SingerLinks
Connecting The Dots...

Towards a New Age of Conceptual Data Modeling

Enterprise Data World 2021

Presenter: John Singer

SingerLinks
Connecting The Dots...

Towards a New Age of Conceptual Data Modeling

John Singer

- Recovering Data Architect - 4 decades IT experience in various Data Architecture roles
- Founder of NodeEra Software – Property Graph Data Modeling Software get your free copy https://github.com/jsinger0420/NodeEra
- Author Property Graph Database Fundamentals and Modeling Course on ElearningCurve

- Contact Me:
  - My website - www.singerlinks.com
  - LinkedIn - https://www.linkedin.com/in/singerjohn/
  - Email – john@singerlinks.com

SingerLinks
Connecting The Dots...

Agenda

Can Conceptual Models Save IT From Itself?

- The current state – what are the gaps?
  - Data Modeling
  - Databases – aka the persistence layer
- A look into the future – how this might work
  - Meaning Based Modeling
  - Semantic/Conceptual Persistence

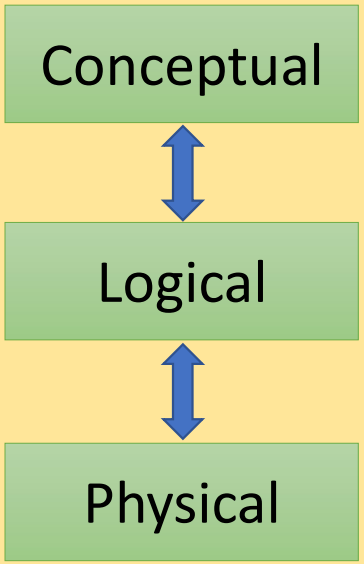
IT technology is bigger/faster but we are still building “unit record” processing systems

SingerLinks
Connecting The Dots...

Current Data Modeling Tools

Confessions Of A Physical Data Modeler

- Data model reflects the physical design
- Conceptual/Logical/Physical model sounds good but doesn't work
- Modeling tools support the approach but
  - Difficult to maintain
  - Polyglot Persistence Layer
  - **Nobody cares**



Abstract
Many to Many OK
Keys / Properties optional
Business oriented
Technology Independent

Add keys/properties
Add "technical" elements
More complete
Not database specific

Database specific
DDL generation
Resolve Sub/Super Types

Nice Process – but E/R Modeling is not sufficient for conceptual models

SingerLinks
Connecting The Dots...

We Need A Conceptual Data Model

Most of this conference deals with topics that exist to fix the lack of conceptual data design:

- Data Catalog / Glossary / Dictionary
- Data Quality
- Data Governance / Strategy
- Data Lineage

Solution Requirements – the Conceptual Database

- Model = Data – model and data defined using the same language
- Technology neutral - easily map to any persistence architecture
- Mirror human behavior - intuitive

