

# Ontologies, Data Models, and Ontology

**Bill Andersen**

Ontology Works, Inc.

Baltimore, MD, USA

**InterOntology 08**

February, 2008

Tokyo

# Outline

- Intuitions – ontologies and data models
- Motivating the distinction
- Attempts at drawing the distinction
- A proposed distinction
- Conclusions

# Motivating the distinction

- What's the difference?
- No good answers so far
- Issues of change, identity, versioning, type/token, use/mention, provenance, security... have been largely ignored
- Ontology of information artifacts in general is needed and **ontology of information system models** in particular

# Ontologies & data models

- Qua formal linguistic **artifacts**
- Similar functions in info systems
  - structuring data
  - storage & retrieval of data
  - data exchange
  - documentation of design decisions

# Distinctions – first attempt

- Spyns, Meersman & Jarrar (2002)
  - Data models are **application-specific** / ontologies are application-independent
  - Ontologies have more **expressive power**
- Meersman (2001)
  - Data models are about the **lexical** / ontologies are not
  - Refutation is bulk of this talk

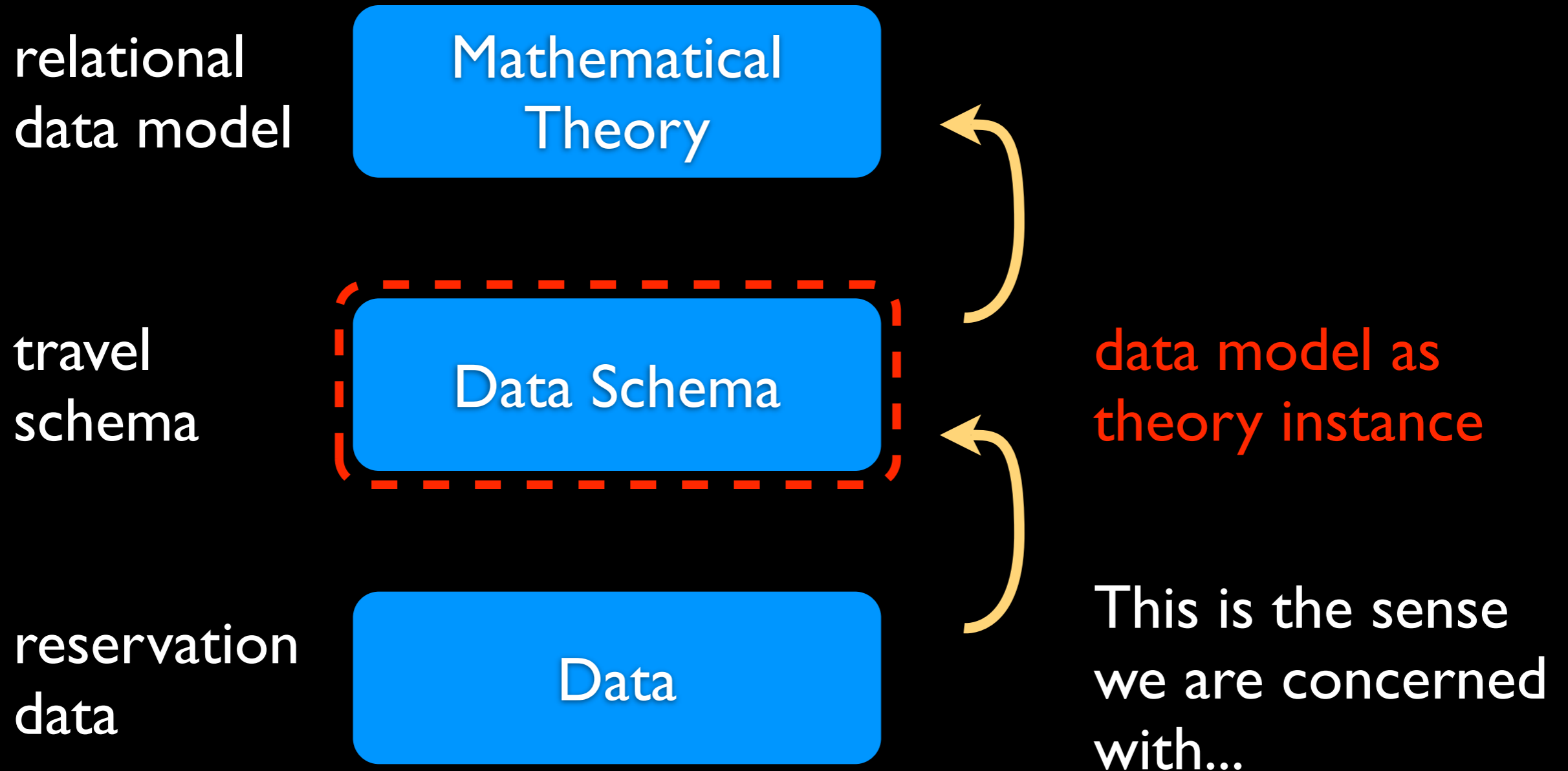
# Application specificity

- Data models are application-specific while ontologies are application-neutral
- Counter-examples are easily found
  - Dublin Core – a “bad” ontology developed for search applications
  - DMTF CIM – a “good” data model developed to support diverse applications

