Plan for INFO Lecture #10

- Modeling across the "Document Type Spectrum"
- Document models \{and, or, vs\} data models
- "Berkeley Event Calendar Network" case study
- How much modeling is necessary?
Many people have contrasted "documents" and "data" and concluded that documents and data cannot be understood and handled with the same terminology, techniques, and tools.

This document vs. data distinction is embedded and reinforced in courses, textbooks, technology, and product marketing.

And it doesn't help.

Mixing Data and Documents

**PUBLISHING**

- Requirements
- RFI
- RFQ
- Product
- Catalogs & Brochures

**TRANSACTIONS**

- Orders
- Invoices

**Technical Documentation; User, Installation, Troubleshooting Manuals**
Catalog: Data (Document)

Industrial Or Light Weight Bags On A Roll

Bags are perforated allowing easy tear off for in-store or assembly line use. Choose industrial 2-mil or extra heavy 4-mil for parts fittings and hardware. Light weight bags are 5 mil and ideal for producing and lighter weight items. Tabs or no-vail dispensers available below.

<table>
<thead>
<tr>
<th>Size (in)</th>
<th>Bags Per Roll</th>
<th>Part No.</th>
<th>Price 2 Mil</th>
<th>Part No.</th>
<th>Price 4 Mil</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 x 6</td>
<td>1000</td>
<td>BS600LU</td>
<td>25.55</td>
<td>BS6354LU</td>
<td>46.55</td>
</tr>
<tr>
<td>6 x 9</td>
<td>1000</td>
<td>BS6034LU</td>
<td>59.15</td>
<td>BS6354LU</td>
<td>85.63</td>
</tr>
<tr>
<td>8 x 10</td>
<td>1000</td>
<td>BS6066LU</td>
<td>67.25</td>
<td>BS6046LU</td>
<td>125.30</td>
</tr>
</tbody>
</table>

Discount per quantity: Less 5% 12-23 mats; 10% 24 or more.

Light Weight Bags On A Roll

<table>
<thead>
<tr>
<th>Size (in)</th>
<th>Bags Per Roll</th>
<th>Part No.</th>
<th>Price Per Carton of 2 Rolls</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 x 12</td>
<td>2500</td>
<td>BS6017LU</td>
<td>120.80</td>
</tr>
<tr>
<td>12 x 15</td>
<td>1500</td>
<td>BS6019LU</td>
<td>120.80</td>
</tr>
<tr>
<td>Dispenser</td>
<td>BS60194LU</td>
<td>17.50</td>
<td></td>
</tr>
</tbody>
</table>

Discount per quantity: Less 5% 5-11 mats; 10% 12-23 mats; 15% 24 rolls or more.

Reference Book: Document (Data)

1.103 Range of Light Intensities Confronting the Eye

Key References

Graphs: The relationship between light intensity and the eye's ability to perceive it is illustrated through various graphs and diagrams. These graphs demonstrate the eye's sensitivity to different levels of light intensity and how it adapts to changes in light. The data is supported by empirical evidence and is crucial for understanding the interaction between light and human vision.

Key Facts:
- The human eye is sensitive to a wide range of light intensities, from extremely low levels to extremely high levels.
- The eye's ability to adapt to changes in light intensity is crucial for survival, allowing humans to function in various lighting conditions.
- The range of light intensities that the human eye can perceive is vast, with some species capable of perceiving light intensities far beyond the range of human capability.
- The eye's ability to adapt to changes in light intensity is influenced by various factors, including age, health, and environmental conditions.
Contrasting Methodologies for Documents and Data

Documents and data have had two different disciplines or methods of analysis that have had little intersection

*Document-centric* analysis

*Data-centric* analysis

Document Analysis

Documents are *Artifacts or Renditions* that combine content, structure and appearance

The goal of document analysis is a model of a document's content and structure that is separate from its presentational characteristics

The optimal prescriptive schema for a set of documents is one that best satisfies the requirements of current and prospective users for carrying out specific tasks with new instances

Finally, one or more stylesheets can be used to assign formatting or rendering characteristics in a consistent manner to any valid document
Data-Centric Analysis

Goal is to understand and describe the properties and relationships between information components or objects.

This understanding is represented in conceptual models that organize the components efficiently to support a broad range of contexts or applications.

The conceptual model is also typically called a schema, but this is generally meant to be a "database schema" rather than a "document schema"
The Document Type Spectrum – "Narrative Publications"

Authored by people
Highly designed, with rich presentational characteristics correlated with semantics and structure
Heterogeneous in structure and content
Weakly datatyped – "just text"

The Document Type Spectrum – "Transactional Documents"

Created mechanically
Few and somewhat arbitrary presentational characteristics
Homogeneous in structure and content
Strongly datatyped
It's Obviously A Continuum

There is systematic and continuous variation in document instances and types and there is no clear boundary between documents and data. But the traditional tools, terminology, and techniques for analyzing documents and data have made it into a chasm.