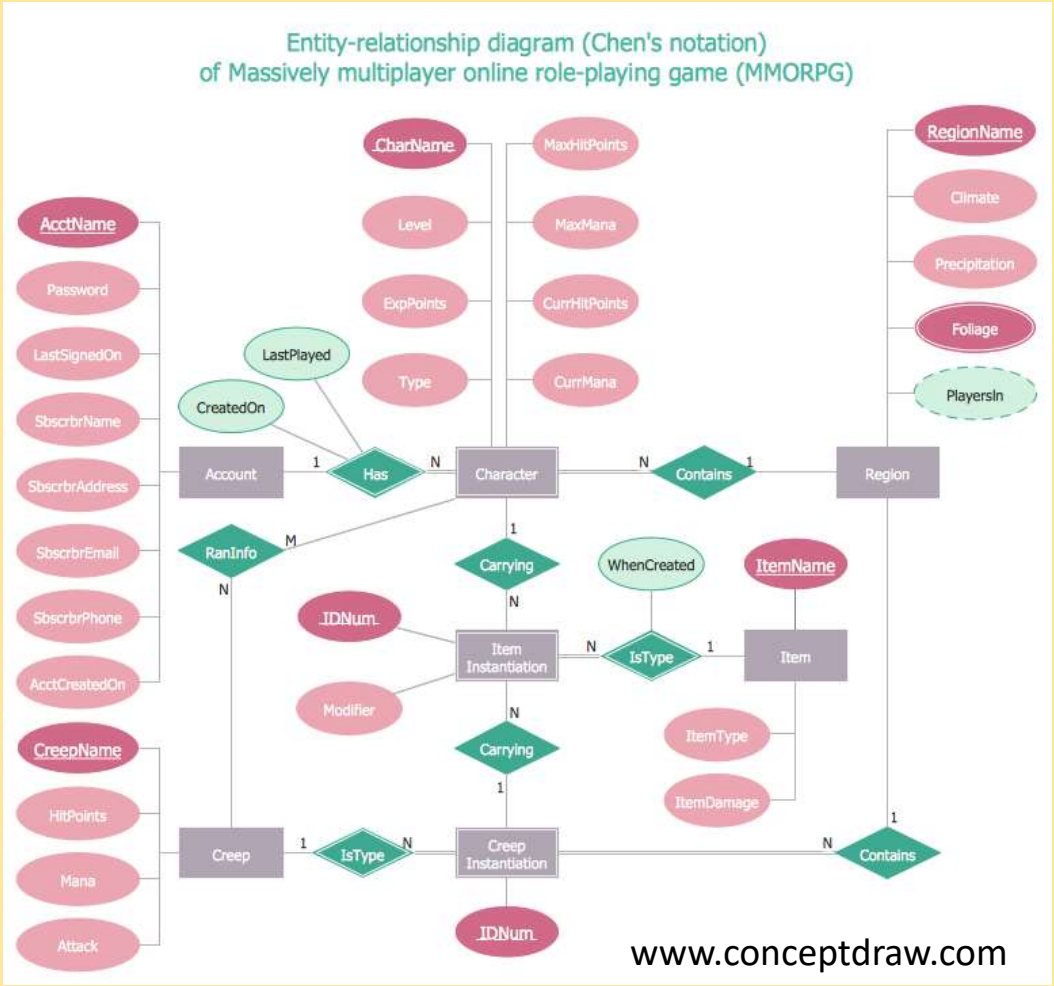
Existing Technologies Exhibit Some Of These Requirements

SingerLinks
Connecting The Dots...

Property/Relationship Centric Data Modeling

" The Entity-Relationship Model-Toward a Unified View of Data "
  - 1975 ACM Publication
- Unify the "Network", "Relational", and "Entity/Set" data models
- "The relational model is based on relational theory … but it may lose some important semantic information about the real world."
- Attributes on Entities and Relationships
- Attribute/Value Set/Value
- Relationship roles

Maybe Chen was right!



SingerLinks
Connecting The Dots...

Early Linguistic Based Modeling

NIAM / ORM

- Natural Language Information Analysis Model
- Object-Role Model (not to be confused with Object Relational Mapping)
- Reflects Language needed to describe the model directly on the diagram
- Domains are modeled directly (Value $)
- Attribute Roles are labeled on the diagram
- Relationships have properties

Rich semantic detail doesn't persist in RDB!

