RESO

#RESO16

# Using OWL for the RESO Data Dictionary

*Chime Ogbuji*

# SPEAKER


## Chime Ogbuji

Property Panorama Inc.

Chief Technical Officer

chime@propertypanorama.com

http://www.linkedin.com/in/chimezie

http://bit.ly/2eOw4DT


**Check out our conference app for more information about this speaker and to view the handouts!**

# Objective

- Introduce a knowledge representation technology (Ontology Web Language) to the Data Dictionary to prevent misunderstanding in communication and ensure semantic interoperability

- Tighten Data Dictionary terminology into a vocabulary of terms which will characterize their meaning by stating how they are interrelated and grouped

http://bit.ly/2eOw4DT

Article Talk Read Edit View history Search

# Tower of Babel

From Wikipedia, the free encyclopedia

*This article is about the Biblical story. For other uses, see Tower of Babel (disambiguation).*

The **Tower of Babel** (Syriac: □□□□ □□□□□, *Maḡdlā d-Bāḇēl* Hebrew: מִגְדַּל בָּבֶל, *Migdal Bāḇēl*) is a Near Eastern etiological myth that is recorded in the Jewish Tanakh's first book (Genesis); it is meant to explain the origin of different languages.[1][2][3][4] According to the story, a united humanity of the generations following the Great Flood, speaking a single language and migrating eastward, came to the land of Shinar (Hebrew: שנער). There they agreed to build a city and a tower "tall enough to reach heaven"; seeing this, God, viewing such behavior as rude and disrespectful, confounded their speech so that they could no longer understand each other and scattered them around the world.

The Tower of Babel has been associated with known structures according to some modern scholars, notably the Etemenanki, a ziggurat dedicated to the Mesopotamian god Marduk by Nabopolassar, king of Babylonia (c. 610 BCE).[5][6] The Great Ziggurat of Babylon was 91 metres (300 ft) in height. Alexander the Great ordered it demolished circa 331 BCE in preparation for a reconstruction that his death forestalled.[7][8] A Sumerian story with some similar elements is told in *Enmerkar and the Lord of Aratta*.[9]

*The Tower of Babel* by Pieter Bruegel the Elder (1563)

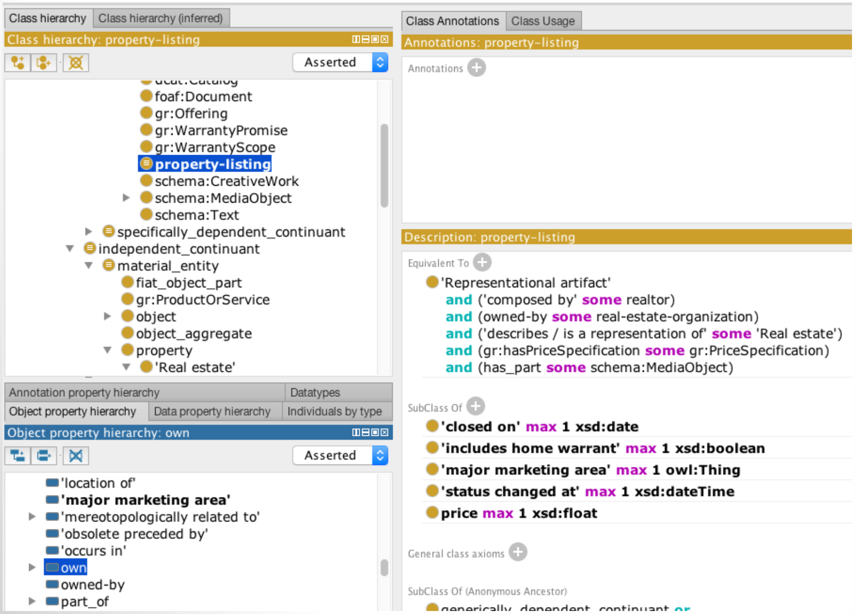
https://en.wikipedia.org/wiki/Tower_of_Babel_(disambiguation)

