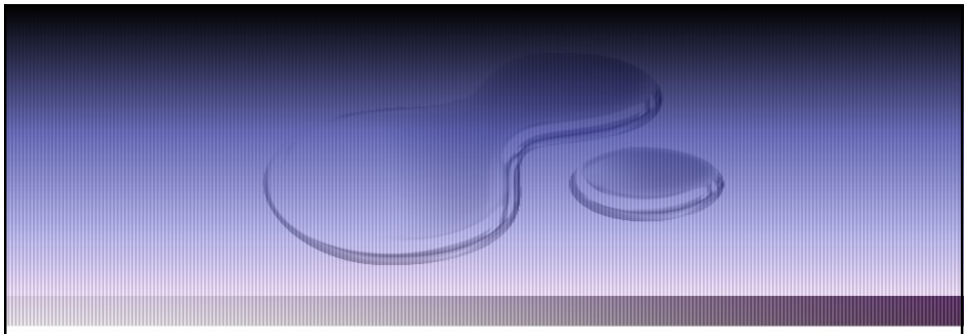


NETWORK INFERENCE, INC

Inference-Driven Business Systems:
A Look at How Artificial Intelligence Technology is Changing the Ways Business Uses Software

Making Data Meaningful

© 2004, Network Inference, Inc. All rights reserved.

network inference

Agenda

INTRODUCTION AND SURVEY
- Is "AI" a dirty word?
- Role of standards in AI proliferation
- Rules-based systems
- Logic-based systems

INTERESTING USE CASES
- Collaborative HealthWare Portal for NeoNatal Care
- Battlespace Awareness in Military Operations
- Financial Reporting in a Fortune 100 Electronics Manufacturer
- Vehicle Configurator in a Global Automaker

ENTERPRISE ARCHITECTURES
- Layered Systems – Enterprise Semantic Web
- Entailment tradeoffs
- Legacy data and user interfaces

© 2004, Network Inference, Inc. All rights reserved.
network inference

Introduction
Is "AI" a dirty word?

- **Artificial Intelligence still has a bad rap**
  - Inflated expectations from the 1970's and 80's still cause heartburn and skepticism in the minds of most IT buyers in commercial businesses
- **Expert systems are perceived as failures**
  - CLIPS never really went big commercially
  - Rule systems are little understood outside small niche communities of interest
  - Core principles of declarative programming have not made their way into any popular languages
- **"AI" as a term, is just not sellable**
  - Popular wisdom disassociates "intelligence" from "artificial"

© 2004, Network Inference, Inc. All rights reserved.

Introduction:
Role of Standards in AI Proliferation

- **Global standards are making AI vogue again, by avoiding the AI baggage. A few are:**
  - W3C – semantic web knowledge representation standards
  - OMG – extensions to UML with greater KR fidelity
  - OASIS – registry interfaces that can be reasoned upon
  - PLCS – domain specific product lifecycle ontologies
  - HL7 – health care ontologies
- **Core patterns are emerging, in uses, and in how they are described in popular literature**
  - Semantic Interoperability – interoperable systems of systems
  - Semantic Web – giant virtual knowledge base
  - Model-Driven Architecture – automated application building
  - Adaptive Systems – sense-and-respond application behavior
  - Autonomic Computing – self-healing, self-organizing behavior

© 2004, Network Inference, Inc. All rights reserved.

Big Picture:
Why is AI important? ORDER TO CHAOS.

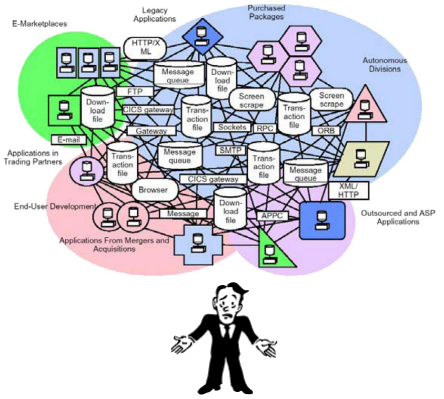
Enterprise data, created over decades, is chaotic, lacking meaning, and increasing at exponential rates.

Popular approaches to solving the problems of visibility and interoperability are incapable of coping with the complexity and scale of the root problems.

Leaving organizational decision-makers confused, and unable to see the forest for the trees.

Inference-driven systems solve these problems…





©2004, Network Inference, Inc. All rights reserved.
network inference

Big Picture:
Why is AI important? DO MORE WITH LESS

Deduce implicit facts on explicit data at runtime. Get more application knowledge with less coding. Keep your business knowledge in models, not algorithms. Maybe learn a few new things too…

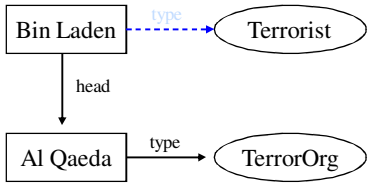
```
<owl:Property rdf:ID="head">
  <rdfs:subPropertyOf
        rdfs:resource="member" />
</owl:Property>

<owl:Class rdf:ID="Terrorist">
  <owl:sameClassAs>
    <owl:Restriction>
      <owl:onProperty
            rdf:resource="member" />
      <owl:someValuesFrom
            rdf:resource="TerroristOrg" />
    </owl:Restriction>
  </owl:sameClassAs>
</owl:Class>
```

The head of an organization is also a member of it

A member of a terror organization is a terrorist

Therefore, the head of a terror organization is a terrorist



©2004, Network Inference, Inc. All rights reserved.
network inference

Big Picture:
Why is AI important? LESS CODE, LESS $$$

INFERENCE-DRIVEN DEVELOPMENT
- Identify Problem
- Assemble Information
- Tea Break
- Model Ontologies
- Encode Instance as Facts
- Ask Queries
- Tune Models

MAINTAIN
- Requirements
- Tee Time
- Model
- Tune

ORDINARY-PROGRAMMING
- Identify Problem
- Assemble Information
- Figure out Solution
- Program Solution
- Encode Instance as Data
- Apply Program to Data
- Debug Software

MAINTAIN
- Requirements
- Design
- Code
- Test

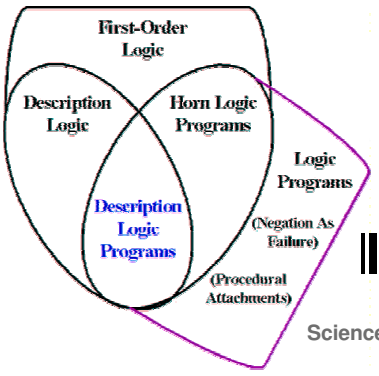
© 2004, Network Inference, Inc. All rights reserved.
network inference

Survey of Approaches:
Inference-Driven Systems in Business

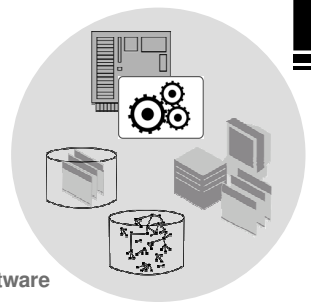
Inference-driven systems employ rules and logics to help solve business problems.



