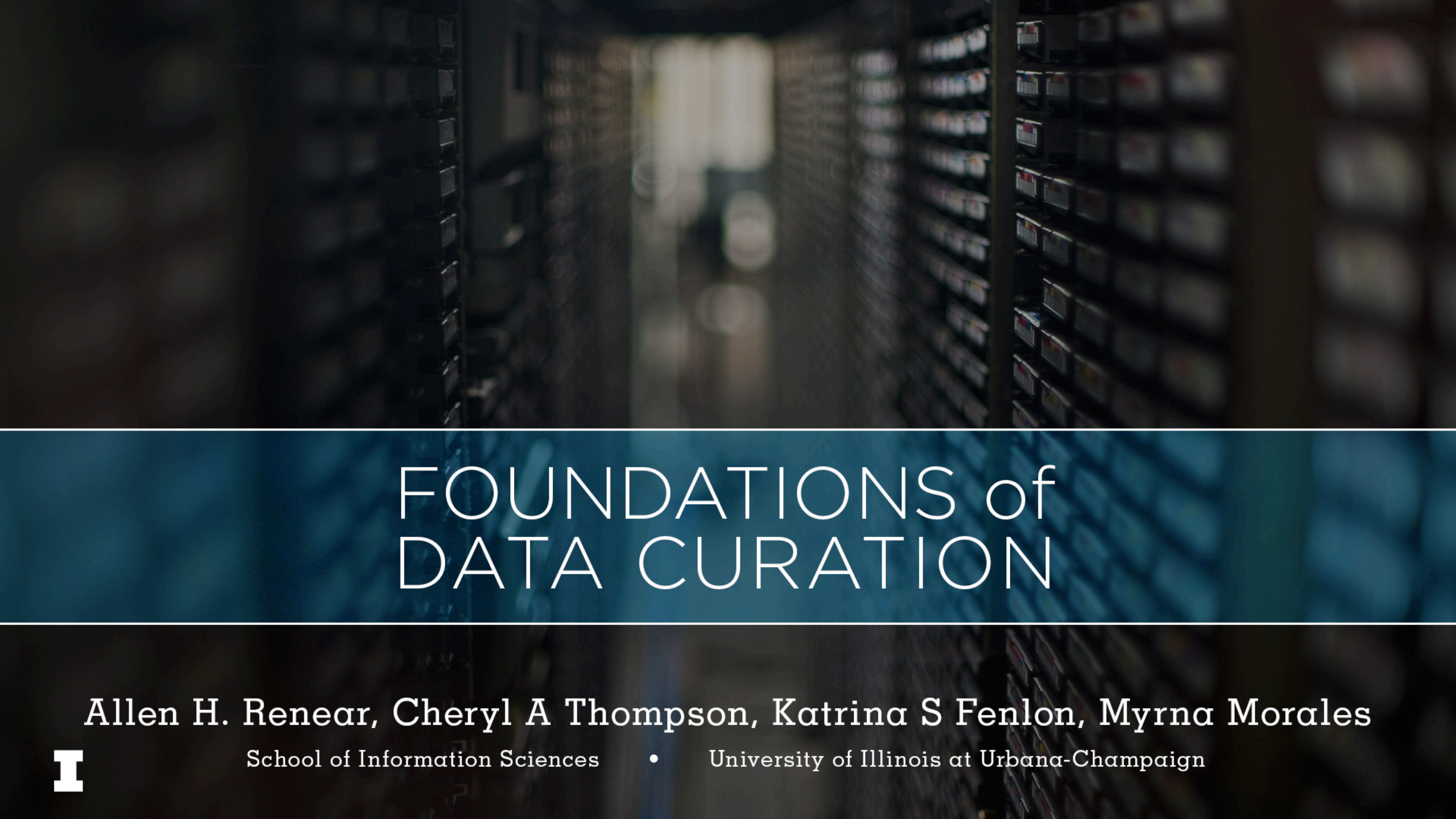