# Logical power

- Ontologies are logically more powerful than data models
- Here we should speak of logical and data modeling **systems** and not the products generated within those systems
- Counter-examples are likewise easily found
  - Dublin Core has no more logical power than ER or OO data models\*

# The term “data model”





# Information artifacts

- Technical artifacts are created to fulfill **designed functions**
- The designed function of both ontologies and data models (theory instance sense) is to express truths about entities by reference to **universe of discourse**
- To what kind of entities do ontologies and data models refer?

# XML(S) example

Data Schema

## Sales.xsd

```
<xsd:complexType name="SaleType">
  <xsd:sequence>
    <xsd:element name="shipTo" type="USAddress"/>
    <xsd:element name="billTo" type="USAddress"/>
  </xsd:sequence>
  <xsd:attribute name="orderDate" type="xsd:date"/>
</xsd:complexType>
<xsd:complexType name="USAddress">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="country" type="xsd:NMTOKEN"
    fixed="US"/>
</xsd:complexType>
```

# XML(S) example

## JanuarySales.xml

```
<sale orderDate="2007-1-20">  
  <shipTo country="US">  
    <name>Alice Smith</name>  
    <street>123 Maple Street</street>  
  </shipTo>  
  <billTo country="US">  
    <name>Robert Smith</name>  
    <street>8 Oak Avenue</street>  
  </billTo>  
</sale>
```