Business



Science

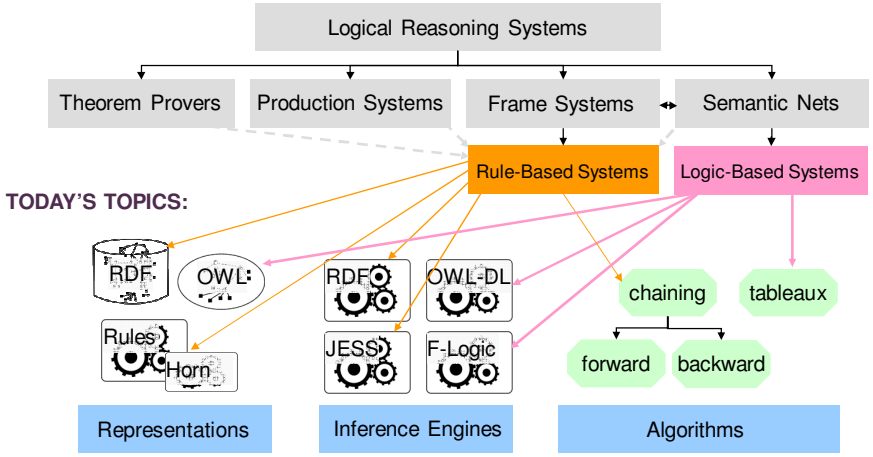


Software

© 2004, Network Inference, Inc. All rights reserved.
network inference

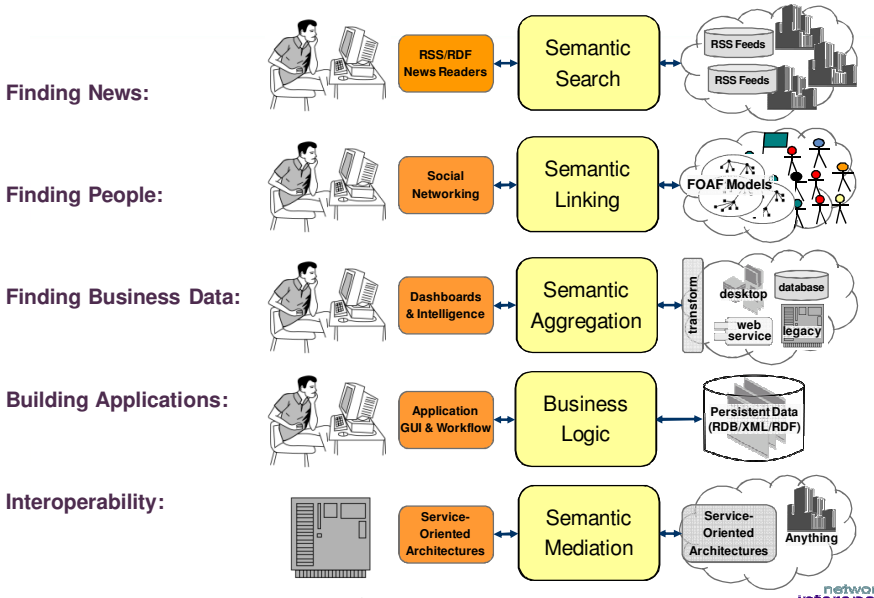
Survey of Approaches:
(over) Simplified Ecosystem of Modern AI Approaches

THE UNIVERSE OF REASONING SYSTEMS:



©2004, Network Inference, Inc. All rights reserved.
network inference

Survey of Approaches:
What are modern AI systems used for?



©2004, Network Inference, Inc. All rights reserved.
network inference

Survey: Rule-Based Systems
Frame-Logic (F-Logic)

F-LOGIC is very OO, based on frames and slots
- Frames are a concept or instance; Slots are binary relations between frames

F-LOGIC is a non-classical logic with different core approaches
- FLORA-2 is an actual programming languages, akin to prolog
- FLORID is a query language, reliant on procedural languages (C++, etc.)
- Ontobroker, a commercial implementation, follows the FLORID pattern

F-LOGIC has many inherent advantages
- Actual OO programming environment, as opposed to engine-driven
- Rule (inference) engine finds efficient path of execution
- Reasoning about the effects of actions, it's "temporal-aware"
- Complete chain-based reasoning – when Horn clauses are used

F-LOGIC also has known limitations
- Complicated due to mixing of representation, programming and reasoning
- Unintuitive for application programmers to shift mindset and syntax
- Natively, its behavior is to treat negation as failure
- Application behavior and knowledge representation are mixed

© 2004, Network Inference, Inc. All rights reserved.

Survey: Rule-Based Systems
Resource Description Framework (as a KR) Overview

RDF is a highly efficient directed graph representation
- Effective at specifying basic semantics in ways that XML cannot
- Basic structure is a "triple" (Object -> Predicate -> Subject)

RDF has a rich history
- 1995, Ramanathan Guha creates "Meta-Content Framework" at Apple
- Guha and Tim Bray, at Netscape now, work to standardize at W3C
- W3C, in 1998, starts the standardization effort for general metadata spec

RDF has a number of powerful advantages
- Easy to optimize in databases (relational or native)
- Simple to annotate and classify knowledge from any source
- Reification is a powerful way to abstract triples as objects or subjects
- Data schemas need not be specified in advance

RDF also has known limitations
- It is not object-oriented, or simple to model real-world concepts with
- Semantic model is weak, especially in comparison to OWL
- Unrestricted graph modeling, unsound, incomplete reasoning

© 2004, Network Inference, Inc. All rights reserved.

Survey: Rule-Based Systems
Chain-Based Inference Techniques

forward chaining (data-driven, deductive logic)

- reasoning from facts to the conclusion
- when facts are available, they are used to match rule antecedents
- a rule can be activated if all parts of the antecedent are satisfied
- often used for real-time expert systems in monitoring and control
- examples: Jena, Intellidimension, JESS

backward chaining (goal-driven, inductive logic)

- starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
- often used in diagnostic and consultation systems
- examples: Jena, MetaLog

inference checking

- Given some antecedent expressions and a consequent expression, determine whether or not the inference rule can deduce the consequent from the antecedents

©2004, Network Inference, Inc. All rights reserved.

Survey: Logic-Based Systems
Description Logics Overview (OWL-DL flavor)

Description Logics are based on concepts and roles

Concepts (classes) are interpreted as sets of objects
Roles (relationships) are interpreted as binary relations on objects

DLs have a rich, deep history

Descendants of KL-ONE and Semantic Networks
Also known as terminological logics, concept languages etc.
Closely related to propositional modal logics

DLs have significant advantages over pure rules systems

Object-oriented modeling characteristic is highly favorable to users
Easier to model the actual real-world (as opposed to rules)
Sound, complete, provable and decidable reasoning is possible
Tableaux algorithm ensures same/same results across implementations
Rules may be expressed as axioms (facts) in concept models
Efficient, highly optimizable reasoning on concepts

DLs also have known limitations

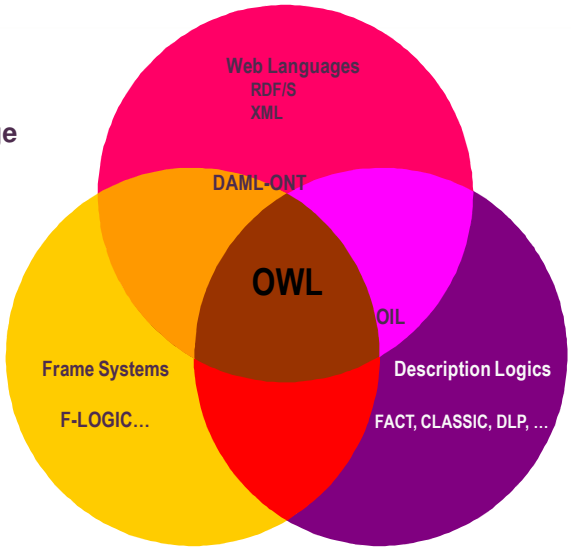
They don't elegantly handle traditional rules well (If…Then…Else)
They don't scale well for instance data due to very high deductive power

©2004, Network Inference, Inc. All rights reserved.

