Lightweight English

Heavyweight Inference

and a Semantic Distance Measure

Adrian Walker

www.reengineeringllc.com

Presentation for the NIST / NSF Semantic Distance Workshop, November 2003, with an added RDF example

Please see also the more recent materials:

http://www.semantic-conference.com/program/sessions/S2.html

http://www.reengineeringllc.com/Internet_Business_Logic_e-Government_Presentation.pdf

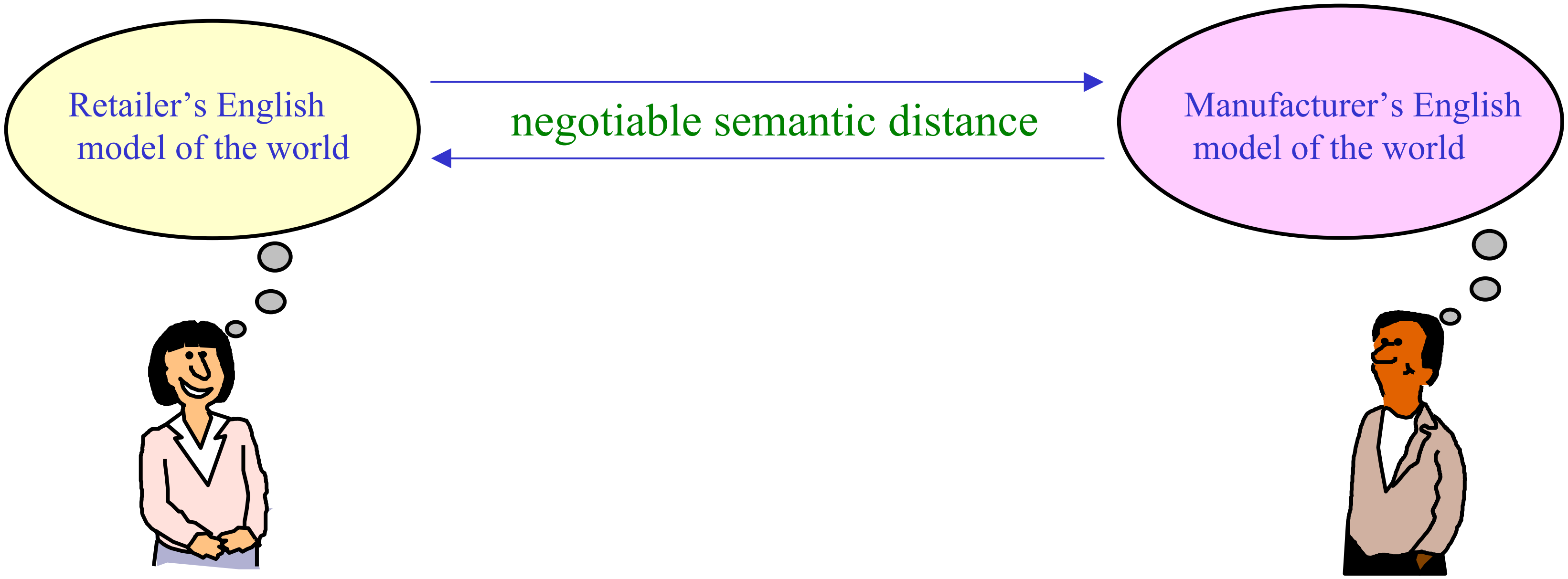
http://www.reengineeringllc.com/Oil_Industry_Supply_Chain_by_Kowalski_and_Walker.pdf

http://www.w3.org/2004/12/rules-ws/paper/19

Outline

- Why we need Natural Language
  - even for simple semantic tasks
- NLP need not be a resource sink
  - A Resource Description Framework (RDF) example
  - A manufacturing example
  - A Process Specification Language example
- A Semantic Distance Measure
  - An abstract example
  - A definition
- Conclusions

Why we need Natural Language, even for simple semantic tasks

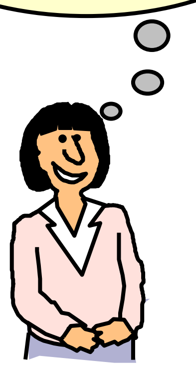


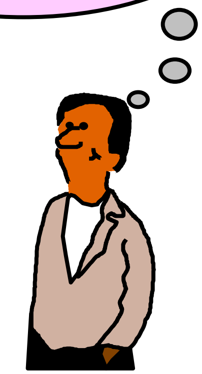
Why we need Natural Language, even for simple semantic tasks

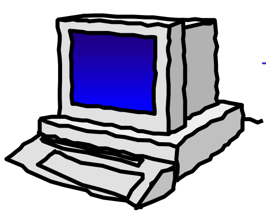
Retailer's English model of the world

negotiable semantic distance

Manufacturer's English model of the world



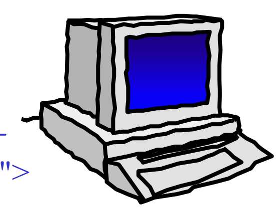




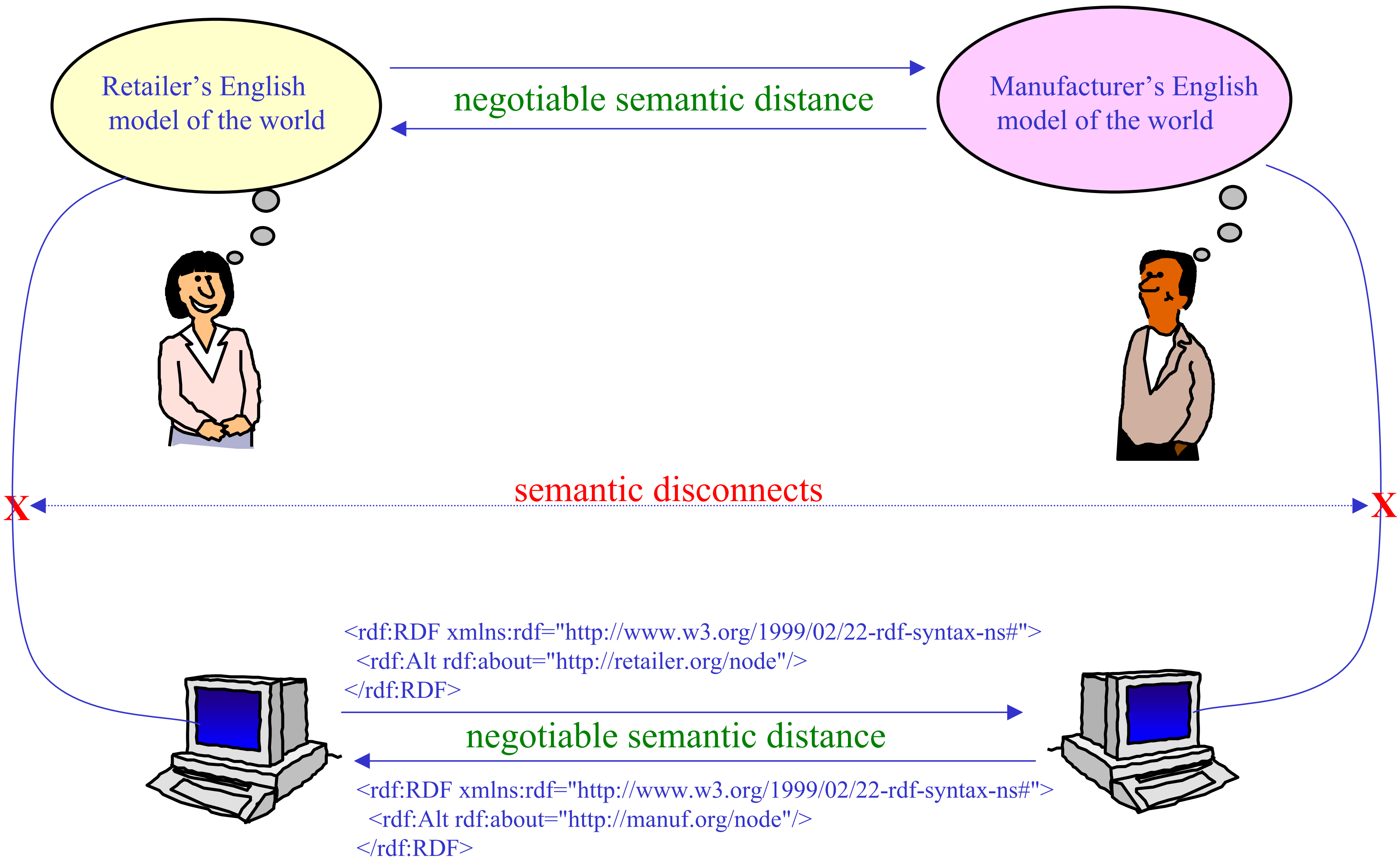
```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Alt rdf:about="http://retailer.org/node"/>
</rdf:RDF>
```

