



# No SQL Metadata Management

October 20<sup>th</sup>, 2010

Dan McCreary  
President  
Dan McCreary & Associates  
[dan@danmccreary.com](mailto:dan@danmccreary.com)

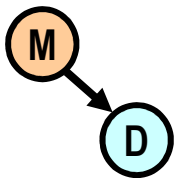
# Presentation Description

---

Metadata, or data that describes data, is fundamentally different than data itself. The management of metadata is becoming a strategic area for many organizations and the topic of data governance is also becoming central to the data strategies for many organizations.

This presentation will look at how the requirements of enterprise metadata management dictate that new schema-free web application architectures be better suited to the task of metadata management. These new “zero translation” architectures combine some of the best aspects of document management systems and traditional tabular data management but without the complexity of traditional multi-tier architectures.

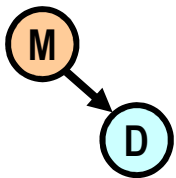
We will give examples of how these new XML-centric architectures are being used to solve metadata management challenges and how they empower non-programmers to build and maintain metadata registries.



# Outline

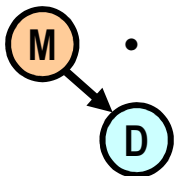
---

- **Part 1**
  - Background on NO-SQL
  - What is metadata?
  - Enterprise Metadata Management (EMM) requirements
  - Role of agility
  - Why **XRX** systems are agile
- **Part 2**
  - Tour of a native XML system and a metadata registry
  - XQuery, REST and XForms
  - XML web services
  - How empower Bas and other non-programmers
  - How to start a pilot project
  - Questions



# After This Presentation Users Will Be Able To:

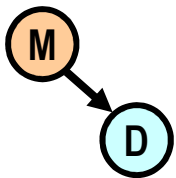
- **Define** metadata and compare and contrast metadata management with data management
- Describe the high-level **features** of enterprise metadata management systems
- Differentiate between metadata **repositories** and metadata **registries**
- Understand the role of **duplication** in managed metadata environments
- Understand the role of **ISO-Standards** in metadata management
- Describe the major **application architectures** and the number of **data translations** used in each architecture
- Define "Zero translation" application architectures
- Define metadata **agility** and the metrics used to measure metadata agility
- Describe the **XRX** architecture and XML search
- Understand the role of native xml systems and the **XQuery** language
- Access resource for creating a **pilot metadata registry project**




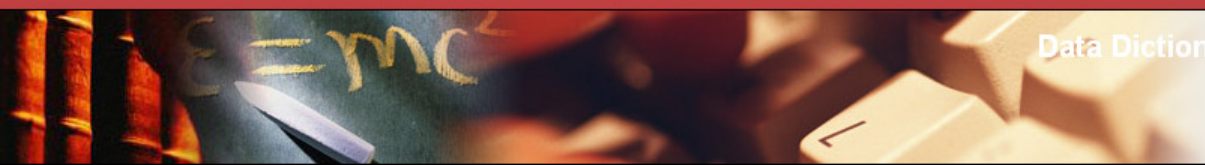
# Background for Dan McCreary

---

- Enterprise data architecture consultant based in Minneapolis
- Strong interest in enterprise metadata management and semantic web
- Builds metadata registries using ISO/IEC 11179 and US Federal XML standards (NIEM.gov)
- **Customers:** CriMNet/BCA, MN Dept. of Education, MN Dept. of Revenue, Thrivent Financial, Patriot Data Systems, US Department of State, MN Historical Society, US Library of Congress, Mindware, Syntactica, Surescripts



# Origins: The XML Data Dictionary

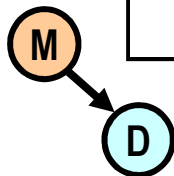
## MDE Data Dictionary

- Data Dictionary Home
- Search Dictionary
- Elements By Team
- Elements By Namespace
- About the Data Dictionary
- Frequently Asked Questions

## Data Dictionary Home - Browse Data Elements

Search by Data Element Name:  
[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

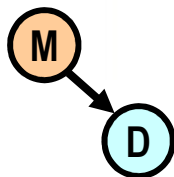
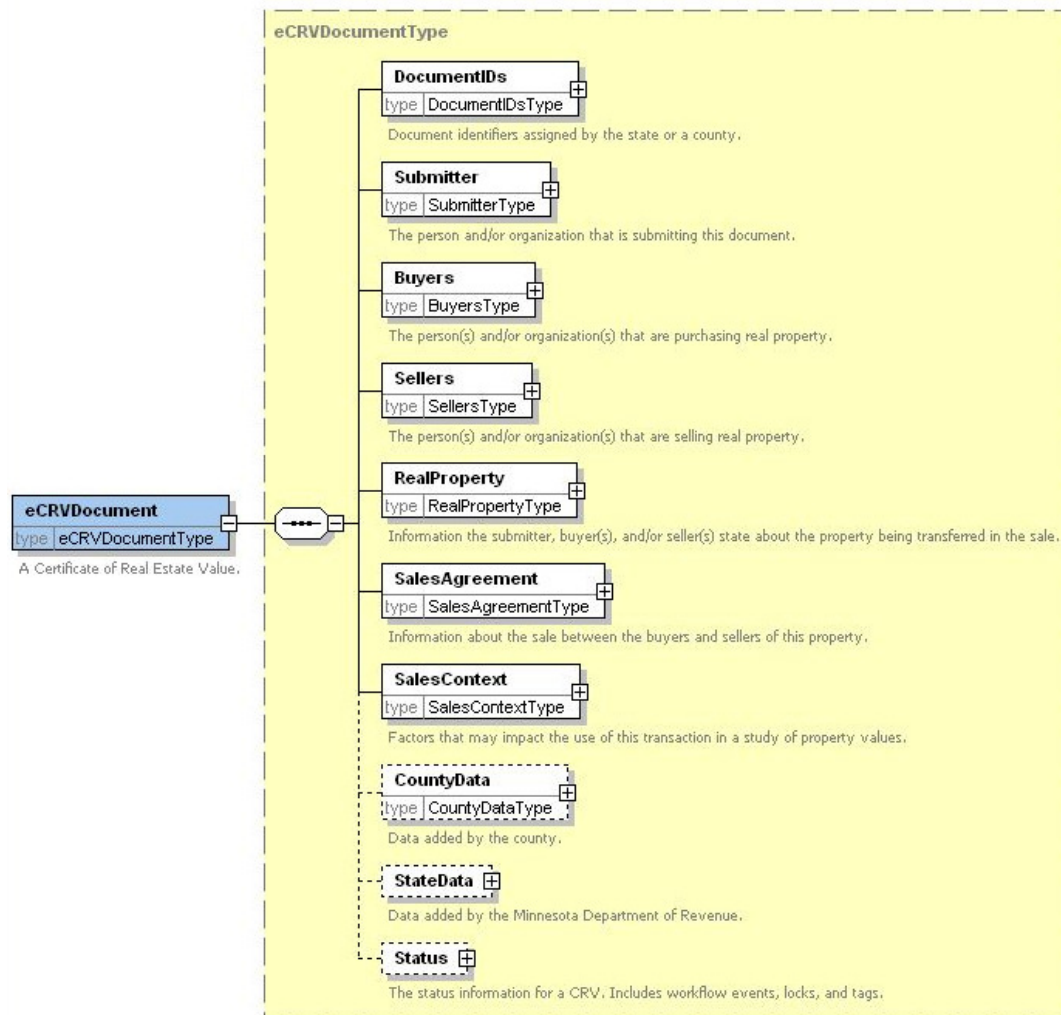
	Data Element Name	Owner	Status
1	<a href="#">ActivityAthleticsCode</a> A code to indicate the type of athletic activity in a program. Often referred to as a sport.	DataModelingTeam	initial-draft
2	<a href="#">ActivityDate</a> The date that an activity occurred upon.	DataModelingTeam	initial-draft
3	<a href="#">ActivityEndDate</a> An end or completion date of an activity that occurs over a duration of time.	DataModelingTeam	initial-draft
4	<a href="#">ActivityFiscalYearNumber</a> The fiscal year that the Activity is classified under as determined by the accounting policy of the data owner.	DataModelingTeam	initial-draft
5	<a href="#">ActivitySchoolYearCode</a> A five character string that specifies the school year such as 04-05.	DataModelingTeam	assigned-to-review-team
6	<a href="#">ActivityTest</a> A generic moment-interval container for data associated with an event that occurs at a specific point in time or time interval.	DataModelingTeam	initial-draft
7	<a href="#">Address</a> A postal location to which paper mail can be directed.	DataModelingTeam	approved-for-publication
8	<a href="#">AlternativeAssessment</a> The Minnesota Alternative Assessment Test.	SpecialEducation	assigned-to-review-team
9	<a href="#">AngularMinute</a> A structure that describes a location based on latitude and longitude of a Location.	DataModelingTeam	initial-draft
10	<a href="#">AngularSecond</a> A structure that describes a second of a degree, with a restricted range of 0 (inclusive) to 60 (exclusive).	DataModelingTeam	initial-draft
11	<a href="#">Assessment</a> A person having a possible assessment administered.	DataModelingTeam	initial-draft



