

Semantics and Ontology in IT Data Management



It all started with Aristotle.

But that's a story for another time....

**Semantic Technology Conference
Wilshire Conferences
San Jose, CA.
May, 2007.**

**Dr. Tom Johnston
Mindful Data, Inc.
MindfulData.org
tjohnston@mindfuldata.org
404.819.7604**

Agenda



Part I Basic Concepts

- **Controlled Vocabularies**
- **Taxonomies**
- **Thesauruses**
- **Ontologies**
- **Inference Engines**
- **Semantics**
- **Semantic Interoperability**

Part III Discussion

Part II Ontologies and Databases: an Action Plan

- **Develop a *rigorous* semantics.**
- **Develop a *formalized* semantics.**
- **Develop an *integrated* semantics.**
- **Develop a *late-bound* semantics.**

Part IV Final Thoughts

- **Relational database semantics are what runs the business. They are where the action is.**
- **But those semantics must be improved, formalized, integrated and late-bound.**
- **Specific projects/organizations must not be allowed to establish semantic silos.**



Part I

Basic Concepts

- **Controlled Vocabularies**
 - The foundation for clear definitions.
- **Taxonomies**
 - The backbones of ontologies.
- **Thesauruses**
 - Translating searches for text strings into searches for concepts.
- **Ontologies**
 - Data models, and business rules, on FOPL steroids.
- **Inference Engines**
 - How ontologies are put to work.
- **Semantics**
 - The meaning behind the data.
- **Semantic Interoperability**
 - For federated queries: “Are we speaking the same language?”



Controlled Vocabulary: Definition

Controlled Vocabulary: a set of business terms precisely enough defined that it is clear what they do and do not apply to.

use

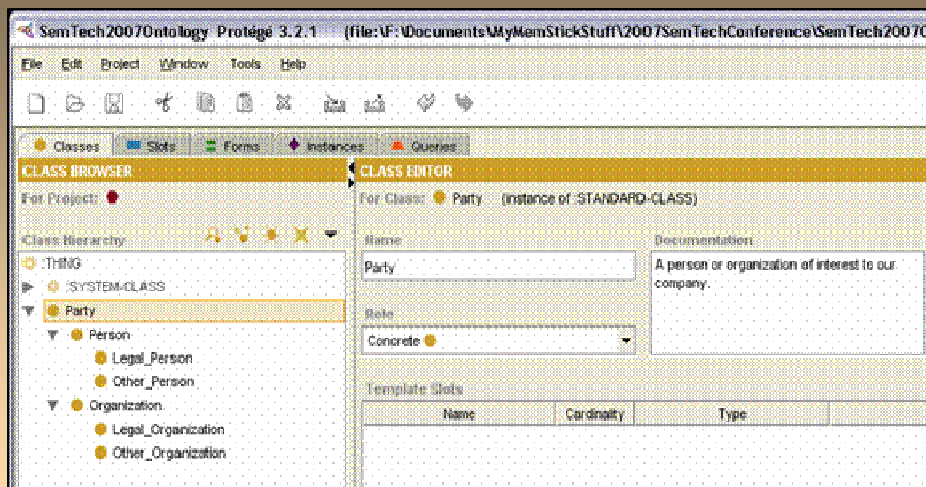
use

to define the elements which make up a taxonomy or ontology in an ontology management tool.

to define the entities, attributes and relationships of a data model.

contains

Data Dictionary: a collection of definitions of the entities, attributes and relationships of a data model.