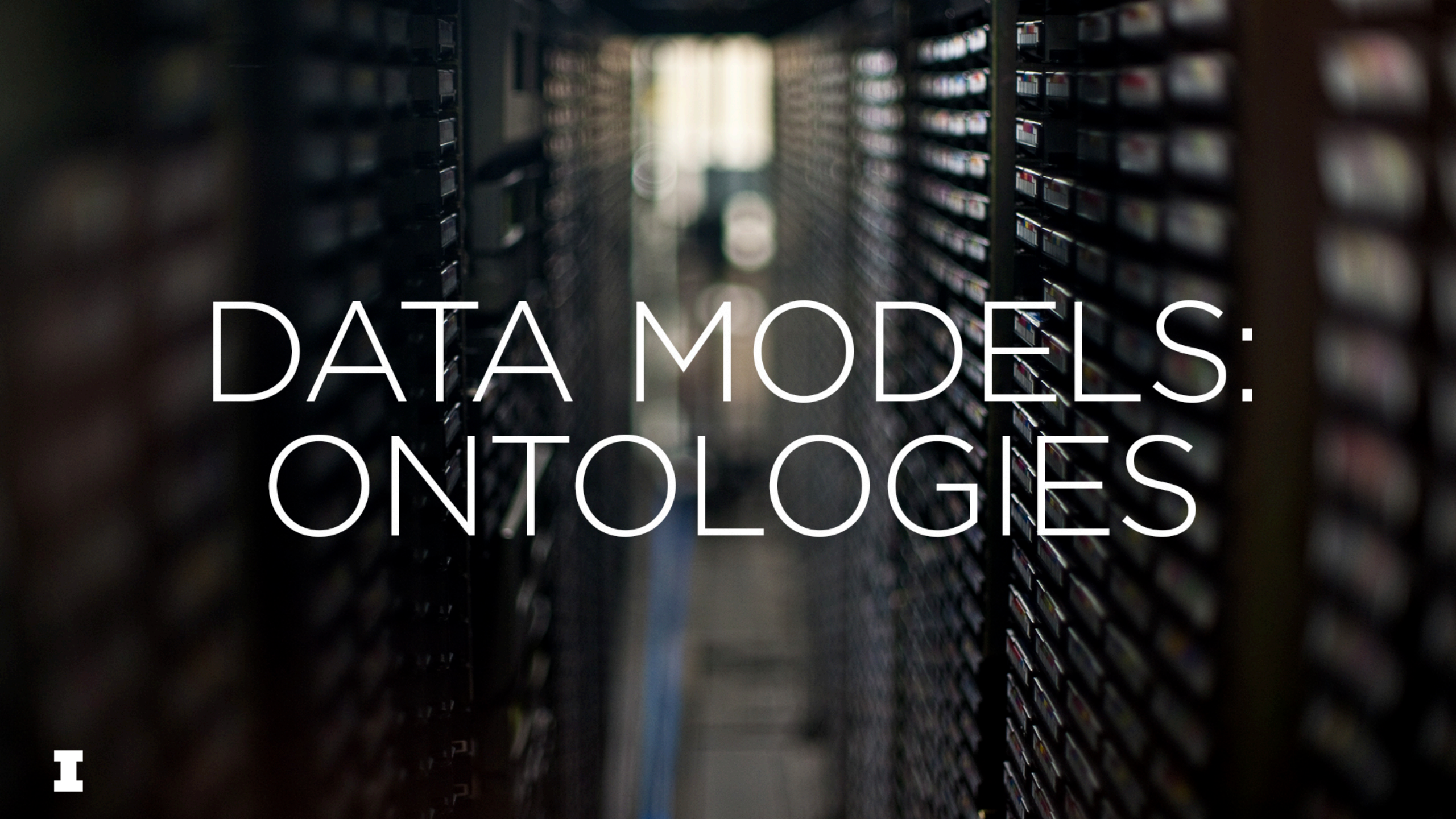