Data

# OWL/RDF example

## Sales.owl

Ontology

```
<owl:Class rdf:ID="SaleType"/>
<owl:Class rdf:ID="USAddress"/>
<owl:ObjectProperty rdf:ID="shipTo">
  <rdfs:domain rdf:resource="#SaleType">
  <rdfs:domain rdf:resource="#USAddress">
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="billTo">
  <rdfs:domain rdf:resource="#SaleType">
  <rdfs:domain rdf:resource="#USAddress">
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="orderDate">
  <rdfs:domain rdf:resource="#SaleType">
  <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="name">
  <rdfs:domain rdf:resource="#USAddress">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="street">
  <rdfs:domain rdf:resource="#USAddress">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

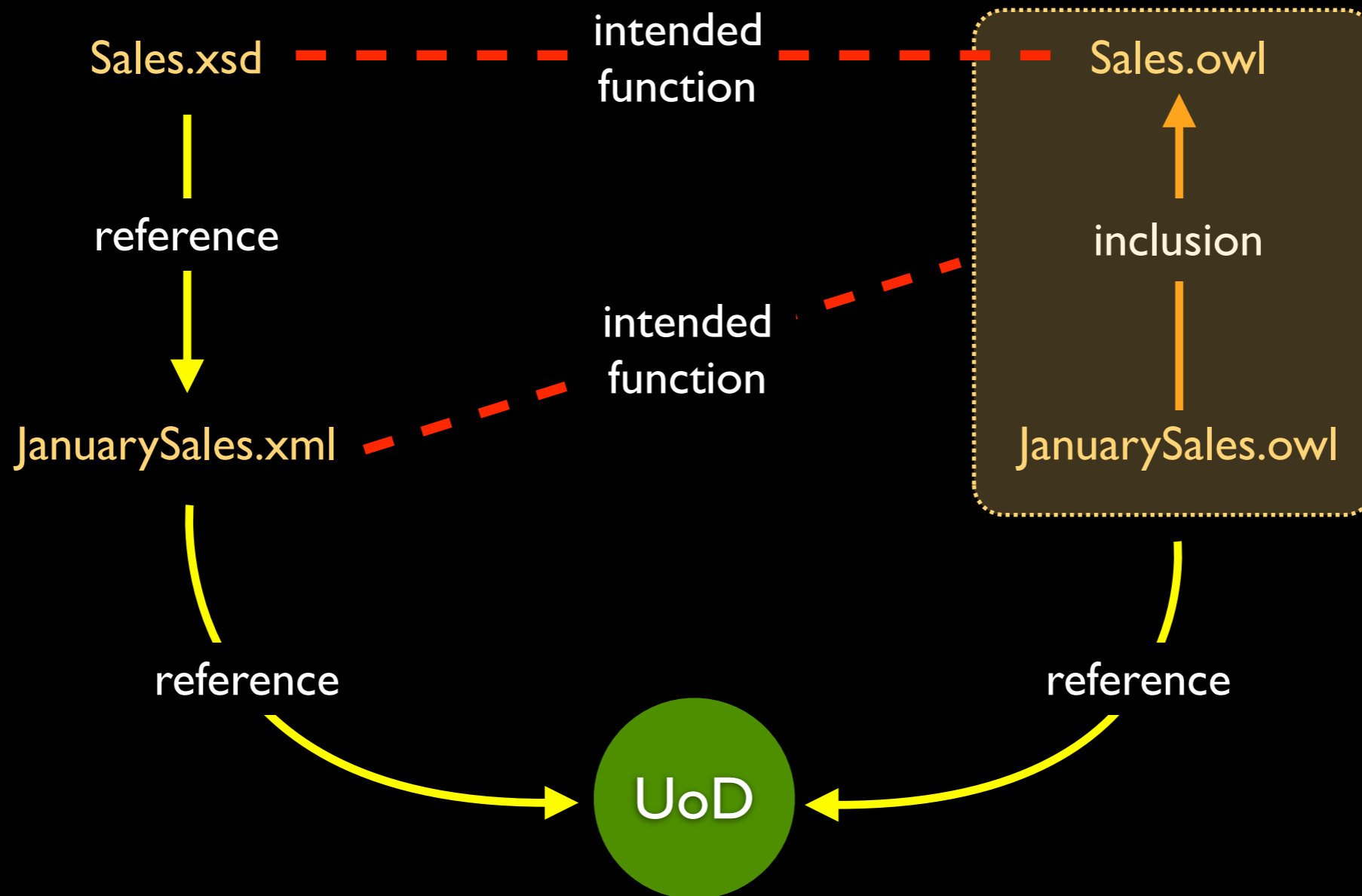
# OWL/RDF example

## JanuarySales.owl

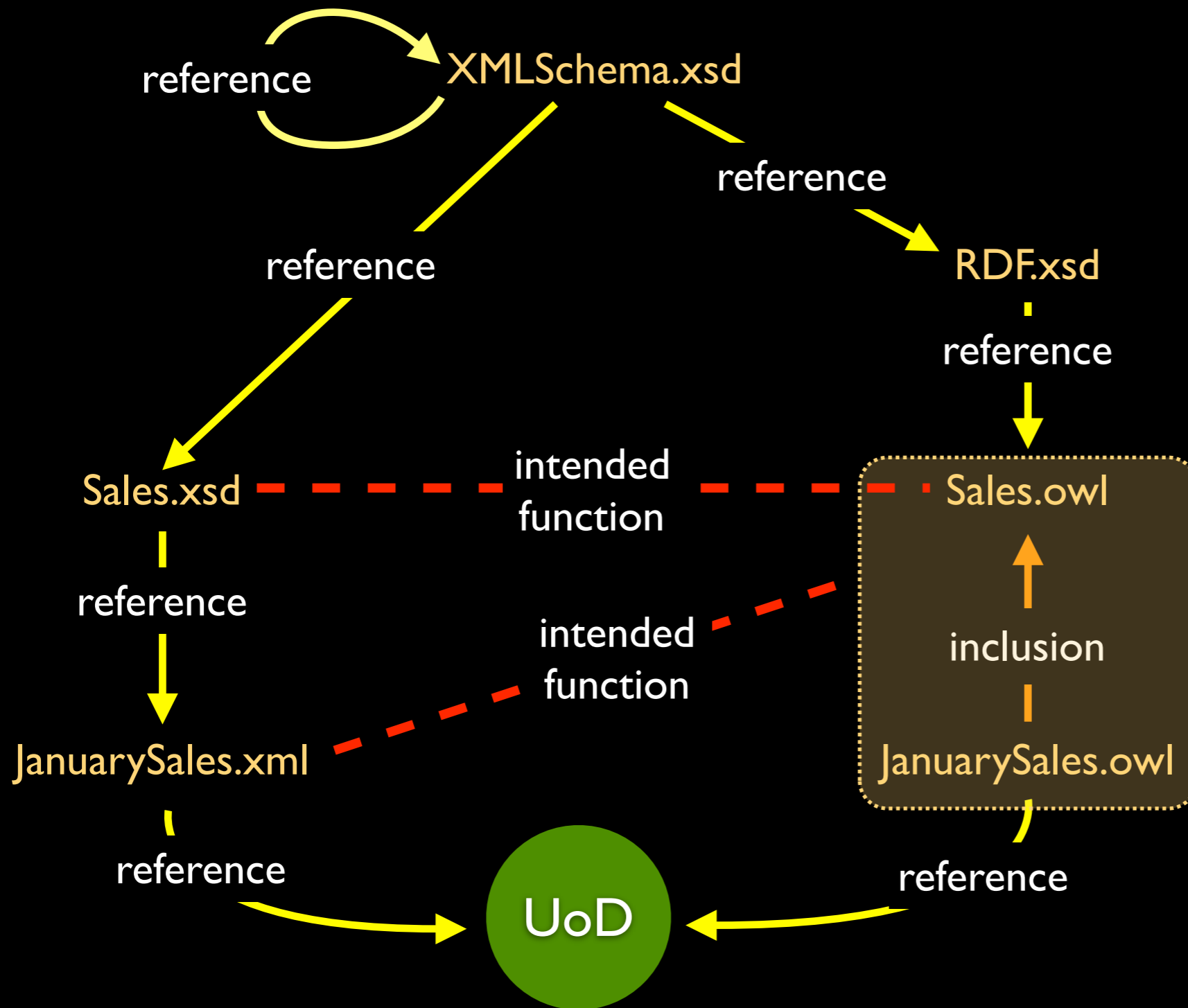
```
include "Sales.owl"
_:s rdfs:type sales:SaleType.
_:s orderDate xsd:dateTime^^"2007-1-20".
_:a1 rdfs:type sales:USAddress.
_:s sales:shipTo _:a1.
_:a1 sales:country "US".