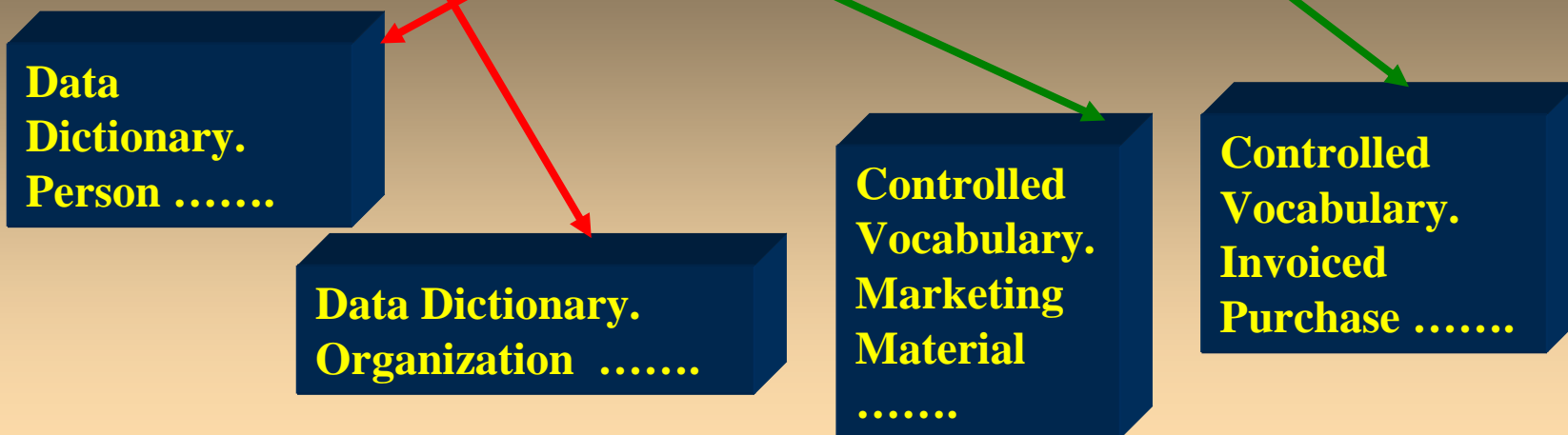




Controlled Vocabularies and Data Dictionaries

Data Dictionary: a bad definition. Customer: a person or organization who has purchased or may purchase a product or service from our company.

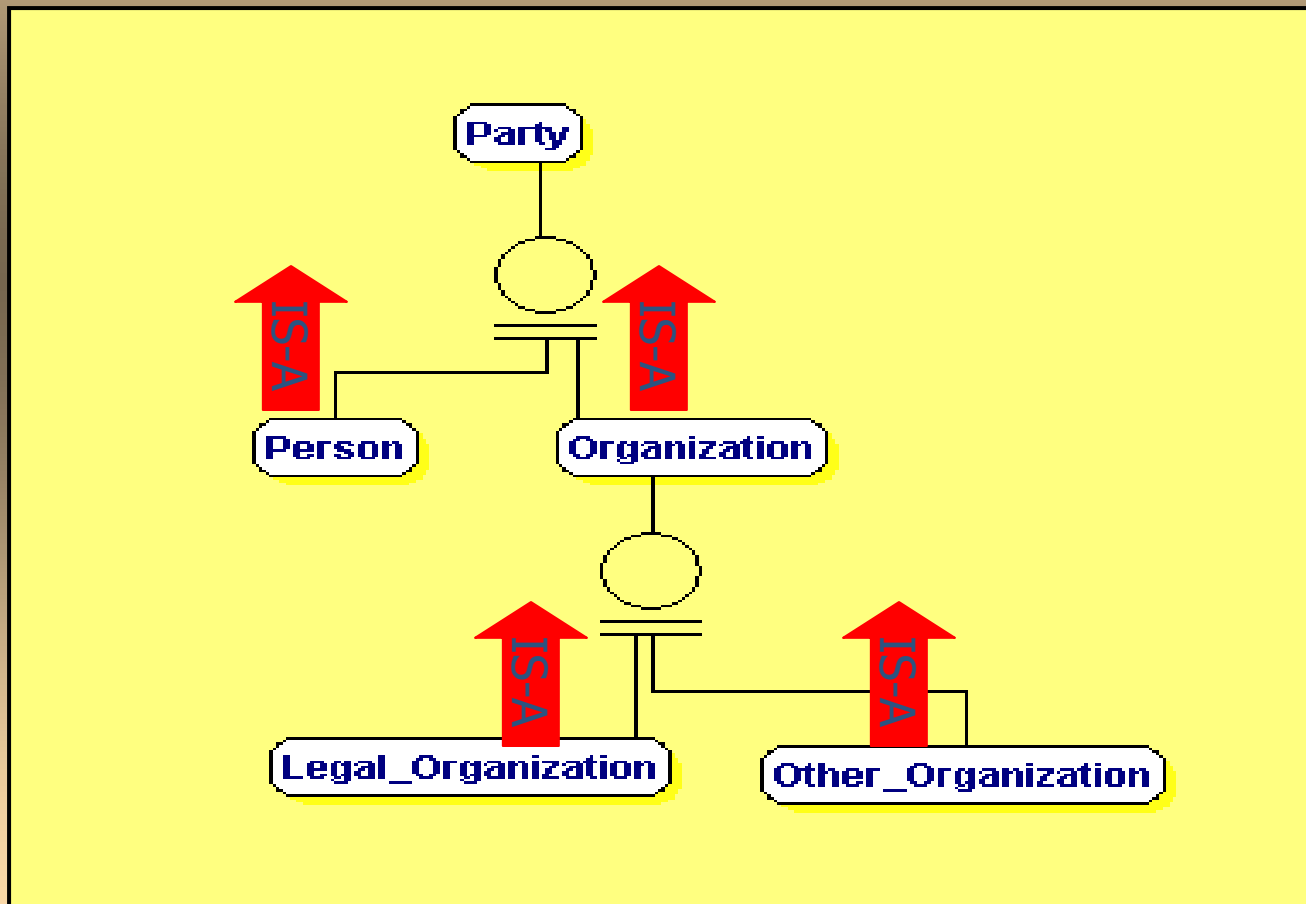
Data Dictionary: a good definition. Customer: a [person] or [organization] who, within the last five years, has either responded to {marketing material} or made an {invoiced purchase} from our company.