FOUNDATIONS of DATA CURATION

Allen H. Renear, Cheryl A Thompson, Katrina S Fenlon, Myrna Morales

School of Information Sciences • University of Illinois at Urbana-Champaign


DATA MODELS:
ONTOLOGIES

2

THE SOLUTION:
ONTOLOGIES

The Solution: Ontologies

The new problem (review)

The solution (historically):

Entity Relationship Modeling.

What it is

Why it works

Ontologies generally

What they are

How they look today (RDFS/OWL style)

Ontologies provide a way to abstract away from data structures such as tables and trees and focus on *information*: facts about things and relationships in the domain of interest.

And many data curation advantages ensure

The heart of the matter

We lack a shared framework
that could explicitly and formally map the relevant features in the domain of interest
to the relation or tree schemas holding data about those things

Such a framework could be used to guide relation and tree schema development and revision
and to identify the common domain features reflected in different relation and tree schemas.

The framework would also provide relation and tree schemas with a *semantics*
and as a consequence their instances, relations and trees, would have meaning and assert propositions

Without that we really don't know, formally, what a relation (or an XML document) is telling us

Yes, it's in our head, but that not good enough

We need yet *another level of abstraction*

The Solution

The old problem / solution:

There are many different ways to use *storage methods* to store *data*,

so, we need a single *data abstraction* that allows us to work with data regardless of what storage methods are used.

The solution: the relational model or the tree model

The new problem:

There are many different ways use *data abstractions* to record *information*

So we need a single *information abstraction* that allows us to work with information regardless of how the data expressing that information is stored.

The solution: . . . ?

The Solution

In 1976 Peter Chen proposed a simple solution.

Here's my interpretation:

Conceptualize your domain of interest
in terms of its things, relationships, etc.,
and then map that conceptualization
to whatever logical model schemas are being used

This (also) changed the world.

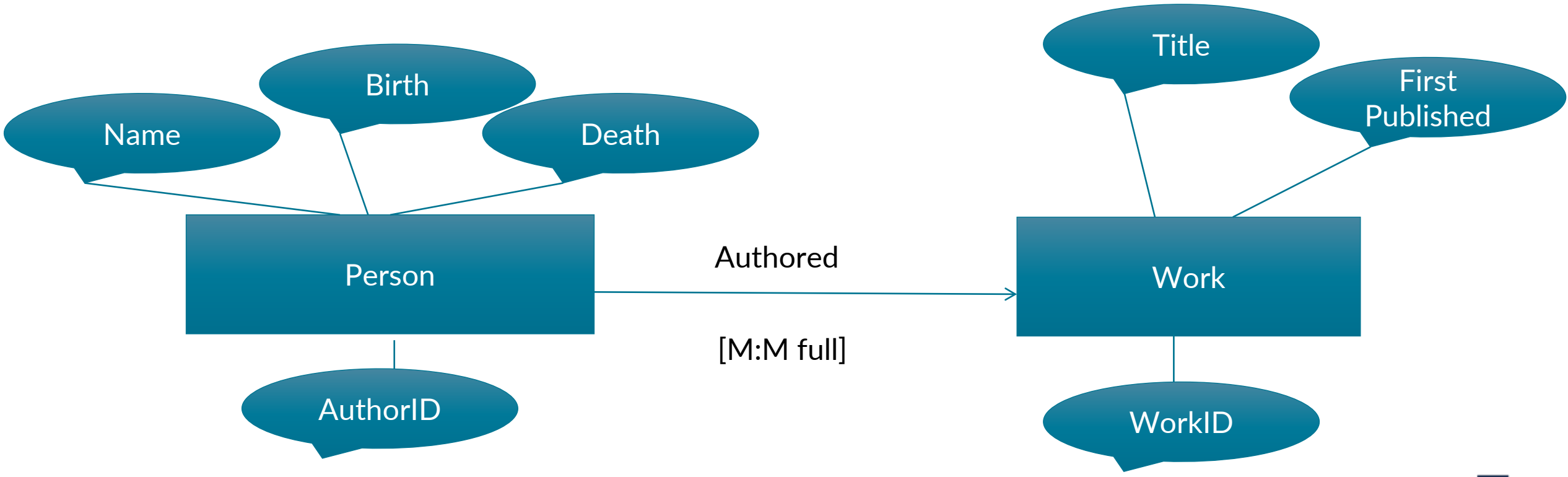
Peter Chen

Peter Pin-Shan Chen, "The Entity-Relationship Model-Toward a Unified View of Data" *ACM Transactions on Database Systems* (1970). *Also one of the most influential papers in computer science.*

Traditional ER Diagrams

An Entity Relationship diagram

Compare also UML class diagrams, etc.