# Electronic Certificate of Real Estate

Summer 2006

1 Document  
= 44 SQL inserts



# 250 Data Elements

## MINNESOTA E CRV Project

You are currently logged in as **dakota** of **dakota** and you have the role[s] of: **auditor assessor supervisor auditor-dakota**

### Minnesota Certificate of Real Estate Valuation Form - editing CRV# 19-08-49

Auditor ID#

County Edit Form Version 1.0 created on December 17th, 2007

[Back to Dashboard \(abandon edit\)](#) | [Save form information \(not available for demo\)](#) | [Send feedback to the team](#)

#### CRV Status and lock preview

County Accepted:	<input checked="" type="checkbox"/>
Auditor Complete:	<input type="checkbox"/>
Assessor Complete:	<input type="checkbox"/>
County Final (and Lock):	<input type="checkbox"/>
State Review Complete:	<input type="checkbox"/>
State Study Lock:	<input type="checkbox"/>
County Edit Lock:	<input type="checkbox"/>
State Edit Lock:	<input type="checkbox"/>

## XForms Mockup

SummaryBuyersSellersPropertySales AgreementSupplementaryCountyWorkflow

Current Workflow for this CRV Document

ID	Date/Time	User	Org	Code	Activity	Comment
1	2008-01-14, 08:42	anonymous	anonymous	original		
2	2008-01-14, 08:44	dakota	dakota	county-accepted		County Accepted CRV
3	2008-01-14, 08:45	dakota	dakota	activity		CRV edited and saved
4	2008-01-14, 08:47	dakota-as	dakota	assessor-assigned		assigning to assessor
5	2008-01-15, 10:39	dakota	dakota	activity		CRV edited and saved

Add new Workflow event

CRV Workflow Code:

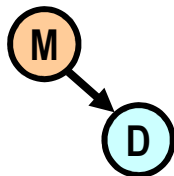
County Code Text (only relevant on county code selected):

Comment:

Add Workflow Code

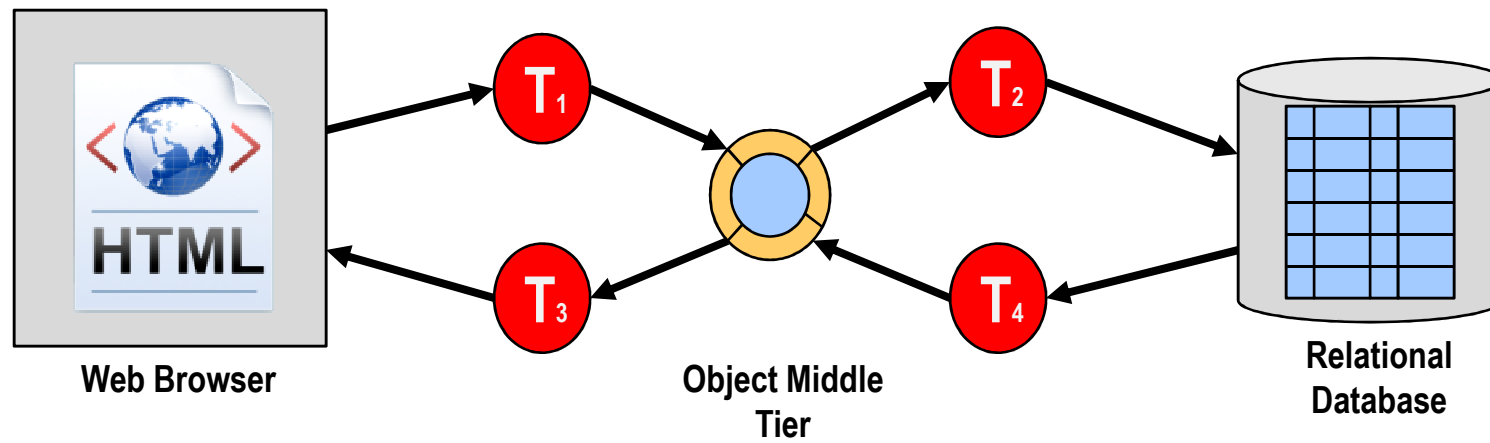
Tags

ID	Org	Date/Time	Keyword	Comment	Work
dakota	dakota	2008-01-14, 08:44	county-accepted	CRV has been accepted and automatically tagged by the accept process	2

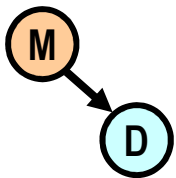




# Four Translations



- $T_1$  – HTML into Java Objects
- $T_2$  – Java Objects into SQL Tables
- $T_3$  – Tables into Objects
- $T_4$  – Objects into HTML

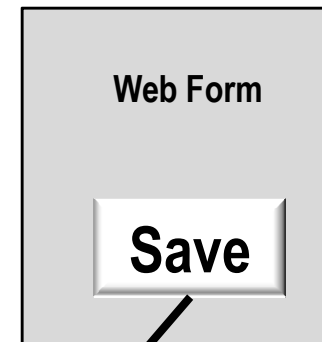


# Kurt's Suggestion

Use a  
A Native XML  
Database!



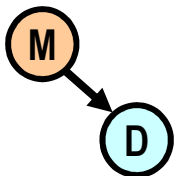
Kurt Cagle



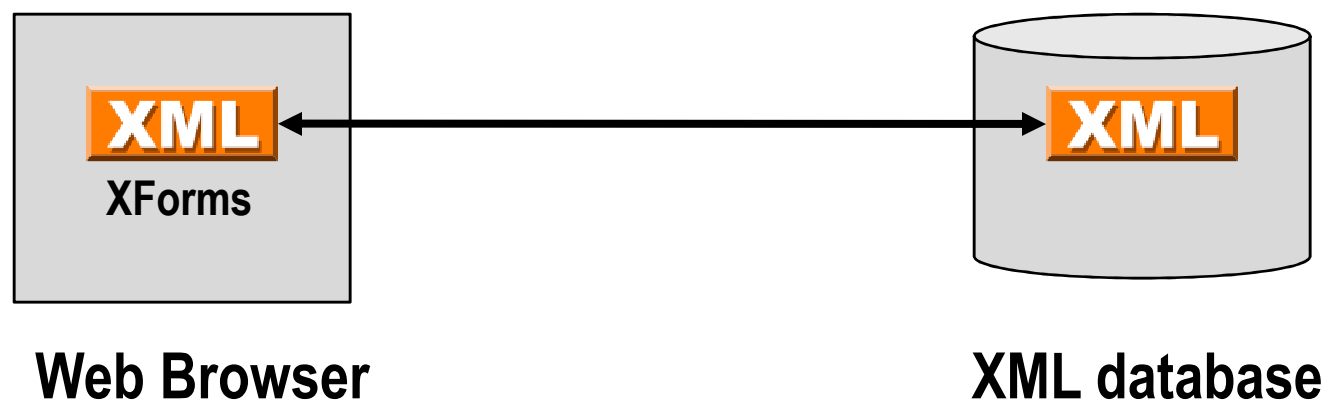
Web Browser

```
store($collection, $file-name, $data)
```

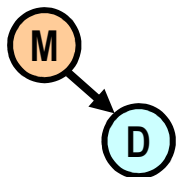
eXist



# Zero Translation

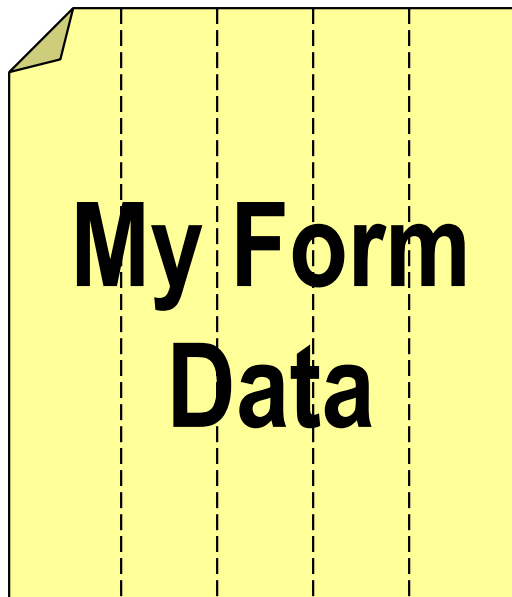


- XML lives in the web browser (**X**Forms)
- **REST** interfaces
- XML in the database (Native XML, **X**Query)
- **XR**X Web Application Architecture
- No translation!

