

Cerebra®

Using XQuery with OWL

Jeffrey Pollock, Cerebra, Inc.
June 17, 2005



Agenda

- Introductions
  - Who we are
- Cerebra XQuery Implementation
  - Overview
  - XQuery Extensions
  - Future Directions
- Query Demonstration
- Pocket Slides
  - Why are Semantic Technologies Different?
  - What are Semantics?
  - What are Ontology?
  - What is Inference?

Cerebra®

Who We Are:
Company Facts

Products	Cerebra Server – Runtime query/retrieval platform Cerebra Repository – Runtime metadata platform Cerebra Construct – Design-time development IDE
Customers	20+ customers in 2004 from F100, Global 50, Federal and Commercial; Direct and Indirect. Short list: webMethods, Cisco, Department of Homeland Security, Lockheed Martin, SAIC, General Electric, NASA…
Channels	BearingPoint, Lockheed Martin, SAIC, MITRE, QinetiQ, WebMethods (OEM), TopQuadrant, Baseline Consulting, Network Effects.
Alliances	Establishing partnerships w/ leaders in selected key horizontal markets, i.e. Oracle, Documentum, Composite Software, MetaMatrix, ILog.
Standards	Active in OWL, RDF, ebXML, and UDDI groups, Chief Scientist (Prof. Ian Horrocks) and advisory board responsible for advancement all significant semantic web standards.
Locations	HQ: San Diego, Regional offices: Menlo Park CA, Boston MA, Washington DC, Manchester UK
Financing	Series-B January 2004 from Palomar Ventures, BlueRun Ventures (formerly Nokia Venture Partners)

Cerebra®



Cerebra®

XQuery Implementation

Overview, Extensions, Directions



Overview

- Cerebra's XQuery Implementation is:
  - A subset with extensions and a future direction
  - Based on function calls to underlying Description Logic model
  - Handles *provable* situations and *possible* situations
- A Subset
  - No built-in XQuery Data Model
  - No built-in XPath Expressions
  - No built-in XSLT Transformations
- Supported
  - XQuery FLWOR Syntax
  - XML Result Sets
  - XML Datatypes
  - XML Namespaces

Cerebra®



Summary Characteristics

- Fixed set of fundamental DL functions
- As foundation underlying FLWOR Syntax
- Each function call returns either:
  - Array of XML elements
  - Array of names
  - A boolean
- Not forcing DL model as XML Doc
- Symmetry between provable and possible scenarios
- Complex classes cater for dynamic querying requirements
- Class based disjunctions, negations separate from boolean logic
- Extensions (External Functions) Straight Forward
- Easy to understand
- Largely avoided co-mingling DL and XQuery semantics
- Well-defined meaning in both DL and XQuery

Cerebra®



A Simple (Non-FLWOR) Example

```
declare default function namespace
   "http://www.networkinference.com/cerebra/functions/3.1"
declare default element namespace
   "http://www.healthy_food.com/pizza”

subClasses(
   intersectionOf(
       Pizza,
       someValues(hasTopping, unionOf(Meat, Fish))
       someValues(hasSize, greaterThan(7))
   ), “all” )
```

Cerebra®



Extensions – "TBox"

- Taxonomic
  - isSubClassOf()
  - subClasses()
  - equivalentClass()
  - superClasses()
  - satisfiableClasses()
  - containingClasses()
  - isDifferentIndividual()
  - isSameIndividual()
  - isDisjointWith()
  - whatIs()
  - subProperties()
  - equivalentProperties()
  - entails()
  - superProperties()
  - classProperty()
  - isFunctional()
  - isInverseFunctional()
  - isTransitive()
  - isSymetric()
  - isInverseOf()
  - isConsistent()
  - describeClass()
  - allIntersectingClasses()
  - subClassesPath()
  - superClassesPath()
  - ontologyProperties()

Cerebra®



Extensions – "ABox"