# Semantic Interoperability

- Requires that all recorded data conforms to some reference terminology […] to interpret and reuse it uniformly in all [participating] information systems
  - Qamar et. al. 2008. *Interoperability of Data Models and Terminology Models: Issues with using the SNOMED CT terminology*

# What is OWL2 and What Are Ontologies?

- OWL2 are mathematical, description logic languages for expressing ontologies
- An **ontology** is a computational artifact made up of precise, descriptive statements about some part of the world: the domain of interest
- The goal is to prevent misunderstanding in communication and to ensure software behaves in a uniform, predictable, and interoperable manner
- An essential part of an ontology, a **terminology** defines a vocabulary of terms and characterizes their meaning by stating how they are interrelated and grouped
- Ontologies are increasingly being used by domain experts in various scientific fields to facilitate representing their domains
- See: W3C's *OWL 2* *Web Ontology Language Primer*




**Definition 5** (*$\mathcal{SROIQ}$* **Concepts, Tboxes, and Aboxes**)
*The set of $\mathcal{SROIQ}$-**concepts** is the smallest set such that*

- *every concept name (including nominals) and $\top, \bot$ are concepts, and,*
- *if $C$, $D$ are concepts, $R$ is a role (possibly inverse), $S$ is a simple role (possibly inverse), and $n$ is a non-negative integer, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, $\exists R.C$, $\exists S.\mathsf{Self}$, $(\geqslant nS.C)$, and $(\leqslant nS.C)$ are also concepts.*

*A **general concept inclusion axiom** (GCI) is an expression of the form $C \mathrel{\dot\sqsubseteq} D$ for two $\mathcal{SROIQ}$-concepts $C$ and $D$. A **Tbox** $\mathcal{T}$ is a finite set of GCIs.*

*An **individual assertion** is of one of the following forms: $a : C$, $(a, b) : R$, $(a, b) : \neg R$, or $a \neq b$, for $a, b \in \mathbf{I}$ (the set of individual names), a (possibly inverse) role $R$, and a $\mathcal{SROIQ}$-concept $C$. A $\mathcal{SROIQ}$-**Abox** $\mathcal{A}$ is a finite set of individual assertions.*

Horrocks, Kutz. & Sattler. 2006.
The Even More Irresistible SROIQ

If you have Protégé, you can load this example ontology from:
http://propertypanorama.com/static/rdf/reso_dd_1.5_core.owl

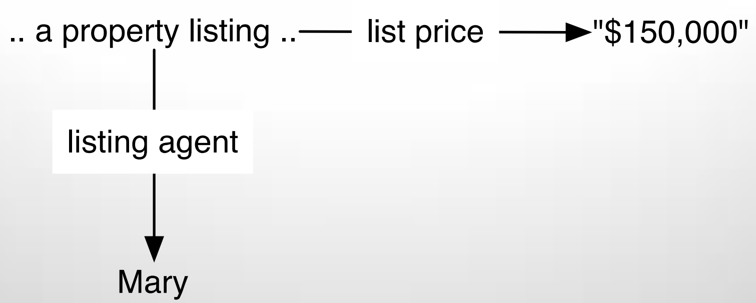
#RESO16

# Classes and Properties

- Statements in OWL ontologies refer to **objects** (or **individuals**) of the world and describe them by putting them into categories
  - "Mary *is a* real estate agent"
- Characterizing relationships between them:
  - "Every real estate agent *sells* property *owned* by an entity"
- These categories are called **classes** and relationships are called **properties**
- **Object properties** relate objects to objects and **datatype properties** associate data values with objects via a named relation:
  - The *price* of a property, The *owner* of a home, the *listing agent* of a property listing
- All individuals, classes, their instances, properties, and their instances are identified by URLs:
  - OWL is primarily meant for use on the web

# Objects of a Property vs. an Object (Individual)

- Every instance of a property connects a **subject** to an **object**
- Below, the property listing is the *object of* the **list price** property and is an object or an individual