negotiable semantic distance

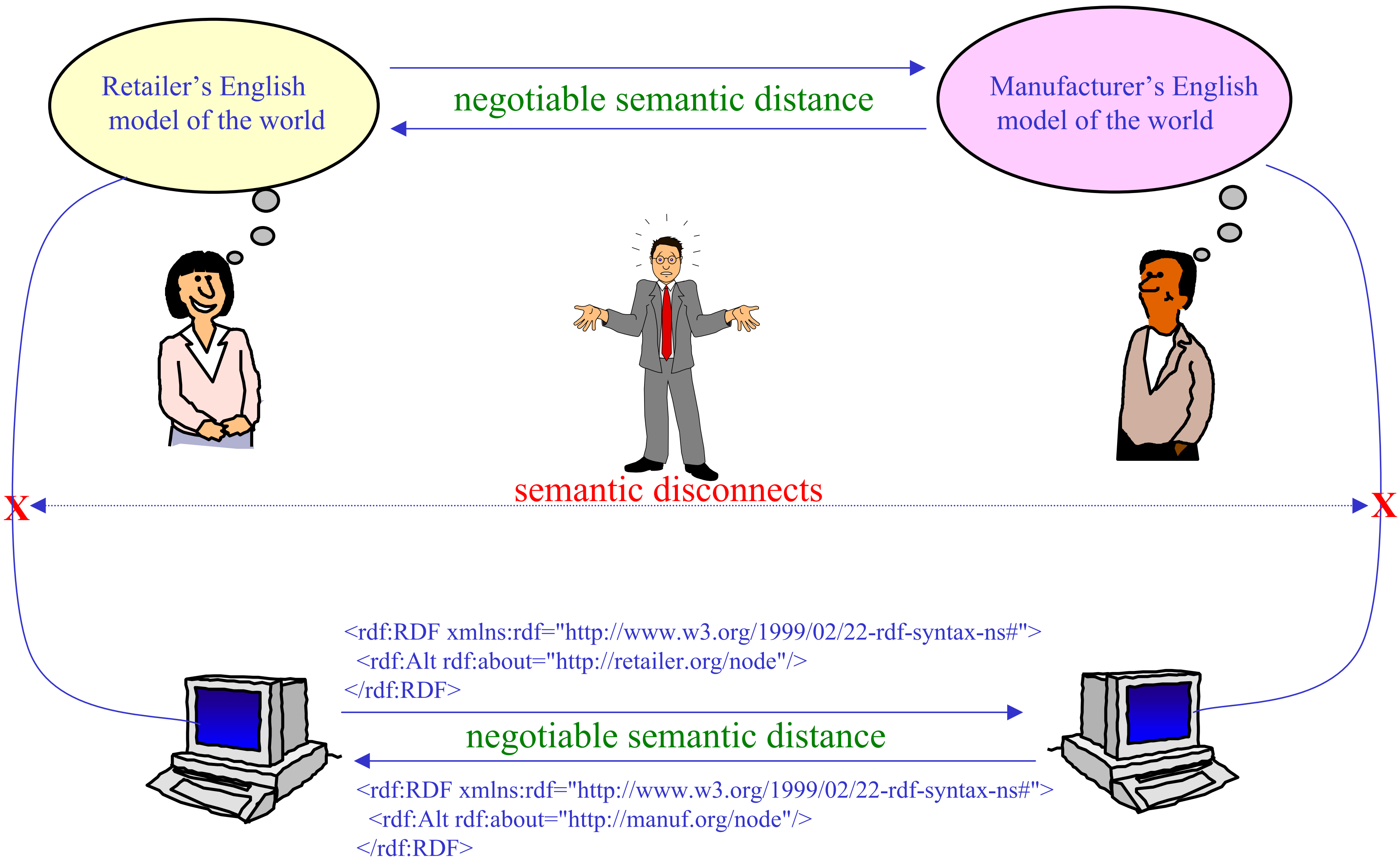
```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Alt rdf:about="http://manuf.org/node"/>
</rdf:RDF>
```



Why we need Natural Language, even for simple semantic tasks



Why we need Natural Language, even for simple semantic tasks



Why we need Natural Language, even for simple semantic tasks

- Ontologies currently contain words and short phrases
  - plus related-by-taxonomy info
  - plus machine-friendly KIF, RDF, etc...

Why we need Natural Language, even for simple semantic tasks

- Ontologies currently contain words and short phrases,
  - plus related-by-taxonomy info
  - plus machine-friendly KIF, RDF, etc...

- A term is defined by the set of its superclasses in the taxonomy, and by its properties

Why we need Natural Language, even for simple semantic tasks

- Ontologies currently contain words and short phrases,
  - plus related-by-taxonomy info
  - plus machine-friendly KIF, RDF, etc...

- A term is defined by the set of its superclasses in the taxonomy, and by its properties

- Ontologies do not contain human-friendly sentences, or relations between sentences

  -- Except as comments that are ***not*** used by machines

Why we need Natural Language, even for simple semantic tasks

"...the current KIF-based syntax [of PSL] is not easily

understandable for 'nongeeks' and in the future a more

human readable language representation is needed."

-- *Ontology-Based Translation: A Case of Process Specification Language (PSL)*
Teppo.Pirttioja@hut.fi

Why we need Natural Language, even for simple semantic tasks

“... As we read and write N3, communicating in RDF, we need to share an understanding of what each URI means. We often pick URIs which offer clues about meaning, such as

http://www.w3.org/2000/10/swap/test/demo1/biology#Dog

but the text of the URI still gives only a clue…..

---*http://www.w3.org/2000/10/swap/doc/ontologies*

Why we need Natural Language, even for simple semantic tasks

"... As we read and write N3, communicating in RDF, we need to share an understanding of what each URI means. We often pick URIs which offer clues about meaning, such as

http://www.w3.org/2000/10/swap/test/demo1/biology#Dog

but the text of the URI still gives only a clue. Would a wolf qualify as a one of these? How about a Dingo? We can't tell just by looking at the name. It's even possible the URI text is misleading, and the intended meaning has nothing to do with dogs."

---*http://www.w3.org/2000/10/swap/doc/ontologies*

Why we need Natural Language, even for simple semantic tasks

John Sowa's example :

Clyde is an elephant,
elephant is a species
==> Clyde is a species   Wrong !

Why we need Natural Language, even for simple semantic tasks

John Sowa's example :

Clyde is an elephant,
elephant is a species
==> Clyde is a species   Wrong !