_:a1 sales:name "Alice Smith".
_:a1 sales:street "123 Maple Street".
_:a2 rdfs:type sales:USAddress.
_:s sales:billTo _:a2.
_:a2 sales:country "US".
_:a2 sales:name "Robert Smith".
_:a2 sales:street "8 Oak Avenue".
```

“Data”

# Reference and intention



# The full(er) picture



## Formal Systems:

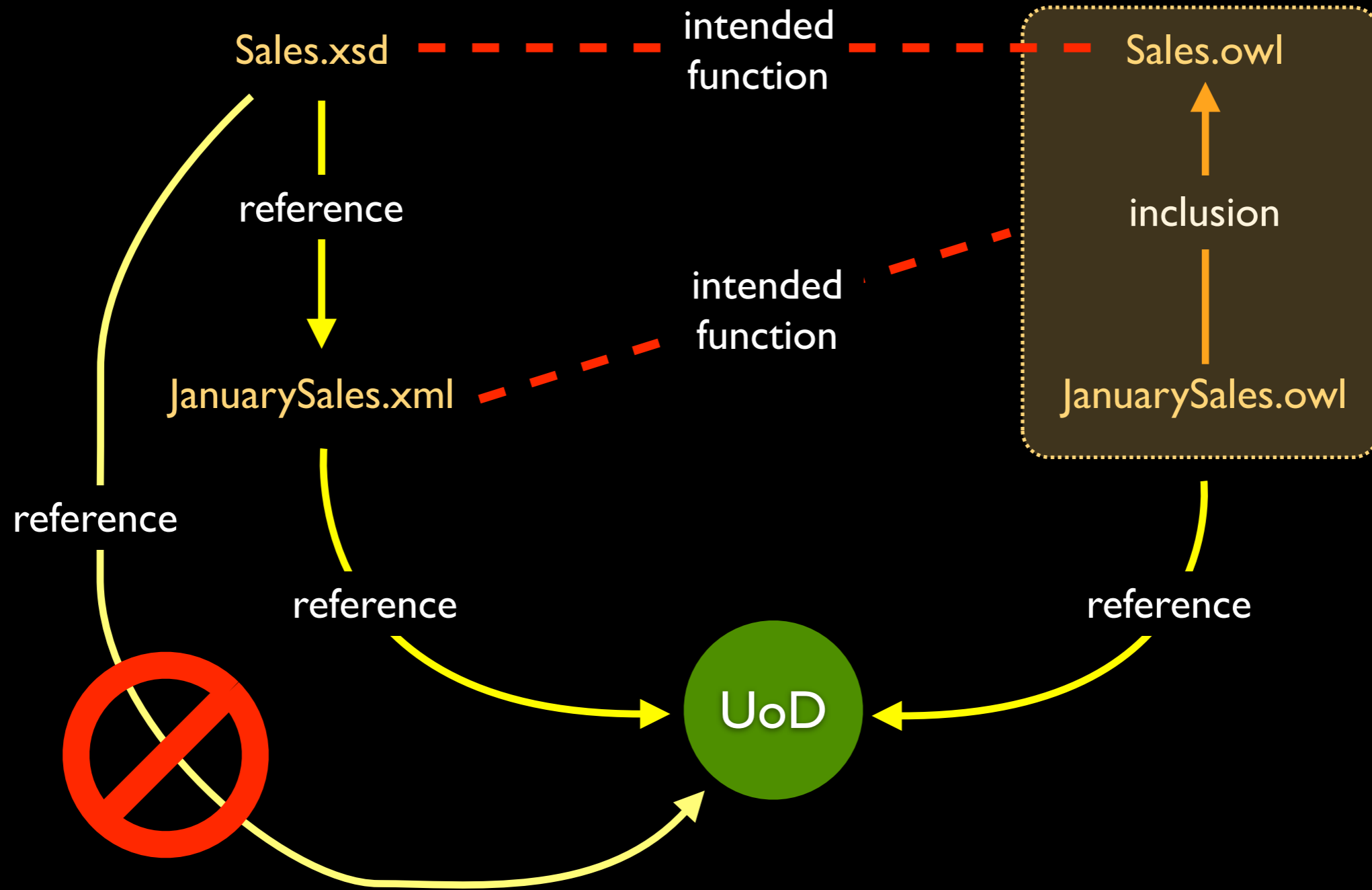
XML Schema  
RDF (syntax)  
OWL/RDF

# Dilemma

- The door is open to interpret the class and property terms of Sales.owl as universals in the UoD
