Builds knowledge base for users
  - Exploiting HTTP/HTML access to servers
- INTELLIGENCE:
  - Discovers new Web servers
  - Has some limited understanding of HTML structure and Document

## Agency: Access



- Action/Event frameworks - What's going on?
- Launching actions from the Machinery
- Hooks into 'legacy' applications
- Distributed systems: distributed object frameworks, communication protocols, itinerant agents

## Intelligent Agent as Glue: After

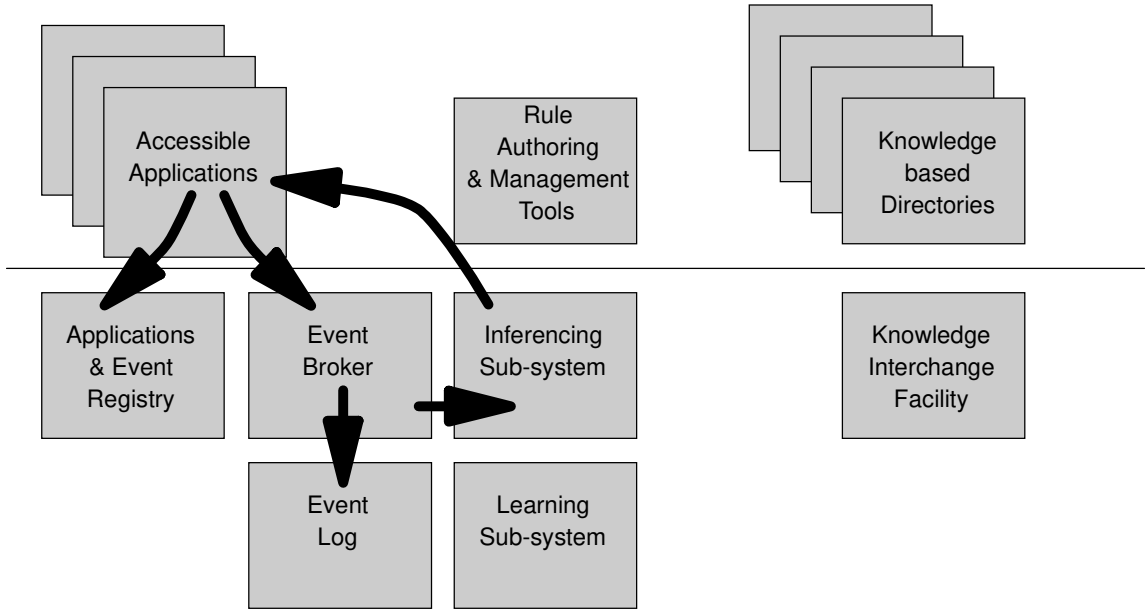



IBM Research/Communication Systems & Networked Services

## Intelligent Agent Frameworks

- Making agent services (Machinery, Content, Access & Security) available to all applications. Making applications available to agent services.
- Operating system or middleware functions that provide a common set of Application Program Interfaces for agent-based services:
  - Application registration: capabilities, Events of interest, Events produced, Actions supported
  - Local and remote application capabilities, Events, Actions
  - Broker for Events, 'software bus', Event log
  - Common knowledge-based Directories: user Persona, information service & product catalog...