The ER model has A foundation in first order logic (FOL)

Intensionally, *rectangles* indicate a kind (type) of entity.
Extensionally, they are a set of entities of a type.

In FOL: $\{ x: Fx \}$

Intensionally *attributes* are dyadic predicates.
Extensionally they are a function mapping entities to values in a domain

In FOL: $\{ \langle x,y \rangle : Rxy \ \& \ (\forall z)(Rxy \supset z=y) \}$

Intensionally *relationships* are (also) dyadic predications, but subject to a range possible cardinality and participation constraints.

Why is that logical basis important

We now have a description of the structure of the domain, a specification of the real world things, attributes, and relationships, in that domain.

Such as description can been called a *conceptual schema*.

A conceptual schema really is about the world:

its predicates correspond to real world properties (they are not simply names of domains); its variables range over real world individual things, explicitly, not by informal interpretation.

Connecting a conceptual schema to a logical level schema can give the instances of that logical schema *meaning*, formalizing how they express propositions, how they express *information about the world*.

Conceptual schemas solve the problem described earlier: they can be used to guide the development of logical schemas, or characterize the common of two different schemas that partially or totally equivalent.

And finally, perhaps most importantly, conceptual schemas provide critical *documentation* of the dataset.

Where does the mapping occur?

The most common use of conceptual models is to support the creation of logical schemas

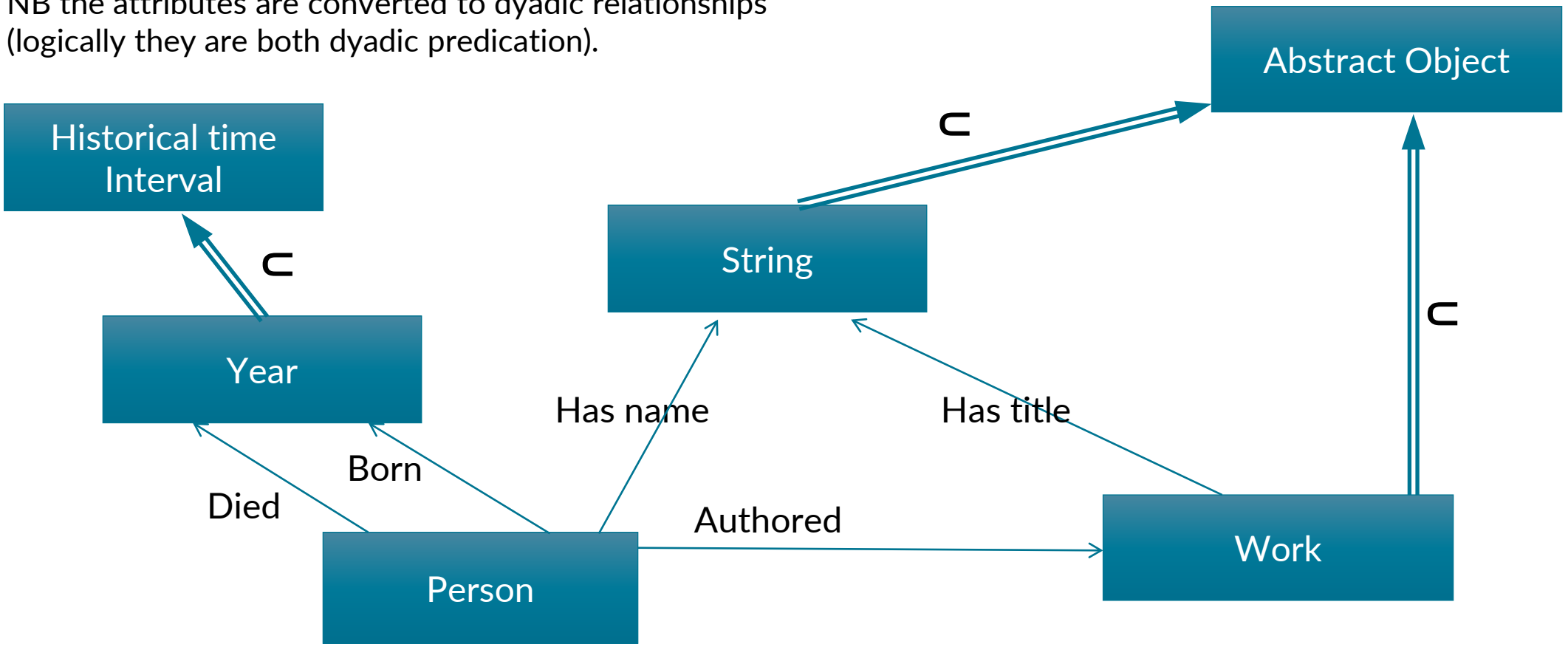
This typically involves the step by step generation of relational schemas from an ER schema