# Classes as Folders

By: Quinn Dombrowski

#RESO16

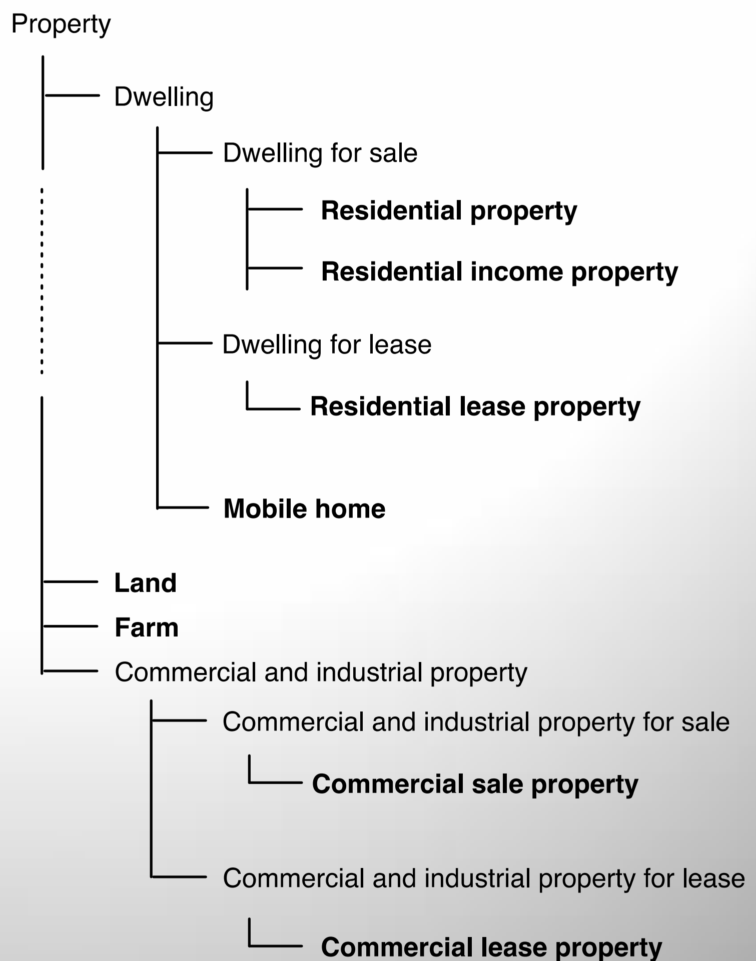
# Class Hierarchies

- Classes group individuals (the **members**) that have something in common in order to refer to them [collectively] as a concept
- Can specify that one class (A) is subsumed under another (B), implying that every member of A is a member of B (General Concept Inclusion)
- These (subsumption) assertions are used to create a hierarchy of classes
- The classes below specialize those above and the classes above are generalizations of those below

# A Property Type (Class) Hierarchy

Extrapolated from [Data Dictionary Terms and Meta Definitions v1.5]()

Any individual that is a *Residential property* is also a *Dwelling for sale*, *Dwelling*, and *Property*



# Property Range Definitions

- Some property definitions imply logical conclusions about the objects of instances of the property. For example:
  - A property can have a range definition, such that object of instances of the property, are either of:
    - A member of a particular class (for object properties)
    - A data value from a particular XML Schema datatype