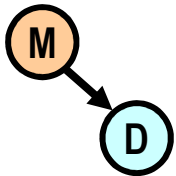


# No-Shredding!

---



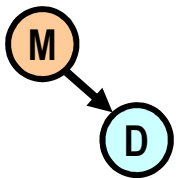
- Relational databases take a single hierarchical document and shred it into many pieces so it will fit in tabular structures
- Native XML databases prevent this shredding



# Is Shredding Really Necessary?

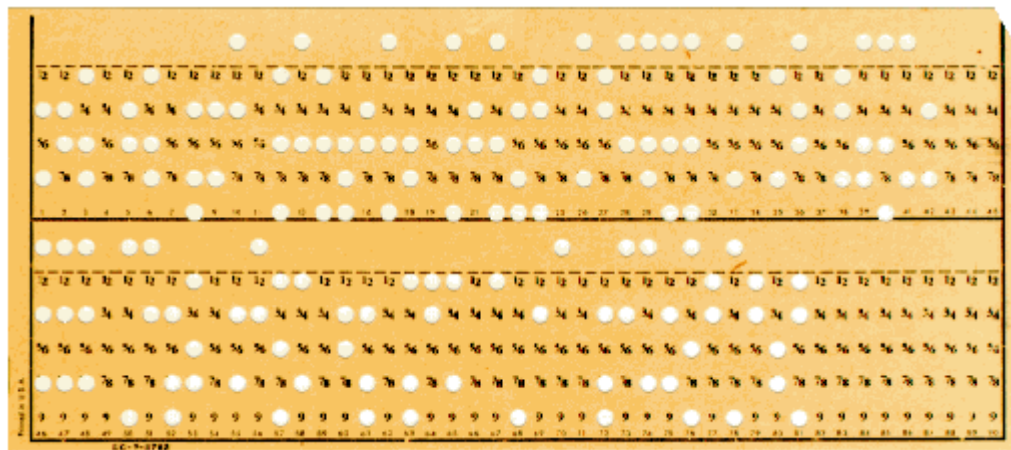


- Every time you take hierarchical data and put it into a traditional database you have to put repeating groups in separate tables and use SQL “joins” to reassemble the data



# Many Processes Today Are Driven By...

The constraints of yesterday...



## Challenge:

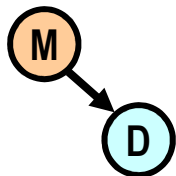
Ask ourselves the question...

Do our current method of solving problems with tabular data...

Reflect the storage of the 1950s...

Or our **actual** business requirements?

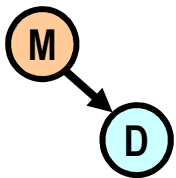
What structures best solve the actual business problem?



# "Schema Free"

---

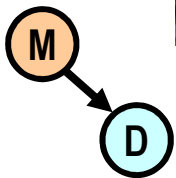
- Systems that automatically determine how to index data as the data is loaded into the database
- No *a priori* knowledge of data structure
- No need for up-front logical data modeling
  - ...but some modeling is still critical
- Adding new data elements or changing data elements is not disruptive
- Searching millions of records still has sub-second response time



# Monoculture and Mono-architecture



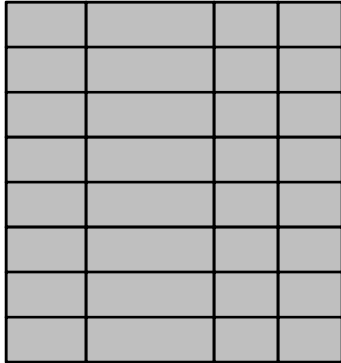
Image Source: Wikipedia



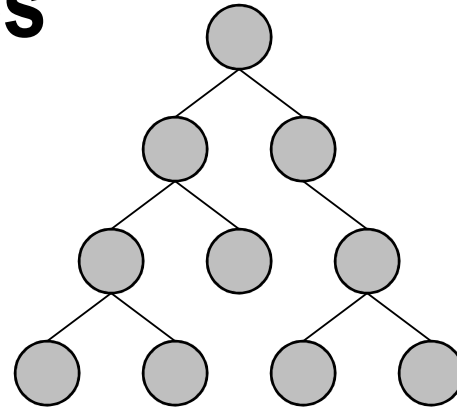


# Storage Architectural Patterns

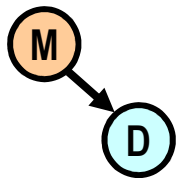
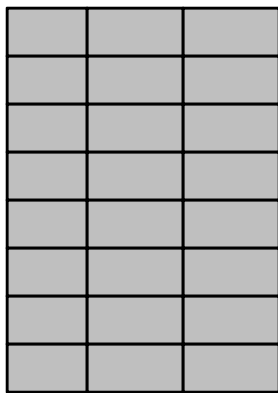
## Tables