Most of you are familiar with the algorithm for this (there are many examples available on the web).

A data curation point: if your relational schema was generated mechanically from an ER schema then that ER schema reliably documents, at least partially, the relationship between your view of the domain and your relational schema. This is, obviously, very valuable.

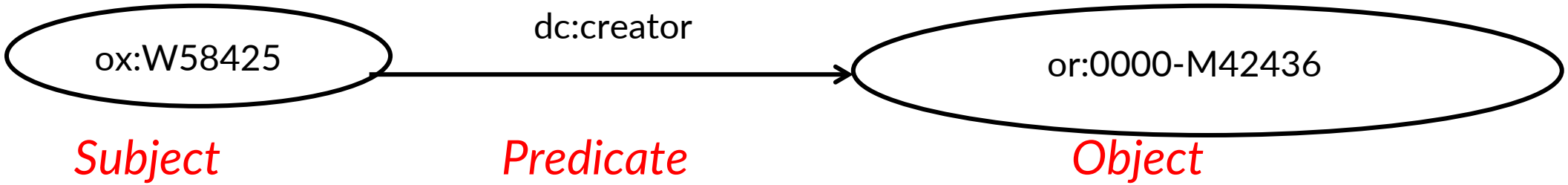
Generic Conceptual Modeling Today

NB the attributes are converted to dyadic relationships (logically they are both dyadic predication).