## Framework for Agent Access

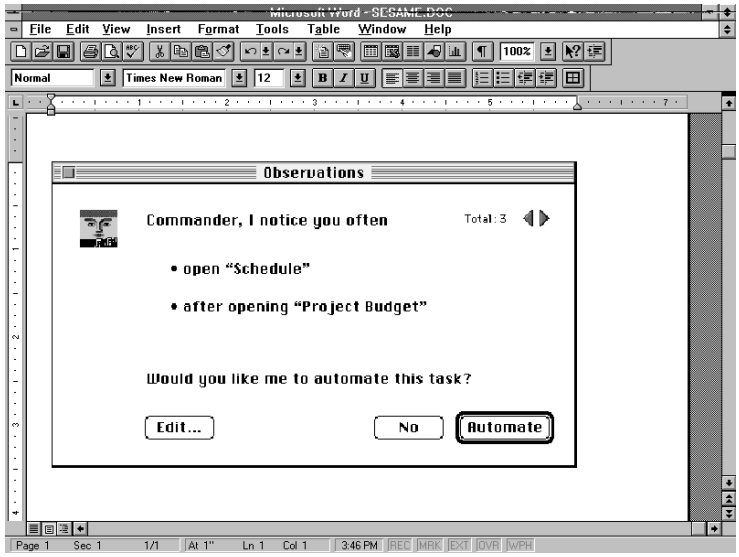



IBM Research/Communication Systems & Networked Services

## Apple Open Scripting Architecture (TM)

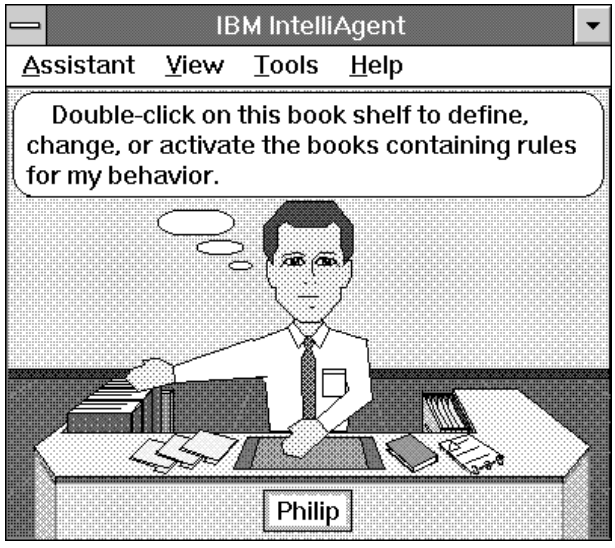
- Based on Interapplication Communication (IAC)
- Apple Events - high-level event that comply with the Apple Event Interprocess Messaging Protocol
  - Real-time messages between applications
  - Appear in the applications' main event loop
  - Example:
    - *Open Application, Open Documents, Print Documents, Quit Application (Required suite)*
    - *Cut, Copy, Paste*
    - *Get Data, Set Data, Move, Delete*
- Apple Event Object Model
  - A way to access application objects
- Apple Event Suites
  - A consistent vocabulary across applications
    - *Required suite, Text suite, Database suite, Telephony suite, Mail suite, and many others*
- Apple Events User Terminology
  - A mechanism to announce supported Events

## Charles River Analytics Open Sesame! (TM)






## IBM Intelliagent (TM)

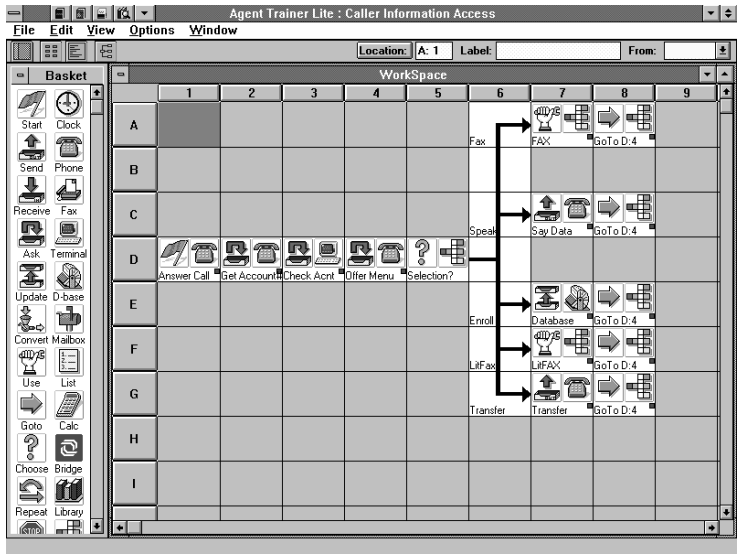


## Example: Edify (TM)

- Provides system solutions for customer service market that incorporate intelligent agent components for message (e-mail, phone, fax) filtering.
- AGENCY:
  - Message filters represent user persistently
  - Filtering occurs asynchronously from user actions
- INTELLIGENCE:
  - Uses rules based technology and inferencing to capture user preferences and some domain knowledge.
- BUSINESS MODEL:
  - Delivery of systems solutions featuring agent capability as componentry.