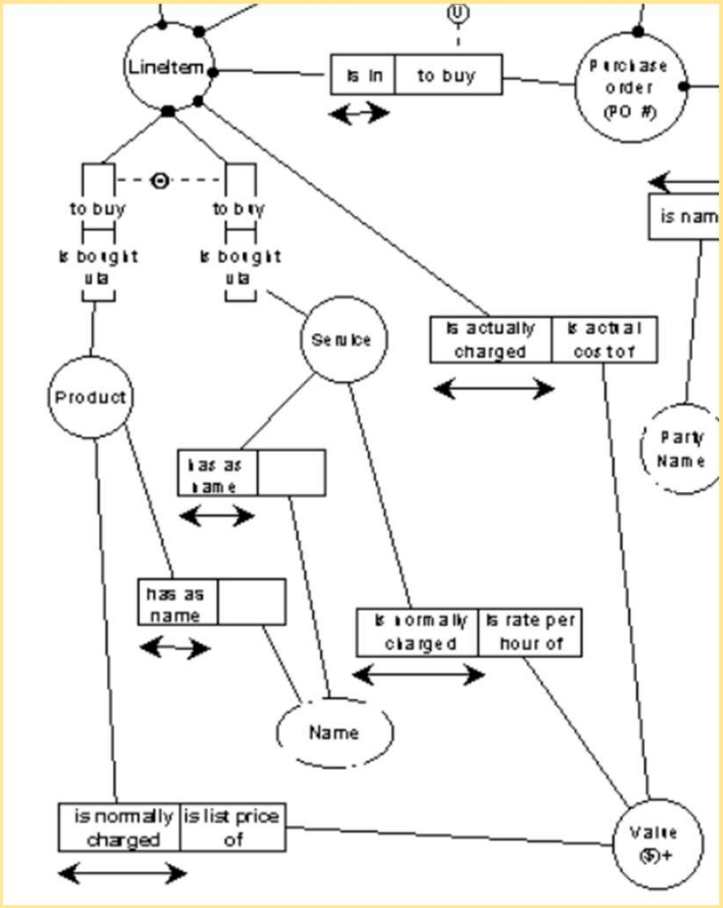
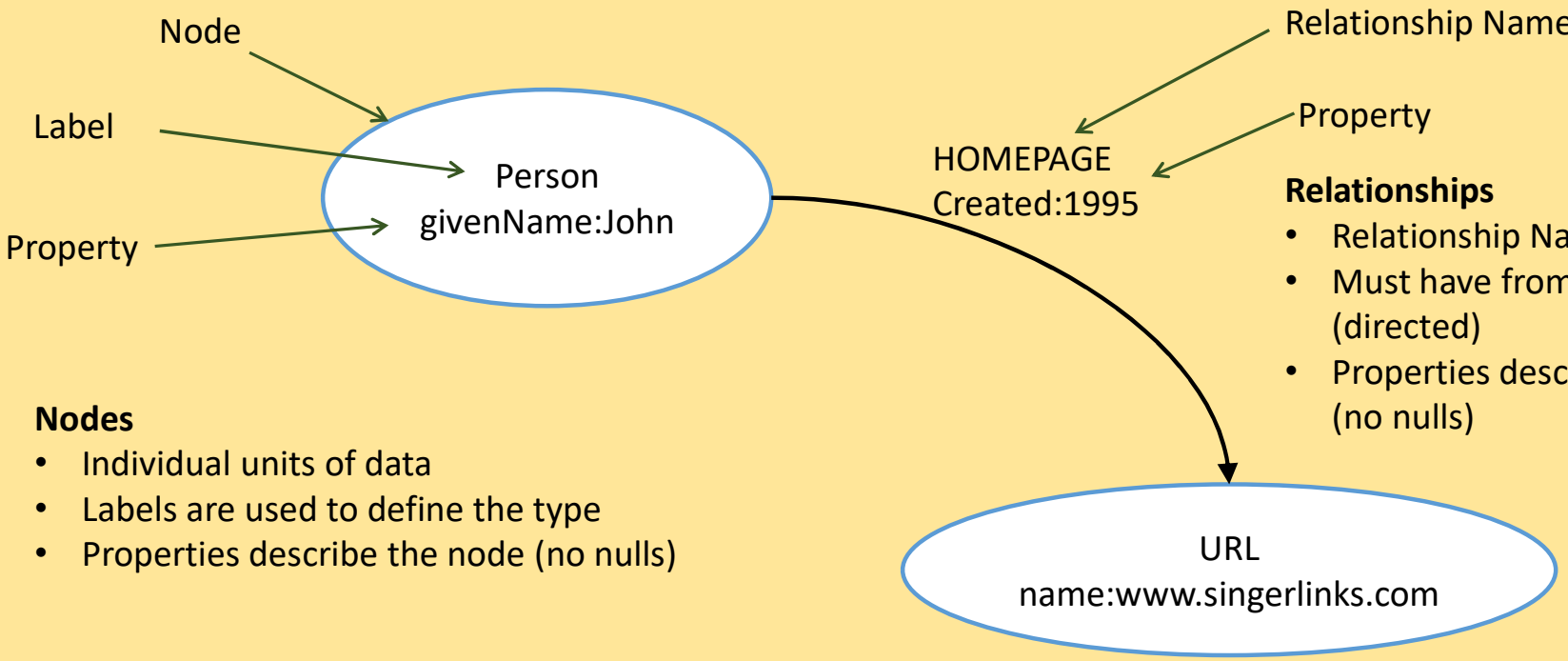


