Review:

Data Models
Data Integration
Data Concepts

Data model relationships

Entities, Relationships

Conceptual models, UML or ER models, ontologies

Schemas: ER, UML . . .

Schemas: RDFS, OWL . . .

Conceptual Level

Logical Level

Relations

e.g., Relational databases

Schemas: column and key descriptions

Trees

e.g., XML Documents

Schemas: grammars (e.g. DTDs),

Triples

e.g., RDF triple stores

Schemas: serialization descriptions.

Physical Level [or: Storage]

[files, records, delimiters, data structures, indexes, etc.]

Data curation actions wrt data models

Data curation involves:

Selecting data model types

Selecting data model schemas

Developing data model schemas

Revising data model schemas

Documenting data model schemas

Validating dataset instances with schemas*

Transforming data in one model (type) to another data model (type)*

Transforming data in one model (schema) to another data model (schema)*

Transforming data from one representation (e.g. serialization) to another (with same schema)*

Integrating data from two different data models (schema or type)*

[*and documenting that]

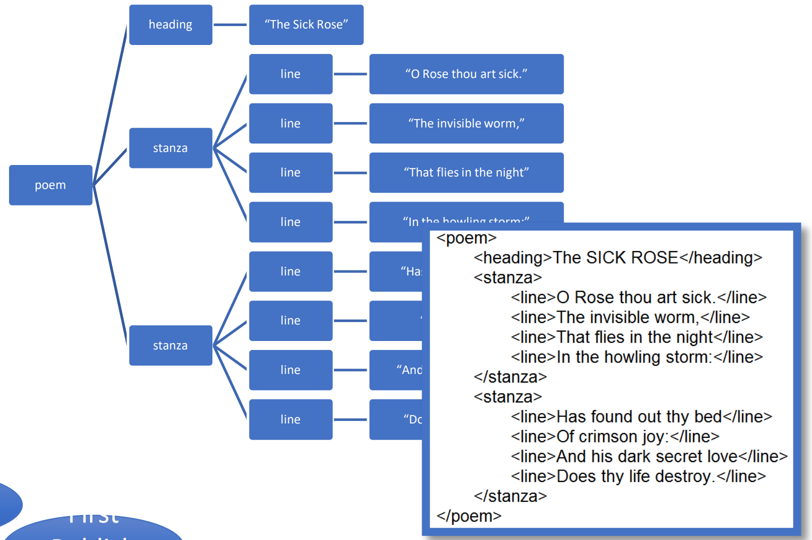
Some data models you know and love

Relations (good for attribute value pairs)

Logical:

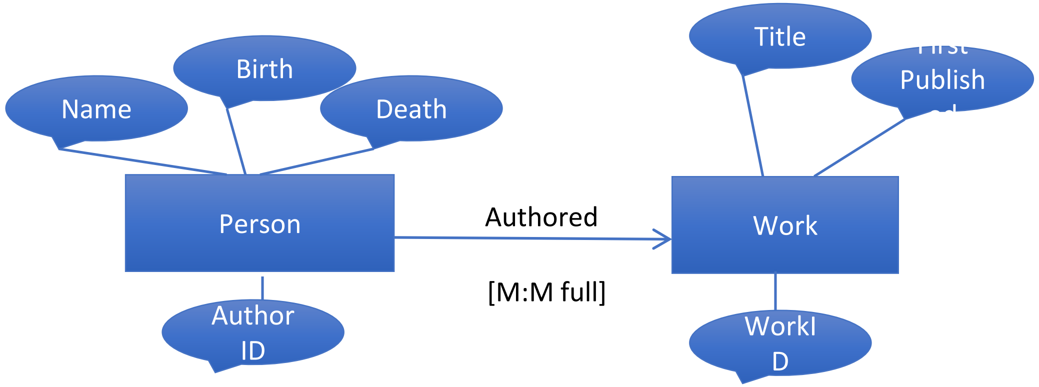
Work	Author	Title	Date
W58425	P42425	Moby Dick	1851
W85246	P24246	The Scarlett Letter	1860
W55427	P24246	Fanshawe	1828

Trees (good for text and documents)



Entity/Relationship (ontologies)

Conceptual:



Two fundamental problems in data management

Programs and users often interact with data directly via its storage structure

(And those storage structures vary wildly).

The intrinsic nature of the information is often not reflected in the management systems.

These systems to not explicitly reflect the attributes, relationships, etc. that are the genuine components of the information being stored and managed.