Crossing the Chasm with "Document Engineering" Methods

Document Engineering harmonizes the terminology and emphasizes what they have in common rather than highlighting their differences:

- Identifying the presentational, content, and structural components
- Eliminating synonymy and homonymy
- Identifying and organizing the "good" content components
- Assembling hierarchical document models to organize components to meet requirements for a specific context for information exchange
Harvesting and Consolidation

HARVESTING: Create a set of candidate content components by extracting them from the information sources while removing presentation and structure

• For each component, record its properties (or metadata or attributes or behaviors) that enable us to understand and distinguish it

CONSOLIDATION: Identify synonyms and homonyms among the candidate content components, assigning a unique name to each unique meaning as part of a controlled vocabulary

• Names might follow precise rules to ensure that they can be reliably stored and located in a data dictionary (e.g., a la ISO 11179 part 5)

"Good Models and "Better Models"

Definitions
Definitions in a controlled vocabulary
Data types
Metadata
Metamodels
Formal assertions
Ontologies and thesauri
The Simplest Component Model

The simplest or minimal information component model is a GLOSSARY – a list of the words used to describe or name the "things of significance" and what they mean.

This simple data model is augmented as attributes or characteristics of the significant things are identified and recorded.

The model is further developed as relationships or associations or links between the "significant things" are identified and recorded.

Component Metadata

What attributes about each type of content might we record in our analysis?

- Names/synonyms/homonyms (what it is called)
- Definition (what it "means")
- Identifiers
- Cardinality/Optionality (occurrence rules)
- Restricted values, code sets, defaults
- Data Type (text, numbers, date, video)
- Relationships/Associations (participation in structures and "ontology")
Modeling "Events" for The Berkeley Calendar Network

The first published Document Engineering case study whose "snapshots" illustrate the analysis, modeling, and schema encoding approach

The problem - scores of calendars on berkeley.edu with overlapping coverage and audiences but incompatible data models

No automated reuse of information; you need to submit events to multiple calendars or copy events from them

Each calendar has a different event submission form and a different model of an event

The UC Berkeley Event Calendar, 2004
The UC Berkeley Event Calendar, 2008

Typical Incompatibility of Event Models

- **U.C. Berkeley Gateway Site**
  - Admission: Registration Required, Ticket Required
  - Cost: 
  - Open to: Public, Campus, Alumni, Students, Faculty, Staff

- **Haas School of Business**
  - Contact Person: 
  - Contact Email: 
  - Repeat: Never
  - Remind: Never
  - E-mail: 
Event Calendar Network: Conceptual Architecture

Information Sources

User Interviews (18)
Event specifications/standards (iCalendar [IETF RFC 2445], SKiCal)
Existing Calendars (23)
Event Calendars: Analysis Strategy

What can we learn from a specific calendar instance?
What can we learn from an "add new event" forms?
But you also have to look at instances and forms in combination
Kept analyzing new calendars until "law of diminishing returns" kicked in

Event Calendars: Harvesting and Consolidating Components

Synonyms:
- Start Date
- Commencement

Homonyms:
- Contact (person submitting an event)
- Contact (person to contact about an event)
- Category / Type (disjoint domains: events, attendees)

Harvesting took on average 2 hours per calendar
Event Calendars Harvest of Candidate Components

Event Calendars Component Consolidation (Simplified)

<table>
<thead>
<tr>
<th>Name</th>
<th>Semantic Description</th>
<th>Source 1</th>
<th>Source 2</th>
<th>Source 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>The title of the event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Date</td>
<td>The date of the event, or</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the first date of a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>recurring event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End Date</td>
<td>The last date of the event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>The location of the event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speaker</td>
<td>Name(s) of the person(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>speaking at the event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The description of the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speaker Title</td>
<td>The title of the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>speaker</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- (merged with SYNONYM, Place)
Event Calendars: The Conceptual Model

When we've analyzed all of the candidate components for dependencies, we've created a conceptual model for event calendars.

From this model we can assemble any of a set of related document types for different varieties of event calendars.

The Complete Conceptual Model
A relational model (a set of tables in our example) simultaneously describes all of the associations among the components; put another way, it doesn't highlight any particular association.

But when we exchange information, we do so to satisfy the requirements in some context.

If there are multiple ways to interpret the content we will not achieve interoperability.

So we impose a contextual interpretation when we create a hierarchy on a relational model.
Document model assembly is the process of creating a model of a document type – hierarchical and nested – by drawing on the "pool" or library of content and structural components.

Assembly involves designing (or selecting a pattern for) the top level structure as an entry point and then navigating through the relationships in the conceptual model to order components to satisfy requirements.

Assembly order can differ whenever there is a bi-directional relationship between components.
Alternate Assemblies

Assembling a Time-based Calendar Model
The Time-Based Calendar Model

The Modeling Debate [1]

Some problems and some domains are inherently complex and a careful, rigorous modeling approach is required

- This "heavyweight" position argues that there are no modeling shortcuts

But some people argue that modeling "involves a substantial amount of work that is often political, tedious, and unpleasant" that should be avoided whenever possible

- Some domains and use cases might be simple enough ("Microformats") that less "heavyweight" modeling approaches could suffice
The Modeling Debate [2]

You should always look to see if someone has already modeled your problem domain (Cover Pages and OASIS)

If the underlying conceptual model of an existing vocabulary doesn't fit your requirements and you must develop your own, you have many choices to make about scope, abstraction, and granularity

---

Modeling "Professor Stories"

Bob Glushko is an Adjunct Full Professor at UC Berkeley's School of Information, located in South Hall. He teaches Information Organization and Retrieval (INFO 202), Document Engineering (INFO 243), and other courses. He has a B.A. from Stanford University (California) and a Ph.D. from UC San Diego.