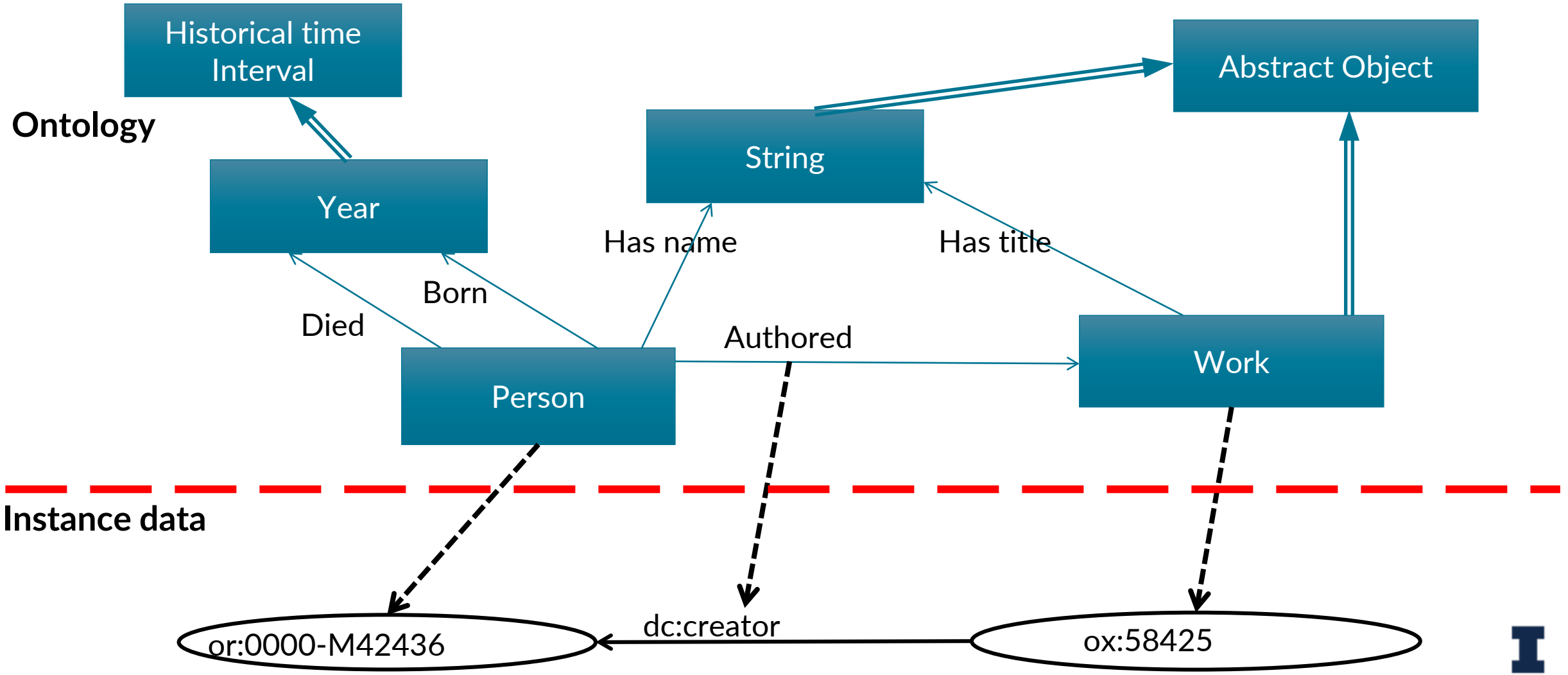


Now we sneak in A third logical level model

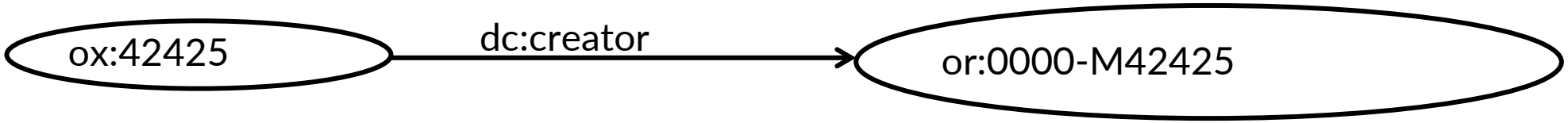
- The emergence of contemporary ontologies sparked interest in a logical level data model that had not been particularly important in the late 20th century: simple dyadic predication
- It makes its appearance as the “RDF triple”, typically represented with the generic graph model



Ontology + Instance Data



RDF Statements in N3 Triples



ox:W42425	dc:creator	or:0000-M42436
ox:W24246	dc:creator	or:0000-H24246
ox:W42427	dc:creator	or:0000-H24236

Compare:

WorkID	dc:creator	Title	First Published
W58425	M42425	Moby Dick	1851
W85246	H24236	The Scarlet Letter	1850
W55427	H24236	Fanshawe	1828

Hmm . . .

What is an Ontology?

“An ontology is an explicit specification of a conceptualization of a domain.”

- A body of formally represented knowledge is based on a *conceptualization*:
  - The objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them.
- A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

  [Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly.]

Tom Gruber (Stanford), "What is an Ontology?”

Conceptual Models vs. Ontologies

I don’t see any useful distinction between conceptual models and ontologies and so will be using “ontologies” to include models like ER.

Nevertheless there are slight differences of emphasis:

- Ontologies almost always have class relationships (and perhaps an extensive class hierarchy)
- Ontologies will often include very abstract concepts at the levels of that hierarchy (*physical object*, *set*, *event*, *time interval*, *property*, etc.)
- Ontologies are usually intended to be relative stable and multipurpose.
- Ontologies are sometimes accompanied by logical axioms that support inferencing.
- Sometimes a specific formal technique is used for developing ontologies
- Often in ontology development minimizing the number of general kinds of entities is a concern.

Conceptual Models for XML Trees?

You will have noticed that most of this discussion is about conceptual models, or ontologies, that can be mapped to relational schemas.