Storage representations

Variable-length fields, differently delimited

W54825,Moby Dick,1851
W85246,The Scarlett Letter,1860
W55427,Fanshawe,1828

W54825 Moby Dick 1851
W85246 The Scarlett Letter 1860
W55427 Fanshawe 1828

Fields indexed byte offsets

00000000	(WorkID)
00000010	(Title)
00000020	(Year)

Fixed-length fields

…counting from the left: 0, 6, 25

W	5	8	4	2	5	M	o	b	y		D	i	c	k											1	8	5	1
W	8	5	2	4	6	T	h	e		S	c	a	r	l	e	t	t		L	e	t	t	e	r	1	8	6	0
W	5	5	4	2	7	F	a	n	s	h	a	w	e												1	8	2	8

…or from the right: 28, 22, 3

As a result. . .

This result is that systems and practices are:

Inefficient

Error-prone

Untrustworthy

Difficult to document

Difficult to repurpose and reuse

Difficult to preserve for future use

Dependent on memory and workplace practices

Dependent on custom tools and applications

Data Independence

One significant consequence of this chaos is a failure of data independence.

This failure comes in two varieties:

> **Type 1: If the storage method changes**, then the end user programs accessing the data will fail to perform as expected.
>
> **Type 2: If new kinds of data need to be represented,** then again end user programs may fail or give the wrong result.

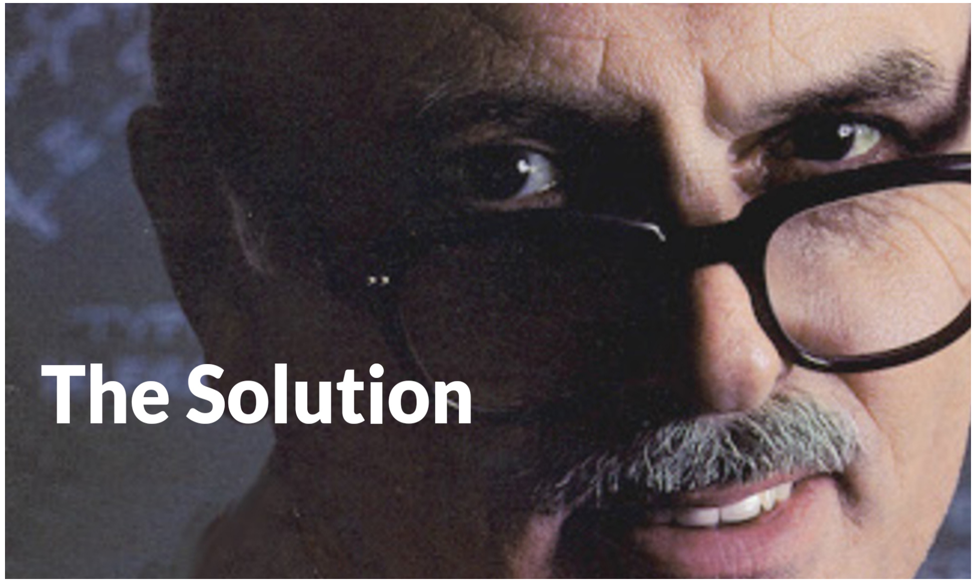
Let's take a closer look

The solution

In 1970 E. F. Codd proposed a simple solution.

Conceptualize data as relations (tables)
and then map those relations
to whatever storage methods are being used

It changed the world.



E. F Codd

EF Codd in "A Relational Model of Data for Large Shared Data Banks" (1970).
Perhaps the most cited paper in computer science.

Abstraction and Indirection

The success of the relational model is based primarily on two related principles:

Abstraction and **Indirection**

The relational representation

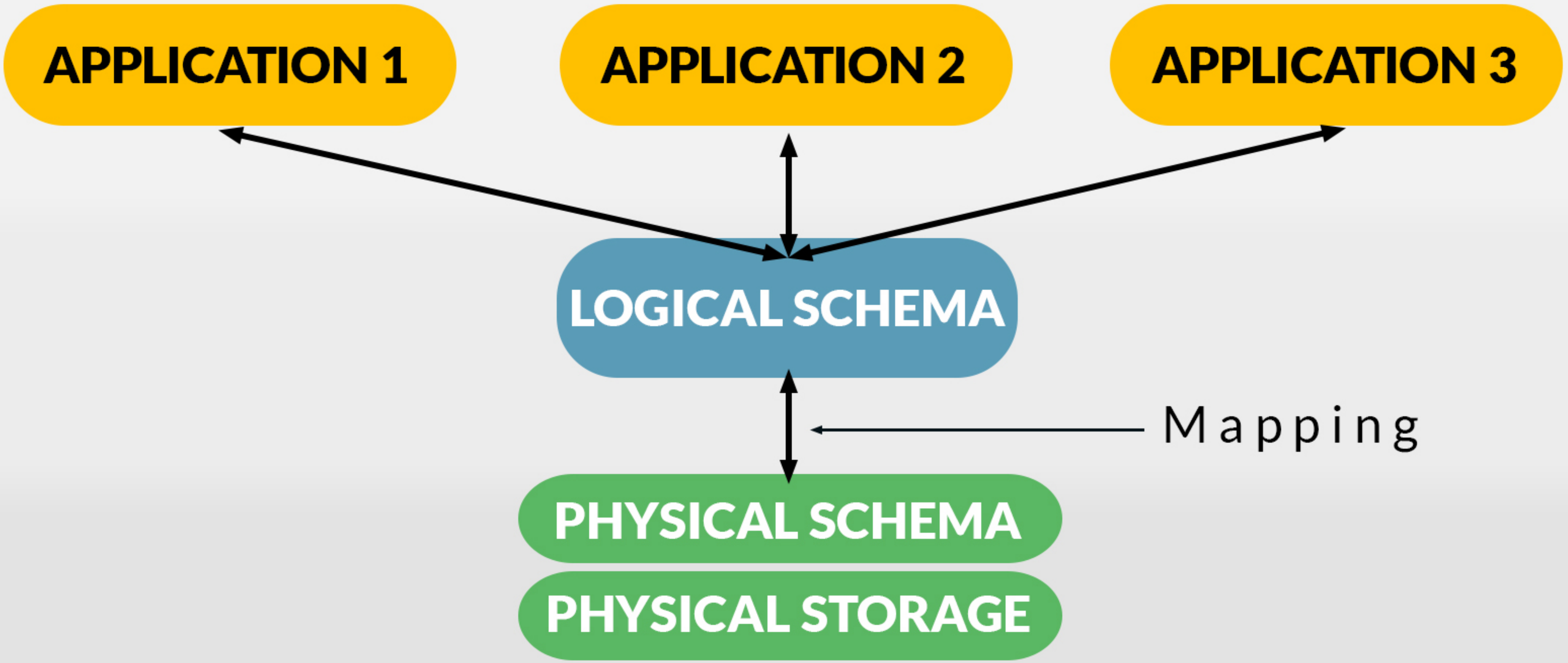
abstracts away from the specific and transient details of storage,

presenting only the intrinsic features of the data itself.

Subsequent interaction with the stored data is then *indirect*

via a mapping from the relational schemas to the stored data

Mapping


APPLICATION 1
APPLICATION 2
APPLICATION 3
LOGICAL SCHEMA
Mapping
PHYSICAL SCHEMA
PHYSICAL STORAGE

The problem for documents and text *(it’s the same problem!!)*

There are many, many ways to represent text and documents

```
.ll 3i
.mk a
.ce
Preamble
.sp
We, the people of the United States, in order to form a more perfect Union
```

```
{\rtf1\ansi{\fonttbl\f0\fswiss Helvetica;}\f0\pard
This is some {\b bold} text.\par }
```

```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
00000000  D0 CF 11 E0 A1 B1 1A E1  00 00 00 00 00 00 00 00  ÐÏ à¡± á
00000010  00 00 00 00 00 00 00 00  3B 00 03 00 FE FF 09 00          ;   þÿ
00000020  06 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00
00000030  11 00 00 00 00 00 00 00  00 10 00 00 02 00 00 00
00000040  01 00 00 00 FE FF FF FF  00 00 00 00 00 00 00 00      þÿÿÿ
00000050  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00001A00  57 00 65 00 20 00 74 00  68 00 65 00 20 00 50 00  W e   t h e   P
00001A10  65 00 6F 00 70 00 6C 00  65 00 20 00 6F 00 66 00  e o p l e   o f
00001A20  20 00 74 00 68 00 65 00  20 00 55 00 6E 00 69 00    t h e   U n i
00001A30  74 00 65 00 64 00 20 00  53 00 74 00 61 00 74 00  t e d   S t a t
00001A40  65 00 73 00 2C 00 20 00  69 00 6E 00 20 00 4F 00  e s ,   i n   O
00001A50  72 00 64 00 65 00 72 00  20 00 74 00 6F 00 20 00  r d e r   t o
00001A60  66 00 6F 00 72 00 6D 00  20 00 61 00 20 00 6D 00  f o r m   a   m
```

```
<w:body>
    <w:p w:rsidR="00146B24" w:rsidRDefault="00146B24">
      <w:bookmarkStart w:id="0" w:name="_GoBack"/>
      <w:bookmarkEnd w:id="0"/>
    </w:p>
    <w:p w:rsidR="00146B24" w:rsidRDefault="00146B24“  w:rsidP="00146B24">
      <w:pPr>
        <w:pStyle w:val="Heading1"/>
      </w:pPr>
      <w:r>
        <w:t>Preamble</w:t>
      </w:r>
    </w:p>
    <w:p w:rsidR="001C180C" w:rsidRDefault="00146B24">
      <w:r>
        <w:t xml:space="preserve">We the People of the United States, </w:t>
      </w:r>
        ...
```

Interaction is typically directly with these storage structures, or via processing instructions

The fundamental principles of *abstraction* and *indirection* are not followed and all the usual problems ensue

The solution

In 1981 Charles Goldfarb proposed a simple solution.