Diagram Copyright © 1997 Essential Strategies, Inc.

SingerLinks
Connecting The Dots...

A New DBMS – The Property Graph



Relationships

- Relationship Name defines the type
- Must have from node / to node (directed)
- Properties describe the relationship (no nulls)

Nodes

- Individual units of data
- Labels are used to define the type
- Properties describe the node (no nulls)

The property graph has intuitive support for Chen/ORM style modeling

SingerLinks
Connecting The Dots...

A New DBMS – The Property Graph

Understanding Property Graph

- Must let go of relational database structure
  - Properties can be modeled as their own entities
  - No Nulls – nulls represented by the absence of a property
  - No Primary Keys required – needed only for performance or uniqueness constraints
  - Relationships are first class citizens – RDBMS's don't store relationships
- Physical Data model is fixed (nodes, relationships, properties)
- Conceptual Data model is not predefined (created at runtime)
- Natural/intuitive support for Chen/ORM style modeling

Semantics defined by convention only

SingerLinks
Connecting The Dots...

Semantic Web Technologies

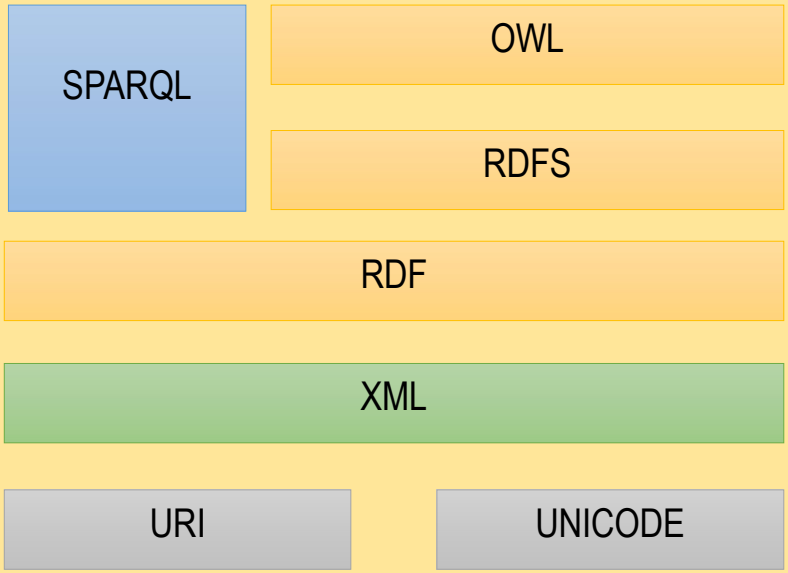
A set of W3C Standards

- Identifiers: URI
- Character Set: UNICODE
- Serialization: XML (others)
- RDF – Resource Description Facility
- RDFS – RDF Schema
- OWL – Web Ontology Language
- SPARQL – Query Language

Goal of the Semantic Web

- Anyone, anywhere can say anything about anything
- Linked Open Data Cloud

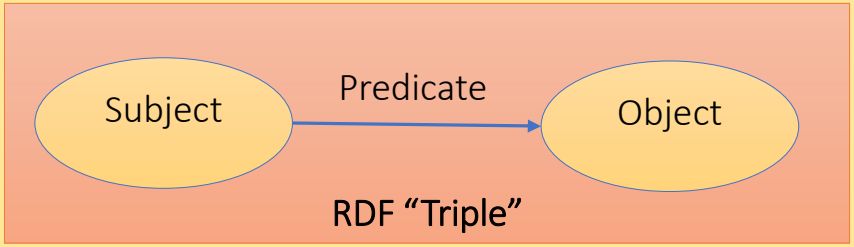
Semantic Web Stack (Simplified)



https://en.wikipedia.org/wiki/Semantic_Web_Stack

SingerLinks
Connecting The Dots...