Logic-Based Systems
OWL, Logical Foundations

- Extends vocabulary of XML and RDF/S
- Rich ontology representation language
- Language features chosen for efficient implementations

- Evolved from joint EU and US efforts (OIL & DAML) – DAML was a DARPA project.
- Originally intended to specialize on a knowledge system for intelligent agents



© 2004, Network Inference, Inc. All rights reserved.
network inference

Survey: Logic-Based Systems
Core Inference Algorithm: *Can I Get a Witness*?

THE **TABLEAUX ALGORITHM** IS USED TO REASON

- **Try to build a model (witness) or concept C**
- **Model represented by tree T**
  - Nodes in T correspond to individuals in a model
  - Nodes labeled with sets of subconcepts of C
  - Edges labeled with role names in C
- **Start from root node labeled {C}**
- **Apply expansion rules to node labels until:**
  - Rules correspond with language constructs
  - Expansion completed (tree represents valid model)
  - Contradictions prove there is no model

© 2004, Network Inference, Inc. All rights reserved.
network inference

Survey: Reasoning & Inference
Some Interesting Hybrid Approaches

Rules (not tableaux) to do Description Logics

Some systems have been implemented that use rule chaining mechanisms to reason with DL's

HP Jena & IBM SNOBASE

Both offer a hybrid reasoning service that is "pluggable" for different reasoning types

F-Logic – sort of

Difficult to classify because it uses logics at core, and also rule chaining mechanisms for reasoning

Layered Rules over Representation

"Complete" system of systems will no doubt use all of these approaches – many architectures are currently envisioned as a pure (undecidable) rules layer on top of a (decidable) logic layer. (W3C Layer Cake)

©2004, Network Inference, Inc. All rights reserved.

Survey: Reasoning & Inference
Other Mentionables

SWRL – Semantic Web Rule Language

Axiom-based approach to rules

Simplistically, to add Horn logic to OWL

Based on OWL XML syntax

Includes mapping to RDF graphs

Rule asserting that wherever an Airport is located, there is some map:Location that is the sameLocation as the Airport's location and that is the location of a map Point:

```
ObjectProperty(map:location (inverseOf map:isLocationOf))

Implies(Antecedent(airports:Airport(I-variable(airport))
                   location(I-variable(airport) I-variable(loc))
                       latitude(I-variable(loc) I-variable(lat))
                       longitude(I-variable(loc) I-variable(lon)))
          Consequent(restriction
                     (sameLocation someValuesFrom
                       (intersectionOf
                         (map:Location restriction
                           (map:isLocationOf someValuesFrom
                             (map:Point)))))))
```

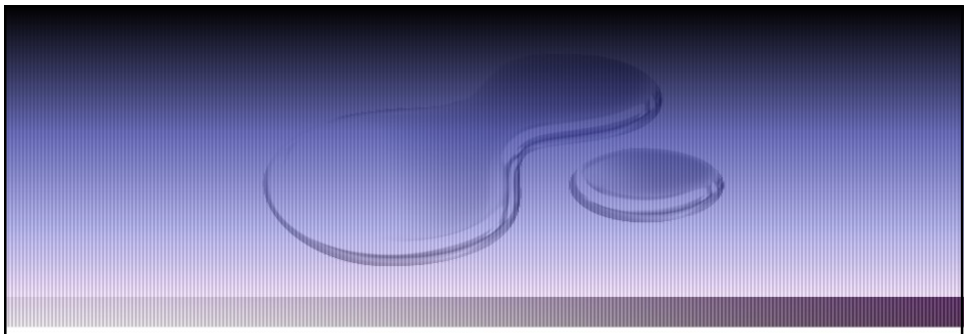
Note use of complex OWL description as predicate in rule; use of OWL:someValuesFrom to assert existence of map location

©2004, Network Inference, Inc. All rights reserved.

Survey: Inference-Driven Systems
Summary of Selected Vendor/Tool Capabilities & Notes

Vendor/Tool	Core Reasoning	Platform & Representation	Notes
JESS	Forward-Chaining	Java platform, non-standard fact representation	Fun to play with, hard to build apps with
Ontobroker	Forward and Backward Chaining	Frame-Logic, non-standard fact representation	The only commercial F-Logic implementation
FLORA-2	Forward and Backward Chaining	Frame-Logic, programmatic execution environment (mix of behavior and data)	Requires adopting a new programming paradigm
Tucana	Uses Jena	Native RDF triples store, J2EE based and app server deployable	Massively scaleable and federated
HP Jena	Forward and Backward Chaining	Java platform, RDF and OWL-Lite supported native	Free, lot's of functionality, hybrid backplane, but slow
Intellidimension	Forward Chaining	Windows platform based, native RDF and OWL-Lite support	Inexpensive, get a lot of functionality for the price
RACER	Tableaux	Lisp-based, support for OWL-Lite, OWL-DL and most of OWL-Full	Great freeware program
Network Inference	Tableaux	Java and C++ based, support for OWL-Lite, OWL-DL and most of OWL-Full	Commercially-proven, scalable, comes with additional infrastructure services

© 2004, Network Inference, Inc. All rights reserved.
network inference



SOME KEY USE CASES:

Collaborative HealthWare Portal
Battlespace Awareness Desktop
Financial Reporting Application
Automobile Configuration Utility

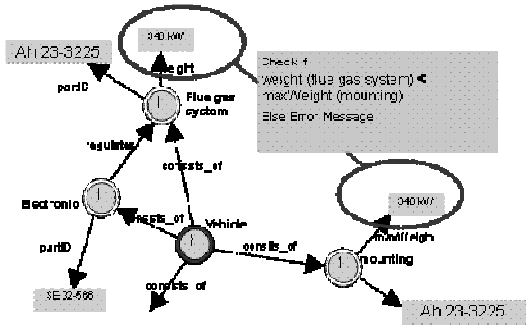
© 2004, Network Inference, Inc. All rights reserved.
network inference

Use Case

Vehicle Configurator – Global Automotive Supplier

Key Technologies:

- F-Logic & Chain-Based Reasoning, C++



Auto manufacturing engineers design, build and test new prototypes as part of the innovation process. The faster this cycle can be completed, the greater the number of innovations that can be brought to market, and the sooner.

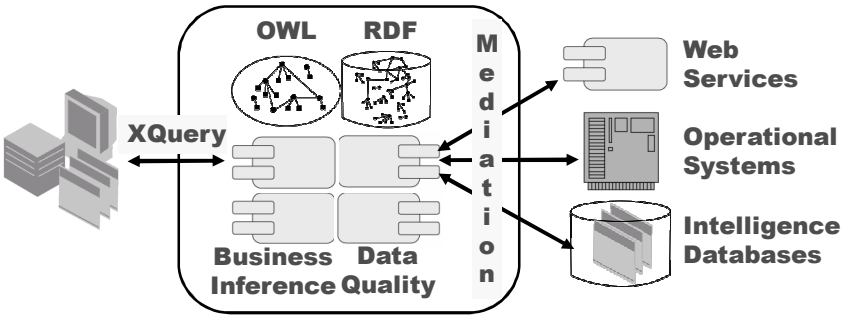
©2004, Network Inference, Inc. All rights reserved.