Conceptualize a document as a tree (rooted and ordered directed acyclic graph) of textual data elements

and then map those elements to
to whatever storage or processing methods as needed

It, also, changed the world. (really, heard of HTML? SGML? XML?)

Charles Goldfarb

Charles Goldfarb, "A Generalized Approach to Document Markup", in *SIGPLAN Notices*, June 1981.

Descriptive markup + trees

This model describes the *logical components* of documents.

It does not specify storage strategies (even though often inline)

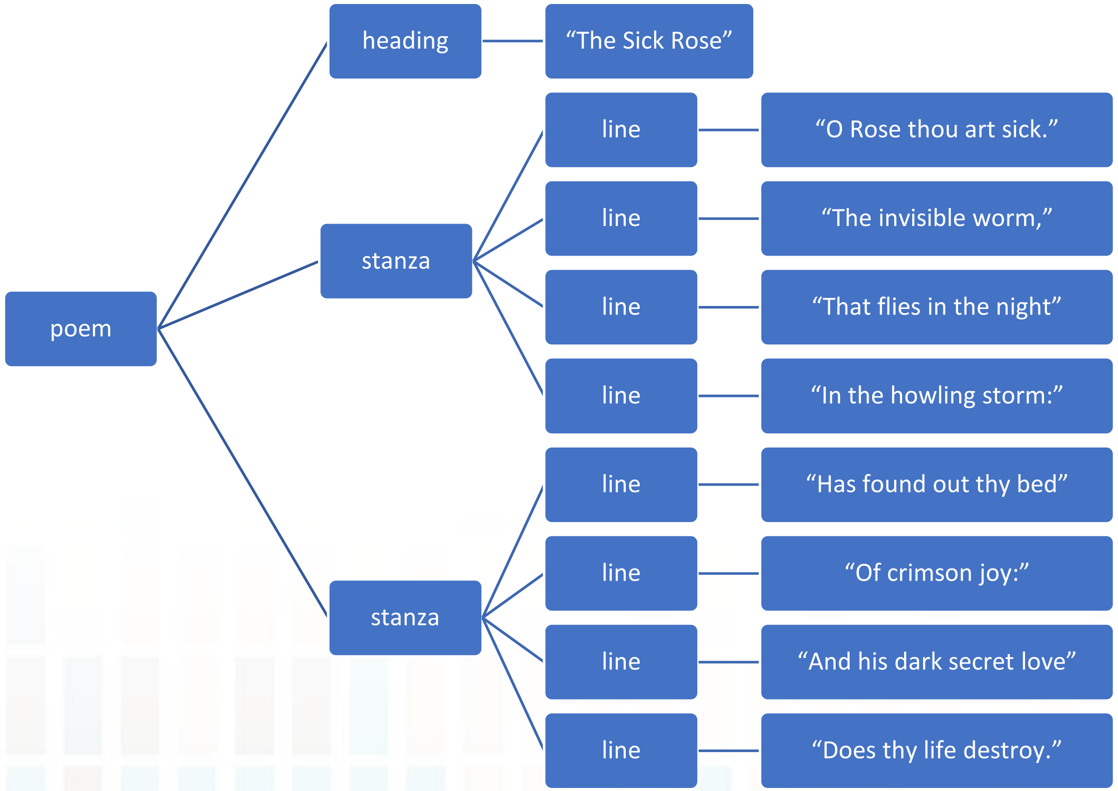
It does not specify processing

It **abstracts** from storage and processing

It connects data to storage and processing by *mappings*, i.e by **indirection**

[And it uses a well understood data structure]

Using XML to Serialize a Tree




```xml
<poem>
    <heading>The SICK ROSE</heading>
    <stanza>
        <line>O Rose thou art sick.</line>
        <line>The invisible worm,</line>
        <line>That flies in the night</line>
        <line>In the howling storm:</line>
    </stanza>
    <stanza>
        <line>Has found out thy bed</line>
        <line>Of crimson joy:</line>
        <line>And his dark secret love</line>
        <line>Does thy life destroy.</line>
    </stanza>
</poem>
```

A tree can be serialized with a formal language defined by a context free grammar, such as an XML language.

So are we done yet? (no)

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

In 1976 Peter Chen proposed a simple solution.