Semantic Web Database



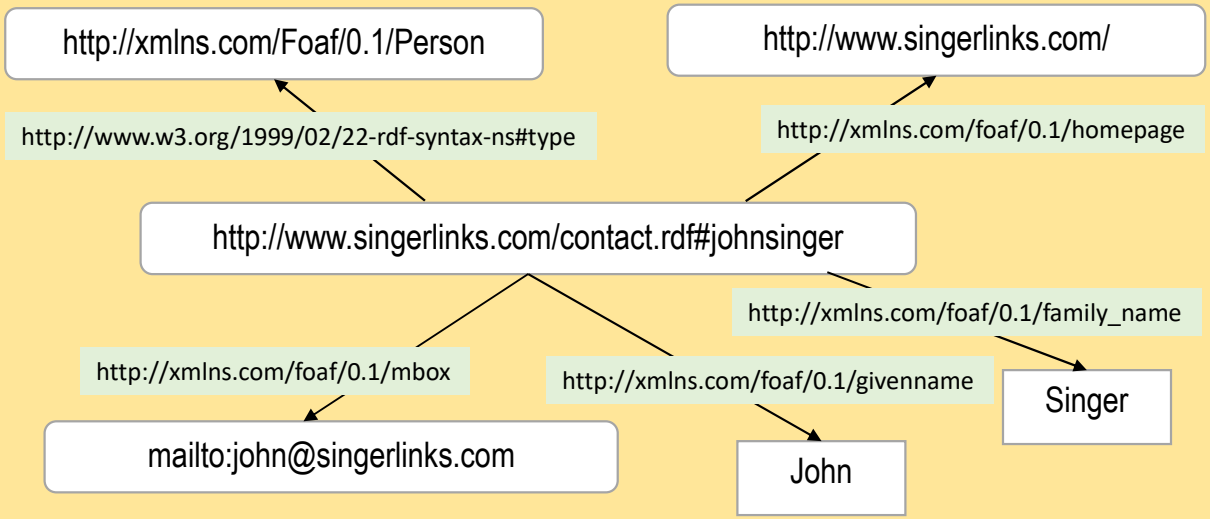
RDF – Resource Description Facility

- A simple graph data model (shown above) is used to express all RDF, RDFS, and OWL statements
- Each subject-predicate-object statement is referred to as a "Triple", hence the term "triple store" when looking at semantic web graph DBMS products
- Each "triple" is an assertion of some fact - a relationship that exists between the Subject and the Object
- The combination of all RDF assertions is called the RDF Graph
- Datatypes are defined by the XML Schema standard

The Semantic Web is based on Logic

SingerLinks
Connecting The Dots...

Semantic Web Graph Database



FOAF – Friend Of A Friend

- Graphical depiction of my contact information using the FOAF vocabulary
- An "Object" can be a literal ("John") or another URI reference to a resource
- Every "property" of the person #johnsinger requires its own predicate and object

```
@prefix : <http://www.singerlinks.com/contact.rdf#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
:johnsinger a foaf:Person ;
   foaf:givenname "John" ;
   foaf:family_name "Singer" ;
   foaf:homepage <http://www.singerlinks.com/> ;
   foaf:mbox <mailto:john@singerlinks.com> .
```

Data model and instance data are defined using the same language

Semantic Web Database

Understanding Semantic Web

- Must let go of relational database structure
- Model properties / domains / ranges – first class citizen
- Instance data can exist without declaring it's "type"
- Physical data model is fixed – "triple"
- Conceptual Data model is rigorously defined – no loss of semantics when persisting data
- Based on Logic:
  - Super-power - Inferencing Engine – infer new facts from given facts
  - Kryptonite – difficult to understand