- Conjunctive
- Variable definitions
  - instances()
  - datavalues()
  - properties()
  - variable()
- Constraints
  - every
  - some
  - models()
  - member()
  - related()
  - matches()
- Semantics
  - models() returns the infinite set of valid models
  - "every" = provable: true for all models in the universe
  - "some" = possible: true for at least one model in the universe

Cerebra®



Extensions – Complex Class

- Complex Class Construction
  - allValues()
  - someValues()
  - unionOf()
  - intersectionOf()
  - complementOf()
  - minCardinality()
  - maxCardinality()
  - equalCardinality()
- Datatype Keywords
  - greaterEqualThan(), lessEqualThan(), greaterThan(), lessThan(), notEqual()

Cerebra®



FLWOR Example
A Conjunctive Query

```
declare default function namespace
   "http://www.networkinference.com/cerebra/functions/3.1"
declare namespace mail ="http://www.labs.com/mail"

for $x in instances()
for $y in datavalues()

where every $model in models() satisfies
        member($x,  mail:MailUser, $model)
    and related($x, mail:emailAddress, $y, $model)
    and matches($y, xs:string,
                    “muzzled@xyz.com”, $model)

return <user>{$x}</user>
```

Cerebra®



Future Directions

- Cerebra is Customer-Driven
  - Our product roadmap is geared to support current and emerging customer use cases
- Our XQuery Roadmap Could Include:
  - Updates/Deletes (for KB changes)
  - Richer XQuery Function and Operator Support
    - String, date, time, duration operators
    - Arithmetic Operators
    - Ordering
    - Result set manipulation (XSLT)
    - Additional External Functions (e.g. sameAs())
  - Full applications can be created using nothing but ontologies and queries - no custom code

Cerebra®



Cerebra®

XQuery Demonstration



Demonstration

- Taxonomic Query – Named Class
- Taxonomic Query – Complex Class
- Taxonomic Query – Super Class of a Named Class
- Conjunctive Query – Named Class
- Conjunctive Query – Complex Classes
- Conjunctive Query – Super Property
- Conjunctive Query – Every vs. Some

Cerebra®



Cerebra®

“Pocket” Slides:

Definitions and Examples



Graphs and Ontology

Q: How can the Semantic Web help?

- "Small Pieces Loosely Joined"
  - Small Pieces: RDF Triples – data graphs
  - Loosely Joined: OWL Models – concept collections **defined with precision**
- Fundamentally Different Than…
  - XML – taxonomic tree structures
  - Java – object structures (frames)
  - RDB – two-dimensional/tabular structures
- From "Trees to Piles of Leafs"
  - RDF Triples, collections of atomic data
  - Ontologies are "views" into the whole
  - "Strict" trees (XML) are not required
    - Ontologies are multi-dimensional, with formal, logically-sound, data semantics