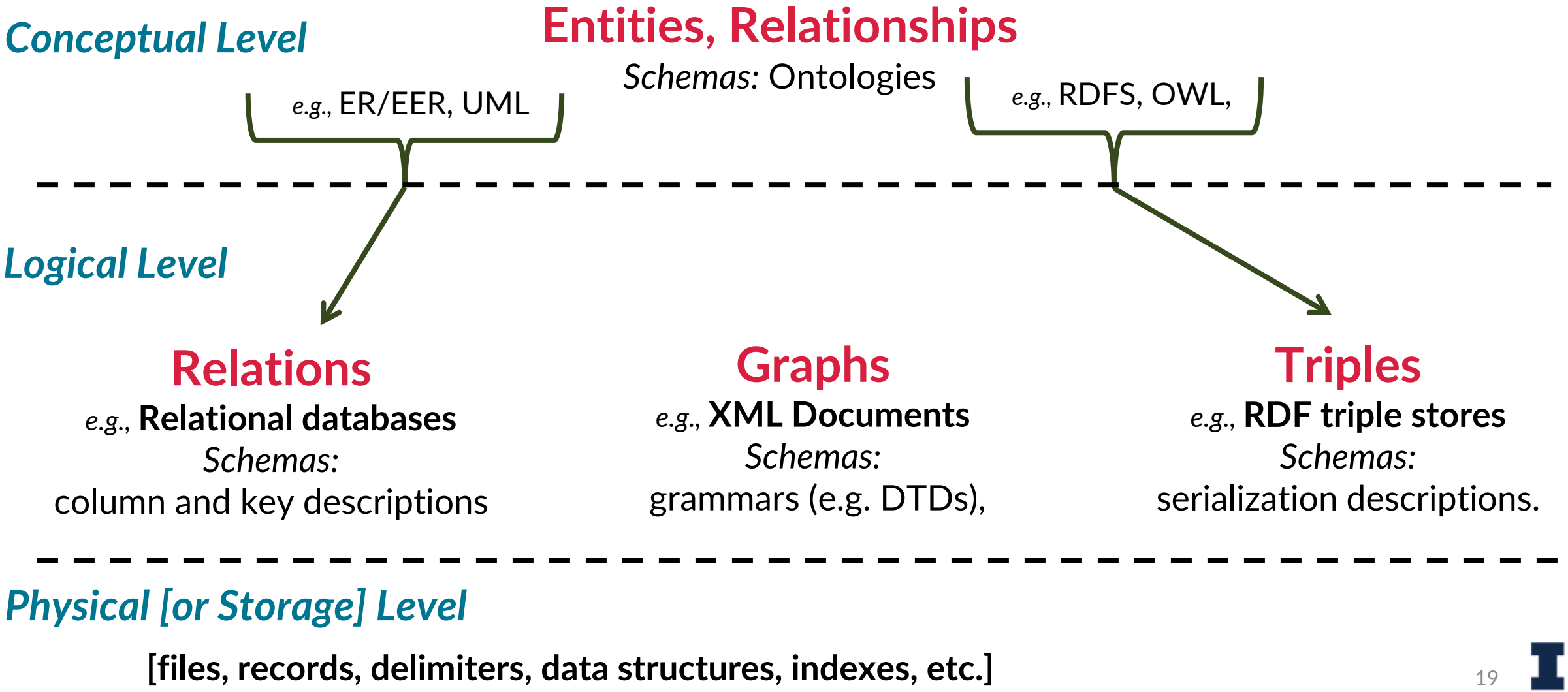
But what about the tree model? What about XML documents with descriptive markup?

Turns out that the semi-structured nature of XML documents makes it very challenging to develop a conceptual model that will play the same semantic role as ER diagrams do for the relational model.

For more on that see:

> “Towards a Semantics for XML Markup.”
> *Proceedings of the 2002 ACM Symposium on Document Engineering*,
> New York: Association for Computing Machinery. 2002.
> Allen H. Renear, David Dubin, C. M. Sperberg-McQueen, and Claus Huitfeldt.

Data Model Relationships



FOUNDATIONS OF DATA CURATION (IS531)

Allen H. Renear, Cheryl A Thompson, Katrina S Fenlon, Myrna Morales
School of Information Sciences
University of Illinois at Urbana-Champaign

Includes material adapted from work by Carole Palmer, Melissa Cragin, David Dubin, Karen Wickett, Bertram Ludaescher, Ruth Duerr and Simone Sacchi.

Comments and corrections to: renear@illinois.edu.