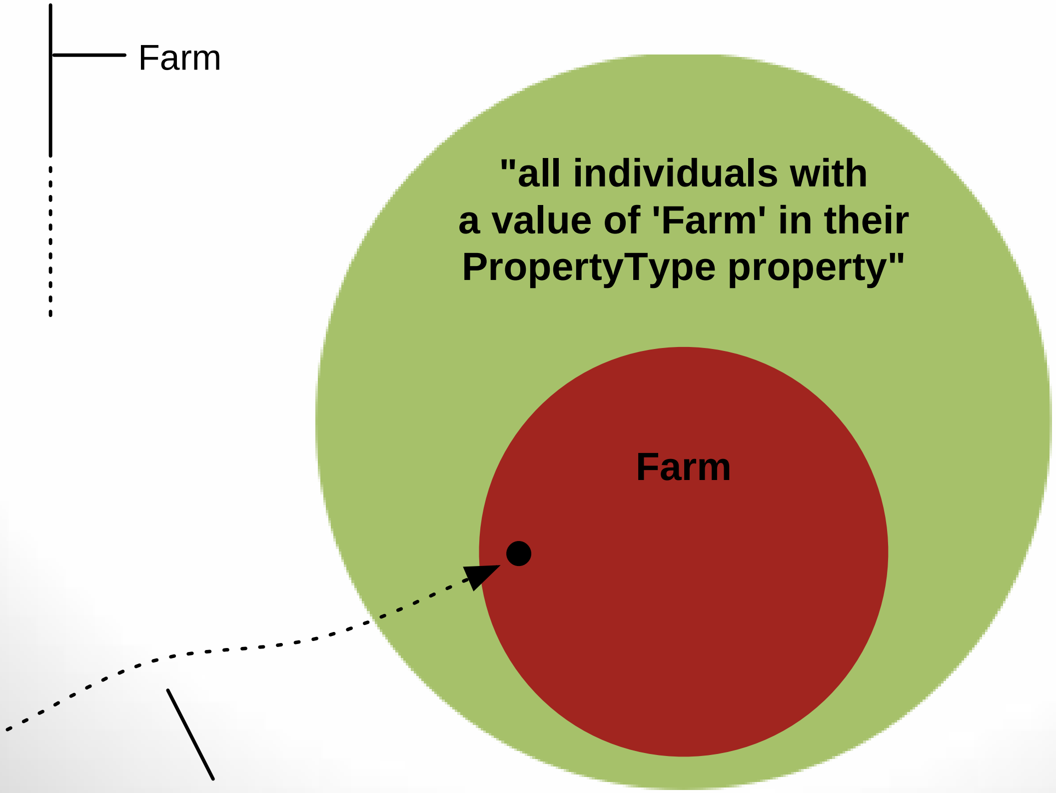
# Data type and Object properties

- Some Data Dictionary **Simple Data Type** field values and range restrictions on the corresponding data type properties:
  - *String* (xsd:string)
  - *Boolean* (xsd:boolean)
  - *Number* (xsd:float)
  - *Date* (xsd:date)
  - *Timestamp* (xsd:dateTime)
- Fields with a Simple Data Type value of ‘*String List*’ can be converted into object properties
  - More on modeling lookup and lookup values later

# Class Restrictions

- An abstract class based on limitations on the properties of its members.  These classes are called **restrictions**
- Named classes can be placed above or underneath restrictions in the class hierarchy in order to specify these requirements on their members

"all individuals with a value of 'Farm' in their PropertyType property"




Any individual that is a member of the class called "Farm" is also a member of the class of all individuals that have a value of "Farm" as their PropertyType property

# Cardinality Restrictions

- Certain restrictions require each member to have at least or no more than a particular number of instances of a particular property
- For each Data Dictionary field with a *RepeatingElement* value of ‘No’, the definition of the resource (property, member, office) can include a cardinality restriction on the property of no more than 1

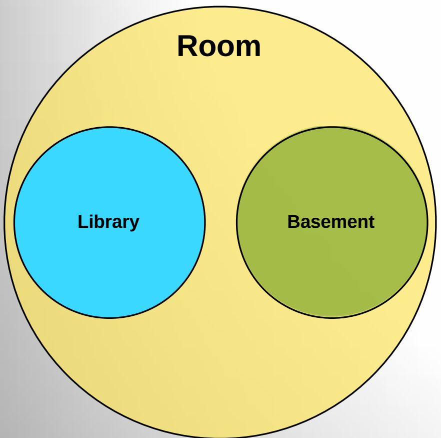

Any individual that is a member of the class called "Property listing" is also a member of the class of all individuals that have no more than one ListPrice property