It looks OK in logic

p(X,Y), p(Y,Z) ==> p(X,Z)   But that's no help

Why we need Natural Language, even for simple semantic tasks

John Sowa's example :

Clyde is an elephant,
elephant is a species
==> Clyde is a species   Wrong !

It looks OK in logic

p(X,Y), p(Y,Z) ==> p(X,Z)   But that's no help

RDF to the rescue ?

X verylongoverloadedURI1 Y
Y verylongoverloadedURI2 Z ==>
X verylongoverloadedURI3 Z

Not much help either

Why we need Natural Language, even for simple semantic tasks

Clyde is an elephant, elephant is a species ==> Clyde is a species Wrong!

So, write and run the example in lightweight, executable English instead:

Why we need Natural Language, even for simple semantic tasks

Clyde is an elephant, elephant is a species ==> Clyde is a species   Wrong!

So, write and run the example in lightweight, executable English instead:

Facts

this-item is a member of the set this-set

================================

Clyde   The Elephants

this-item is a named subset of the set this-set

======================================

The Elephants   All Species Of Animals

Why we need Natural Language, even for simple semantic tasks

Clyde is an elephant, elephant is a species ==> Clyde is a species Wrong!

So, write and run the example in lightweight, executable English instead:

Facts

this-item is a member of the set this-set	
Clyde	The Elephants

this-item is a named subset of the set this-set	
The Elephants	All Species Of Animals

General rule

some-item is a member of the set some-set
that-set is a named subset of the set some-superset
----
that-item is a member of a named subset of that-superset

Why we need Natural Language, even for simple semantic tasks

Clyde is an elephant, elephant is a species ==> Clyde is a species Wrong!

So, write and run the example in lightweight, executable English instead:

Facts

this-item is a member of the set this-set		this-item is a named subset of the set this-set	
Clyde	The Elephants	The Elephants	All Species Of Animals

General rule

some-item is a member of the set some-set
that-set is a named subset of the set some-superset
----
that-item is a member of a named subset of that-superset

Explanation

Clyde is a member of the set The Elephants
The Elephants is a named subset of the set All Species Of Animals
----
Clyde is a member of a named subset of All Species Of Animals

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ClydeElephant1 at www.reengineeringllc.com

NLP need not be a resource sink

Example   Reasoning Using RDF

If some first thing is related by rdf:type to a second thing,
and that second thing is related by rdfs:subClassOf to a third thing,
then that first thing is related by rdf:type to that third thing

some-subject is related by rdf:type to some-subclass
that-subclass is related by rdfs:subClassOf to some-object
rdf:type can be expanded to some-URI1:name1
rdfs:subClassOf can be expanded to some-URI2:name2
ns is shorthand for this-URI
----
that-subject is related by rdf:type to that-object

(Note that this kind of inheritance reasoning does not always seem to be valid in the real world, as indicated in the "Clyde is a species" example. That's why we need lightweight English)

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo RDFreasoning1 at www.reengineeringllc.com

-- Example based on "Using Inference Rules" at http://www.interprise.com

NLP need not be a resource sink

Reasoning Using RDF -- some facts

some-subject is related by rdf:type to some-subclass
that-subclass is related by rdfs:subClassOf to some-object
rdf:type can be expanded to some-URI1:name1
rdfs:subClassOf can be expanded to some-URI2:name2
ns is shorthand for this-URI
----
that-subject is related by rdf:type to that-object

this-subject is related by this-predicate to this-object		
ns:_0123456789	rdf:type	ns:Car
ns:Car	rdfs:subClassOf	ns:LandVehicle
ns:LandVehicle	rdfs:subClassOf	ns:Vehicle

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo RDFreasoning1 at www.reengineeringllc.com

-- Example based on "Using Inference Rules" at http://www.interprise.com

Reasoning Using RDF -- An Answer

this-subject is related by this-predicate to this-object		
ns:Car	rdfs:subClassOf	ns:LandVehicle
ns:LandVehicle	rdfs:subClassOf	ns:Vehicle
ns:_0123456789	rdf:type	ns:Car
ns:_0123456789	rdf:type	ns:LandVehicle
ns:_0123456789	rdf:type	ns:Vehicle

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo RDFreasoning1 at www.reengineeringllc.com

-- Example based on "Using Inference Rules" at http://www.interprise.com

NLP need not be a resource sink

Reasoning Using RDF -- An Explanation/Proof

ns:_0123456789 is related by rdf:type to ns:LandVehicle
ns:LandVehicle is related by rdfs:subClassOf to ns:Vehicle
rdf:type can be expanded to http://www.w3.org/1999/02/22-rdf-syntax-ns#:type
rdfs:subClassOf can be expanded to http://www.w3.org/2000/01/rdf-schema#:subClassOf
ns is shorthand for http://www.reengineeringllc.com/namespaces/ns#
----
ns:_0123456789 is related by rdf:type to ns:Vehicle

ns:_0123456789 is related by rdf:type to ns:Car
ns:Car is related by rdfs:subClassOf to ns:LandVehicle
rdf:type can be expanded to http://www.w3.org/1999/02/22-rdf-syntax-ns#:type
rdfs:subClassOf can be expanded to http://www.w3.org/2000/01/rdf-schema#:subClassOf
ns is shorthand for http://www.reengineeringllc.com/namespaces/ns#
----
ns:_0123456789 is related by rdf:type to ns:LandVehicle

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo RDFreasoning1 at www.reengineeringllc.com*

-- *Example based on "Using Inference Rules" at http://www.interprise.com*

NLP need not be a resource sink

Example A retailer orders computers from a manufacturer

In the retailer's terminology, a computer is called a *PC for Gamers*, while in the manufacturer's terminology, it is called a *Prof Desktop*.

The retailer and the manufacturer agree that both belong to the class Worksts/Desktops

Use semantic resolution to find out to what extent a *Prof Desktop* has the required memory, CPU and so forth for a *PC for Gamers*

-- Example based on "Semantic Resolution for E-Commerce",
by Yun Peng, Youyong Zou, Xiaocheng Luan (UMBC) and
Nenad Ivezic, Michael Gruninger and Albert Jones (NIST)

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- facts

for the retailer the term PC for Gamers has super-class this-class in the this-ns namespace	
Computers to order	retailer
Worksts/Desktops	shared
Computers	shared

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- facts

for the retailer the term PC for Gamers has super-class this-class in the this-ns namespace

==

Computers to order	retailer
Worksts/Desktops	shared
Computers	shared

for the manufacturer the term Prof Desktop has super-class this-class in the this-ns namespace

==

Desktop	manufacturer
Worksts/Desktops	shared
Computer Systems	manufacturer
Computers	shared

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- facts and a rule

for the retailer the term PC for Gamers has super-class this-class in the this-ns namespace
==

Computers to order	retailer
Worksts/Desktops	shared
Computers	shared

for the manufacturer the term Prof Desktop has super-class this-class in the this-ns namespace
==

Desktop	manufacturer
Worksts/Desktops	shared
Computer Systems	manufacturer
Computers	shared

for the retailer the term some-item1 has super-class some-class in the some-ns namespace
for the manufacturer the term some-item2 has super-class that-class in the that-ns namespace
----
the retailer term that-item1 and the manufacturer term that-item2 agree - they are of type that-class

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- answer table

this-result : retailer this-item1 is matched by manufacturer this-item2 on the property this-prop for part this-comp

this-result	this-item1	this-item2	this-prop	this-comp
NEED	PC for Gamers	*missing-item*	Size	Graphics Card
OK	PC for Gamers	Prof Desktop	Size	CPU
OK	PC for Gamers	Prof Desktop	Size	Memory
OK	PC for Gamers	Prof Desktop	Size	Sound Card