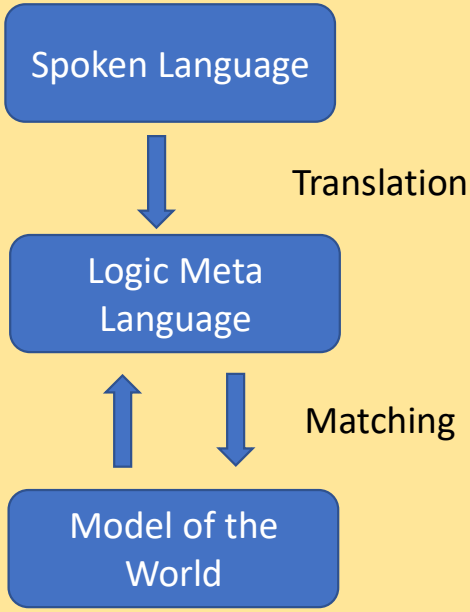
Supports most requirements for Conceptual Database - Ease of Use?

Linguistics - Formal Semantics

Sentence – a grammatically well-formed string of words
Proposition - the meaning carried by the sentence
Formal Semantics – propositions are defined using predicate calculus, a form of logic

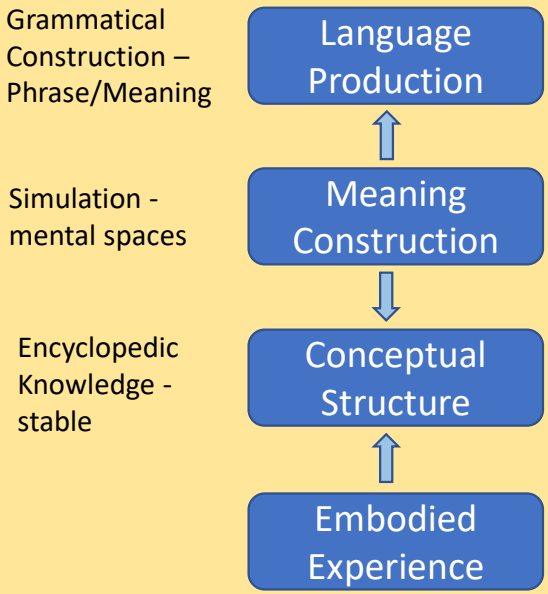
Translation:
- The grammatically well-formed sentence is converted to logic

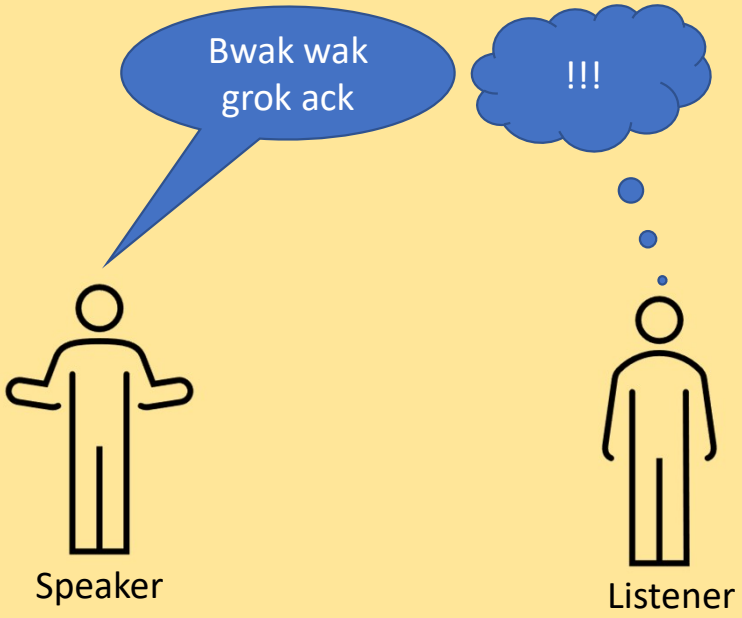
Matching:
- The proposition is compared against the "state of the world" to determine if the statement is true



This approach is computable, but is this really how the brain works?