Use Case

Battlespace Awareness – NATO Military Force

Key Technologies:

- OWL, Description Logics, RDF, Tableaux, XQuery, SOAP/WSDL, Relational Databases, Sensors

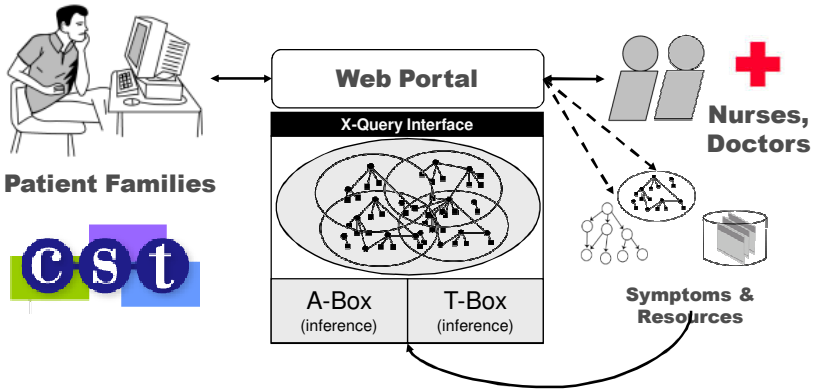


©2004, Network Inference, Inc. All rights reserved.

Use Case
HealthWare Portal – HealthCare Startup

Key Technologies:

- OWL, Description Logics, Tableaux Reasoning, XQuery, Java J2EE (JSP & Servlet)

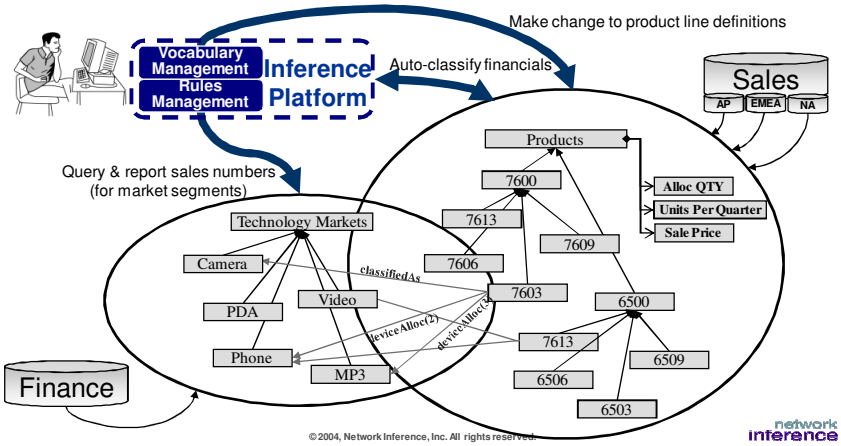


© 2004, Network Inference, Inc. All rights reserved.

Use Case
Financial Reporting – Fortune 100 Electronics

Key Technologies:

- OWL, Description Logics, RDF, Tableaux, XQuery, Workflow, Java J2EE (JSP & Servlet)



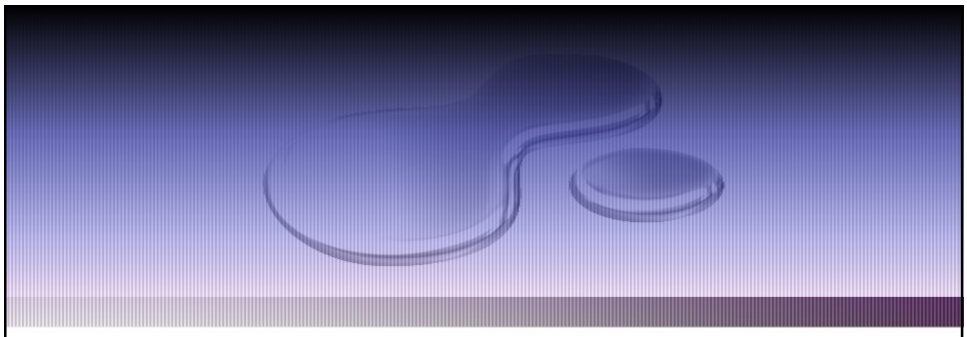
© 2004, Network Inference, Inc. All rights reserved.

More Use Cases
Brief Summary of Some Other Known Systems

More Use Cases:

- **Risk Management**
  - Linking hospital insurance providers with common vocabularies and terminology – database schema mediation via OWL.
- **Security Authorizations**
  - OWL-based security permissions, fine-grained and dynamic controls over access (in place of LDAP)
- **Policy Management**
  - Network and hardware configuration management
- **Vocabulary Management**
  - Business terminology unification and alignment based on OWL inference (reclassification) of terms. Used to unite website and internal systems

©2004, Network Inference, Inc. All rights reserved.
network inference



ENTERPRISE ARCHITECTURES:

Layered Approaches
Entailment Awareness
Legacy Data Integration
User Interfaces

©2004, Network Inference, Inc. All rights reserved.
network inference

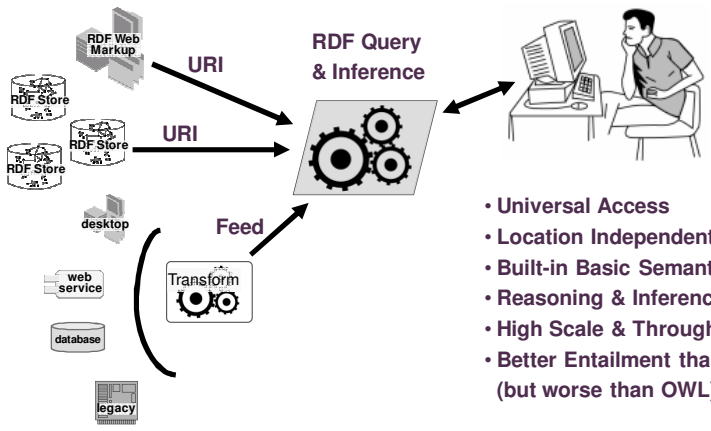
What are Semantic Conflicts?