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com*

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- explanation/proof of an answer

retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

----

OK : retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- explanation/proof of an answer

retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory
----
OK : retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

the retailer term PC for Gamers and the manufacturer term Prof Desktop agree - they are of type Worksts/Desktops
for the retailer the term PC for Gamers has part Memory with property Size >= 256 in the shared namespace
for the manufacturer the term Prof Desktop has part Memory with property Size = 512 in the shared namespace
= 512 meets the requirement >= 256
----
retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com*

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- explanation/proof of an answer

retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

----

OK : retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

the retailer term PC for Gamers and the manufacturer term Prof Desktop agree - they are of type Workststs/Desktops
for the retailer the term PC for Gamers has part Memory with property Size >= 256 in the shared namespace
for the manufacturer the term Prof Desktop has part Memory with property Size = 512 in the shared namespace
= 512 meets the requirement >= 256

----

retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

for the retailer the term PC for Gamers has super-class Worksts/Desktops in the shared namespace
for the manufacturer the term Prof Desktop has super-class Worksts/Desktops in the shared namespace

----

the retailer term PC for Gamers and the manufacturer term Prof Desktop agree - they are of type Worksts/Desktops

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com*

NLP need not be a resource sink

A retailer orders computers from a manufacturer -- explanation/proof of an answer

retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory
----
OK : retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

the retailer term PC for Gamers and the manufacturer term Prof Desktop agree - they are of type Worksts/Desktops
for the retailer the term PC for Gamers has part Memory with property Size >= 256 in the shared namespace
for the manufacturer the term Prof Desktop has part Memory with property Size = 512 in the shared namespace
= 512 meets the requirement >= 256
----
retailer PC for Gamers is matched by manufacturer Prof Desktop on the property Size for part Memory

for the retailer the term PC for Gamers has super-class Worksts/Desktops in the shared namespace
for the manufacturer the term Prof Desktop has super-class Worksts/Desktops in the shared namespace
----
the retailer term PC for Gamers and the manufacturer term Prof Desktop agree - they are of type Worksts/Desktops

512 is greater than or equal 256

= 512 meets the requirement >= 256

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution1 at www.reengineeringllc.com

NLP need not be a resource sink

Example   Process Specification Language -- food service

A food service process must include ordering, preparing, serving, eating, and paying, but not necessarily in exactly that order

The constraints are:

- Ordering, preparing, and serving always happen before eating
- Serving happens after preparing and ordering
- Paying can happen any time in the process

-- *Example based on "PSL: A Semantic Domain for Flow Models"*
by Conrad Bock (NIST) and Michael Gruninger (NIST)

NLP need not be a resource sink

Process Specification Language -- facts

this-activity1 must occur before	this-activity2
ordering	eating
preparing	eating
serving	eating
preparing	serving
ordering	serving

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com*

NLP need not be a resource sink

Process Specification Language -- facts

this-activity1 must occur before this-activity2	
ordering	eating
preparing	eating
serving	eating
preparing	serving
ordering	serving

in scenario this-number step this-step is this-activity

in scenario this-number	step this-step	is this-activity
1	1	ordering
1	2	paying
1	3	eating

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com

NLP need not be a resource sink

Process Specification Language -- rules

Rule for checking a given scenario (1)

in scenario some-number step some-step2 is some-activity2
some-activity1 must occur before that-activity2
not : in scenario that-number that-activity1 occurs before that-activity2
----
in scenario that-number step that-activity1 should have happened before that-activity2 but did not

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com*

NLP need not be a resource sink

Process Specification Language -- rules

Rule for checking a given scenario (1)

in scenario some-number step some-step2 is some-activity2
some-activity1 must occur before that-activity2
not : in scenario that-number that-activity1 occurs before that-activity2
----
in scenario that-number step that-activity1 should have happened before that-activity2 but did not

Simplified rule for finding a new scenario (2)

in scenario 2 step some-step is some-activity
that-activity must occur before some-activity2
not : there is an activity that must occur between that-activity and that-activity2
that-step + 1 = some-step2
----
in scenario 2 step that-step2 is that-activity2

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com*

NLP need not be a resource sink

Process Specification Language - Checking the given scenario (1)

Answer

in scenario this-number step this-activity1 should have happened before this-activity2 but did not

this-number	this-activity1	this-activity2
1	preparing	eating
1	serving	eating

Explanation/proof

in scenario 1 step 3 is eating
preparing must occur before eating
not: in scenario 1 preparing occurs before eating
----
in scenario 1 step preparing should have happened before eating but did not
…..

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com*

Process Specification Language - finding a new scenario (2)

Answer / Process Plan with Parallel Steps

in scenario 2 step this-step	is this-activity
1	ordering
1	paying
1	preparing
2	serving
3	eating

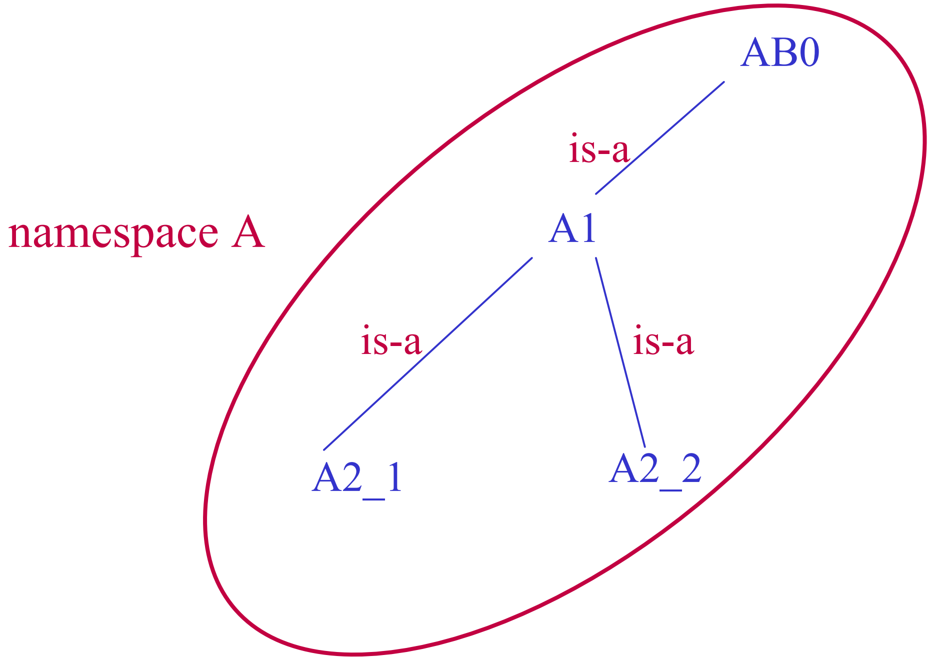
Explanation/proof

in scenario 2 step 2 is serving
serving must occur before eating
not: there is an activity that must occur between serving and eating
2 + 1 = 3
----
in scenario 2 step 3 is eating
…..