Coye Cheshire is an Assistant Professor at the School of Information. He recently received his Ph.D. from Stanford University. He teaches Computer Mediation Communication, Social and Organizational Aspects of Computing, and other courses.
Modeling as "Text Blobs"

<Para> Bob Glushko is an Adjunct Full Professor at UC Berkeley's School of Information, located in South Hall. He teaches Information Organization and Retrieval (INFO 202), Document Engineering (INFO 243), and other courses. He has a B.A. from Stanford University (California) and a Ph.D. from UC San Diego. </Para>

"Content Nuggets" in the Text (aka "Mixed Content")

<Para><Name>Bob Glushko</Name> is an <Rank>Adjunct Full Professor</Rank> at the <Institution>UC Berkeley</Institution> <AcademicUnit>School of Information</AcademicUnit>, located in <Building>South Hall</Building>. He teaches <CourseName>Information Organization and Retrieval</CourseName> (<CourseNum>INFO 202</CourseNum>), <CourseName>Document Engineering</CourseName> (<CourseNum>INFO 243</CourseNum>), and other courses. He has a <Degree>B.A.</Degree> from <Institution>Stanford University</Institution> (<State>California</State>) and a <Degree>Ph.D.</Degree> from <Institution>UC San Diego</Institution>.</Para>
A More Structured Professor Story

Professor

Name: Bob Glushko
Rank: Adjunct Full Professor
Affiliation:
  Institution: UC Berkeley
  Academic Unit:
    Name: School of Information
    Building: South Hall
Courses:
  Course:
    Name: Information Organization and Retrieval
    Number: INFO 202
Degrees:
  Degree:
    Institution State="California": Stanford University
    Type: B.A.

Facts in Tabular Format

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>AFFILIATION</th>
<th>COURSES</th>
<th>DEGREES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Glushko</td>
<td>Adjunct Full Professor</td>
<td>UC Berkeley, School of Information (South Hall)</td>
<td>Info Org &amp; Retrieval (INFO 202); Document Engineering (INFO 243)</td>
<td>BA, Stanford (California); Ph.D. UCSD</td>
</tr>
<tr>
<td>Coye Cheshire</td>
<td>Assistant Professor</td>
<td>UC Berkeley, School of Information</td>
<td>Computer-Mediated Communication; Social &amp; Organizational Aspects of Computing</td>
<td>Ph.D. Stanford</td>
</tr>
</tbody>
</table>
Problems with this Organization of the Facts

It may seem that this way of organizing the facts is useful, but there are some problems with it.

This is a "spreadsheet" style of data organization, with rows and columns defining cells that are just "data buckets" buckets into which we can put almost anything.

Some of the "buckets" contain repeating items rather than "atomic" information components.

Some of the "buckets" contain values that are not of the same type.

What relationships describe how different columns go together?

Normalized Tables

<table>
<thead>
<tr>
<th>PROFESSORS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>RANK</td>
<td>AFFILIATION</td>
</tr>
<tr>
<td>Bob Glushko</td>
<td>Adjunct Full Professor</td>
<td>UC Berkeley, School of Information (South Hall)</td>
</tr>
<tr>
<td>Coye Cheshire</td>
<td>Assistant Professor</td>
<td>UC Berkeley, School of Information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSES</th>
<th></th>
<th>COURSE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTRUCTOR</td>
<td>COURSE NAME</td>
<td></td>
</tr>
<tr>
<td>Bob Glushko</td>
<td>Info Org &amp; Retrieval</td>
<td>INFO 202</td>
</tr>
<tr>
<td>Bob Glushko</td>
<td>Document Engineering</td>
<td>INFO 242</td>
</tr>
<tr>
<td>Coye Cheshire</td>
<td>Computer-Mediated Communication</td>
<td></td>
</tr>
<tr>
<td>Coye Cheshire</td>
<td>Social &amp; Organizational Aspects of Computing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEGREES</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RECIPIENT</td>
<td>DEGREE TYPE</td>
<td>INSTITUTION</td>
</tr>
<tr>
<td>Bob Glushko</td>
<td>B.A.</td>
<td>Stanford</td>
</tr>
<tr>
<td>Bob Glushko</td>
<td>Ph.D.</td>
<td>UCSD</td>
</tr>
<tr>
<td>Coye Cheshire</td>
<td>Ph.D.</td>
<td>Stanford</td>
</tr>
</tbody>
</table>
Readings for INFO Lecture #11

Robert J. Glushko and Tim McGrath, Document Engineering, Chapter 6, "When Models Don't Match: The Interoperability Challenge"

Michael Stonebraker and Joseph Hellerstein, "Content Integration for E-Business"