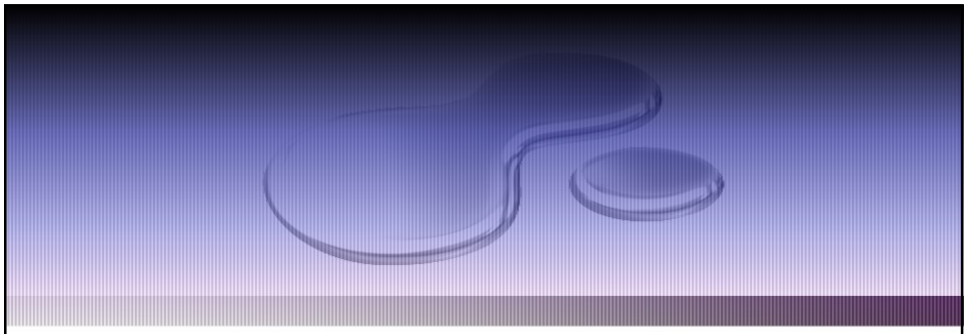


NETWORK INFERENCE, INC

Inference-Driven Business Systems:
A Comparative Look at Emerging Solutions for Managing Change in IT Software Systems

Making Data Meaningful

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

network inference

Agenda

BIG PICTURE
- What are inference-driven systems?
- Why are they important?
- What can they be used for?

MORE DETAILS
- Rules-based approaches
  - Chaining-Algorithms (JESS, F-LOGIC, and RDF)
- Logic-based systems and approaches
  - Graph Matching (Description Logics & OWL)
- Hybrid approaches
  - Rules for deduction, conforming to semantics
- Selected Vendor Summary

INFERENCE AND RULES IN REAL IT ECOSYSTEMS
- Layered Systems – Enterprise Semantic Web
- Do you need it?
- Conclusions

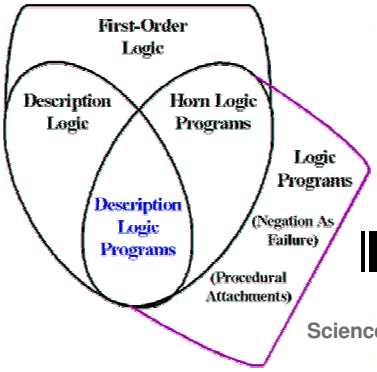
© 2004, Network Inference, Inc. All rights reserved.
network inference

Big Picture:
What are inference-driven systems?

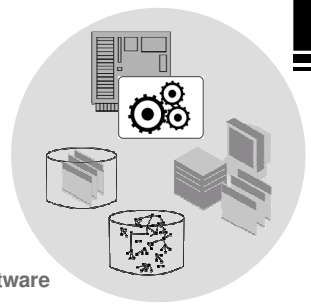
Inference-driven systems employ rules and logics to help solve business problems.