-- To run or change this example, please point IE6, Netscape7 or Mozilla
to the demo ProcessSpecificationLanguage1 at www.reengineeringllc.com

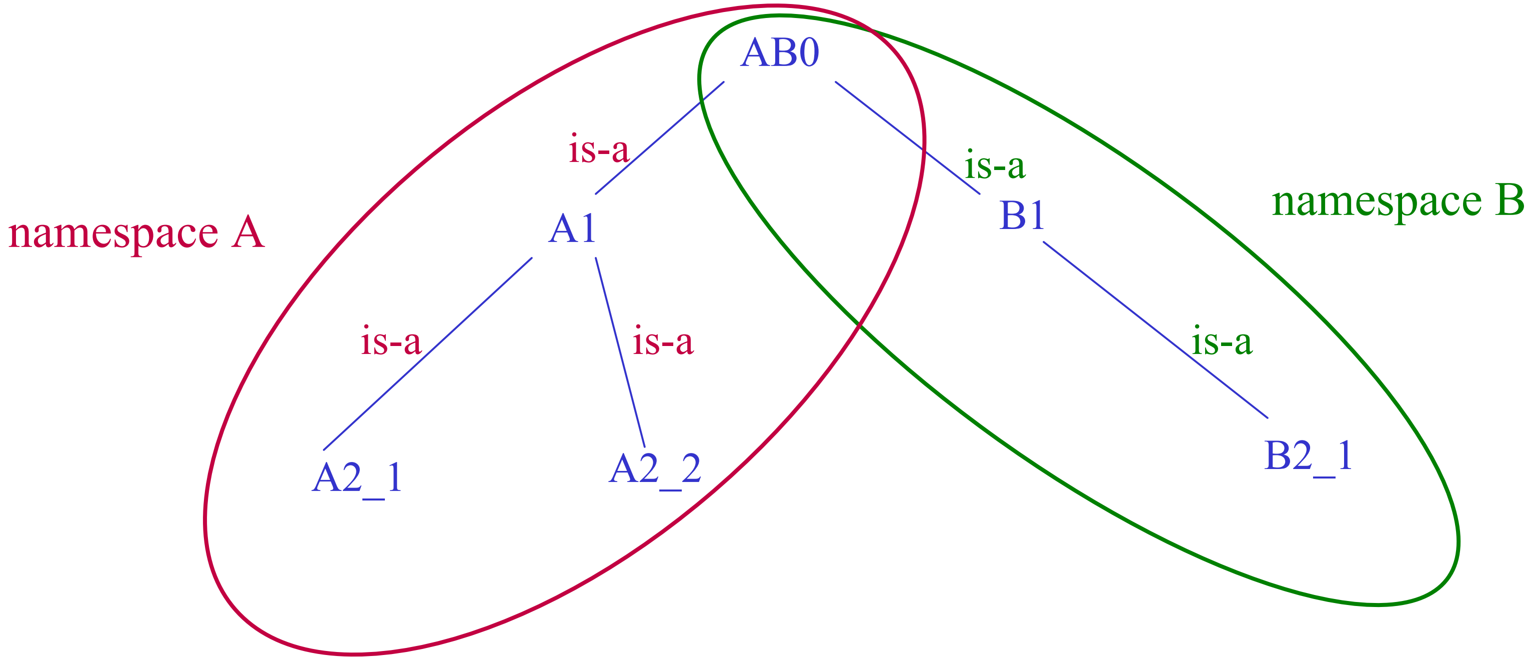
A Semantic Distance Measure

An abstract example



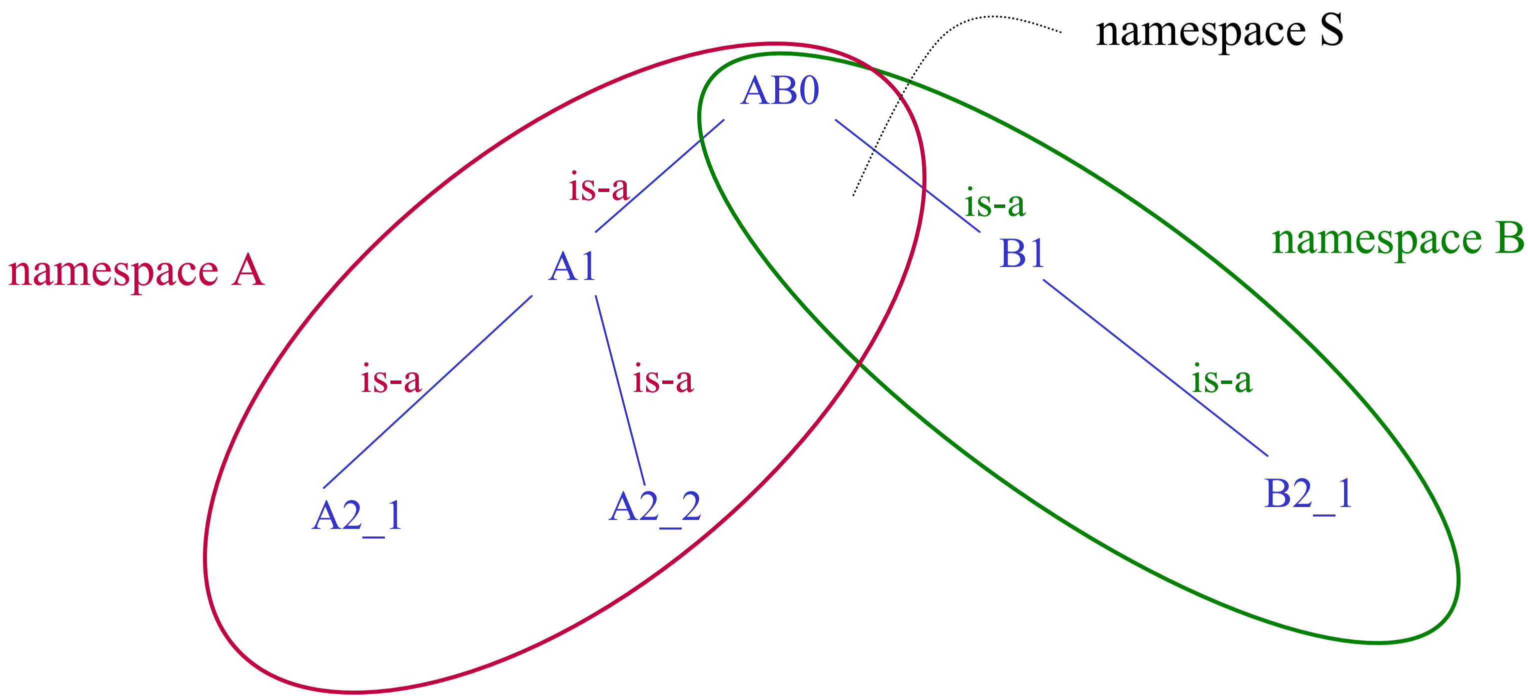
A Semantic Distance Measure

An abstract example



A Semantic Distance Measure

An abstract example

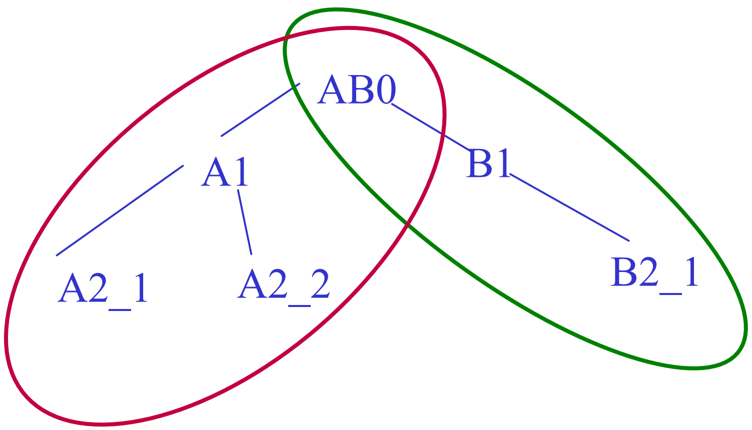


A Semantic Distance Measure

Abstract example -- facts

this-player	uses the name this-name	in namespace this-ns
A	AB0	S
B	AB0	S

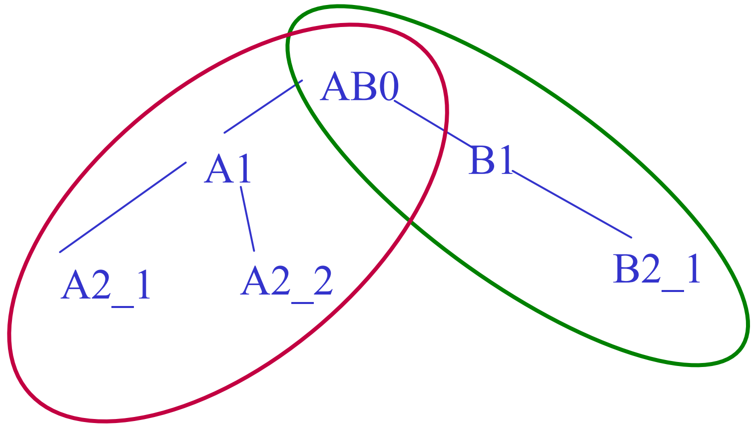
this-name1	is a this-name2	in namespace this-ns
A1	AB0	A
B1	AB0	B
A2_1	A1	A
A2_2	A1	A
B2_1	B1	B



-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution2 at www.reengineeringllc.com

A Semantic Distance Measure

Abstract example -- rules

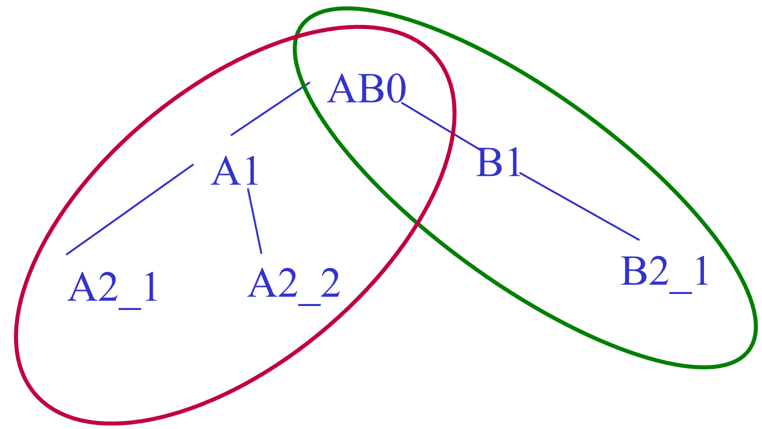


some-name1 is a bottom item in namespace some-ns1
some-name2 is a bottom item in namespace some-ns2
that-name1 and that-name2 are different
that-ns1 and that-ns2 are different
that-name1 is a some-name3 in namespace that-ns1
that-name2 is a that-name3 in namespace that-ns2
the-player1 and the-player2 have agreed on the meaning of that-name3
that-name3 specializes to some-number1 different bottom names in namespace that-ns1
that-name3 specializes to some-number2 different bottom names in namespace that-ns2
----
that-name1 in namespace that-ns1 that-number1 **::** that-number2 matches that-name2 in namespace that-ns2