Data Type	Different primitive or abstract types for same information
Labeling	Synonyms/antonyms have different text labels
Aggregation Structure Cardinality	Different conceptions about the relationships among concepts in similar data sets. Collections or constraints have been modeled differently for same information
Generalization	Different abstractions are used to model same domain
Value Representation	Different choices are made about what concepts are made explicit
Impedance Mismatch	Fundamentally different data representations are used
Naming	Synonyms/antonyms exist in same/similar concept instance values
Scaling and Unit	Different units of measures with incompatible scales
Confounding	Similar concepts with different definitions
Domain	Fundamental incompatibilities in underlying domains
Integrity	Disparity among the integrity constraints

©2004, Network Inference, Inc. All rights reserved.
network inference

The Power of RDF on Networked Data

Dynamic repurposing of distributed information on demand



- Universal Access
- Location Independent
- Built-in Basic Semantics
- Reasoning & Inference
- High Scale & Throughput
- Better Entailment than SQL (but worse than OWL)

©2004, Network Inference, Inc. All rights reserved.
network inference

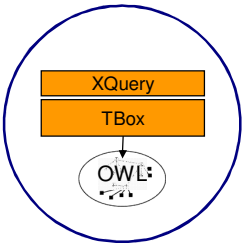
Some OWL Deployment Architectures

OWL Inference Real IT Systems

THREE MAIN USES OF OWL INFERENCE

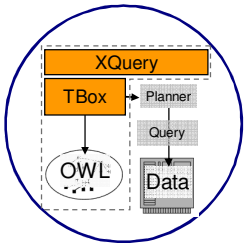
CONCEPT ONLY

- Reason on classes
- "TBox" = class only
- No instances/individuals
- Valuable for building logics & rules that can be used to drive application behavior



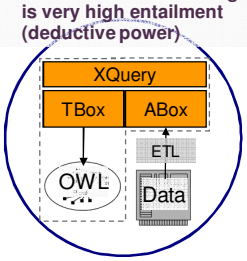
CONCEPT-TO-INSTANCE

- Reason on concepts first
- Then "get" data that belongs to concepts
- Instances come from native source (RDF, RDBMS, WSDL, XMLDB)
- Valuable for accessing native data at high speed and volume



INSTANCE REASONING

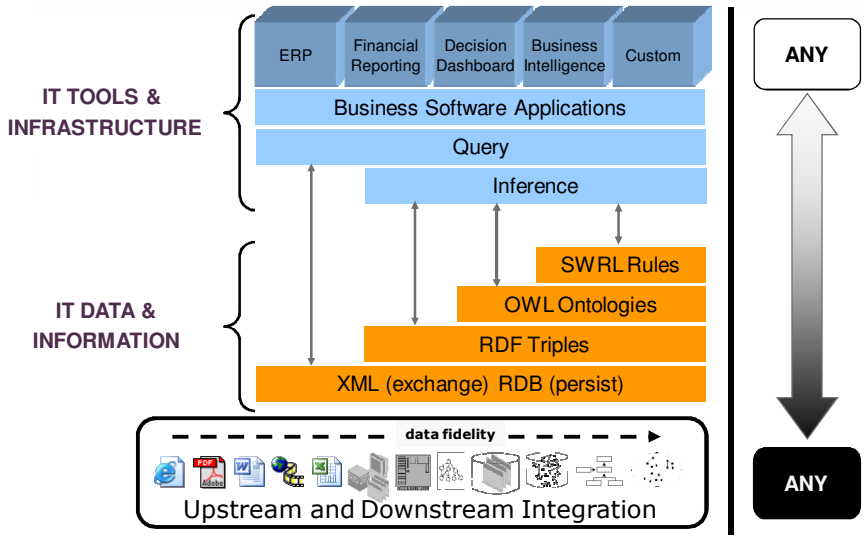
- Concepts and instances
- "TBox" and "ABox"
- Instances may be in active memory, or in persisted memory (DB)
- Valuable for reasoning on real data values
- OWL instance reasoning is very high entailment (deductive power)



©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures

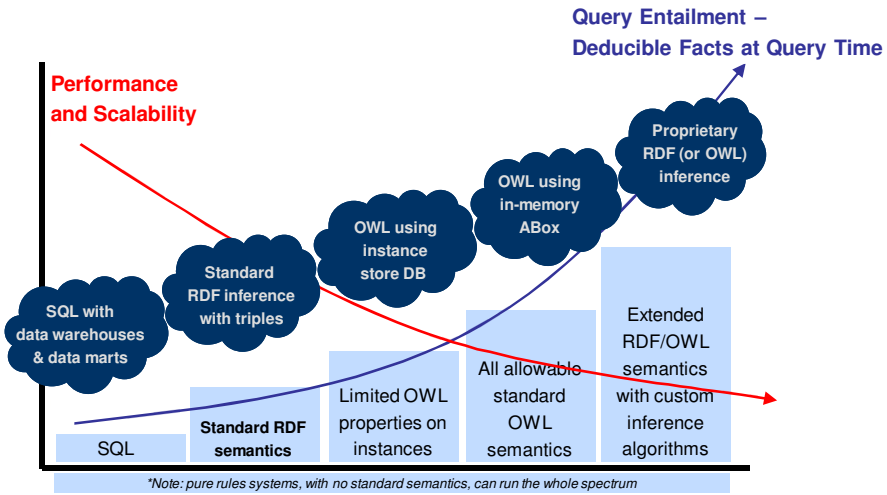
Best Practice: Layer Your Architectures!



©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
Choose Your (deductive) Weapons Wisely

All things being equal, a query with greater entailment will always be less performant than the same query with less entailment.



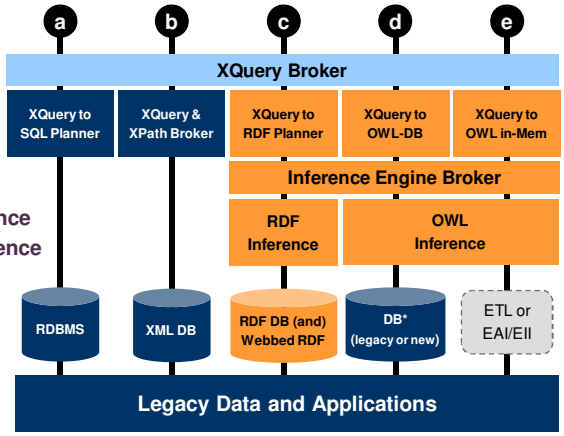
©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
Choose a Pluggable Backbone with a Common Façade

Any IT application ought to be able to get to any data from a common API...

```
Query.getConnection(name,type{RDB,XML,RDF,OWL})
```

KEY:
- a) RDB query only
- b) XML query only
- c) RDF query + inference
- d) OWL ABox in DB inference
- e) OWL ABox inMem inference



©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
Data Integration and the need for Downstream Heavy-Lifting

BUSINESSES CANNOT "RIP AND REPLACE" EXISTING SYSTEMS…

DRIVING THE NEED FOR TRADITIONAL:

- Federated query
  - Relational algebra to do federated heterogeneous joins across instances (Composite Software, MetaMatrix, DB2ii)
- Transformation (EDI, Copybooks, Card Systems)
  - Pragmatic need to convert data of all types to "higher" knowledge representation formats
- Automated model & context building
  - Metadata extraction from legacy sources, ontology "seeding," strong methodologies needed

©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
Addressing the Average, Every-Day Business User

SEX SELLS!!

DRIVING THE NEED FOR GOOD USER INTERFACES

- Normal users editing ontology
  - "Average" business users need to be able to create ontologies without knowing that they are doing so
- Creating applications with dynamic UI
  - Software applications that can "display" the power of inference by demonstrating usability improvements will create the most attention
- Attention to business, non-technical users
  - Generally speaking, AI and Semantic Web technologies need to pay more attention to regular users (killer app?)