Business



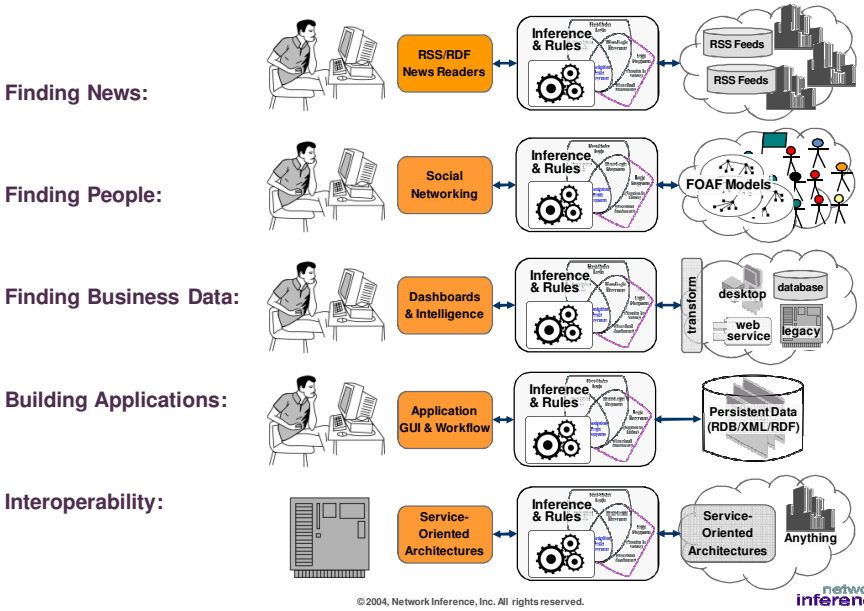
Science



Software

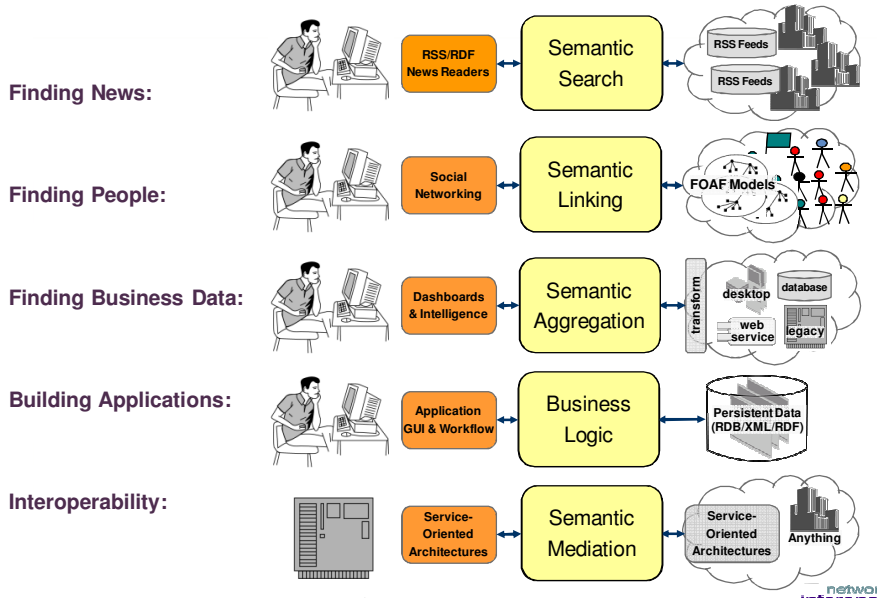
© 2004, Network Inference, Inc. All rights reserved.
network inference

Big Picture:
What can they be used for?



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

Big Picture:
What can they be used for?



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

Big Picture:
Why are they important? ORDER TO CHAOS.

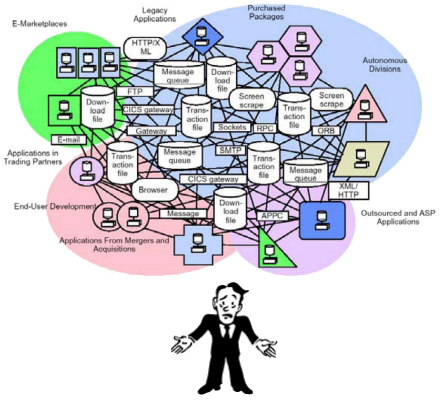
Enterprise data, created over decades, is chaotic, lacking meaning, and increasing at exponential rates.

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 are they 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…

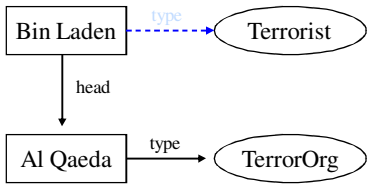
```
<owl:Property rdf:ID="head">
  <rdfs:subPropertyOf
        rdfs:resource="member" />
</owl:Property>

<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.

Big Picture:
Why are they 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.

MORE DETAILS:

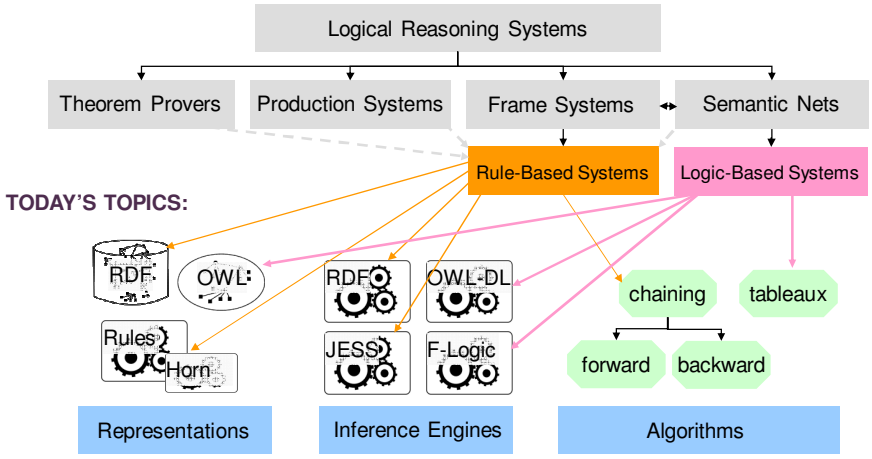
Rules Based Systems
Logic Based Systems
Hybrid Approaches

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

network inference

(over) Simplified Ecosystem

THE UNIVERSE OF REASONING SYSTEMS:



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

network inference

Rule-Based Systems
Java Expert System Shell (JESS) Overview

JESS is based on rules being applied to facts
- A rule can be simple (IF…THEN) or complex (patterns & functions)
- A list is the main data structure for JESS, items in lists are facts

JESS has a rich, deep history (sort of indirectly)
- Java-based descendant of CLIPS, early rules-based AI system
- Rules-based expert systems originated in the 70's & 80's
- Developed by Sandia National Laboratories, late 1990's

JESS has many inherent advantages
- Declarative programming shifts focus from code logic to facts & rules
- Rule (inference) engine finds efficient path of execution
- Pattern matching is easily driven with atomic rules
- Deductive forward chaining reasoning
- Highly efficient and optimized reasoning

JESS also has known limitations
- Large rule systems get very complicated to manage (too many facts)
- Conversion of external data into native JESS facts is intensive

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

Rule-Based Systems
How does JESS Work?

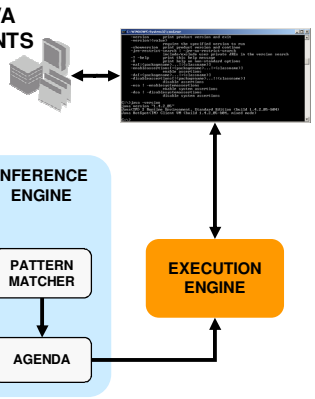
- **Jess matches facts in the fact base to rules in the rule base.**
- **The rules contain function calls that manipulate the fact base and/or other Java code.**
- **Jess uses the Rete (ree-tee) algorithm to match patterns.**
- **Rete network = an interconnected collection of nodes = working memory.**

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

Rule-Based Systems
What is the JESS Architecture?

LISTS

(a b c) ; list of tokens
(1 2 3) ; list of integers
(+ 2 3) ; an expression
("Hello world!") ; a string
(foo ?x ?y) ; a function call



RULES