# Philosophical / Modeling Considerations

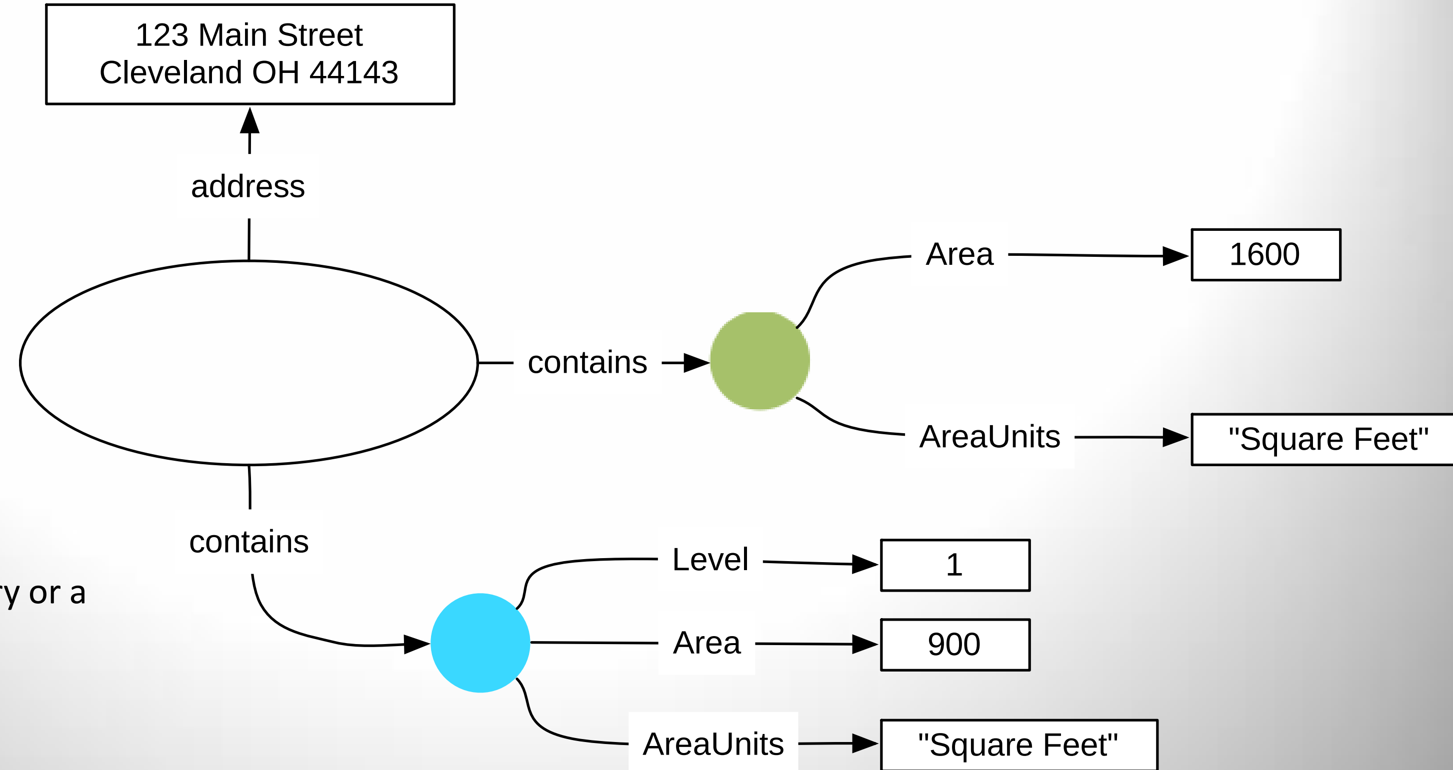
- Possible URIs identifying Data Dictionary classes and properties:
  - http://ddwiki.reso.org/display/DDW/ListPrice+Field
  - http://ddowl.reso.org/ns/100027

- Classes and properties can be designated human-readable labels based on the Standard Field Name ("ListPrice" for example)
  - They are not denoted by the label
- Certain ontological distinctions may need to be explicitly handled
  - For example: a **property** is a physical, independent thing owned by a an entity, but a **property listing** is a representation (physical or digital) of an offer of a property, owned by an agent, for sale or rent

# Representation of Room Fields in OWL




"Everything that is  a library or a basement is a room




"..." The value of a datatype property

An OWL entity that can either be the *subject* of an object or datatype property or the *object* of an object property

An entity that is a member of the OWL class indicated by its color

*Subject* — property → *Object*

# Representing Lookups in OWL

- OWL can be used to represent "value partitions" or "value sets"
- Fields with lookups can be modeled as datatype properties (whose objects are strings) with restrictions on their format
- Or they can be modeled as object properties with range restrictions to a particular enumeration of individuals
  - The object of a **RoomType** property must be Basement, Den, Dining Room, Library, etc.