Taxonomy: Definition

Taxonomy: a hierarchical structure where the only semantic relationships are IS-A and INSTANCE-OF .



Supertypes and their subtypes are linked by the IS-A relationship.

Rows of tables are the INSTANCES OF those types.



Using a Thesaurus: an Example

A search for the string “SOA” in MS Word documents.

But what we really want is to find all references to Service Oriented Architecture.

That would include the text strings “SOA”, “Service-Oriented Architecture”, “Service Oriented Architecture”, “service-oriented architecture”, etc.

A formal thesaurus would specify all these strings as *synonyms*. A search engine using that thesaurus, and given any one of those text strings, could search for the *concept*.

Search by any or all of the criteria below.

All or part of the file name:

*.doc

A word or phrase in the file:

SOA

Look in:

My Computer

When was it modified?



What size is it?



More advanced options



Back

Search





Ontology: Definition

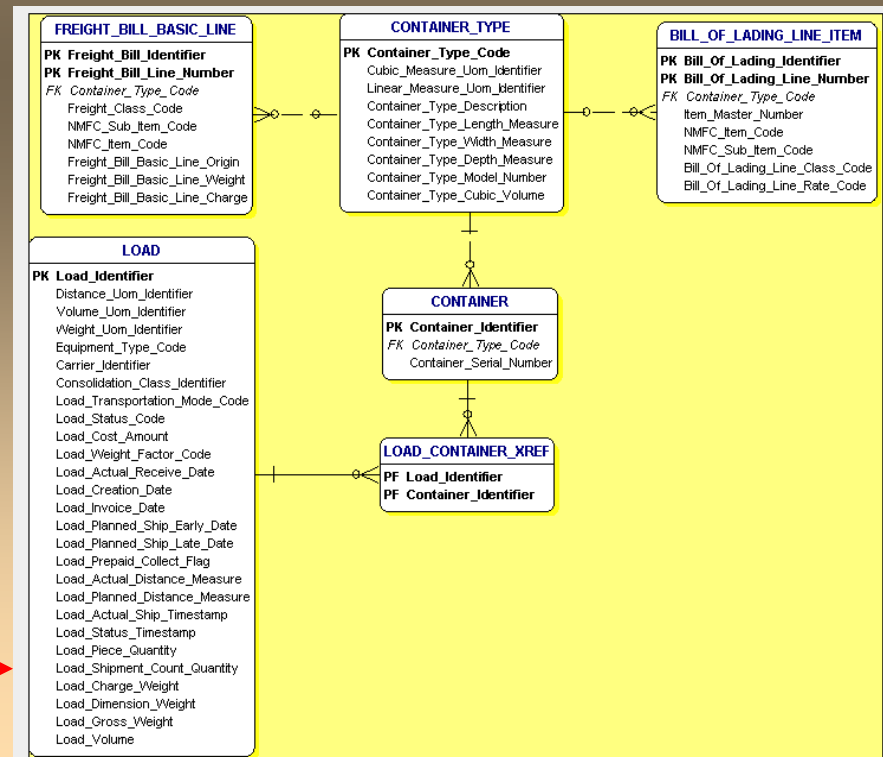
Ontological Commitment: what kinds of things we say there can be instances of. Thus, the tables in our databases. Their rows represent what we say exists.