some-name1 in namespace some-ns1 some-number1 **::** some-number2 matches some-name2 in namespace some-ns2
that-number1 + that-number2 = some-sum
that-sum - 2 = some-number
----
that-name1 in namespace that-ns1 matches that-name2 in namespace that-ns2 with semantic distance that-number

-- To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution2 at www.reengineeringllc.com

A Semantic Distance Measure

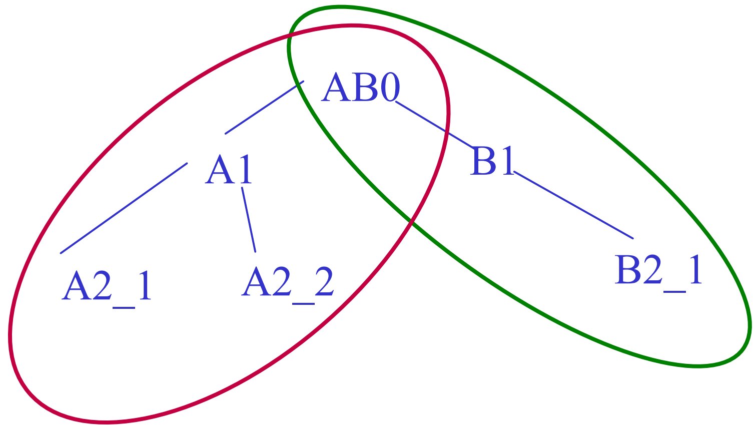


Abstract example -- answer

this-name1 in namespace	this-ns1 matches	this-name2 in namespace	this-ns2 with semantic distance	this-number
A2_1	A	B2_1	B	1
A2_2	A	B2_1	B	1

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution2 at www.reengineeringllc.com*

A Semantic Distance Measure



Abstract example -- explanation/proof

A2_1 in namespace A **2 :: 1** matches B2_1 in namespace B
2 + 1 = 3
3 - 2 = 1

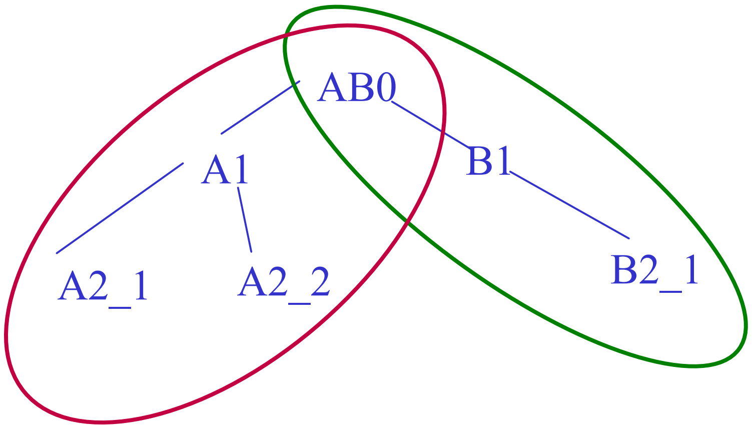
A2_1 in namespace A matches B2_1 in namespace B with semantic distance 1

A2_1 is a bottom item in namespace A
B2_1 is a bottom item in namespace B
A2_1 and B2_1 are different
A and B are different
A2_1 is a AB0 in namespace A
B2_1 is a AB0 in namespace B
A and B have agreed on the meaning of AB0
AB0 specializes to 2 different bottom names in namespace A
AB0 specializes to 1 different bottom names in namespace B

A2_1 in namespace A **2 :: 1** matches B2_1 in namespace B

-- *To run or change this example, please point IE6, Netscape7 or Mozilla to the demo SemanticResolution2 at www.reengineeringllc.com*