Conceptual Linguistics





Listener performs the same process in reverse

- Language (words) "Prompt" for meaning construction (simulation)
- Words do not carry all of the meaning
- If you hear something new it gets added to the conceptual structure

Using language to communicate meaning – ultimate in ease of use

SingerLinks
Connecting The Dots...

Why Language Is Important

Language is easy!

- We do this all day long without thinking about how it works – most of the process occurs in the sub-conscious mind
- Words reflect underlying meaning – they need to be managed
- Close the gap between meaning and persisted data
- The system should be able to explain itself
  - All the Data ??? processes occur during design instead of long after implementation
  - Capture and maintain all business context related to each piece of data

We need to bridge the gap between language and logic

Conceptual Database Future

We started with three high level requirements:

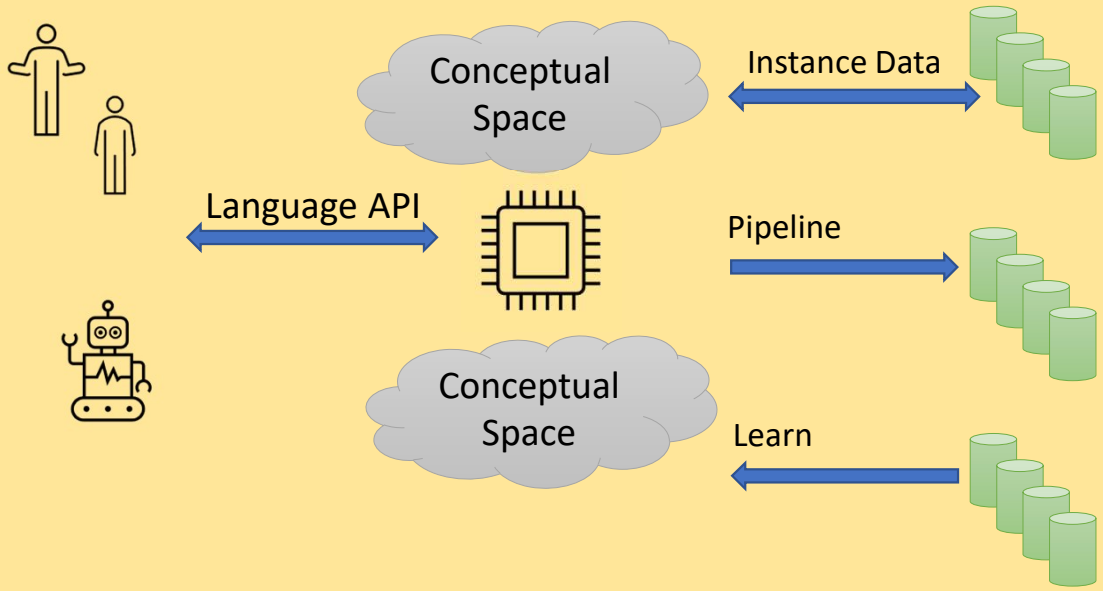
1. Model = Data
2. Technology Neutral
3. Mirror Human Behavior

We looked at existing technologies

1. Conceptual modeling methods – technology neutral
2. Graph and Semantic DBMS – persistence that maps well to cognition/linguistics
3. Language processes – theories of grammar construction

Other related technologies

1. Natural language processing (wordnet)
2. Knowledge Representation Systems



Much of what is needed already exists

All we need is a spark to get this fire started

SingerLinks
Connecting The Dots...

And Finally…

- You can start NOW!
  - Download NodeEra and build a knowledge graph (free)
  - Download Protégé and learn ontology – pizza tutorial (free)

- Self Promotion
  - Hit me up on LinkedIn
  - Visit my webpage – www.singerlinks.com