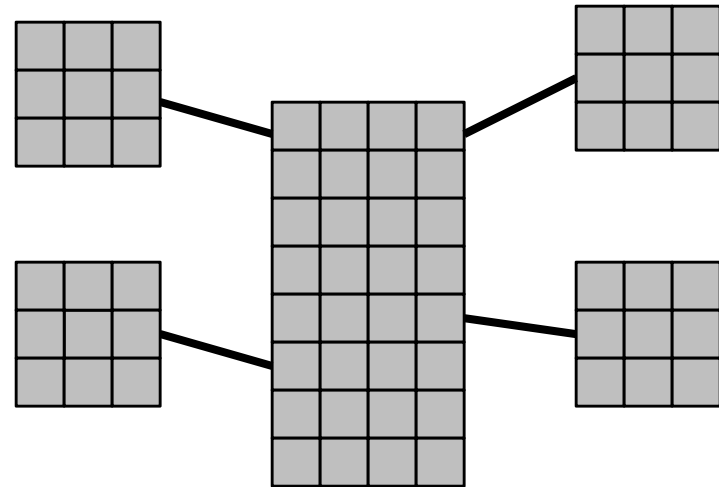
## Trees



## Triples



## Stars



# The NO-SQL Universe

## Key-Value Stores

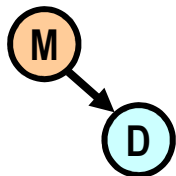


**Project Voldemort**  
*A distributed database.*

## Graph Stores

**AllegroGraph® RDFStore**

## Object Stores

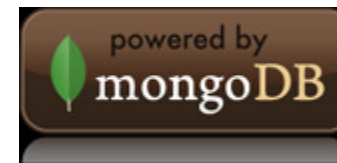


## Document Stores

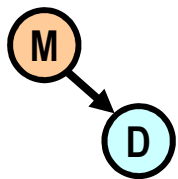
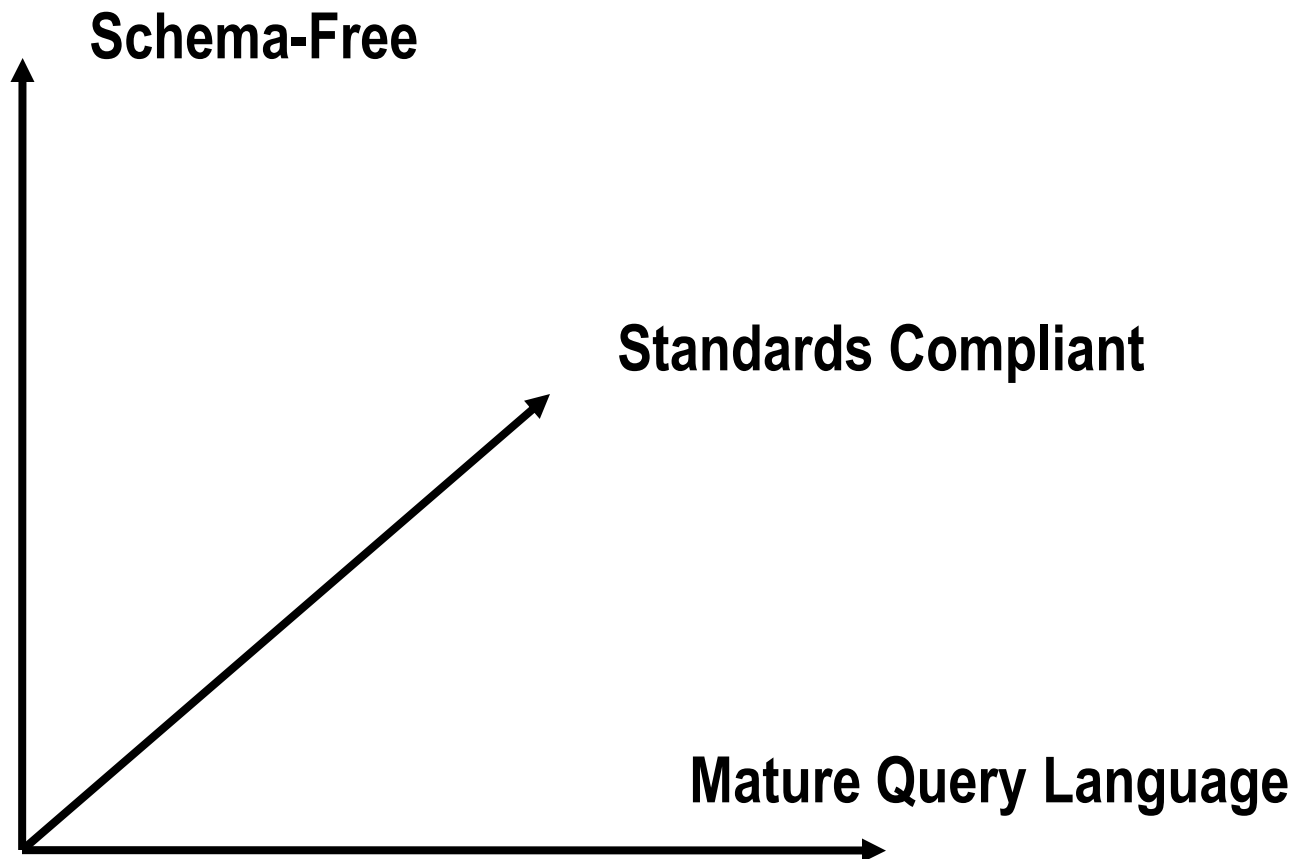
```
<SPEECH>
<SPEAKER>HAMLET</SPEAKER>
<LINE>Rest, rest, perturbed spirit!</LINE>
<STAGEDIR>They enter</STAGEDIR>
<LINE>So, gentlemen, I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
<LINE>I am dead, you are not. I am dead, you are not.
</SPEECH>
```



} XML



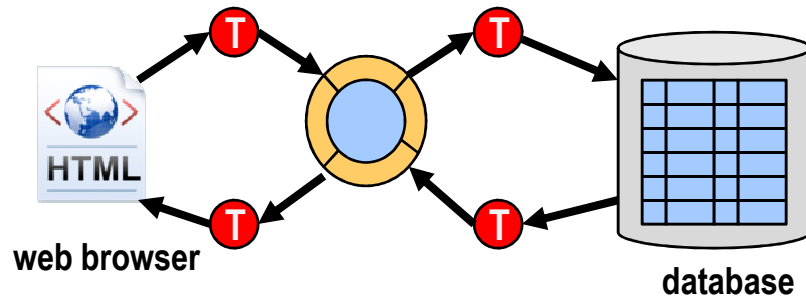
# Finding the Right Match



Use CMU's Architectural Tradeoff and Modeling (ATAM) Process

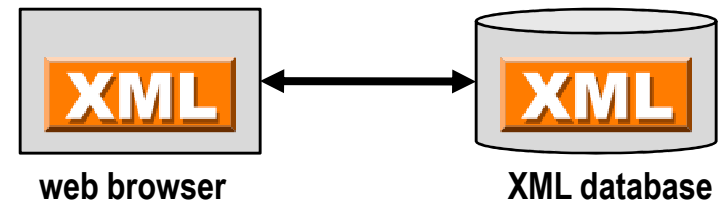
# Architectural Summary

## Four Translation



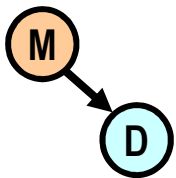
- HTML web pages
- Object middle tier
- RDBMS database

## Zero Translation



- XForms Client
- Native XML Database

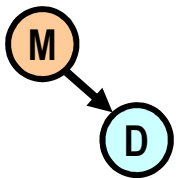
**Which system more agile and by how much?**  
**How can this help us manage enterprise metadata?**



# What Is Metadata?

- Data about data
- Data that describes other data

Last Name	First Name	Title	Phone	} Metadata  Data
Smith	John	BA	x1234	
Anderson	Sue	PM	x4567	
Johnson	Becky	QA	x8765	

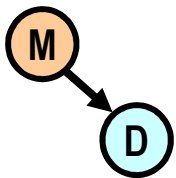


# Data

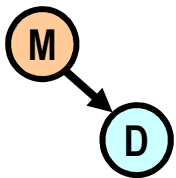
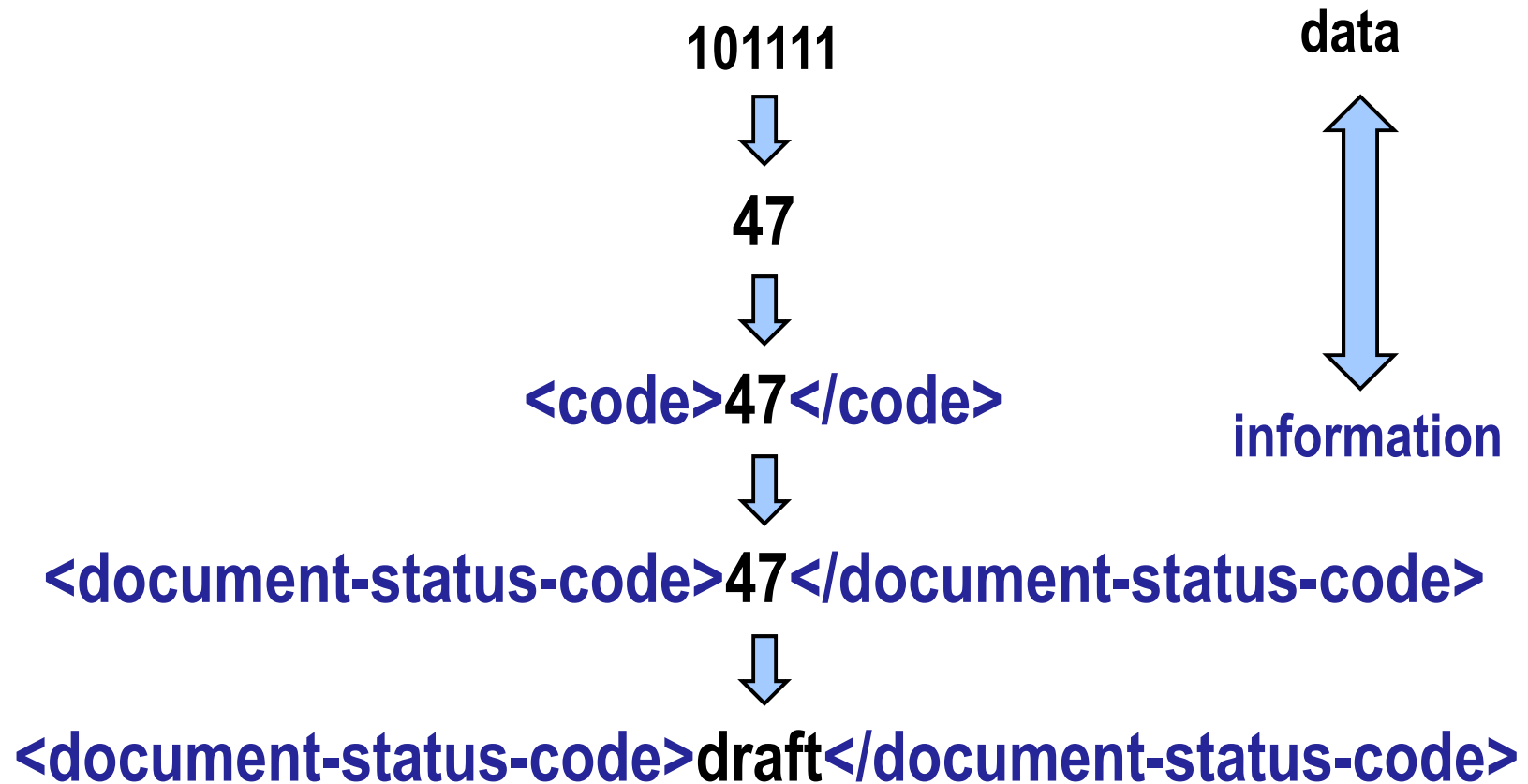
---

```
1010001010001010100101101001000101000101010010
1101000101000101010010110001010001000101010100
10110001110110101000011101001011000101001110110
10100000111010010110001010011101101010111011010
```