©2004, Network Inference, Inc. All rights reserved.
network inference

ENTERPRISE ARCHITECTURES:

Two "Mega-Trends"
(1) Model-Driven SOA
(2) Model-Driven App Servers

© 2004, Network Inference, Inc. All rights reserved.

network inference

Evolution of Integration
Ad Hoc, Point-to-Point Integration



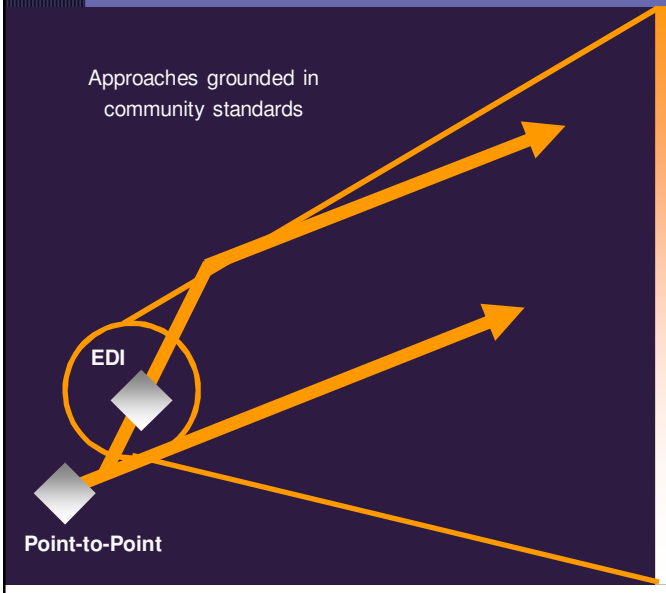
Main Improvements:
- Application connectivity is possible
- Data sharing need not be "swivel-chair"
- Some automation

Key Problems:
- No common protocols (at network)
- No common interfaces
- No tools for development
- Few tools for runtime workflow or process automation
- Too many interfaces to scale well
- $$$$$

© 2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration

Rise of Standards: *Electronic Data Interchange*



Main Improvements:

- Common vocabulary
- Standard protocols

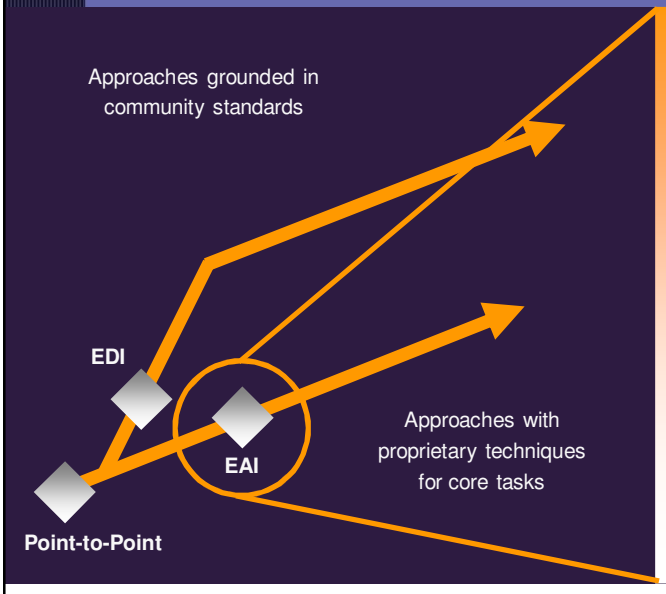
Key Problems:

- Too much time to develop
- Too much time to change
- Fragmented vertical adoption
- Tight-coupling – inflexible
- No machine accessible semantics
- **$$$$$**

©2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration

First COTS: *Enterprise Application Integration (EAI)*



Main Improvements:

- Eliminate N^2 problem (for the plumbing)
- Extra "management" layers
- Support for different transaction models (pub/sub & req/rep)

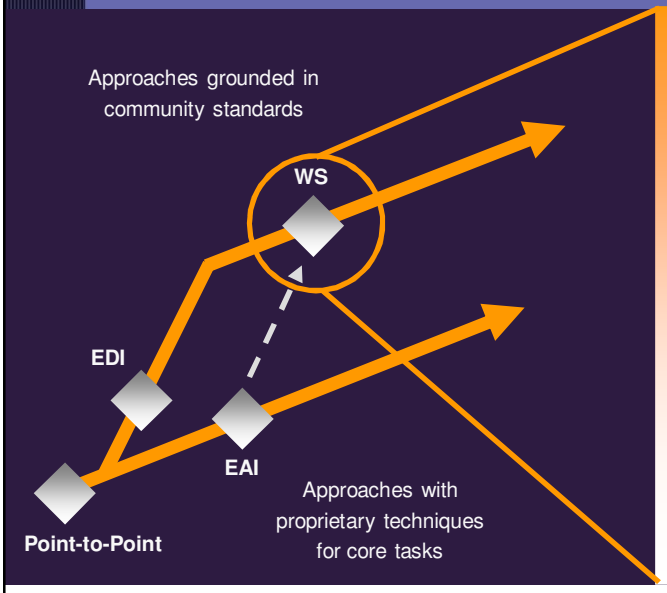
Key Problems:

- Vendor lock-in
- Still no common, standard protocols (at network)
- Proprietary data/info management
- N^2 problem at data layer
- Weak business process
- **$$$$**

©2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration

Standards Finally Catch Up: *Web Services*



Main Improvements:

- No vendor lock-in
- Standard APIs/Services
- Loose-coupled "pipes"
- Service "paradigm" (vs. object or RPC)

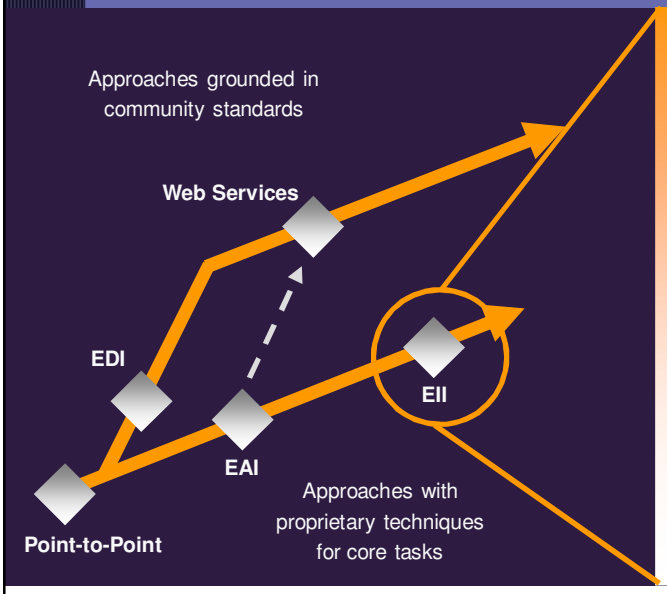
Key Problems:

- Difficult "management"
  - fewer tools
  - evolving vision
- N^2 problem at data layer
  - XSL/T & Java Xform
- Too much hype
  - not dynamic discovery
  - not "process aware"
- **$$$**

©2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration

Information Engineering: *Enterprise Information Integration (EII)*



Main Improvements:

- Focus on core business value = information
- "Virtual" data views
- N^2 data problem solved
- Meta-data management
- Loose-coupled data