Some Definitions

Semantics
Taxonomy
Schema
Ontology
Inference

Cerebra®

Inference?
Logic With Less Code

BACK

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…

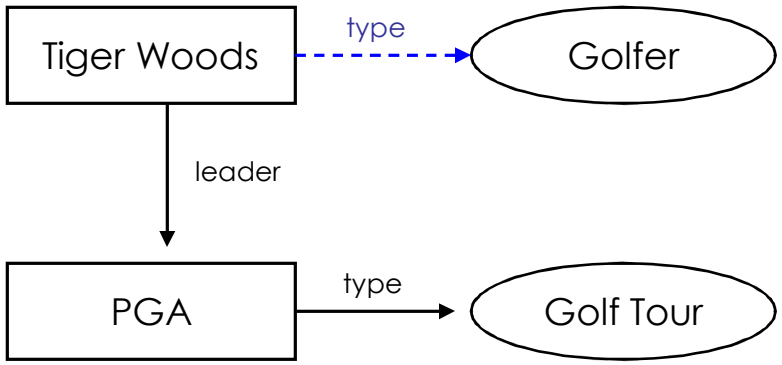
```
<owl:Property rdf:ID="leader">
  <rdfs:subPropertyOf
        rdf:resource="member" />
</owl:Property>

<owl:Class rdf:ID="Golfer">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty
            rdf:resource="member" />
      <owl:someValuesFrom
            rdf:resource="Golf Tour" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

The **leader** of a **tour** is also a **member** of it

A **member** of a **golf tour** is a **golfer**