```
;; An error handler
(defrule report-err
?err <- (is-error (msg ?msg))
=>
(printout t "Error was: " ?msg crlf)
(retract ?err))
```


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

Rule-Based Systems
What are Facts in JESS?

WORKING WITH FACTS IN JESS

- Facts have a head and one or more slots.
- Slots hold data (can be typed).
- Multislots can hold lists.
- You can modify slot values at runtime.
- Facts are constructed from templates.

JESS FACT TYPES

- Ordered – head only.
- Ordered – single slot.
- Unordered – multiple slot, like a database record.
- Shadow – slots correspond to properties of a JavaBean.

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

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.

Rule-Based Systems
Semantics of Frame-Logic (F-Logic)

F-structure $I = (U, <_U, \in_U, I_F, I_{->}, I_{->>}, I_{.->}, I_{.->>}, I_{=>}, I_{=>>})$

U = objects, attributes, classes
$<_U$ = subclass relation (irreflexive partial order)
$\in_U$ = class membership (if $a \in_U b$ and $b \leq_U c$ then $a \in_U c$)
$I_F \approx$ terms (like mary["Mary",ms,…] on last slide) $\rightarrow (U^i \rightarrow U)$

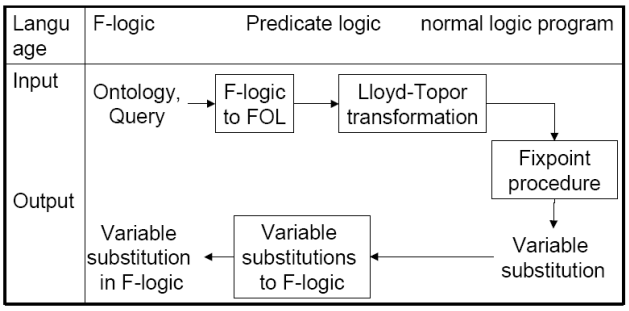
Elements of U are mapped to 4 sets of partial functions:
$I_{->}$/ $I_{->>}$: non-inheritable object attributes/set-valued attrs.
$I_{.->}$/ $I_{.->>}$: inheritable class attributes/set-valued attrs.
$I_{=>}$/$I_{=>>}$: signature of class attrs./set-valued class attrs.
(defining types)

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

Rule-Based Systems
F-Logic and Ontobroker

F-logic	Predicate logic
C1::C2	sub(C_1,C_2)
O: C	isa(O,C)
C_1[A=>>C_2]	att_type(C_1,A,C_2)
O[A->>V]	att_val(O,A,V)
O1<: O2	part_of(O_1,O_2)

Ontoprise's Ontobroker implementation uses F-Logic to accomplish inference on business knowledge. Internally, F-Logic is converted to predicate logic for execution within a regular programming language like C++.



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

Rule-Based Systems
Resource Description Framework 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)

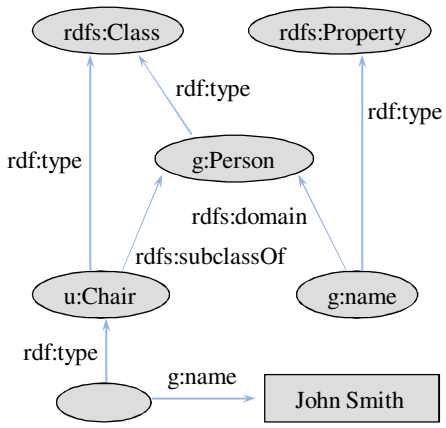
RDF has a rich history
- 1995, Ramanathan Guha creates "Meta-Content Framework" at Apple
- Guha and Tim Bray, at Netscape now, work to standardize at W3C
- W3C, in 1998, starts the standardization effort for general metadata spec

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.
network inference

Rule-Based Systems
RDF Graphs & Syntax




```
<rdfs:Property rdf:ID="name">
   <rdfs:domain rdf:resource="Person">
</rdfs:Property>

<rdfs:Class rdf:ID="Chair">
   <rdfs:subclassOf  rdf:resource=
     "http://schema.org/gen#Person">
</rdfs:Class>