IBM Research/Communication Systems & Networked Services

## Example: Edify Agent Trainer (TM)

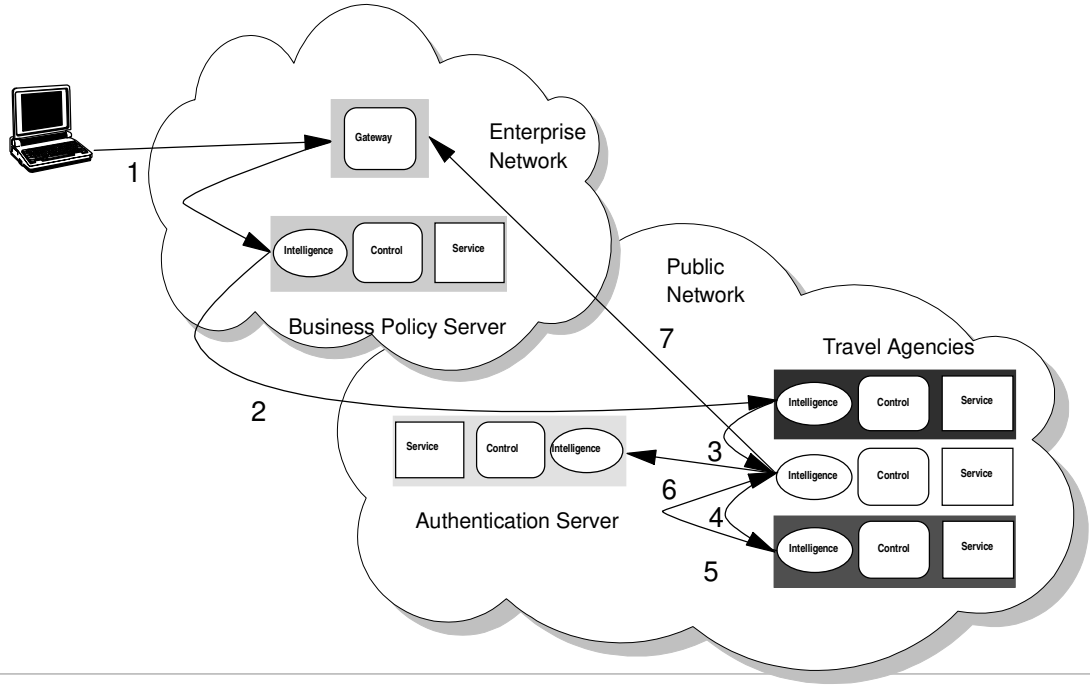


## Itinerant Agents

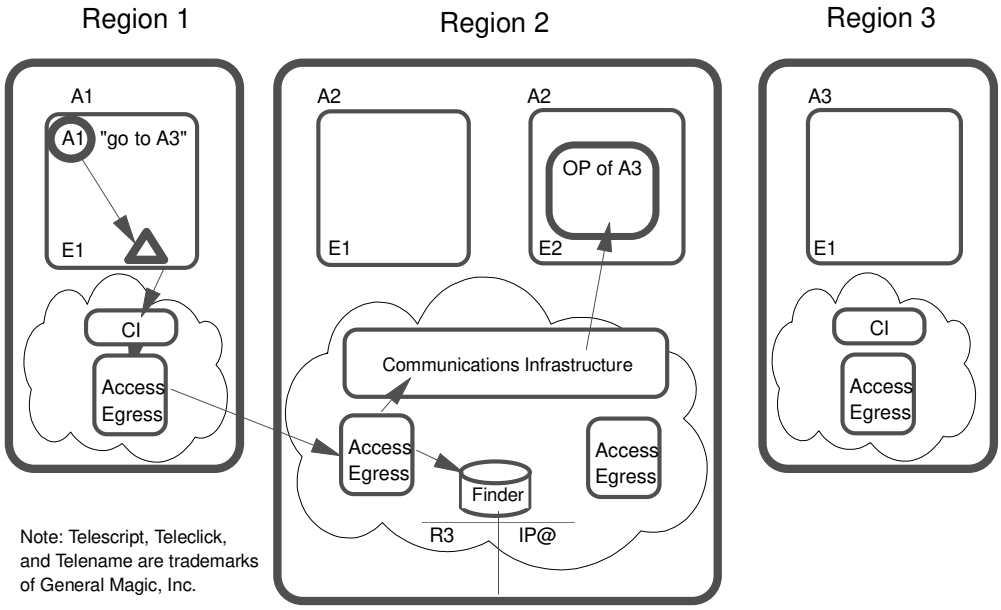
- Concepts:
  - Itinerant Agents are programs, typically written in a script language, that can execute autonomously in special environments which provide access to server functions.
  - Itinerant Agents are capable of migrating their entire state among execution environments, and of forking daughter processes.
- Examples:
  - General Magic Telescript (TM) language
  - TcL script language
  - IBM Object REXX (TM) script language
- Motivations:
  - Disconnected operation from a (lightweight) mobile client device.
  - Remote searching/filtering of very large, free-text databases.
  - Remote, real-time decision-making/process control/network management.
  - Remote, Event-triggered monitors.
- Similar:
  - Sun Microsystems' Java language & Hot Java (TM) execution environment
  - Oracle in Motion (TM)