Raw data is just values without **context**

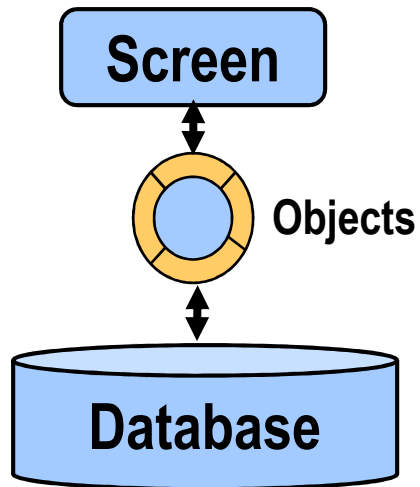


# Adding **Context** Turns Data into Information

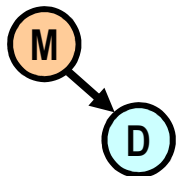


# Two Kinds of Thinking

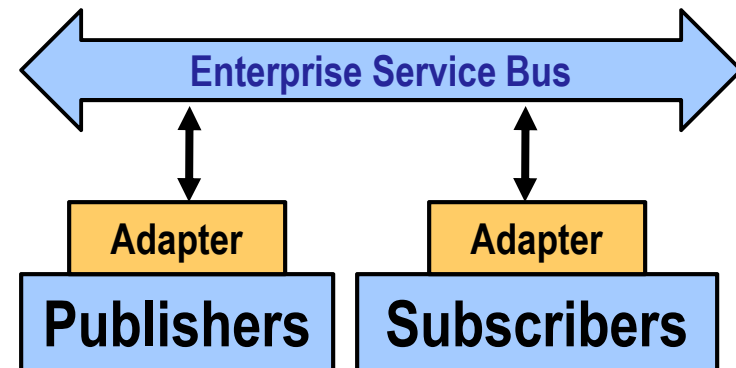
## "In the Can"



- Vertical
- Soloed
- Translation-intensive
- Application-centric
- Good for small teams



## "On The Wire"



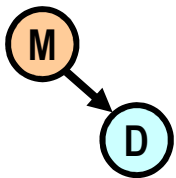
- Horizontal
- Publish/Subscribe
- Messages
- Communication of Shared Meaning (Semantics)
- Good for large organizations



# Managed Metadata

---

- The processes surrounding the creation and management of enterprise metadata and their definitions
  - ISO 11179: "Administered Items"
  - Traceability:
    - Who created data definitions and when and in what context for what purpose?



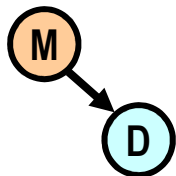
# Repository vs. Registry

## Metadata **Repository**

- Were **any** metadata is stored
- No focus on duplicate element elimination
- No strict controls on removal of imprecise data elements
- Function-specific data

## Metadata **Registry**

- Where carefully controlled metadata is stored
- Focus on elimination of duplicate data elements
- Focus on semantics
- Subject area classification
- Data stewardship
- Follows ISO guidelines



# Empower the Business Analysts!

## Before Registry

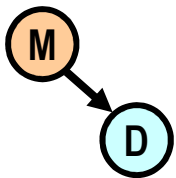


*Sorry, we have no idea  
what code 47 means.*

## After Registry SUPER BA!



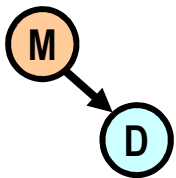
*Let me just search our registry...  
I'll have your answer in 150 milliseconds.*



# EMM Requirements

---

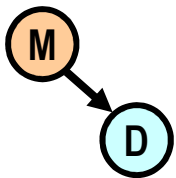
- **EMM** = Enterprise Metadata Management
- Tools to create a "enterprise trust" in data element data definitions (Data Governance)
- Tools to eliminate duplication of data elements
- Powerful search
- Metadata web services
- Controls on who adds and updates definitions
- Support for data stewardship



# ISO/ICE 11179 Metadata Registry

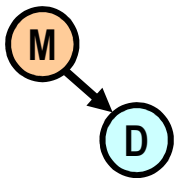
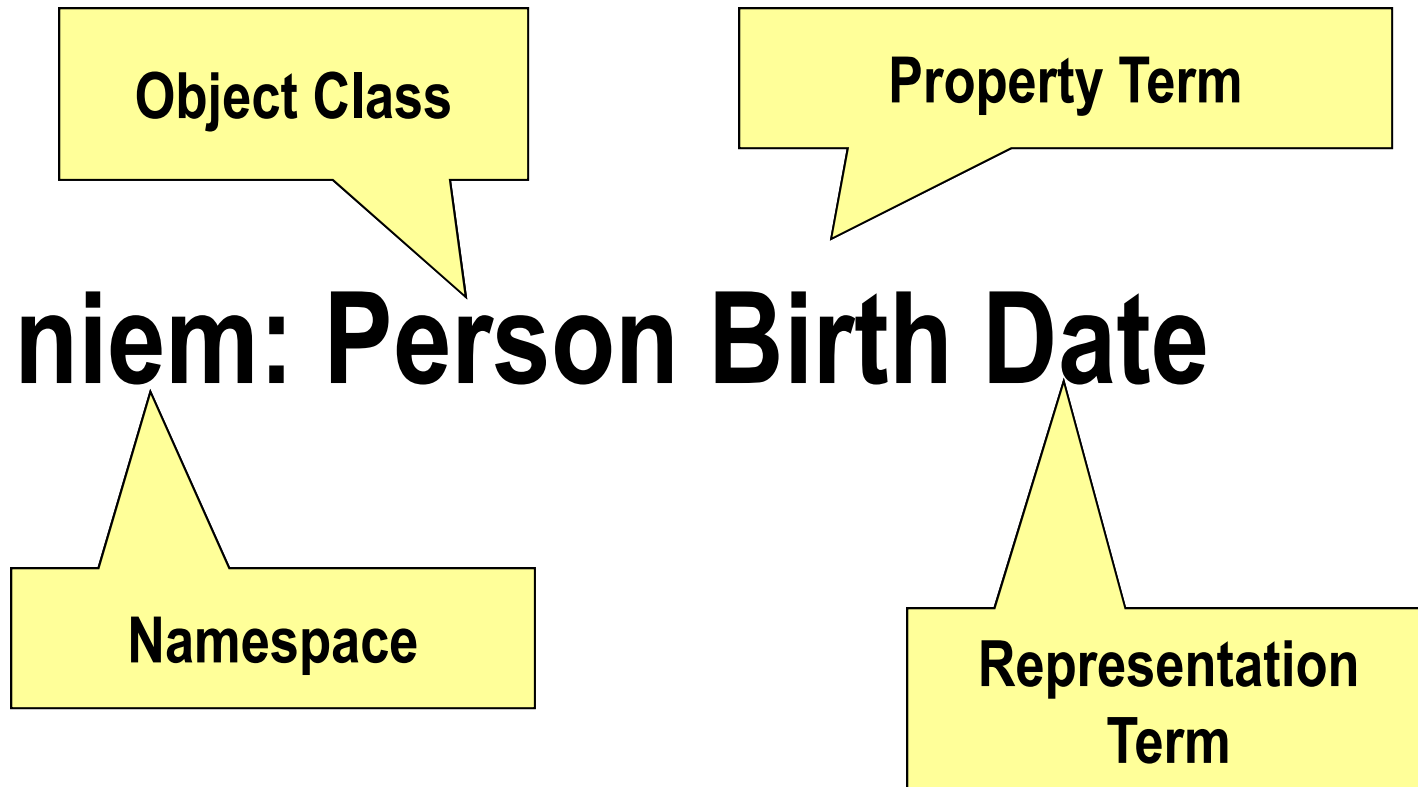
---

- Standards for managing enterprise semantics
- Focus on the management of a "Library" of metadata based on subject headings (like the Dewey Decimal System)
- Guidelines for creating precise data definitions
- Guidelines for classification of data types