<rdf:RDF
     xmlns:g="http://schema.org/gen"
     xmlns:u="http://schema.org/univ">
  <u:Chair rdf:ID="john">
   <g:name>John Smith</g:name>
  </u:Chair>
</rdf:RDF>
```


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

Rule-Based Systems
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)

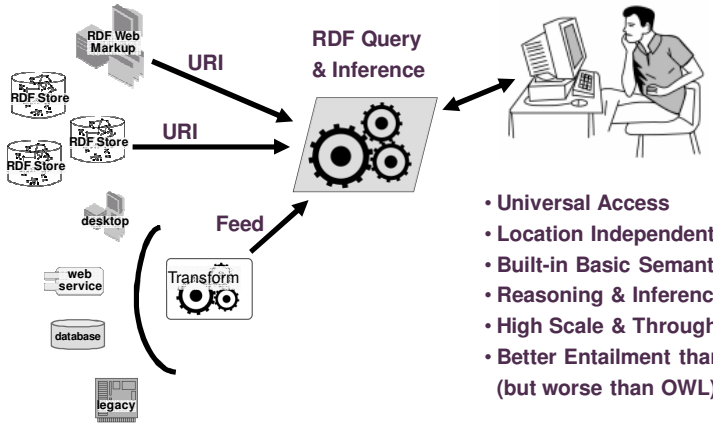
- starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
- often used in diagnostic and consultation systems
- examples: Jena, MetaLog

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.

Rule-Based Systems
Power of RDF Inference on Networked Data



- 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

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

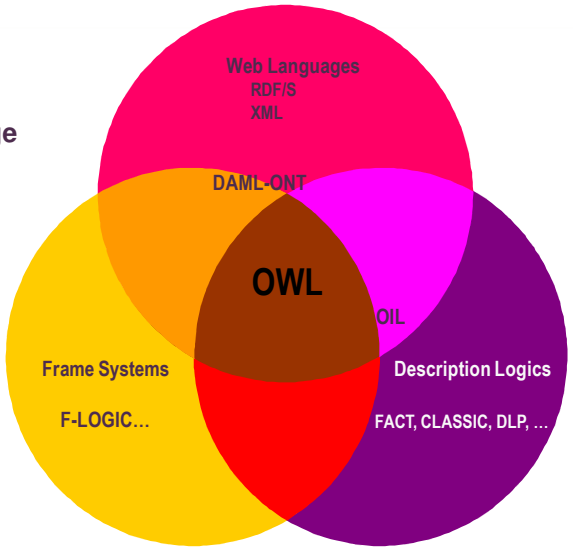
DLs also have known limitations
- They don't elegantly handle traditional rules well (If…Then…Else)
- They don't scale well for instance data due to very high deductive power

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

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

Logic-Based Systems
OWL: Axioms and Class Constructors

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer
complementOf	$\neg C$	¬Male
oneOf	$\{x_1 \ldots x_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	∀hasChild.Doctor
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer
maxCardinality	$\leqslant nP$	⩽1hasChild
minCardinality	$\geqslant nP$	⩾2hasChild

CLASS CONSTRUCTORS

AXIOMS

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human ⊑ Animal ⊓ Biped
equivalentClass	$C_1 \equiv C_2$	Man ≡ Human ⊓ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male ⊑ ¬Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} ≡ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} ⊑ ¬{peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter ⊑ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost ≡ price
inverseOf	$P_1 \equiv P_2^-$	hasChild ≡ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ ⊑ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1P$	⊤ ⊑ ⩽1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1P^-$	⊤ ⊑ ⩽1hasSSN$^-$

network inference

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

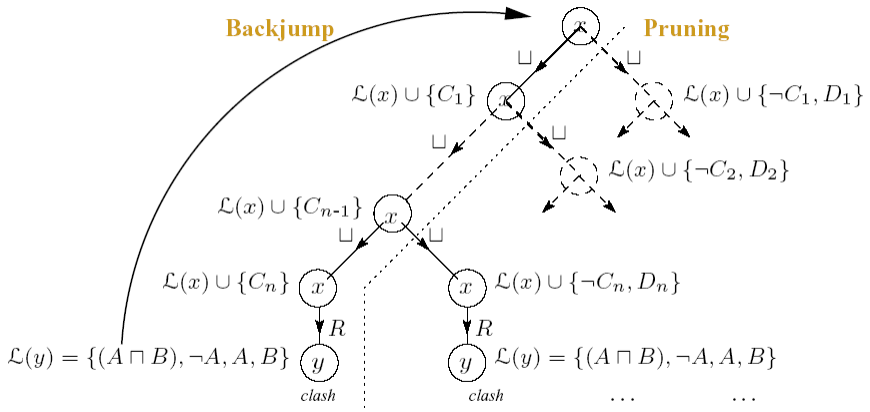
THE **TABLEAUX ALGORITHM** IS USED TO REASON

- **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

Logic-Based Systems
Core DL Inference Efficiencies

BACKJUMPING IS ONE TECHNIQUE TO OPTIMIZE



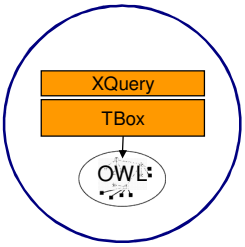
©2004, Network Inference, Inc. All rights reserved.
network inference

Logic-Based Systems
OWL Inference in Plain English

THREE MAIN USES OF OWL INFERENCE

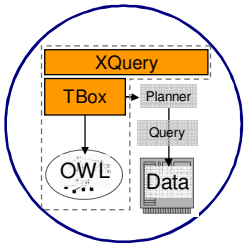
CONCEPT ONLY

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



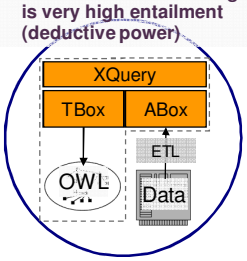
CONCEPT-TO-INSTANCE

- 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

Logic-Based Systems
OWL: Big Advantages for Regular Software

TOP 5 REASONS FOR OWL (ALL SPECIES)



Loose-coupling – semantics may be decoupled from the application code (or parsing algorithms)

Machine-actionable – automated decisions can be made from interpretable inferences

Highly expressive – can capture core elements of EER, UML, and frame-based systems

Precision – language checking available to prevent inconsistent/contradictory model semantics

Fun acronym! – OWL is named for the owl in Winnie the Pooh, who spelled his name WOL

TOP 5 REASONS FOR OWL DESCRIPTION LOGICS

Consistency – query results, across vendor implementations and instances, should be consistent

Performance – Although performance metrics depend on model constructs, OWL-DL supports highly optimized inference algorithms

Predictable – semantics are mathematically decidable within the model, reasoning is finite

Foundational – provides a baseline inside applications for layered semantic models

Reliability – if the answer to a query is implied by any of the model data, it will be found – guaranteed.

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

Reasoning & Inference
Hybrid Approaches

Rules (not tableaux) to do Description Logics

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

Jena & 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.

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)))))))
```