Here's my interpretation:

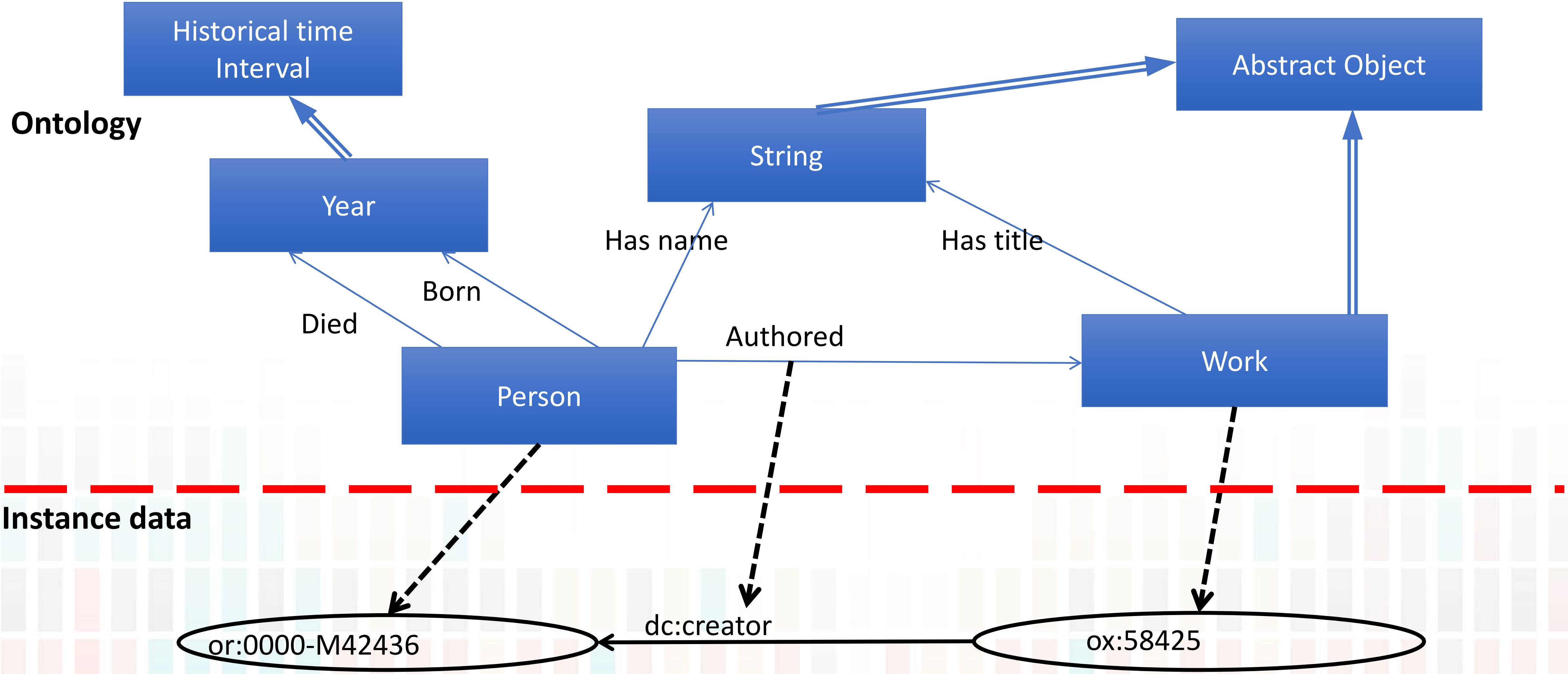
Conceptualize your domain of interest
in terms of its things, relationships, etc.,
and then map that conceptualization
to whatever logical model schemas are being used

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*.

Ontology + Instance Data



Data model relationships

Entities, Relationships

Conceptual models, ER or UML models, ontologies

Schemas: ER, UML . . .

Schemas: RDFS, OWL . . .

Conceptual Level

Logical Level

Relations

e.g., Relational databases

Schemas: column and key descriptions

Trees

e.g., XML Documents

Schemas: grammars (e.g. DTDs),

Triples

e.g., RDF triple stores

Schemas: serialization descriptions.

Physical Level [or: Storage]

[files, records, delimiters, data structures, indexes, etc.]

Data integration

Data integration

Data integration:

"... combining data residing in difference sources
and providing users with a unified view..."

[Lenzerini 2002]

Kinds of heterogeneity

Relatively easy
Often difficult
Usually very difficult

Encoding heterogeneity *
Different mappings from bitstreams into bytes, characters, numbers, or other logical units

Syntax heterogeneity *
Different data description languages for the same model type: e.g. RDF/XML vs N3

Model heterogeneity *
Different model type; e.g., relations vs entities/relationships

Representational heterogeneity *
Different modeling choices within a model type; e.g. relationships vs entities.

Semantic heterogeneity *
Different conceptualization of similar domain features

Processing heterogeneity *
e.g. different maintenance and update regimes

Schema integration

Policy heterogeneity *
e.g. different privacy and security rules, varying ownership and licensing, etc.