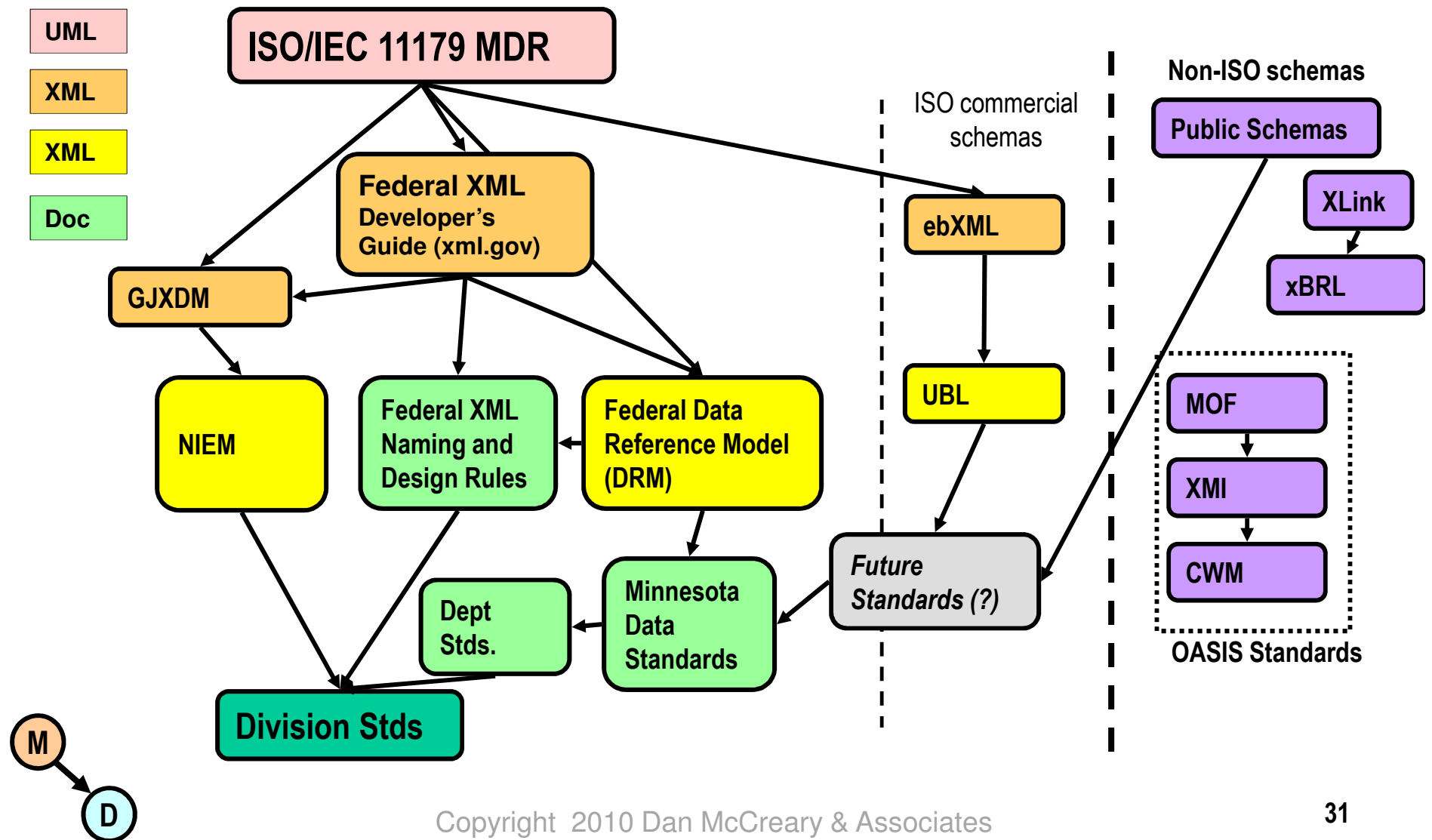


# ISO Naming Conventions

---



# Metadata Standards

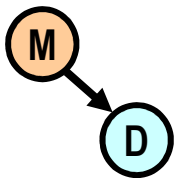


# Why is XRX More Agile?

---

- Importing data
- Querying data
- Creating web services
- Exporting
- Publishing

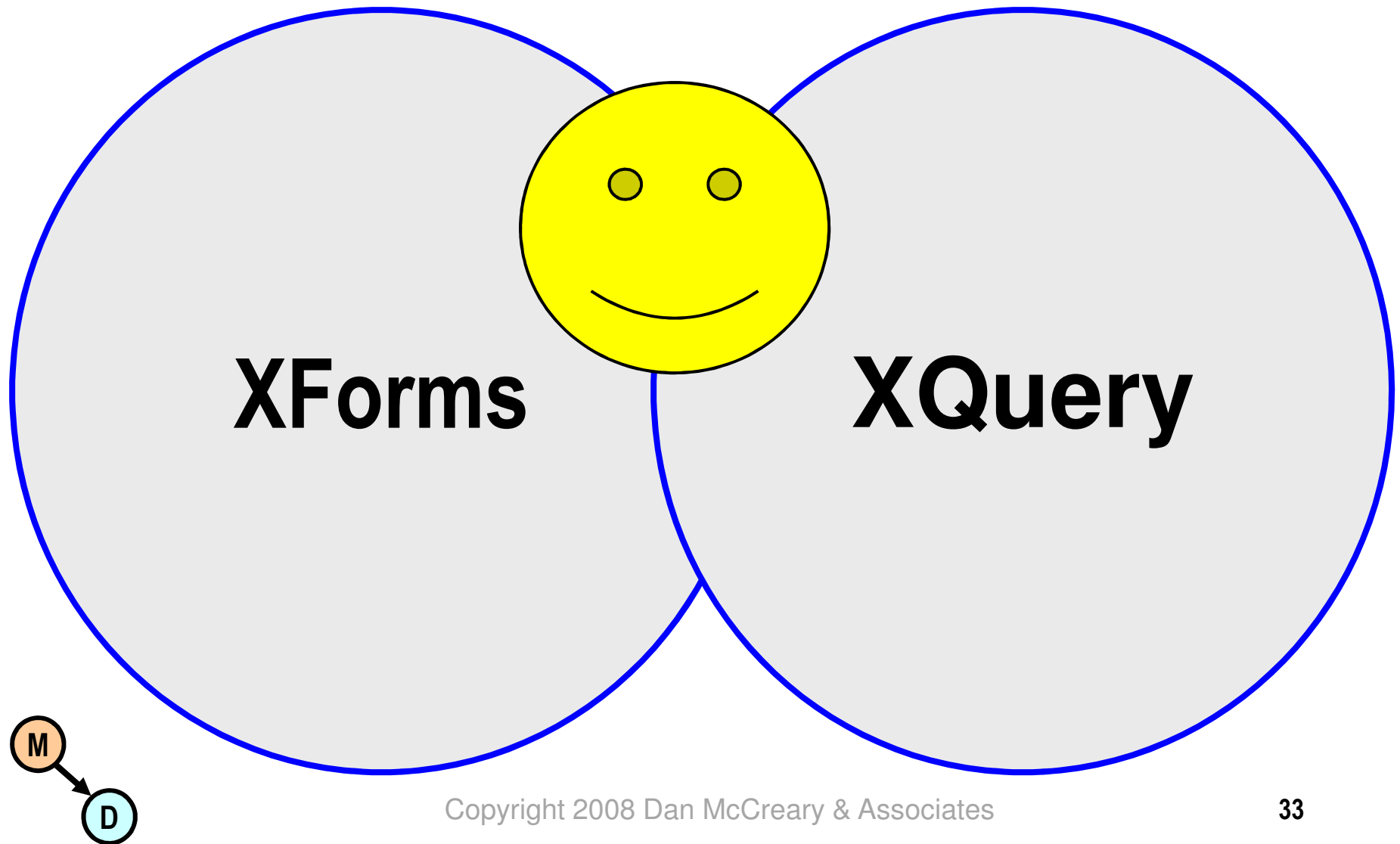
(not to be confused with "Agile Development")





# A Happy Partnership

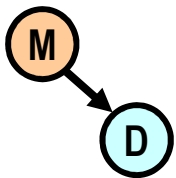
---

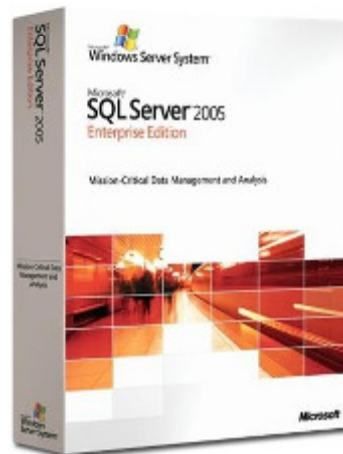
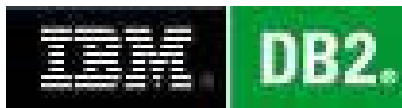


# XQuery

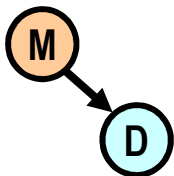
---

- In 1998 Jonathan Robie and Joe Lapp (then the principal architect of WebMethods) created a language called XQL
- In 1998, two query languages, XQL and XML-QL got a lot of interest within the W3C and a working group for XML-based querying languages was formed
- The working group selected around 90 use cases and compared the ability of seven advanced query languages to execute them
- None of the seven were perfect. Each had some defects
- The working we took the best part of each of the seven languages and created the XQuery standard





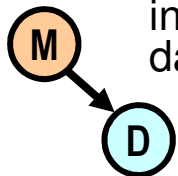
- eXist (open source)
- MarkLogic
- IBM DB2 Version 9  
“PureXML”
- Microsoft SQL Server  
2005
- Oracle 10g Release 2  
Enterprise Edition
- + 50 others...