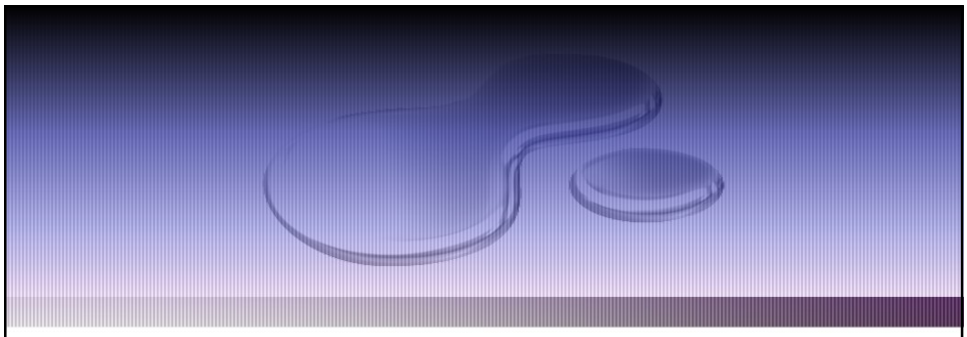
Note use of complex OWL description as predicate in rule; use of OWL:someValuesFrom to assert existence of map location

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

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

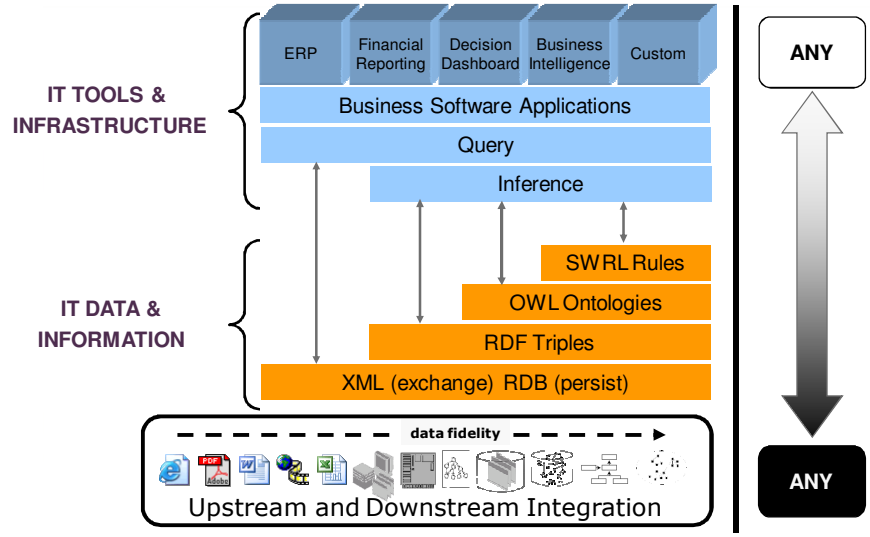


SYSTEMS AND ARCHITECTURES:

HOW DO YOU USE RULES AND INFERENCE INSIDE PRACTICAL IT APPLICATIONS?

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

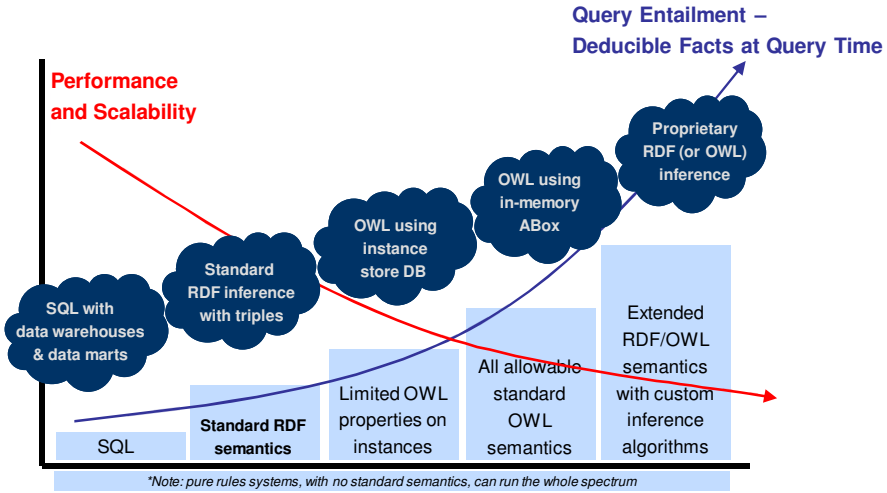
Practical IT Systems
Best Practice: Layer Your Architectures!



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

Practical IT Systems
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.