Ontology: “... an ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to reason about the objects within that domain” (Wikipedia). Uses of ontologies include:

in an ontology management tool.

as an object-oriented data model.

as a relational data model.





Thesaurus: Definition

Thesaurus:

- A book of synonyms, often including related and contrasting words and antonyms.

(Webster's on-line)

Formal Thesaurus:

A thesaurus that supports automated inferencing, i.e. one that can be used by software.

The screenshot shows the thesaurus.com website interface. At the top right is the logo "Thesaurus.com" with a sunburst icon. Below the logo are links for "Register" and "Log In". A navigation menu lists "Dictionary", "Thesaurus", "Encyclopedia", "All Reference", and "The Web", with "Thesaurus" selected. The search results section displays "1 result for: ontology". Below this, it says "View results from: Dictionary | Thesaurus | Encyclopedia | All Reference | the Web". The main entry is "Roget's New Millennium:™ Thesaurus – Cite This Source". The details for "ontology" are:

- Main Entry:** philosophy
- Part of Speech:** noun
- Definition:** principles
- Synonyms:** aesthetics, attitude, axiom, beliefs, conception, convictions, doctrine, idea, ideology, knowledge, logic, metaphysics, ontology, outlook, rationalism, reason, reasoning, system, tenet, theory, thinking, thought, truth, values, view, viewpoint, wisdom

 The source is cited as "Roget's New Millennium:™ Thesaurus, First Edition (v 1.3.1)" with a copyright notice for 2007 by Lexico Publishing Group, LLC. A footer at the bottom of the screenshot reads "Copyright © 2007, Lexico Publishing Group, LLC. All rights reserved."



Ontologies and Relational Databases

In business IT, relational databases and SQL are state of the art in ontology management tools. They have strong *extensional database* management capabilities, but their support for *intensional database* features is minimal.

Extensional Databases.

- What we know of as simply *databases*. The Claims Database, the Shop Floor Management Database, the Personnel Database, etc.

Intensional Databases.

- **The core of each intensional database is a set of FOPL axioms** which fully describe its subset of the enterprise data model.
 - All the information in the DBMS catalog, including **entity, referential and domain integrity constraints**.
 - Augmented with **an FOPL-expressed terminological taxonomy** of the entity, attribute and relationship names for the data model.
 - Including **the state-transformational rules** otherwise hardcoded in programs or stored procedures.
 - Also including **potentially volatile ontological commitments** otherwise hardcoded in database schemas.



Inference Engine: Definition

Inferencing: drawing conclusions from what you already know. More specifically, deriving a conclusion from a set of premises by means of deductive logic.

Example:

- (1) John is older than Mary
- (2) Mary is older than Mike
- (3) John is older than Mike
- (4) John loves Mary
- (5) Mary loves Mike
- (6) John loves Mike

Example:

$[(1 \ \& \ 2) \rightarrow 3]$

BUT

$\sim [(4 \ \& \ 5) \rightarrow 6]$

Inference Engine: software that, without hardcoding, will return (3) to a query, but not (6).

Extensional/Fact Database: a database that contains (1), (2), (4) and (5).

Intensional/Rules Database: a declarative set of rules that defines “older than” as a transitive (and anti-symmetric) relationship, and defines “loves” as an intransitive (and non-symmetric) relationship.



Semantics: Definition

Semantics: in linguistics, what words and sentences mean. In IT, what data means.

Example:

- 885.22. This is just data, a string of characters. We don't know what it means.
- \$885.22. This is a string of characters representing a monetary amount denoted in U.S. dollars. But this meaning is too general to do us much good.
- But in a Customer Invoice table, under the column total-amount, it means the amount due from that customer, for the products on that invoice's line items.

This \$885.22 is still data. But it is data with meaning – some of that meaning formalized, but most of it not formalized. For example, to the DBMS, the label “total-amount” is just a character string associated with that column. The DBMS doesn't “understand” what “total-amount” means. **It can't reason about it.**

But it can manipulate symbols (data) so as to produce the same results as we would produce by reasoning! This is what we mean when we talk about software “reasoning”, or “doing inferencing”.