IBM Research/Communication Systems & Networked Services

## Itinerant Agents: Travel Reservation Scenario



## Itinerant Agents: General Magic Telescript (TM) Concepts




IBM Research/Communication Systems & Networked Services

## Itinerant Agents: General Magic Telescript (TM) Concepts

- Region is a collection of TS engines owned by an Authority
- Domain is a Region and its Outposts
- Outpost is a place owned by a Region in a TS engine of another Region
- Communication Infrastructure
- Transport within a region
  - Access/Egress - between regions
  - PIP, GMTP
  - Security, authentication

## Itinerant Agents: General Magic Telescript (TM) Concepts

- Agents and Places
  - Are processes (actors) within Telescript engines
  - Named objects able to carry out computation autonomously
- Agents
  - Can self-examine and self-modify
  - Can transport themselves between places
  - Can interact with agents in places
  - Can be mobile, like Couriers or stationary, like Mailboxes
  - Named with Authority of originating Region and object Identifier (Telename)
- Places are stationary objects that can interact with agents as they enter, meet and exit
- Ticket is an object used by agent to move to another place
  - Permit
  - Telename and/or Teleaddress of destination
  - Way - immediate destination Region, Means, authentication information
  - Means (PSTN, CDPD, infrared, Portals...)
  - Class of Service
- Permit is an object held by an agent or a place that helps to define the agent or place's capabilities and to control its consumption of resources (Teleclicks)

IBM Research/Communication Systems & Networked Services

## Itinerant Agents: General Magic Telescript (TM)

```
goShopping:   op (warehouse: ClassName)
throws ProductFailure =
{
   //go to the warehouse
   *.go(Ticket(nil, nil, warehouse));

   //show an interest in prices
   *.enableEvents(PriceReduction(*.name));
   *.signalEvent(PriceReduction(), 'responder);
   *.enableEvents(PriceReduction(here.name));

   //wait for the desired price
   actualPrice = desiredPrice+1;
   while actualPrice > desiredPrice
   { *.getEvent(nil, PriceReduction());
   try
   {
     actualPrice = here@Warehouse.getCatalog()[desiredProduct].price
   }
   catch KeyInvalid (throw ProductFailure())
   }
};
```

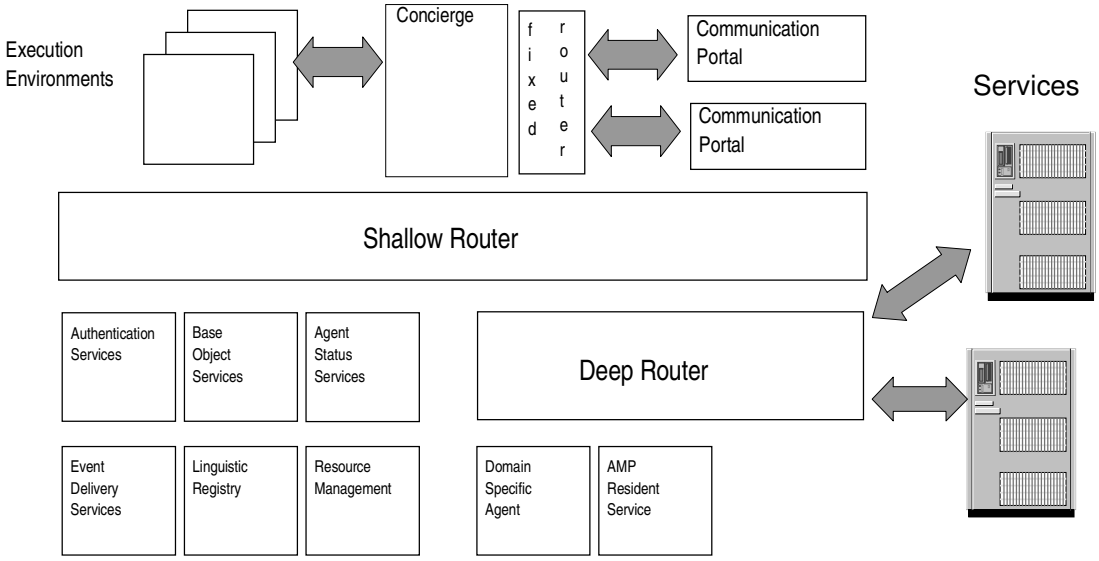
## Itinerant Agents: Tcl-DP

```
#Create price list
set price_list {50 62 75 18 12}

#Define GetPrice procedure for extracting item price from list
proc GetPrice {stock_number} {
     global price_list;
     set item_price [lindex $price_list $stock_number];
     return $item_price;
}

#Open listening on TCP socket #4251
dp_MakeRPCServer 4251

#Establish RPC connection with remote server on socket &
#save identifier in variable 'server'
set server [dp_MakeRPCClient nylan.watson.ibm.com 4251]

#Remote invocation (after transport) of defined procedure
set price [dp_RPC $server {GetPrice 3}]
```