- If the terms in Sales.xsd are used to refer to these same universals, they cannot also refer to the syntactic elements of JanuarySales.xml
- Unless we're willing to dispense with the principle of **univocity**

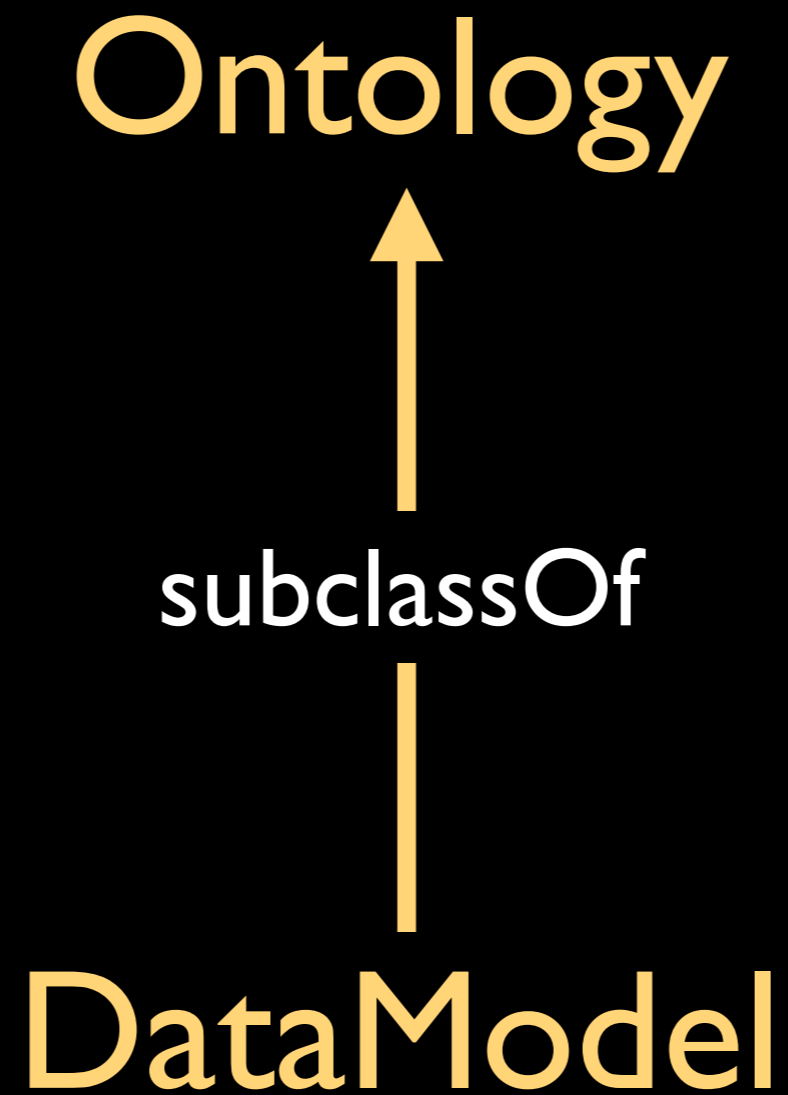


# Can't have it both ways



# Conclusions

- Meersman & Spyns are wrong on
  - The syntactic distinction
  - The logical power distinction
  - The application specificity distinction
- Data models quantify only over syntax.  
That does not make them non-ontologies
- Data models **are** ontologies ... of syntax



# Questions?