Semantics: Commentary

In business IT systems – applications, databases, stored procedures, messages – semantics is spread out all over the place.

Most of the **transformation rules** which govern inserts, updates and deletes are hardcoded, either in database triggers, or in application programs.

Most of the **structural rules** which define entities, attributes and relationships are hardcoded in data schemas.

If we consolidate these rules, express them declaratively, and late bind them, then:

The result will be:

- a **codebase** that is run-time bound to declarative rules which describe the constraints on transformations of business data.
- a **database** that is run-time interpreted by reference to declarative rules which designate highly abstract schemas as specific business objects.
- **The combined set of these declarative rules is the *intensional database*.**



Semantic Interoperability: Definition

Database Interoperability: two databases are interoperable if their data can be assembled by a single SQL statement.

Syntactic Interoperability:

- two **atomic data objects** are syntactically interoperable if SQL can JOIN on them.
- two **composite data objects** are syntactically interoperable if SQL can UNION them.

Semantic Interoperability:

- two **atomic data objects** are semantically interoperable if the label and domain elements of one of them can be fully mapped, using formal semantic relationships, to the label and domain elements of the other.
- two **composite data objects** are semantically interoperable if each composing object in one can be mapped, using formal semantic relationships, to a composing object in the other.



Semantic Interoperability: Commentary

The Semantic Net is all about *inter-company semantic interoperability*. It's about how software can “know” what data out there on the web really means.

- Does my supply chain partner mean the same thing by “customer” that I do? (In other words, will a query ranging across both our Customer tables return a consistent result set, or a bag of apples and oranges?)

But the greatest payback will be in improving *intra-company semantic interoperability*.

- Why don't sales totals ever add up exactly right at executive meetings?
- Are individual loading docks locations, or just the buildings that contain them? (What do we mean by “location”?)
- Is material manufactured by another plant in our company, which is also a leaf node on the bill of materials in our plant, really raw material? (What do we mean by “raw material”?)



Part II

Ontologies and Databases: an Action Plan.

- **Develop a *rigorous* semantics.**
 - Clean definitions, paradigmatic and borderline examples.
- **Develop a *formalized* semantics.**
 - Semantics that DBMSs and other inference engines can interpret.
- **Develop an *integrated* semantics.**
 - Consistent vocabularies, taxonomies and ontologies across the enterprise.
- **Develop a *late-bound* semantics.**
 - Semantics expressed declaratively, in a rules engine / intensional database.



Objective 1. Develop a Rigorous Semantics: Data Dictionaries and Controlled Vocabularies

Data Dictionary: a good definition. Customer: a [person] or [organization] who, within the last five years, has either responded to {marketing material} or made an {invoiced purchase} from our company.