## Itinerant Agents: IBM Object REXX (TM)

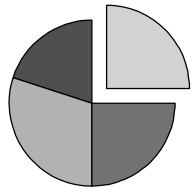
```
Ike=.reporter~new("Ike","IBM",120,"cgh.watson/BeepMe"}
return Ike               /* Return agent to calling moderator */
::requires 'sockets.frm' /* Use TCP wrapper classes */
::requires 'agents.frm   '       /* Use agent classes         */
::class Report subclass Agent    /* Inherits from Agent   */
::method init                    /* Object initialization method  */
expose Stock Trigger Home        /* Available to all methods  */
user arg Name, Stock, Trigger, Home
self~init:super          /* Run super-class init method   */
self~objectName-Name     /* Object-String returns nickname    */
self~HangUp              /* Close transporting link ('agents' method) */
::method Stock; expose Stock; return Stock   /* Report target to Moderator*/
::method Notice          /* Moderator reports action on target stock  */
expose Stock Trigger     /* Access object variables                   */
use arg NameIn, PriceIn
if PriceIn<Trigger then
   self~PhoneHome('From Ticker Service:' Stock 'at' PriceIn)
::method PhoneHome Private   /* Alert sponsor of Event; private method*/
expose Home              /* Beeper service    */
use arg TheMessage
TheBoss=.SyncLink~New(Home)      /* TCP proxy (Sockets Class) */
TheBoss~Send(TheMessage)
```

## Example: IBM Research Agent Meeting Place




IBM Research/Communication Systems & Networked Services

## Agency: Security



- Authority, authentication, access control
- Privacy, integrity
- Payment reconciliation
- Padded cells for itinerant agents
- Emergent phenomena
- Trust, reputation

## General Security Issues for Agents

- Mutual authentication of agent and server
  - Universal certification under global certification hierarchy
  - Mobile agent carries certificate with it
  - Server may require no authentication/formal authentication
  - How to authenticate the server itself?
  - Access control to server resources
- Payment
  - Based on agent's owner's credit or electronic cash
  - Potential for defrauding by server
- Reputation
  - Validate reputation of a server before visiting
  - Reputation servers, e.g. Web Advisory
- Most of these problems are being addressed independently for electronic commerce

IBM Research/Communication Systems & Networked Services

## Security Issues for Itinerant Agents

- Access rights of agent
  - Padded execution environments (Java, TcL, Telescript)
  - Right to execute, access to specific server resources, limitations on resource consumption
- Privacy
  - Encrypt agent with public key of server to be visited
  - Impossible to hide agent contents from server
  - Impossible to communicate secretly with a large, anonymous group of servers
  - Impossible to prevent tampering without specific hardware, but is detectable
- The Virus Question: *"Is this a virus I see before me?"*
  - Not simply a question of Turing-completeness of the script language, except for "spread" verb
  - Can avoid the virus problem by limiting language to only:
    - *Alter its own internal variables*
    - *Make database queries at the server*
    - *Move to another server*
    - *Send messages back to its owner*