Adapted from Bertram Ludäscher, Kai Lin, Shawn Bowers, Efrat Jaeger-Frank, Boyan Brodaric, Chaitan Baru, "Managing scientific data: From data integration to scientific workflows", *Geoinformatics: Data to Knowledge, 2006*; and Amit Sheth. "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics". In M. Goodchild, M. Egenhofer, R. Fegeas,and C. Kottman, editors, *Interoperating Geographic Information Systems* Kluwer, 1998.

Data Concepts

Some identity Problems in Data Curation

Archiving: Is this dataset already in the archive?

Preservation: Was the information preserved in the new file format?

Security: Has this dataset been tampered with?

Authentication: Is this the data we think it is?

Reproducibility: Does this XML file have the same information as that JSON file?

Provenance: Were these datasets derived from the same data?

Conversions: Does the converted file have the same data as the original?

and on and on. . .

"… there are an unknown number of transformations
that are invariant in the sense of *preserving the scientific meaning . . .*
different scientific communities use different tools that require different representations.

Ruth Duerr, National Snow and Ice Data Center
Data Conservancy wiki, December 2010

Same, different, same, different. (but same/different *what?*)

So in a successful conversion

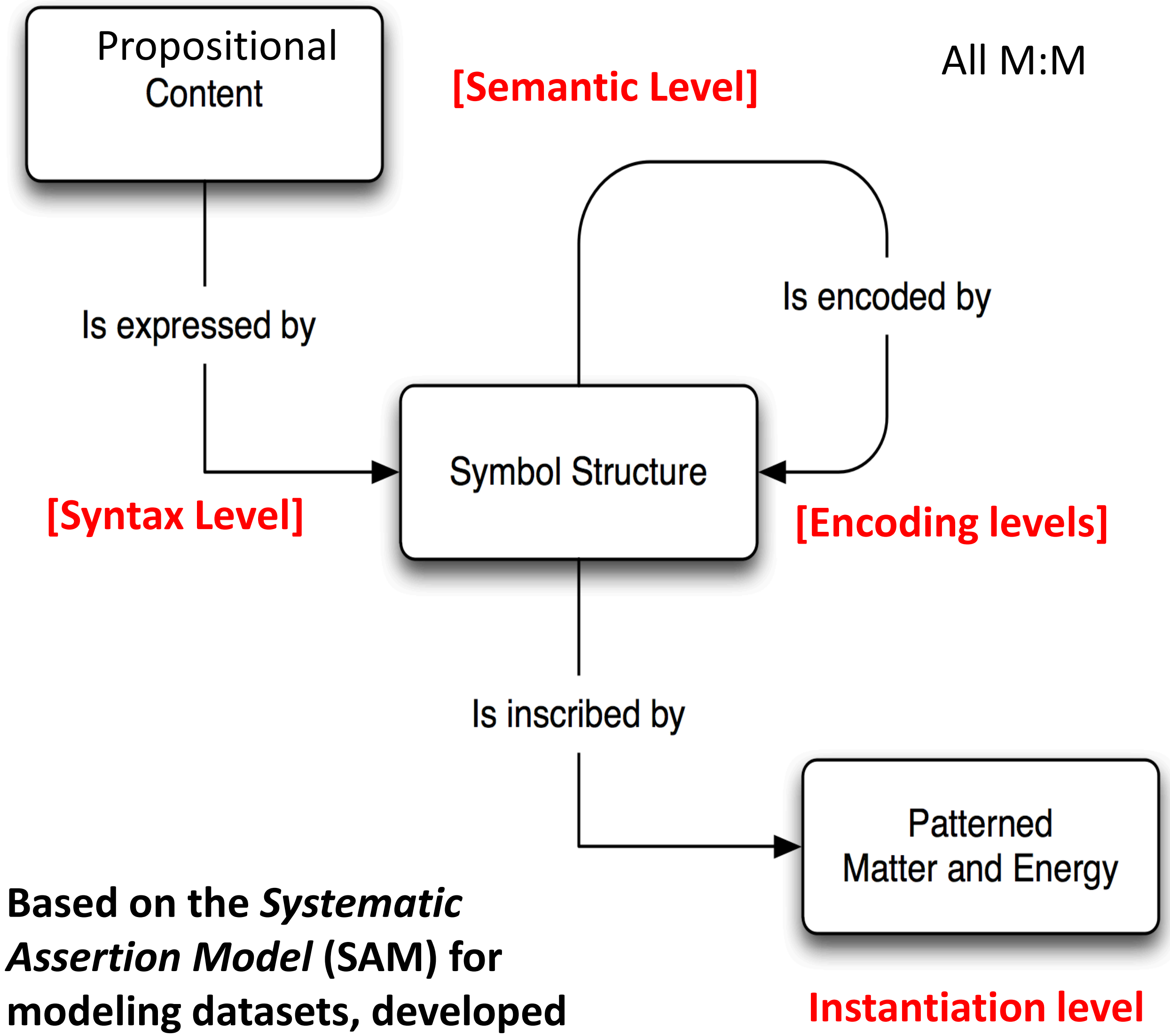
something changes;

and

something remains the same.