Data Dictionary.
Person

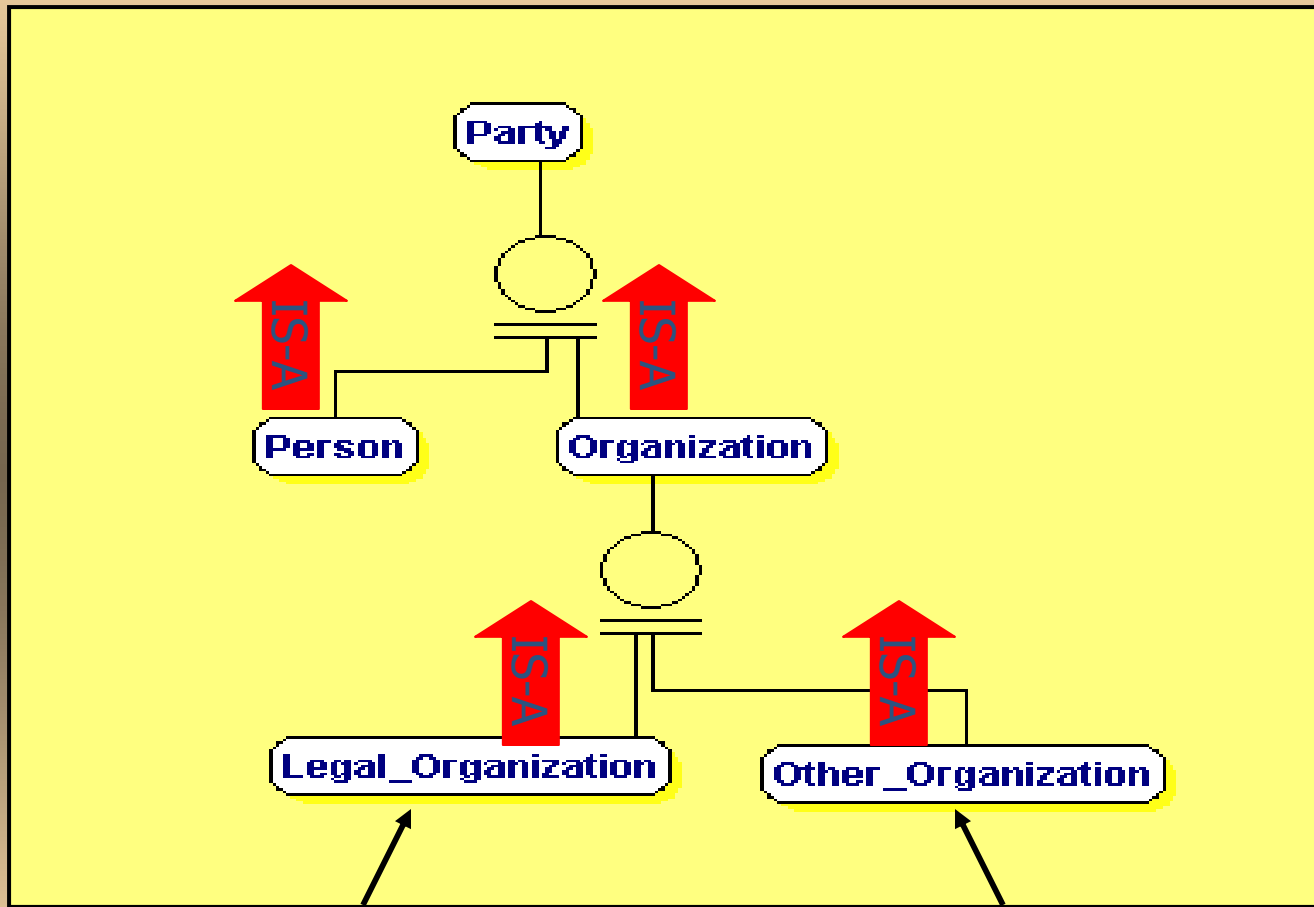
Data Dictionary.
Organization

Controlled Vocabulary.
Marketing Material
.....

Controlled Vocabulary.
Invoiced Purchase



Objective 1. Develop a Rigorous Semantics: Type Hierarchies and Taxonomies



Root and Intermediate Nodes.

Rule-based construction.

Leaf nodes.

List-based construction.

For-profit, non-profit, regulatory. Divisions, departments.



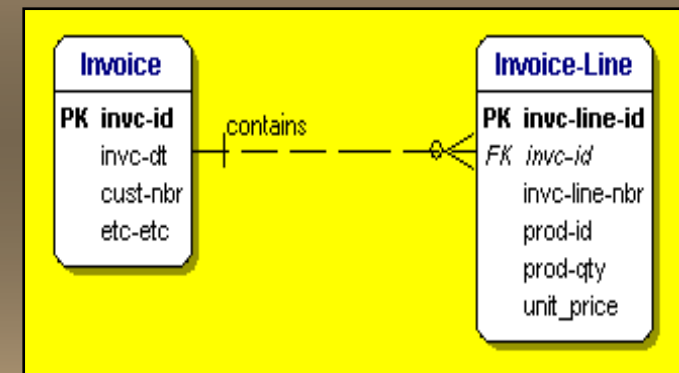
Objective 2. Develop a Formalized Semantics: Database Schemas in FOPL

Example: a minimum cardinality constraint.

In English: “Every invoice line must be contained by one and only one invoice”.

In SQL DDL: expressed as a non-nullable foreign key from invoice line to invoice.

In English/logic: “For all x , if x is an invoice line, then there exists a y such that y is an invoice and y contains x ; and for all z , if z is an invoice and z contains x , then $z=y$ ”.