## Emergent Phenomena

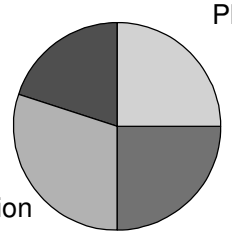
- Unforeseen collective behaviour of large, networked populations, where everything is soon or later connected to everything else (*6 Degrees of Separation*).
- Examples:
  - Mail forwarding loop
  - Modified mail forwarding loop
  - Computational economies
- Treatment:
  - Prevention
    - *Have I seen this message/agent or 'something like it' before?*
    - *Is this program/agent contain a potential for destructive behaviour?*
  - Cure
    - *Limit numbers of forwarding cycles*
    - *System management monitors to detect coherent behavioural explosions & isolate infected hosts*

IBM Research/Communication Systems & Networked Services

## Trust & Reputation

- Trust
  - Trust is developed by experience under controlled circumstances.
  - New agents or new users of agent should keep the Agents on a short leash, verify frequently and examine carefully what was done & why.
  - As trust develops, reduce the amount of verification and increase the authority of the Agent.
- Reputation
  - Itinerant Agents from unknown sources should be trusted less than those from friends and family.
  - The e-community will develop Reputation Servers which can be consulted about the source of an unknown agent.
- The e-community will eventually develop all or most of the institutions of human communities: a monetary system, a legal system, a Better-Business-Bureau, a Notary...