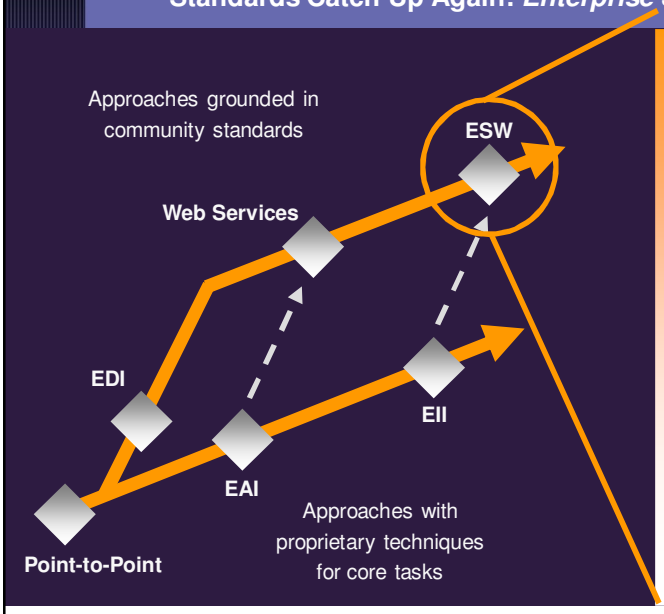
Key Problems:

- Vendor lock-in
- Proprietary metadata
- Fragmented approaches:
  - Relational
  - XML
  - Object
- Minimal automation
- **$$$**

©2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration

Standards Catch Up Again: *Enterprise Semantic Web (ESW)*



Main Improvements:

- Standardized metadata
- New "smart" capabilities
- Much lower TCO
- Focus on core business value = information
- "Virtual" data views
- N² data problem solved
- Meta-data management
- Loose-coupled data
- Loose-coupled "pipes"
- Standard APIs/Services
- SOA Architecture

Key Problems:

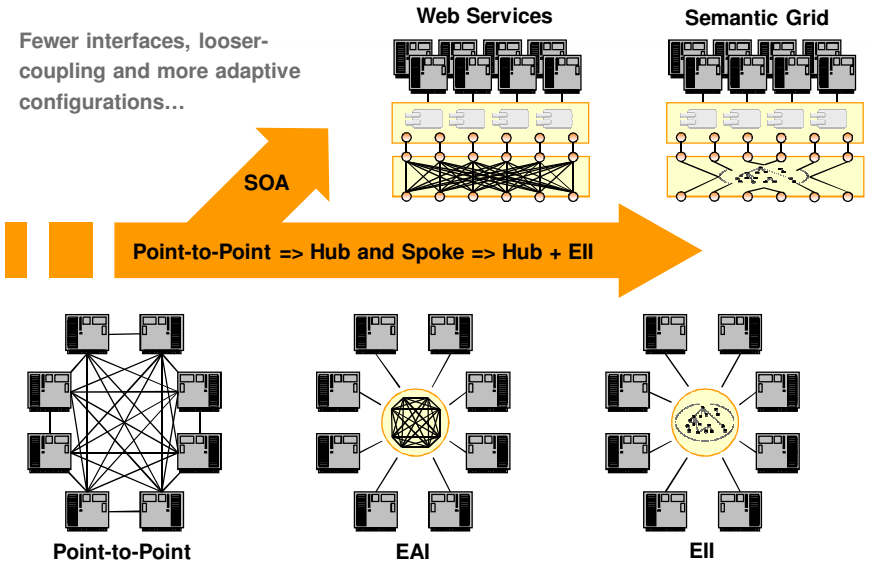
- Emerging vendor support
- Use cases still evolving
- Up-front modeling costs
- $$

©2004, Network Inference, Inc. All rights reserved.
network inference

Evolution of Integration Architectures

Steady Progression Towards Adaptive Capabilities

Fewer interfaces, looser-coupling and more adaptive configurations…



©2004, Network Inference, Inc. All rights reserved.
network inference

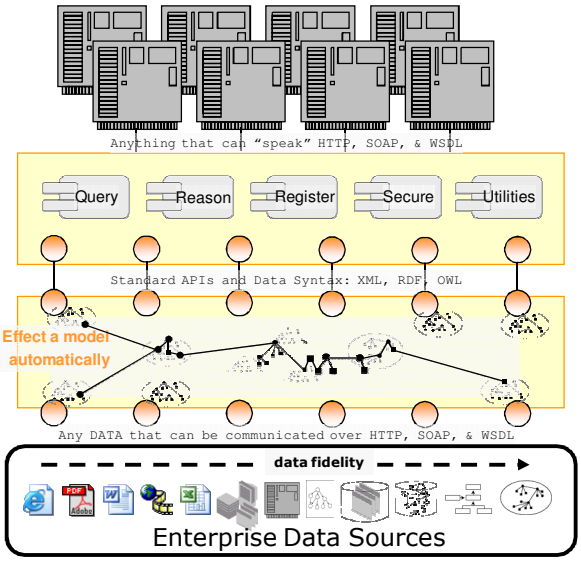
Enterprise Architectures

Incorporate Reasoning Into SOA's to Effect a Common Data Model

Component driven Service-Oriented Architectures, using Web Services, are ripe for inference-driven systems.

Problems agreeing to vocabularies are political and cultural in nature.

Reasoning systems (using RDF and OWL) provide a way around the problem by allowing each environment to have local view – while the overall system can have many common views.



©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures

Re-envisioning the Application Server, Part 1…

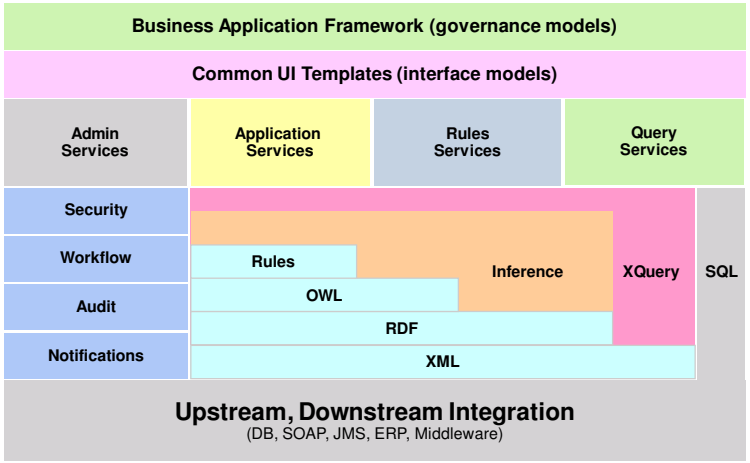
Today's application server centric world drives applications with CODE, the following example is a typical J2EE example, but .NET is very similar...