# Representing Lookups in OWL (continued)

- The advantage of the second approach: the actual human-readable, string representation of a lookup value doesn't have to be specified
  - Each MLS and implementation of the DD can use their own *labeling scheme* for a particular lookup
  - *See:* rdfs:label, skos:prefLabel, skos:altLabel

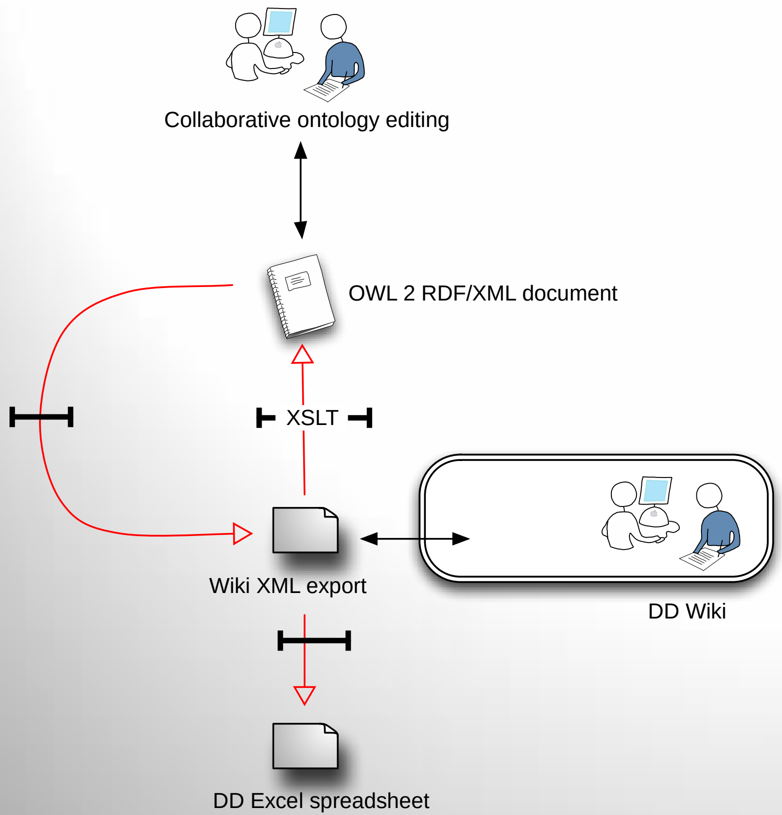
# Using OWL: Benefits & Opportunities

- Reduce or eliminate semantic ambiguity of Data Dictionary terms
- Facilitate the use of logical reasoners or rule-based systems that can:
  - interpret the semantics of a DD OWL ontology and instance data
  - make logical inferences from their interpretation
  - validate data to determine if they conform to the semantics specified by the ontology
- Other OWL tools
- Leveraging other "Semantic Web Technologies"
- Conformant MLS systems can extend a core DD ontology, specifying extensions relevant to local circumstances
- Property listings and virtual tours can be annotated with structured data using OWL vocabulary to facilitate use by third-party web applications and crawlers

# Challenges with Using OWL

- With modeling sophistication comes a steep learning curve
- It is a recent schema technology (compared to relational databases)
  - It is relatively well-supported and widely used in many scientific domains, but not to the extent that relational databases are
- Ontologies are only as useful for standardization as the extent to which they are well-designed in collaboration with domain experts
- Applicable to all new technologies: fear of change

# Using OWL with DD Wiki and Spreadsheet




- DD Wiki XML export file can be transformed to an RDF/XML OWL 2 ontology
- OWL 2 ontology can be edited locally in Protégé or via web-based interface (WebProtege)
- An OWL 2 ontology can be converted to a DD Wiki XML file
- The DD Wiki XML export file is already being used with DD excel spreadsheet and DD wiki

# Questions?