## Intelligent Agent Technology



Inferencing

Learning

Probabilistic Reasoning

Knowledge Representation

User-Computer Interaction

Searching & Filtering

Natural Language Support

Free text

Ontologies

Negotiation Strategies

Agent Communication Languages

Emergent Phenomena

Padded Cells

Mobile Scripting

Agent Frameworks

e-commerce

Authentication

Virus Detection

Distributed Object Frameworks

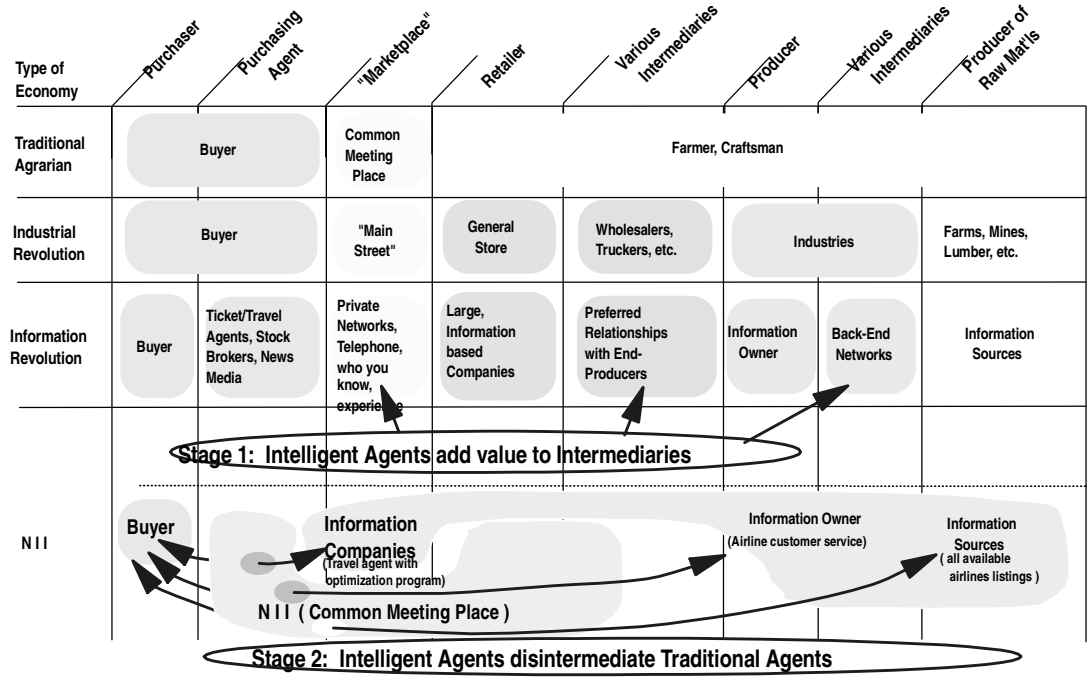
Networking Protocols

IBM Research/Communication Systems & Networked Services

## 3Ds: Why use Intelligent Agents?

- Dialog
  - Agent accumulates knowledge progressively until able to execute task
  - Multiple, overlapping contexts, application-specific knowledge
  - Persona mediates agent 'personality'
  - Avatars - 'talking heads', speech-based user interfaces
- Delegation
  - PC as a partner not a tool
  - Asynchronous execution of service requests
  - Agent provides integration over heterogeneous applications & databases, adds value to new applications & agent-enable applications
  - Agent development environment - equivalent of spreadsheets for personal & business monitors
  - Draws on Persona
- Disintermediation
  - Elimination of intermediaries in distribution of goods, services and information
  - Consumer & provider exploit each other's knowledge bases
  - Dissemination, searching, negotiating, auctions

## Disintermediation




IBM Research/Communication Systems & Networked Services

## The IS Perspective

- Stage 1: Opportunities to add value with information throughout value chain
  - Leverage for human agents
  - Requires information infrastructure
  - Requires applications that add value
  - Enabled by systems technologies (network-centric, O-O, intelligent agents )
  - Rapidly propagates new procedures through enterprise
  - Reduces bureaucracy staff
- Stage 2: Substantial threats to major industries
  - Banking, insurance agents, travel, brokerage, media broadcasting, etc.
  - Timing of movement to stage 2 depends on accessibility of information and complexity of optimization required
- Stage 3: Substantial opportunities to aid customers
  - Value of personalization, subscriber retention
  - Must provide right package of information at the right place
  - Direct Access to enterprise services reduces cost of 'human agents', performance may be improved.

## What have I missed?

- Dialog agents, interfaces
- Agent societies, agent negotiation
- Agents on the Internet
- Agents in network & system management
- Agents for mobility

IBM Research/Communication Systems & Networked Services

## Summary

- Not so much a new technology, as a new way of designing applications.
- Agent-based computing will have impacts as powerful as that of Graphical User Interfaces or Client-server. Some of these will be on the way we as individuals work with computers and some will be on the way that businesses operate.
- Personal impact:
  - "The future of personal computing is 100% about delegation." Nicholas Negroponte, Director, MIT Media Lab.
  - Personalization - adding value to applications & exploiting knowledge
- Commercial impact:
  - Disintermediation, Direct Access by customers
  - Business re-engineering, delegation of process to agents, extension of workflow
  - Customer retention through personalization
- Active research area. Lots of experimentation going on, leading to next generation applications.
- Early commercial successes:
  - Edify
  - Sandpoint
  - Verity

## Entry Points

- Communications of the ACM, July 1994
- Mailing-list: Mail to `listserv@sunlabs.sun.com` with '`subscribe agents`'
- Useful URLs
  - `http://cs.wpi.edu/Research/airg/Agents-hotlist.html`
  - `http://www.cs.umbc.edu/agents/`
  - `http://www.ai.mit.edu/`
  - `http://www.media.mit.edu/MediaLab/Welcome.html`
  - KSE: `http://ksl.stanford.edu/`
  - KIF: `http://logic.stanford.edu/kif.html`
  - Itineracy: `http://www.genmagic.com/`
  - `http://www.research.ibm.com/xw-D953-mobag-ps`
  - UIF: `http://cdr.stanford.edu/`
  - `http://www.research.microsoft.com/`
- Books
  - Wooldridge, MJ & Jennings, NR, *Intelligent Agents*, Springer-Verlag, 1994
  - Rosenschein, JS & Zlotkin, G, *Rules of Encounter*, The MIT Press, 1994
  - Bradshaw, J (ed.), *Software Agents,* MIT Press, (to be published 1995?)
  - Harrison, CG & Janca, P: *Business Applications of Intelligent Agents,* John Wiley, (to be published 2Q96).

IBM Research/Communication Systems & Networked Services

## Acknowledgments

- I am delighted to acknowledge the learning I have gained from the following agenteers: Walter Bender, Stephen Brady, David Chess, Davis Foulger, Don Gilbert, Benjamin Grosof, Peter Janca, Jeff Kephart, Aaron Kershenbaum, David Levine, the OREXX team, Colin Parris, Abhay Parekh, the late Philip Rosenfeld, Ted Selker, Rick Spagna, Steve White, Robin Williamson, and the Magicians.