*Note: pure rules systems, with no standard semantics, can run the whole spectrum

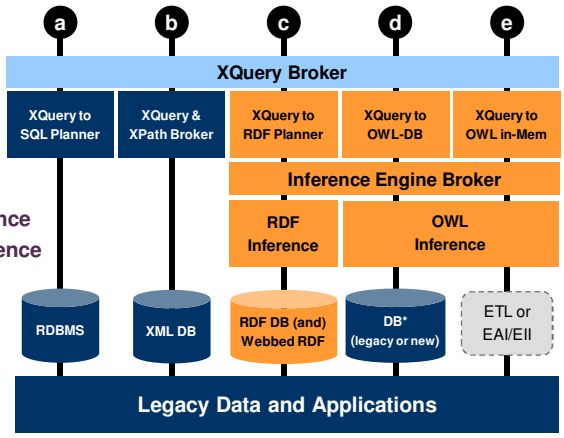
©2004, Network Inference, Inc. All rights reserved.
network inference

Practical IT Systems
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})
```

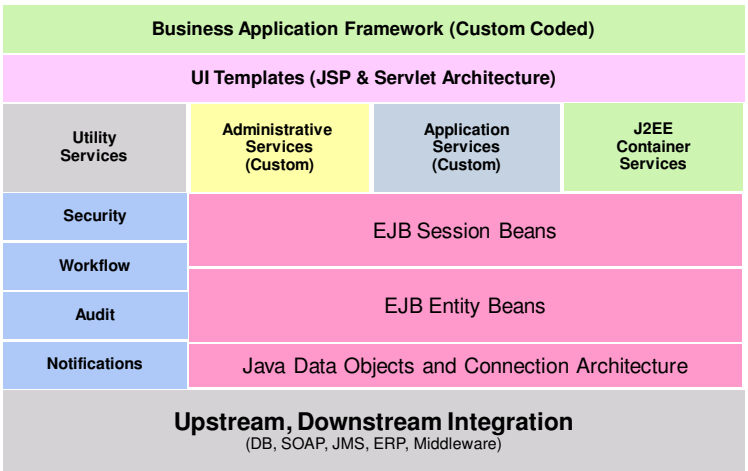
KEY:
- a) RDB query only
- b) XML query only
- c) RDF query + inference
- d) OWL ABox in DB inference
- e) OWL ABox inMem inference



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

Practical IT Systems
Escape from Hard-Coded Applications…

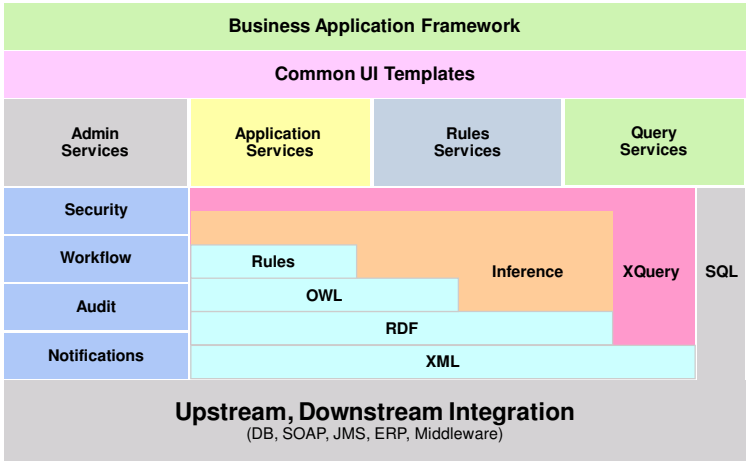
Today's application server centric world drives applications with CODE, the following example is a typical J2EE example, but .NET is very similar...



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

Practical IT Systems
…Build Model-Driven Applications Instead

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...



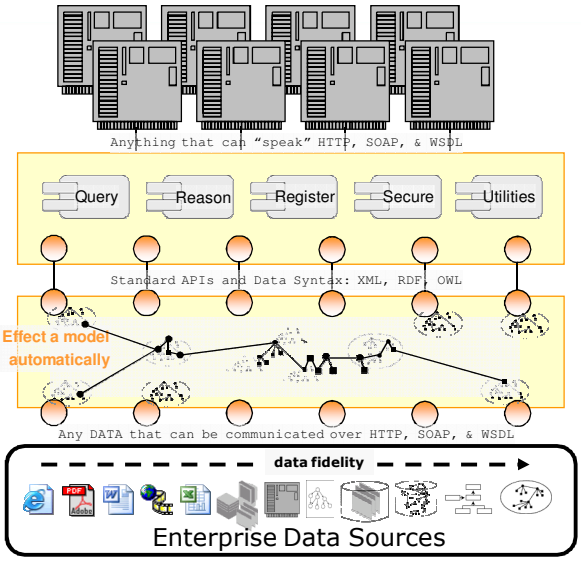
©2004, Network Inference, Inc. All rights reserved.

Practical IT Solutions
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.

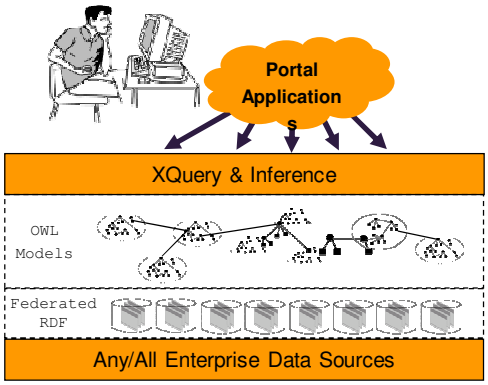


©2004, Network Inference, Inc. All rights reserved.

Practical IT Solutions

Layer Portals on Inference/Rules Systems for Agility

Portals, often seen as a way to tie together disparate business processes and content, can benefit greatly by using an inference and rules driven infrastructure…enabling portals to source, repurpose and manage content in a distributed and federated knowledge space.

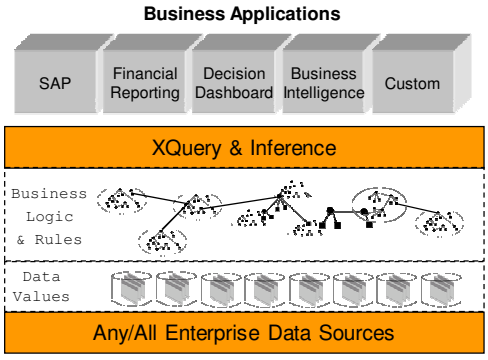


©2004, Network Inference, Inc. All rights reserved.
network inference

Practical IT Solutions

Decouple Your Software from Your Core Information

Business applications, long the culprit of monolithic inflexible application code, can be decoupled from enterprise knowledge – enabling you to plug different applications in and out, or even outsource the software while maintaining control and ownership over business rules, process, logic, and information.

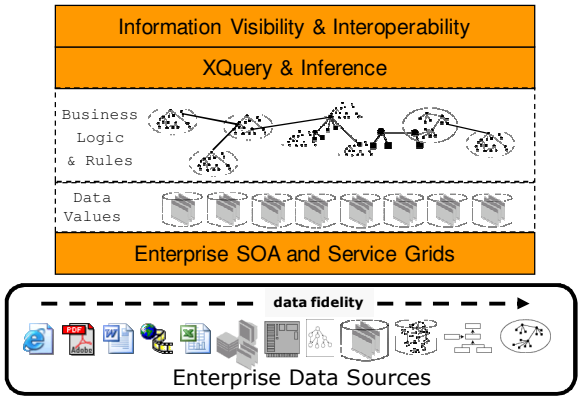


©2004, Network Inference, Inc. All rights reserved.
network inference

Practical IT Solutions