# It is Easy to Import Data

## SQL

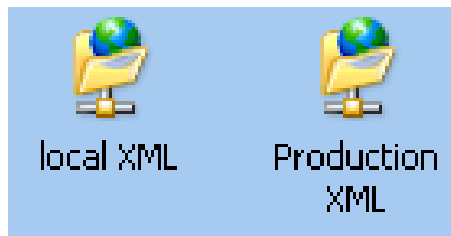
1. Analyze data for all parent child relationships and repeating groups
2. Design logical and physical ER diagrams
3. For each table create a Data Definition File using a data definition language (DDL)
4. Create indexes using DDL
5. Create one table for each set of repeating set of data
6. Run DDL on database creating tables using the appropriate data types
7. Create indexes
8. Create Insert statements
9. Create separate insert statements for each repeating group
10. Run Insert statements on primary structures in database
11. Use primary keys of the first data inserts as foreign keys of dependant data structures



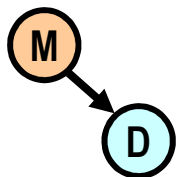
## XQuery

1. Drag XML files into folder

# XML File system



- XML File system – a way of storing information in XML that can be quickly searched
- You can drag and drop almost any files onto this file system
- You access it by using the Microsoft Windows “My Network Places” function (WebDAV)
- But... You can query the file system like a relational database

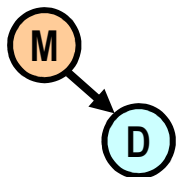


# Functional Programming

---

$$y = f(x)$$

- Computer programs are like mathematical functions
- Developers do not manipulate states and variables (things that change value), but focus entirely on constants and functions (things that never change)
- Functions are treated as first class citizens
- Functions that take other functions as input
- Makes it very easy to build modular programs
- Software written in FP languages tend to be very concise and easy to port to parallel systems

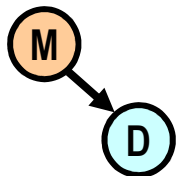


[http://en.wikibooks.org/wiki/Computer\\_programming/Functional\\_programming](http://en.wikibooks.org/wiki/Computer_programming/Functional_programming)

# It's Easy to Query XML Data

```
SELECT COL1, Col2  
FROM TABLE  
WHERE COL1=1
```

Col1	Col2
1	A
1	B
1	C
1	D

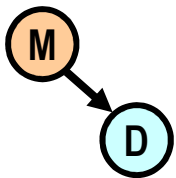


```
for $r in doc('t.xml')//row  
where col1=1  
return $r/col1, $r/col2
```

```
<root>  
  <row>  
    <col1>1</col1><col2>A</col2>  
  </row>  
  <row>  
    <col1>1</col1><col2>B</col2>  
  </row>  
  <row>  
    <col1>1</col1><col2>C</col2>  
  </row>  
  <row>  
    <col1>1</col1> <col2>D</col2>  
  </row>  
</root>
```

# SQL is similar to XQuery

Function	SQL	XQuery
Selecting Distinct Values	<b>SELECT DISTINCT</b>	distinct-values(\$doc)
Row Restriction	<b>WHERE COL=value</b>	where \$r/element=value
Sorting	<b>SELECT C1, C2</b> <b>FROM TABLE</b> <b>ORDER BY C1</b>	for \$r in \$doc/r order by \$r/element



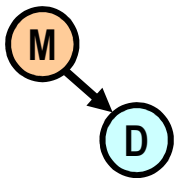


# It is Easy to Create A Web Service

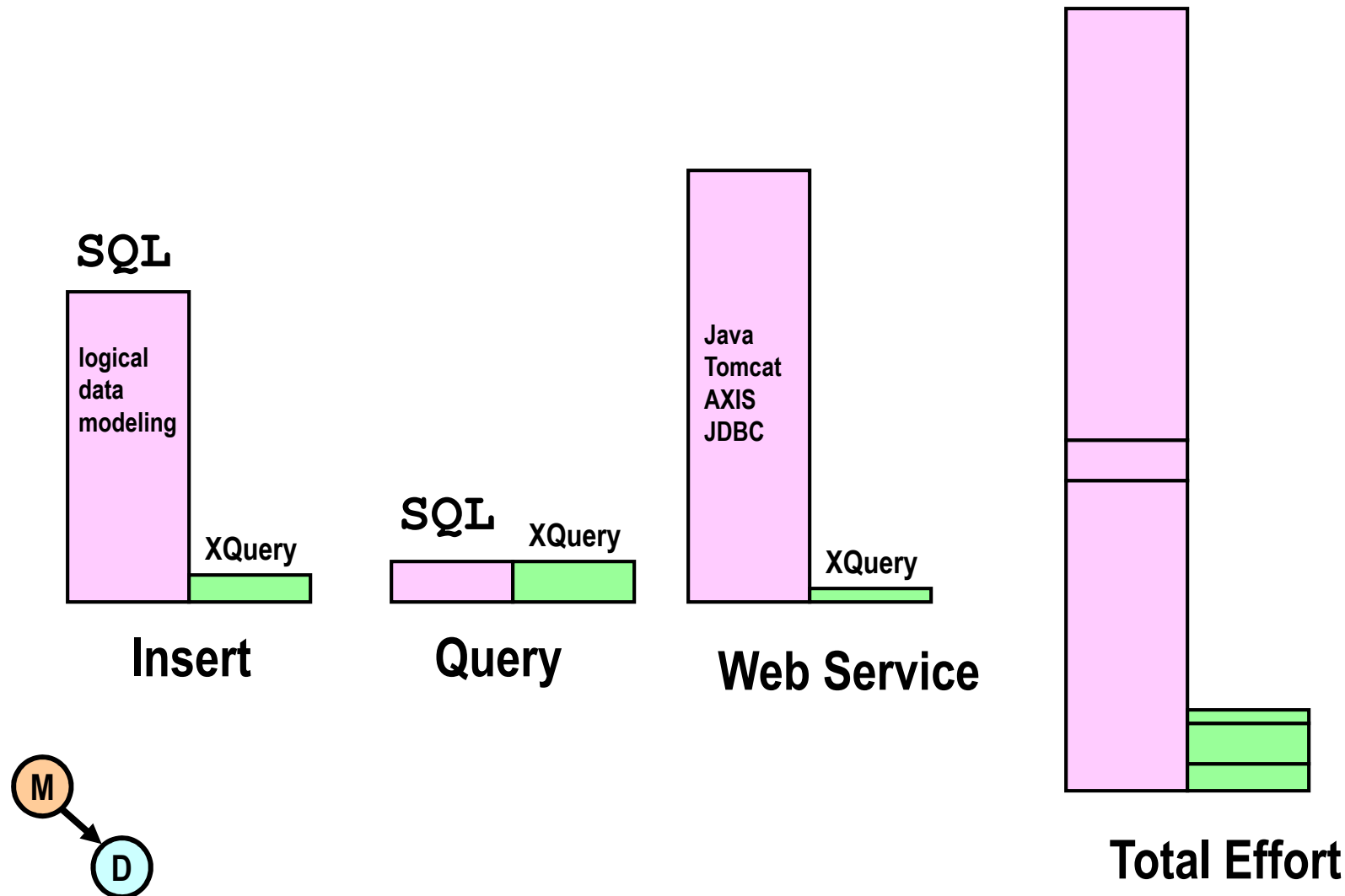
## Java/JDBC/SQL

1. Learn Java or find a Java Developer
2. Install TomCat Web Server
3. Install Java AXIS Web Server
4. Write a JDBC program that sends SQL queries to a database
5. Get the results back in Java Result Object structures
6. Go through the Java Results Structures and use print statements to wrap XML tags around the strings in the result objects
7. Rename your class files to .jws files
8. Add the .jws files to the TomCat deploy folders
9. The WSDL files will automatically be generated

All XQuerys **are** web services



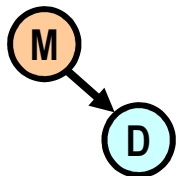
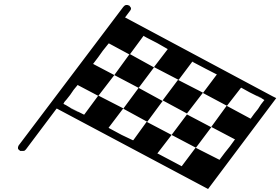
# Insert/Select/Publish Comparison



# High Level Comparison

	SQL	XSLT	XQuery
Query tabular data	Yes	Yes	Yes
Query hierarchical data	No	Yes	Yes
Easy for people to learn	Yes	No	Yes

The winner!