But ***what*** exactly changes? And ***what*** remains the same?

The Basic Representation Model (or FRBR refactored)



Based on the *Systematic Assertion Model* (SAM) for modeling datasets, developed by David Dubin et al.

For example:

C1: propositions

expressed by…

S1: RDF triples

encoded by…

S2: RDF/XML

encoded by …

S3: Unicode characters

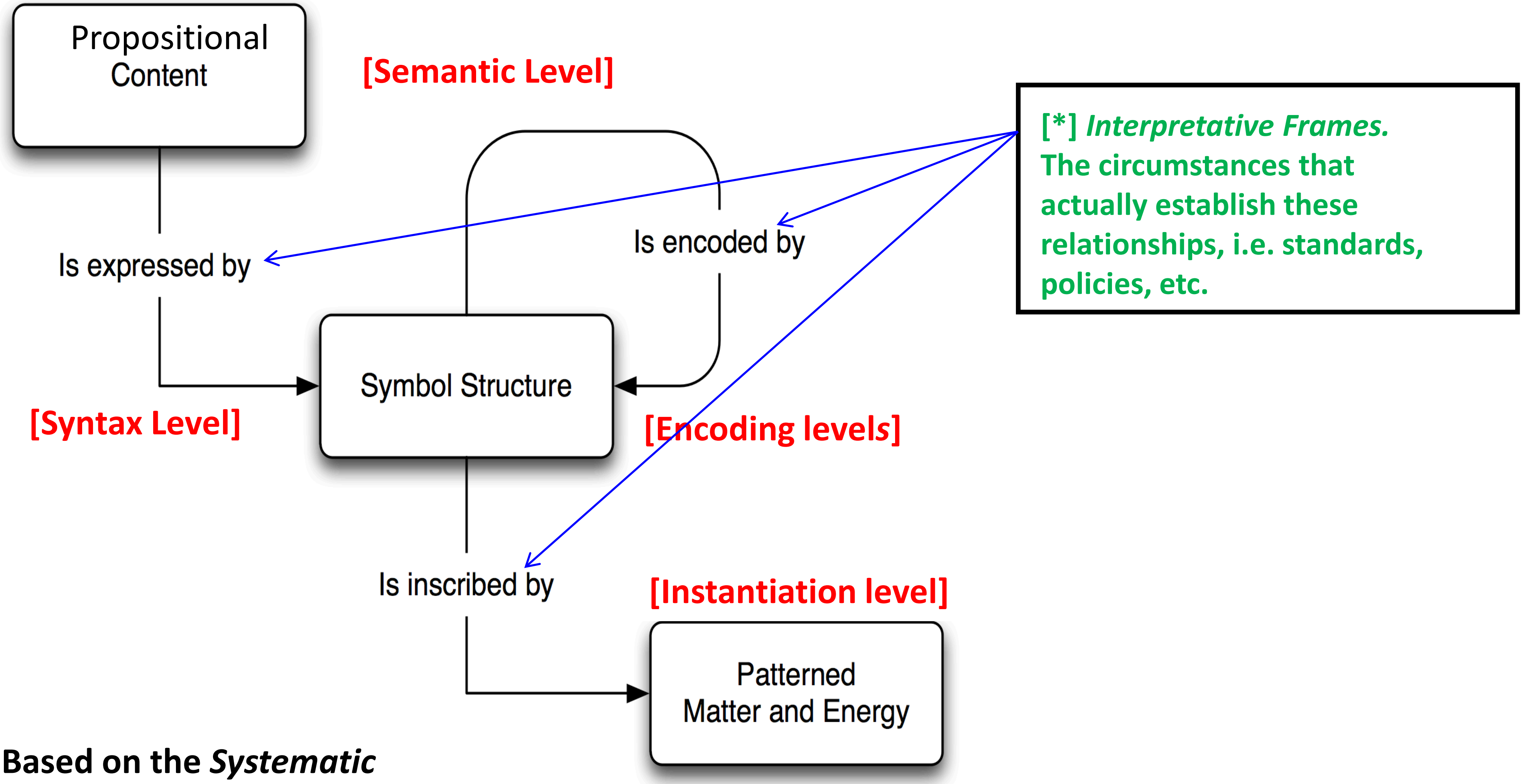
encoded by…

S4: UTF-8 bit streams

inscribed in…

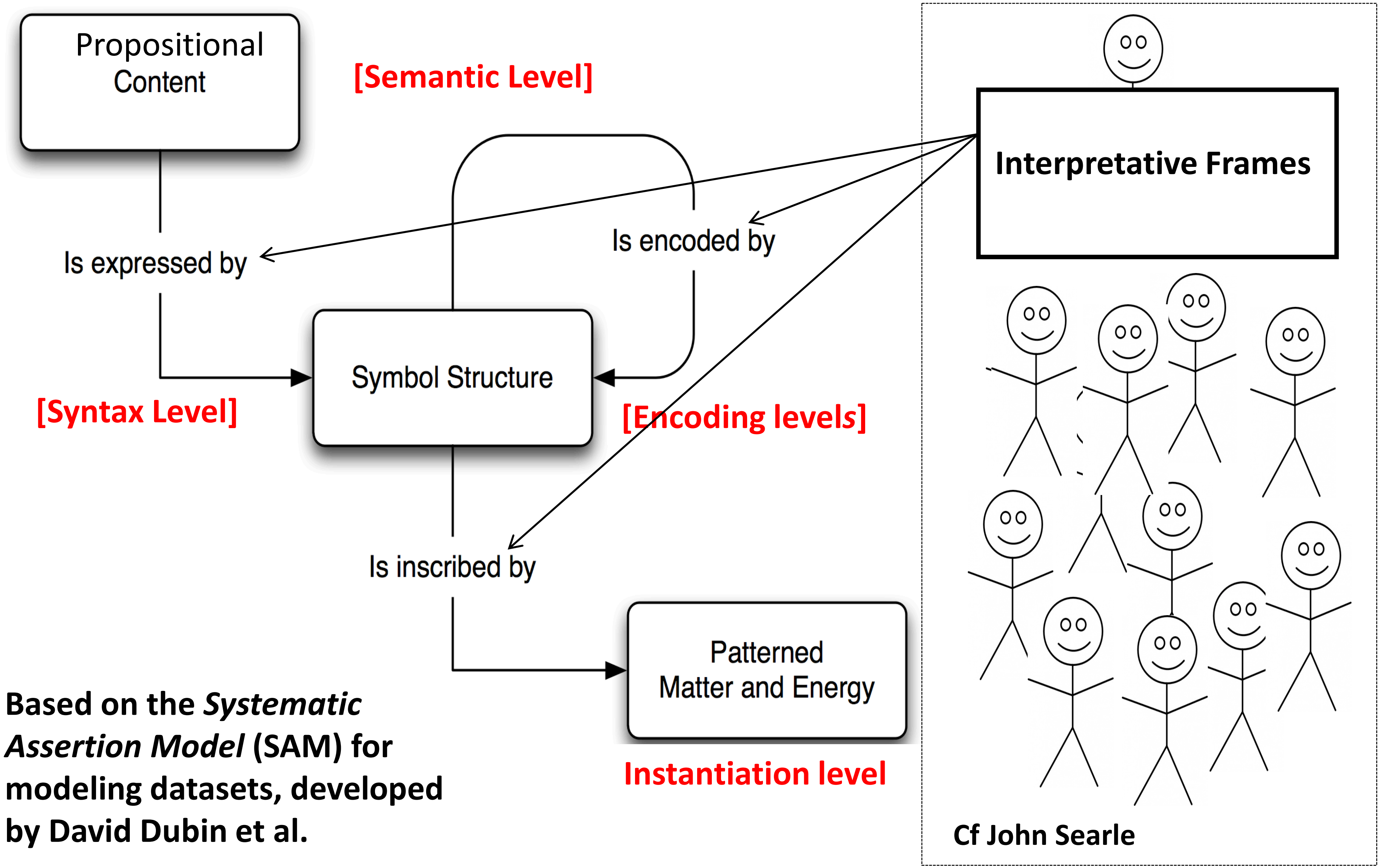
M1: actual RAID array state

FRBR refactored and extended. What’s still missing? [*]



Based on the *Systematic Assertion Model* (SAM) for modeling datasets, developed by David Dubin et al.

Actually "it takes a village" (i.e. *collective* intentionality)



Based on the *Systematic Assertion Model* (SAM) for modeling datasets, developed by David Dubin et al.

Data, our definition

So our answer to the vexed question "What is data?" is:

Data are **propositions**

(i) *asserted* *(via symbols . . . and matter and energy)*

(ii) as *evidence*

Dubin et al. 2009-2014

Data is not a *type* of thing, it is a *role*

Just as persons are students when enrolled in a school

propositions are data when asserted as *evidence*

Being asserted as evidence is a contingent (and social) circumstance
(just like being enrolled)

And so data is:

a role that propositions have in certain contingent social circumstances

Data is *relative*

So whether propositions are data or claims depends upon what is intended.
And propositions can be data in one circumstance, claims in another.

In fact, science as a whole depends on this. For instance:

For a climate scientist,

growth rings on tree rounds may be *evidence*
for theories about temperature changes

But for an evolutionary ecologist

those theories about temperature changes may in turn be *evidence*
for theories about competitive advantages

In a slogan:

one person's data is another person's theory