In FOPL, with IL = “Invoice Line”; I = “Invoice”; and C = “Contains”:

$\forall x[IL(x) \rightarrow \exists y[I(y) \& C(y,x) \& \forall z[I(z) \& C(z,x) \rightarrow (z=y)]]]$

Objective 3. Develop an Integrated Semantics



No semantic silos!

The foundation is the semantics in the data that runs the company. Where is that data? In the production databases, both transactional and analytical.

Once *rigorously* expressed, call the result the “**enterprise semantic base**”.

Once *formally* expressed, call the result the “**enterprise ontology**”.

After this is underway, and not before, is the time to look at industry ontologies, and even upper ontologies. Because now we will have our company’s “run the business” ontology, the foundation on which we will:

- build up to industry ontologies, and then on to SUMO, Cyc or other upper ontologies; and
- build out to other collections of data within our enterprise whose ontologies will likely extend our foundational ontology.



Objective 4: Develop a Late-Bound Semantics

- An intensional database, in which all “rules” are expressed declaratively, in FOPL, and are consequently late-bound to both code and physical schemas.
- Which can therefore be used by a DBMS (or by software for managing unstructured data) to increase inferencing power.
- **Those rules include:**

- **Terminological taxonomies** of the entities, attributes and relationships of the enterprise data model.

- **Fine-grained ontologies** specified in database schemas, extracted from those schemas.

- **Schemas from the database catalog.**

- **Transformation rules** extracted from hardcode.

- **Synonyms and antonyms** expressed as terminological equivalences or as term/negation pairs.

Expressed declaratively, in the intensional/rules database, these semantics are late-bound to the DBMS and to any other inference engine.



Part III

Discussion

How are taxonomies expressed in logical data models?

How are ontologies expressed in logical data models?

How do relational databases and the SQL language constitute an inferencing engine?

What's different about the inferencing engines that are appearing as taxonomy and ontology management tools for business IT?

How can my company get started benefitting from taxonomies and ontologies right away?

How can my company get started preparing for the enhanced inferencing power of ontologically-informed DBMSs?

What's the first thing I should do when I get back to work next week?



Part IV

Final Thoughts.

- **Relational database semantics are what runs the business. They are where the action is.**
- **But those semantics must be improved, formalized, integrated and late-bound.**
- **Specific projects/organizations must not be allowed to establish semantic silos.**



Final Thoughts – 1

Relational database semantics are what runs the business. They are where the action is.

- These are the ontologies that will enable semantic interoperability with our suppliers, customers, regulators and other parties.

- Do these two tables mean the same thing by product? By customer? By calendar? By invoice? By purchase order? By return? Etc, etc.
- If we don't know, we can't coordinate with these other parties.

- Do they mean the same thing? Currently, the burden of insuring this falls primarily on people.

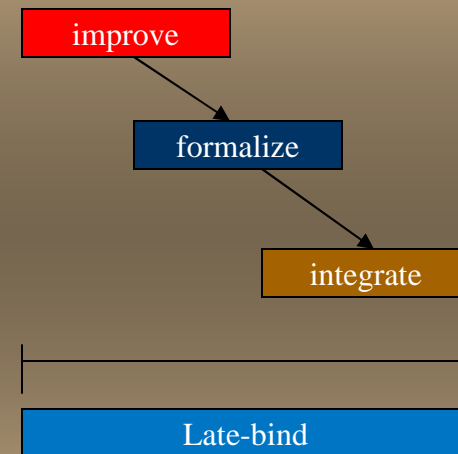
- If we can formalize the ontologies currently imperfectly expressed in our production databases, then the burden of insuring semantic interoperability will fall primarily on inference engines (like enhanced DBMSs).



Final Thoughts – 2

Relational database semantics must be:

- improved (Objective 1)
- formalized (Objective 2)
- integrated (Objective 3)
- late-bound (Objective 4)





Final Thoughts – 3

- An enterprise semantics, based on a controlled vocabulary, taxonomy and ontology, is the first order of business.
 - Its source is the semantics already expressed in the enterprise's run-the-business relational databases.
-
- Projects to apply semantic management to *unstructured* data – scanned and OCR'd documents, emails, pdf files, doc files, etc – must not be allowed to acquire and/or develop their own semantics.
 - They must not become ontology silos.



End of Presentation

Thank you.

I hope this was helpful.

Now it's time to go back home and begin creating an enterprise semantics.

If there's anything else I can do to help, please give me a call.