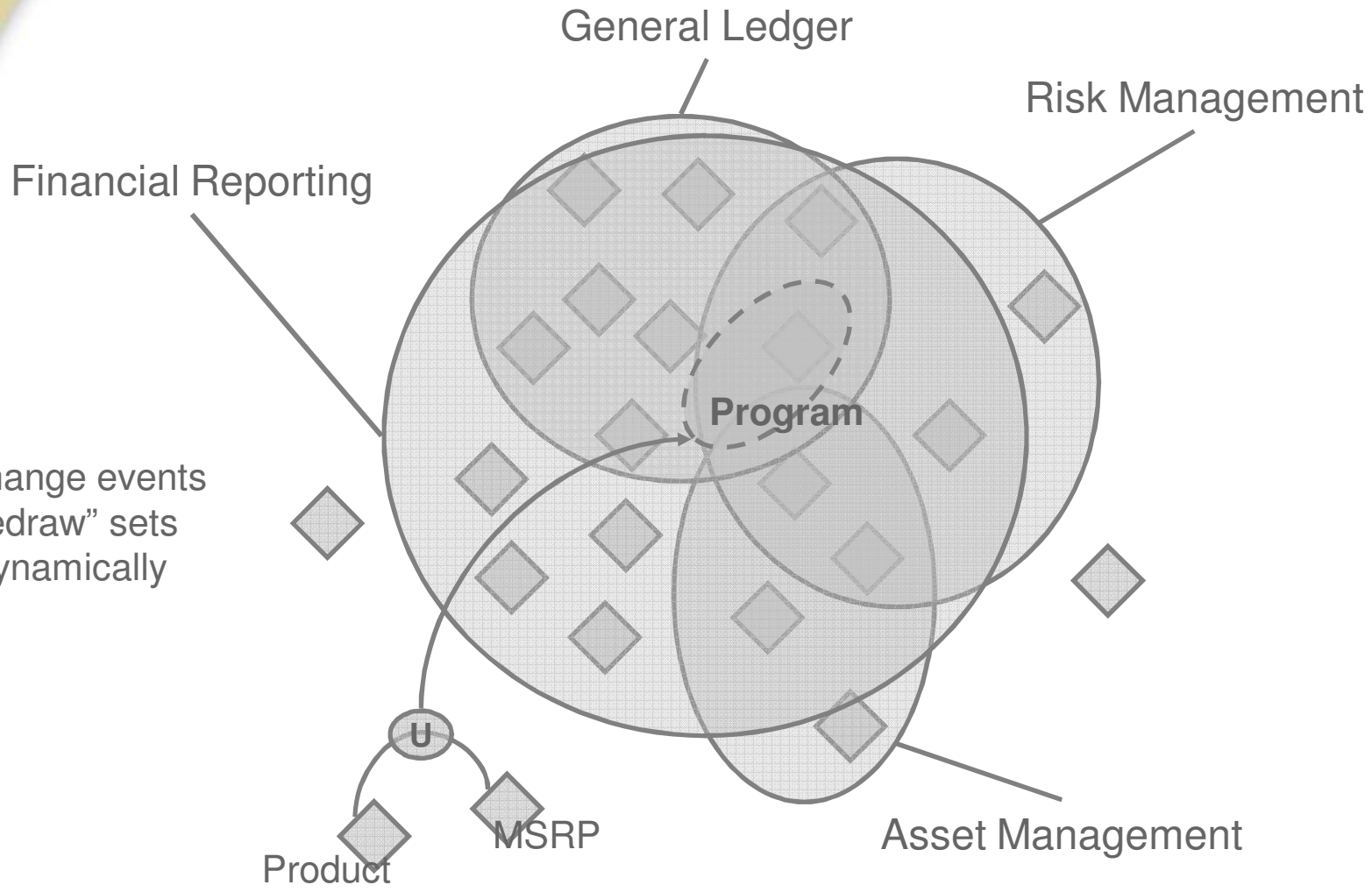
Therefore, the **leader** of a **golf tour** is a **golfer**



Cerebra®

Inference (2)

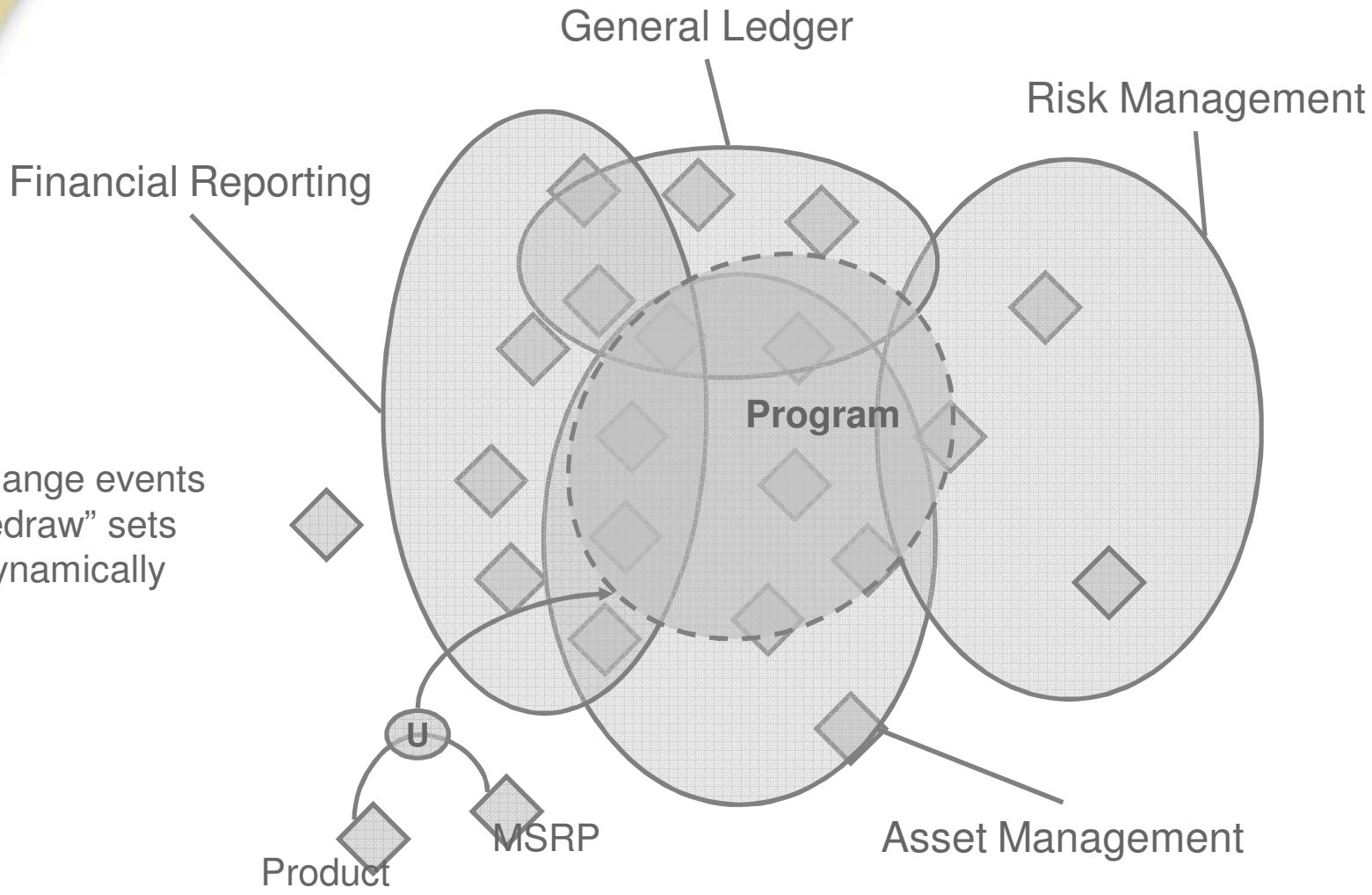
BACK



Cerebra®

Inference (3)

BACK



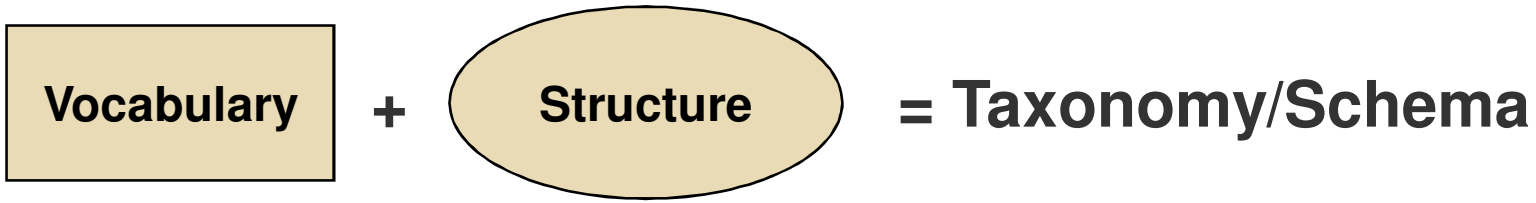
Cerebra®

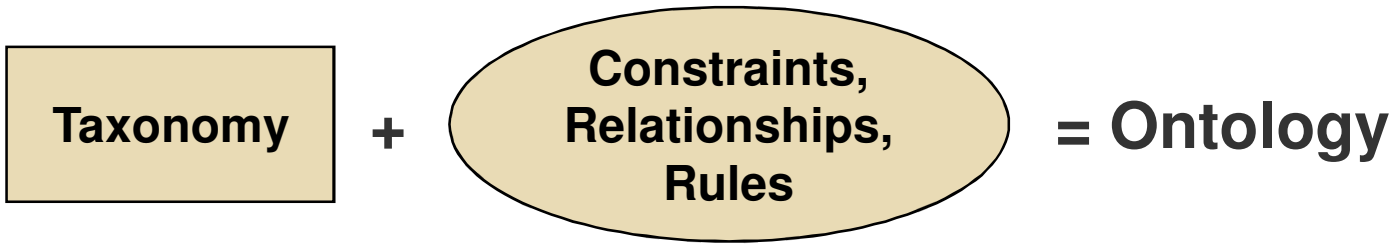


Ontology?

BACK

- Many definitions:
  - Vague – "a specification of a conceptualization"
  - Narrow – "the explicit meaning of terms in a vocabulary and relations between those terms"





Cerebra®

A Little Ironic…

Q: What are semantics?

BACK



"I don't know what you mean by 'glory,'" Alice said.

Humpty Dumpty smiled contemptuously. "Of course you don't—till I tell you. I meant 'there's a nice knock-down argument for you!'"

"But 'glory' doesn't mean 'a nice knock-down argument,'" objected Alice.

"When I use a word," Humpty Dumpty said, in a rather scornful tone, "**it means just what I chose it to mean—neither more nor less**."

Lewis Carroll, Through The Looking-Glass

Cerebra®



Semantics? (OWL Formal)

BACK

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	$\leq nP$	≤1hasChild
minCardinali		

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$^-$

Cerebra®



Cerebra®

Questions?

Alfredo Morales, Eng., MS
Senior Consultant, Global Services
alfredo.morales@cerebra.com