Make "Loose-Coupling" a Requirement – for Pipes and Data

All the hype over SOA is partially justified, a standards-based approach to loose-coupling interfaces is optimal. But SOA approaches usually ignore information or assume common vocabularies. Inference and rules based systems can help you make your data and logic loosely-coupled too.

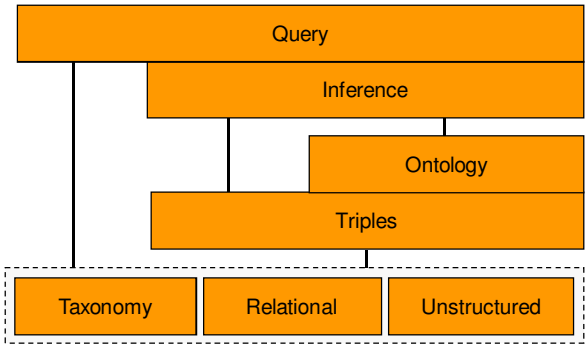


©2004, Network Inference, Inc. All rights reserved.
network inference

Practical IT Solutions

Unify Your Data Semantically, Structurally, and Syntactically

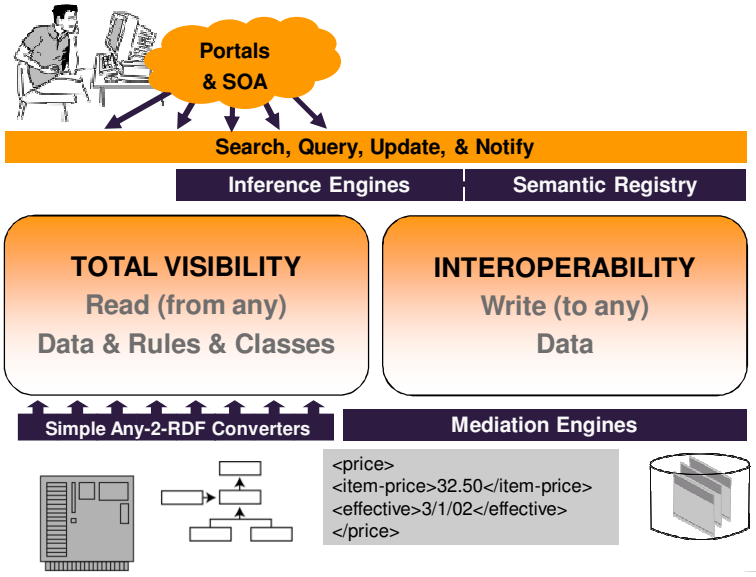
Of all the content inside your organization, there is no single format. Financial data can come in databases, XML documents, spreadsheets or even HTML portal pages. How can you ever make sense of it all? It's easy to do with a model-driven OWL approach and simple any-to-RDF converters.



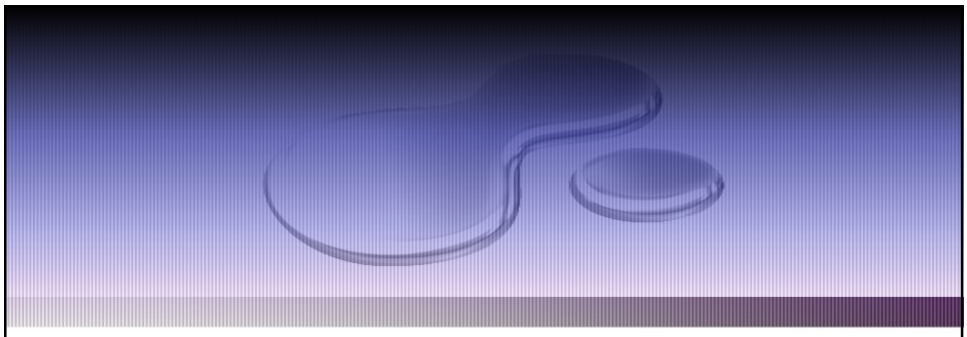
©2004, Network Inference, Inc. All rights reserved.
network inference

Practical IT Solutions

Building a Flexible Application and Middleware Framework



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



HOLISTIC VIEWPOINT:

SOME PITFALLS AND WORDS OF CAUTION

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

Reasoning, Inference, and Rules
Industries in Need

"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

Reasoning, Inference, and Rules
When to Use These Systems

If you thought that XML was going to solve your problems, and you found out that you were wrong.

You are ready to start building toward the future

You need your IT systems to act smarter

You can't cope with the complexity in your information

When you need something fundamentally different, you've been solving the same thing over and over again and getting the same poor results

Information overload prevents your leadership from making the best decisions possible

Your business is information-driven

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

Reasoning, Inference, and Rules
When NOT to Use These Systems

You are looking for a quick-fix, cheap solution

Information is not core to your business success

You cannot train your people to work with semantically rich information