A Semantic Distance Measure

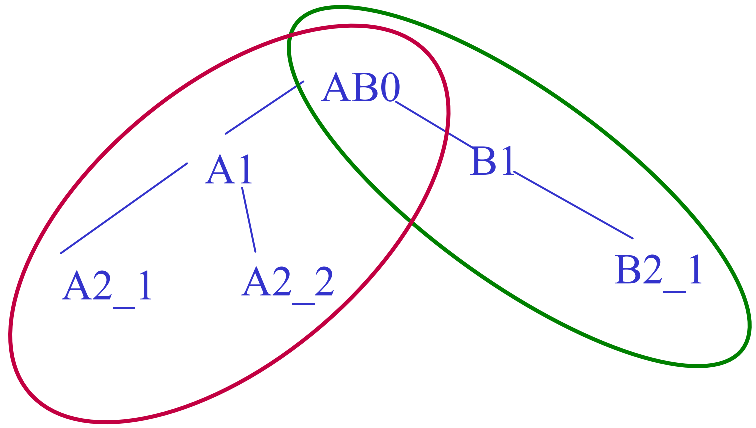


Definition:

g(n,x,y) in namespace n, x generalizes to y (transitively)

shared(n1,n2,y) y is shared between the namespaces n1 and n2

A Semantic Distance Measure



Definition:

g(n,x,y) in namespace n, x generalizes to y (transitively)

shared(n1,n2,y) y is shared between the namespaces n1 and n2

SemanticDistance(x,n1,z,n2) =

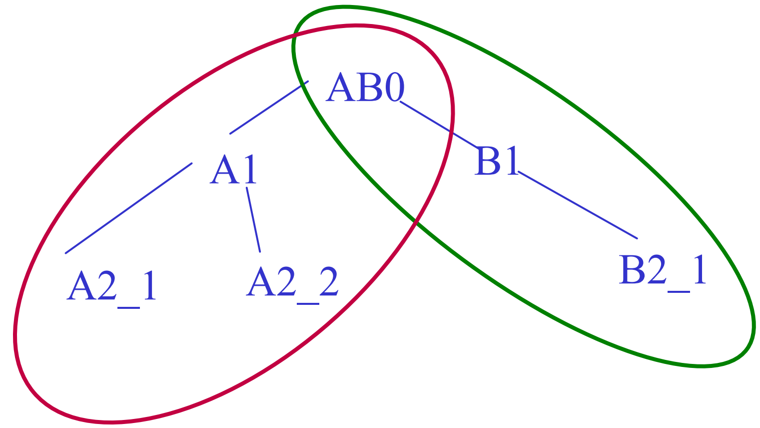
min m [exists(y) such that shared(n1,n2,y) &

g(n1,x,y) & g(n2,z,y) &

((|{x1 : g(n1,x1,y)}| + |{z1 : g(n2,z1,y)}|) = m)] - 2

if such an m exists, else undefined

A Semantic Distance Measure



Definition:

g(n,x,y) in namespace n, x generalizes to y (transitively)

shared(n1,n2,y) y is shared between the namespaces n1 and n2

SemanticDistance(x,n1,z,n2) =

min m [exists(y) such that shared(n1,n2,y) &

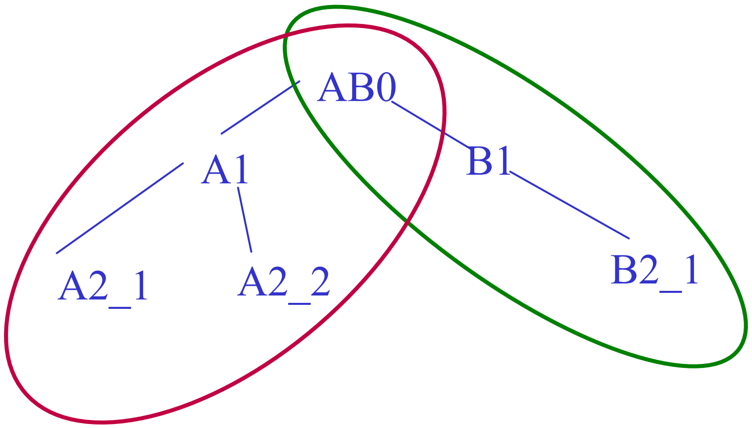
g(n1,x,y) & g(n2,z,y) &

((|{x1 : g(n1,x1,y)}| + |{z1 : g(n2,z1,y)}|) = m)] - 2

if such an m exists, else undefined

Note that SemanticDistance = the number of matches to be ruled out to get unique match

A Semantic Distance Measure



Definition:

g(n,x,y) in namespace n, x generalizes to y (transitively)

shared(n1,n2,y) y is shared between the namespaces n1 and n2

SemanticDistance(x,n1,z,n2) =

min m [exists(y) such that shared(n1,n2,y) &

g(n1,x,y) & g(n2,z,y) &

((|{x1 : g(n1,x1,y)}| + |{z1 : g(n2,z1,y)}|) = m)] - 2

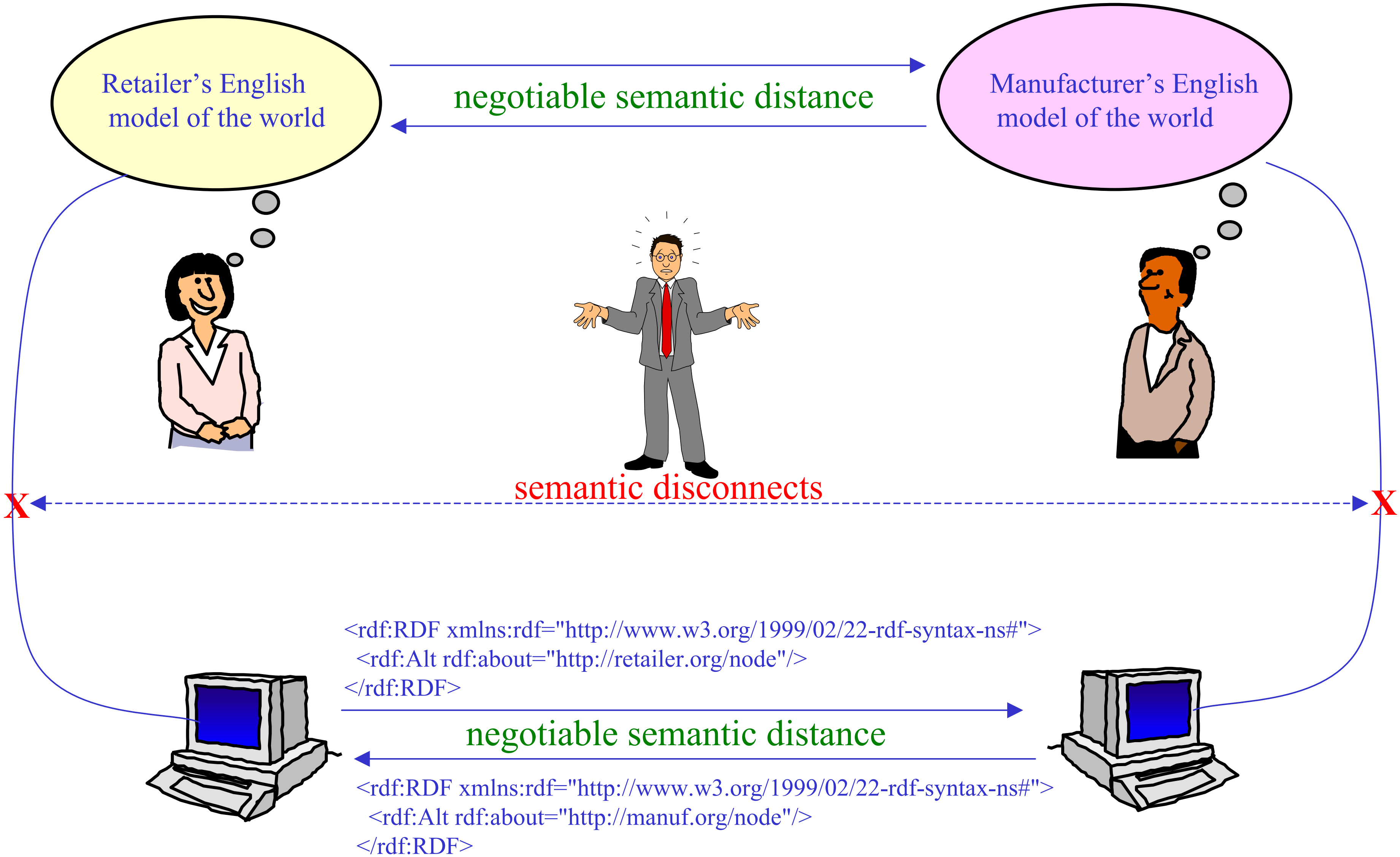
if such an m exists, else undefined

Note that SemanticDistance = the number of matches to be ruled out to get unique match