Business Application Framework (Custom Coded)			
UI Templates (JSP & Servlet Architecture)			
Utility Services	**Administrative Services (Custom)**	**Application Services (Custom)**	**J2EE Container Services**
Security	EJB Session Beans		
Workflow			
Audit	EJB Entity Beans		
Notifications	Java Data Objects and Connection Architecture		
Upstream, Downstream Integration (DB, SOAP, JMS, ERP, Middleware)			

©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
…Tomorrow's Application Servers will be Model-Driven

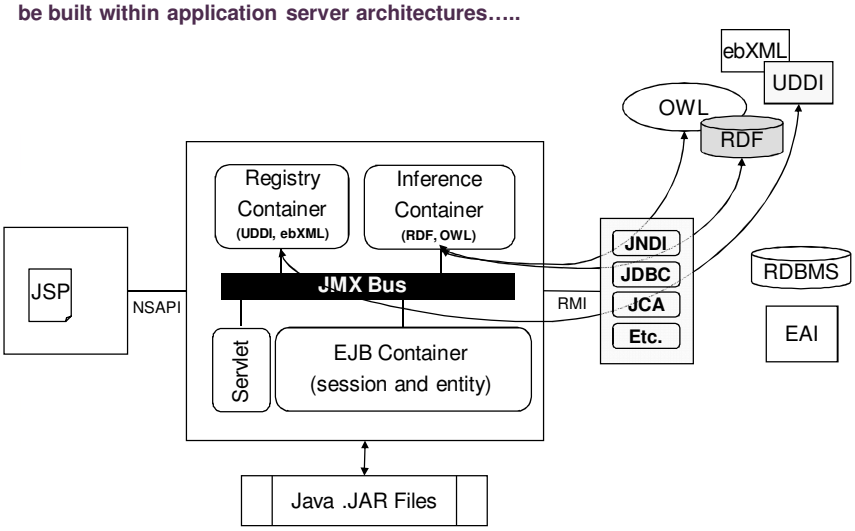
Tomorrow's world will be model-centric, driven by schemas, ontologies, and rule lists – enabling applications to change at the pace of business without having to re-code, re-compile, and re-deploy when change happens...



©2004, Network Inference, Inc. All rights reserved.
network inference

Enterprise Architectures
…More Simplistically Put

Lookup and Inference "containers" will be built within application server architectures…..



©2004, Network Inference, Inc. All rights reserved.
network inference

Data Semantics Call to Action!

SEMANTICS WANT TO BE MODEL-DRIVEN, NOT CODED

Custom written software programs capture semantics in a tightly-coupled way, software models are a loosely-coupled alternative.

SEMANTICS ARE LOCAL, NOT GLOBAL

The meaning of anything, especially data, must be taken in context – and context is always local to a specific user or community.

SEMANTICS SCALE BEST WHEN FEDERATED

Centralized meanings limit adaptability and stifle adoption of new innovations, a flexible framework of meanings with multiple small centers is better.

SEMANTICS EVOLVE IN REALTIME

Static data models can't keep pace with business change; a software infrastructure that leverages dynamic and evolutionary data models is superior.

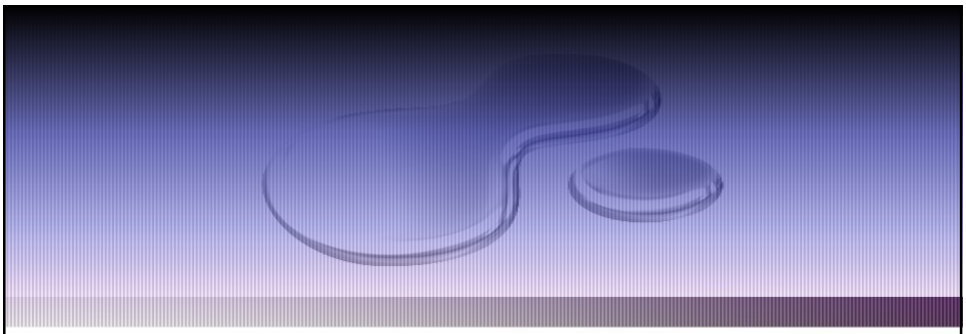
SEMANTICS ALWAYS OCCUR AT THE EDGES

Data meanings are most important when the edges of two "things" (human eyeballs to a monitor or one software layer to another) meet and exchange information.

SEMANTICS NEED NOT BE "STANDARDIZED"

Standard, shared vocabularies can artificially constrain the evolution of data communications because meaning will change over time and in new contexts.

©2004, Network Inference, Inc. All rights reserved.
network inference



HOLISTIC VIEWPOINT:

SOME CONCLUSIONS

©2004, Network Inference, Inc. All rights reserved.
network inference

We Have Not Yet Arrived
Will We Ever?

ALL THIS ARTIFICIAL INTELLIGENCE STUFF ADDRESSES DEEP, SYSTEMIC, AND HIGHLY COMPLEX PROBLEMS…

WHICH ARE NOT USUALLY AT THE TOP OF YOUR AVERAGE BUSINESS EXECUTIVE'S MIND!

BUT THEN AGAIN, ISN'T THAT WHY UNIVERSITIES DO RESEARCH?

©2004, Network Inference, Inc. All rights reserved.
network inference

We Have Not Yet Arrived
Many Industries are in Need (But Not All)

"Imperfect interoperability [of digital data] imposes at least $1 billion per year on the members of the U.S. automotive supply chain. By far the greatest component of these costs is the resources devoted to repairing or reentering data files that are not usable for downstream applications."

– RTI / NIST Report, March 1999 ["Interoperability Cost Analysis of the U.S. Automotive Supply Chain"]

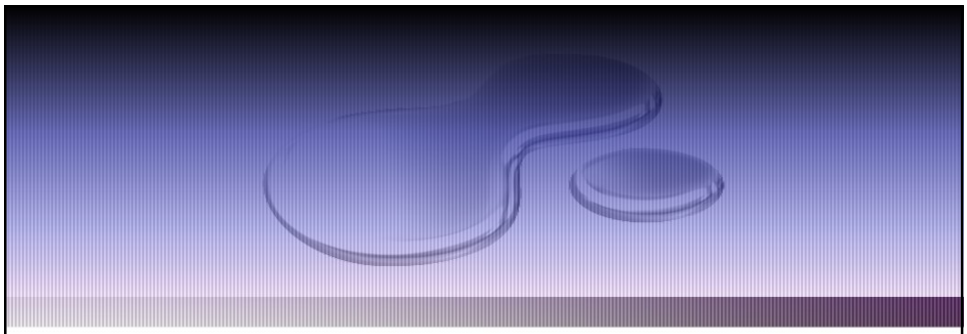
- **Pain Killers?**
  Oftentimes, there is a short-term need that predicates a fast, "stop the bleeding," type approach. Usually custom, one-off solutions are chosen because they are known quantities – at the cost of significant long-term issues like TCO and inflexibility.
- **Penicillin?**
  Certain industries are experiencing decades long systemic pain that software "pain-killers" can no longer help with. These industries are in need of solutions that solve the root of the problem – not the surface "bleeding."
  - Defense Industry
    - Network-Centric Enterprise Services (NCES)
    - Department of Homeland Security (DHS)
  - Manufacturing
    - Automotive Supply Chain
    - Electronics & CPG Supply Chain
  - Life Sciences
    - Process Automation / Risk Analysis

©2004, Network Inference, Inc. All rights reserved.
network inference

Research Challenges
Technologies and Approaches we Could Use NOW!

- **Scaling**
  - Inference for large taxonomy
  - Inference for large amounts of instance data
- **Simplicity**
  - Ease-of-use factors (intuitive ontology creation methodologies – natural language??)
- **Ontology Seeding**
  - Auto metadata extraction
- **"Standard" Methodologies**
  - Information engineering, ontology engineering
- **Meaningful Examples to Commercial Space**
  - Look beyond R&D funding…(is this possible?)

© 2004, Network Inference, Inc. All rights reserved.
network inference



Jeff Pollock

network inference

Vice President, Technology

jeff.pollock@networkinference.com

© 2004, Network Inference, Inc. All rights reserved.
network inference