You want to sole-source to a major software vendor like SAP, PeopleSoft and Microsoft

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

Reasoning, Inference, and Rules
Training is Essential

THE BAD NEWS:

TRAIN YOUR PEOPLE!

THE GOOD NEWS:

DBA's may already know most of this stuff…
Object-Oriented analysts have a BIG head-start…

Tools are available:

- Modeling – GUI driven, Visio-based, Rational-based, etc.
- Inference – Java based, C++, App Server based, etc.
- Rules – GUI driven, J2EE integration, .NET components, etc.
- Database – RDF databases, open source, triples stores

Trainers are available:

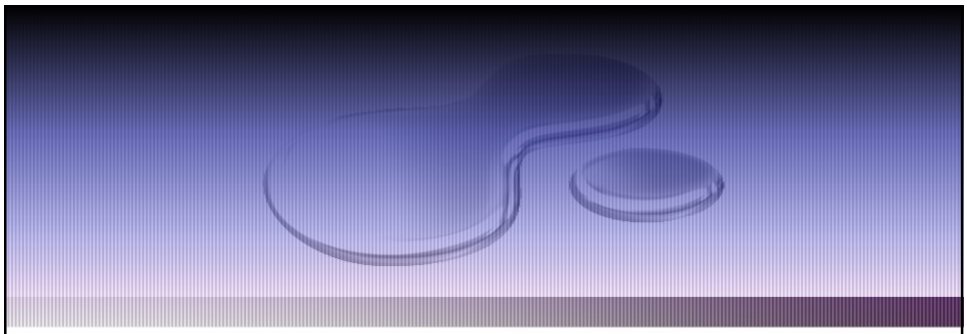
- Software Vendors (Network Inference, Tucana, etc.)
- TopQudrant – Dedicated RDF and OWL training

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

Reasoning, Inference and Rules
Conclusion: Consider Adopting Now!!!

- **Standards Leading the Momentum**
  - W3C – core data and web technologies are semantically enabled
  - OASIS – ebXML is moving rapidly towards an OWL/RDF vision
  - PLCS – manufacturing lifecycle ontologies soon to be OWL ready
  - HL7 – healthcare and medical ontologies already OWL and/or DL
- **Vendors Moving Rapidly With Support**
  - Network Inference – 20 major customers installed w/OWL platforms
  - Sun Microsystems – Swordfish RDF data tools nearing production
  - IBM – commitment to RDF interfaces for DB2ii middleware
  - HP – market-leading open source RDF toolkit – Jena
  - Adobe – embeds RDF metadata in every binary file
- **Your Friends/Competition are Adopting**
  - NATO members adopting OWL net-centric warfare technologies, Fortune 500 electronics companies using OWL to drive financials, Fortune 500 medical companies using OWL/RDF for hospital maintenance systems, plus many, many more…

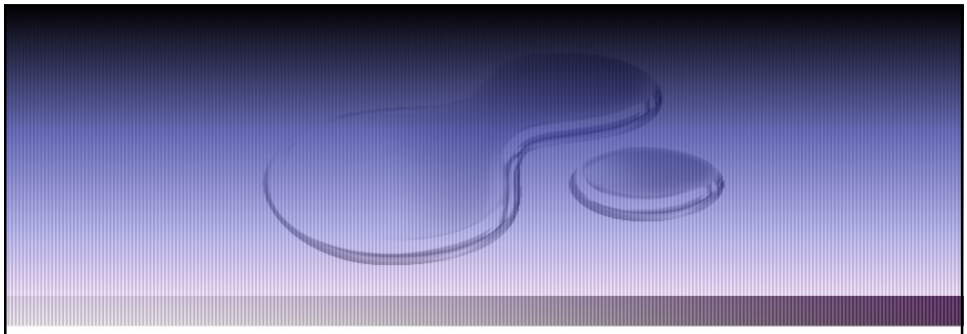
© 2004, Network Inference, Inc. All rights reserved.
network inference



Special Word of Thanks:

This presentation uses slides, pictures, and concepts derived from the hard work of many. In particular, Professor Ian Horrocks, Dr. Deborah McGuinness, Dr. Michael Kifer and Jason Morris deserve many thanks for their contributions.

© 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