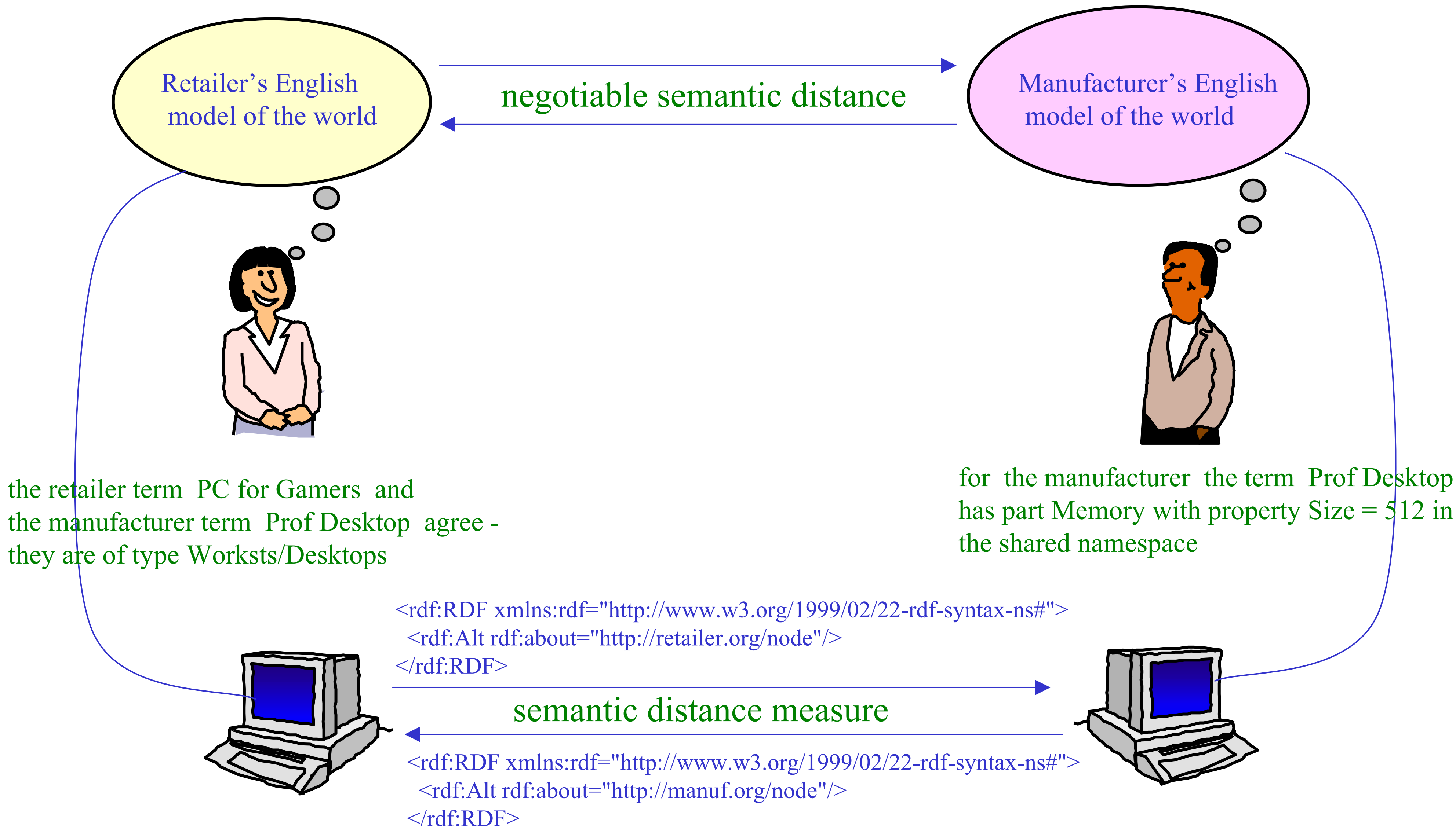
SemanticDistance(Ont1,Ont2) = min sum pairs(x,z) SemanticDistance(x,n1,z,n2)

(Also, set of pairs(x,z) that did not match at all)

Conclusions



Conclusions



Conclusions

- We need Natural Language
  - even for simple semantic tasks, like “Clyde is an elephant”
  - to remove the semantic disconnect between people and ontology notations

Conclusions

- We need Natural Language
  - even for simple semantic tasks, like “Clyde is an elephant”
  - to remove the semantic disconnect between people and ontology notations
- NLP need not be a resource sink, even with *un*limited vocabulary
  - “Just enough” lightweight NLP, combined with heavyweight inferencing
  - Can use the lightweight NLP for English, German… no grammar or dictionary work
  - Can use standard information retrieval for (parts of) lightweight English ontologies
  - A Resource Description Framework (RDF) example
  - A manufacturing example
  - A Process Specification Language example
  - Run and change examples by pointing a browser to www.reengineeringllc.com

Conclusions

- We need Natural Language
  - even for simple semantic tasks, like “Clyde is an elephant”
  - to remove the semantic disconnect between people and ontology notations
- NLP need not be a resource sink, even with *un*limited vocabulary
  - “Just enough” lightweight NLP, combined with heavyweight inferencing
  - Can use the lightweight NLP for English, German… no grammar or dictionary work
  - Can use standard information retrieval for (parts of) lightweight English ontologies
  - A Resource Description Framework (RDF) example
  - A manufacturing example
  - A Process Specification Language example
  - Run and change examples by pointing a browser to www.reengineeringllc.com
- A Semantic Distance Measure
  - An abstract example
  - Semantic distance definition ~ ambiguity
  - Number of meanings that must be removed to get an exact match

Links

1. Focus Scenario for the NIST/NSF Workshop on Semantic Distance, working paper by Ted Goranson, <tedg@sirius-beta.com>, October 2003

2. The NIST / UMBC papers listed in the presentation can be downloaded from :

http://www.mel.nist.gov/msidlibrary/publications.html

3. The English inferencing examples

ClydeElephant1

RDFreasoning1

SemanticResolution1

ProcessSpecificationLanguage1

SemanticResolution2

can be run, changed, and re-run as follows:

1. Point Internet Explorer 6, Netscape 7 or Mozilla to www.reengineeringllc.com
2. Click on Internet Business Logic
3. Click on the GO button
4. Click on the Help button to see how to navigate through the pages
5. Select *ClydeElephant1*

Similarly, you can write and run your own examples.

Reengineering

Reengineering LLC is a privately held company, located in Bristol, CT, USA.

Dr. Adrian Walker is the CTO of the company. His experience includes: Assistant Professor -- Rutgers University, Member of Technical Staff -- Bell Laboratories, Murray Hill New Jersey, Manager, Principles and Applications of Logic Programming -- IBM Yorktown Heights Research Laboratory, Manager, Internet Development -- Eventra (a manufacturing supply chain company).

We work on the Internet Business Logic system, to support flexible, self explaining database programs, written in English. There is an article about the system in the *Software Development Times*, see http://www.sdtimes.com/cols/industrywatch_086.htm .

We have patents pending on: Semantic Encoding, a method and system for securing the contents of relational databases that is immune from conventional cryptological attack (joint work with Professor Paul Benjamin), and on

Confusion Encryption, a novel encryption method having the property that an attacker who finds a plausible plaintext from a ciphertext cannot know whether or not the plaintext is a correct decrypt.