XQuery can be as easy to learn as SQL but also works with hierarchical data structures.

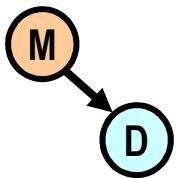
Copyright 2008 Dan McCreary &  
Associates

# XQuery is Easier To Learn Than XSLT

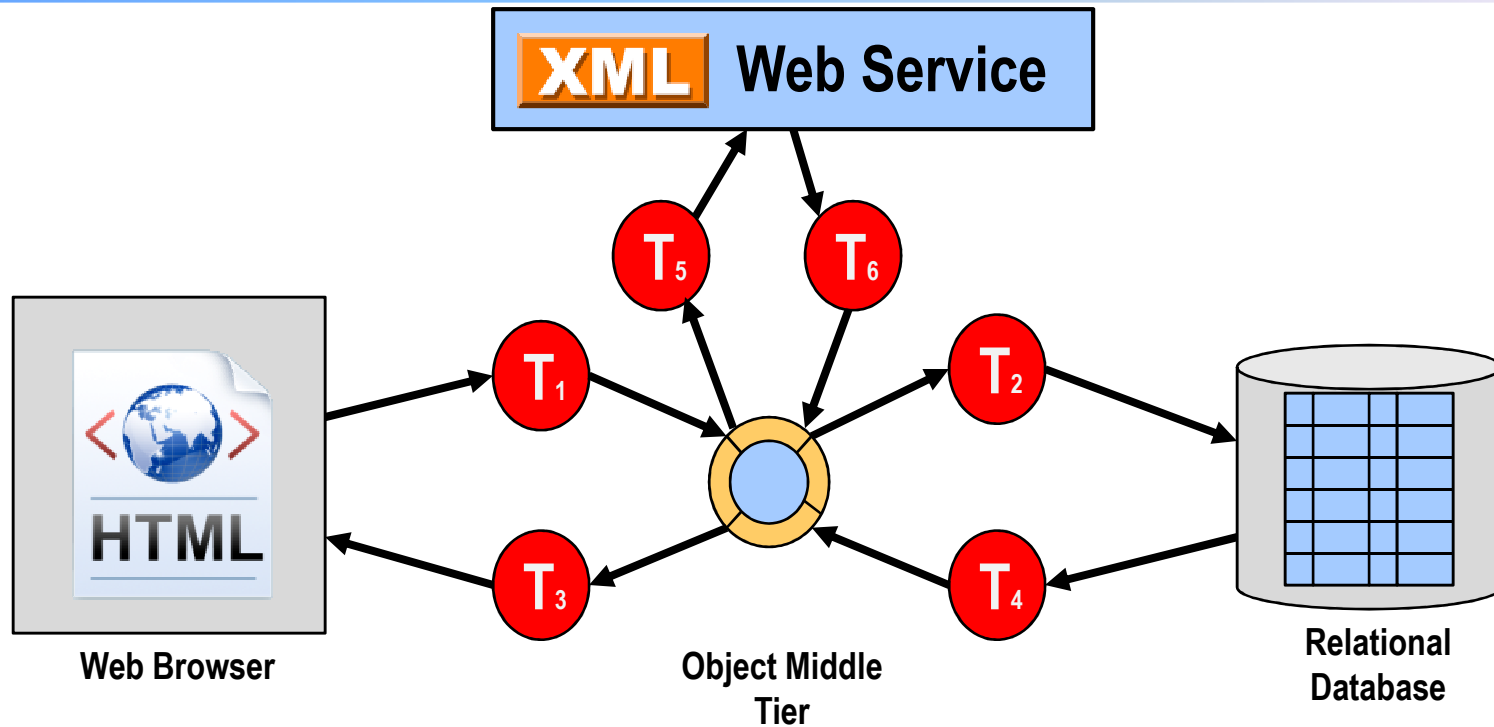
---

- Studies have shown that XQuery is much easier to learn than XSLT, especially if users have some SQL background

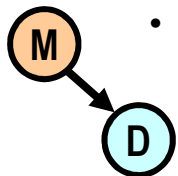
**Usability of XML Query Languages**  
**Joris Graaumans**  
**SIKS Dissertation Series No 2005-16,**  
**ISBN 90-393-4065-X**




# Six Translation



- T<sub>1</sub> – HTML into Java Objects
- T<sub>2</sub> – Java Objects into SQL Tables
- T<sub>3</sub> – Tables into Objects
- T<sub>4</sub> – Objects into HTML
- T<sub>5</sub> – Objects to XML
- T<sub>6</sub> – XML to Objects



# Requirement Lister

 Minnesota Historical Society

[MHS Home](#) [Apps](#) [Requirements Manager](#)

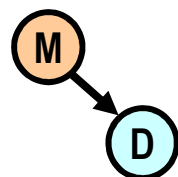
All Requirements

Total Count=57

Sort by Requirement Name

**Count Sort** **Table Header**

id	Name	Parent	Description	Priority	Status	Edit
50	<a href="#">Administrative Reports</a>	6	Reports on who did what and when. These are usage audit trails. .	very-high	approved	<a href="#">Edit</a>
42	<a href="#">BAG-IT Standard</a>	5	The system should support the BAG-IT standard..	medium	removed	<a href="#">Edit</a>
48	<a href="#">Backup and Restore</a>	9	This system must be able to backup and restore collections. Both manual and automatic bac...	high	approved	<a href="#">Edit</a>
34	<a href="#">Batch Uploads</a>	3	The system must be able to support the ability to upload batches of XML files from a contr...	critical	approved	<a href="#">Edit</a>
51	<a href="#">Compare XML Schema Versions</a>		The system must be able to compare versions of XML Schema to indicate what elements have c...	medium	approved	<a href="#">Edit</a>
10	<a href="#">Controlled Vocabulary</a>	9	The system must be able to publish a controlled vocabulary of business terms..	low	approved	<a href="#">Edit</a>
20	<a href="#">Cost Effective</a>	1	The system must be cost effective..	critical	approved	<a href="#">Edit</a>
26	<a href="#">Customizable by Non Programmers</a>	20	Consideration should be made to allow non-programmers the ability to enhance and modify bu...	high	approved	<a href="#">Edit</a>
30	<a href="#">Display as HTML</a>	19	The system must be able to display all documents as HTML documents..	very-high	approved	<a href="#">Edit</a>



**Click to View Item**

**Click to Edit Item**

# Item Viewer



[MHS Home](#) [Apps](#) [Requirements Manager](#) > [List Requirements](#)

## View Requirement 16

**ID:** 16

**Parent ID:** 3

**Name Short Name:** Import XML Data

**Description:** The system must be able to import well-formed XML data.

**Name Short Name:** Import XML Data

**Priority:** critical

**Completion Status:** approved

**Approved By:** robert.horton@mnhs.org

**Approval Date:** 2009-09-04

**Notes:**

**Classifiers:** architecturally-significant

**Tags:**

**Testable:** unknown

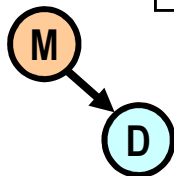
**XML:** </db/cust/mhs/apps/requirements/data/16.xml>



[Edit Item](#) [Delete Item](#)

# View XML Data

```
- <Requirement>
  <id>16</id>
  <RequirementParentID>3</RequirementParentID>
  <RequirementShortName>Import XML Data</RequirementShortName>
  - <RequirementDescriptionText>
    The system must be able to import well-formed XML data.
  </RequirementDescriptionText>
  <ProjectID>1</ProjectID>
  <RequirementLevelCode>0</RequirementLevelCode>
  <RequirementPriorityCode>critical</RequirementPriorityCode>
  <RequirementCompletionStatusCode>approved</RequirementCompletionStatusCode>
  <RequirementApprovedBy>robert.horton@mnhs.org</RequirementApprovedBy>
  <RequirementApprovalDate>2009-09-04</RequirementApprovalDate>
  <RequirementTestableCode>unknown</RequirementTestableCode>
  <RequirementClassifierCode>architecturally-significant</RequirementClassifierCode>
  <RequirementTag/>
  <RequirementAssignedToPersonID/>
  <RequirementBusinessRuleID/>
  <RequirementNoteText/>
</Requirement>
```

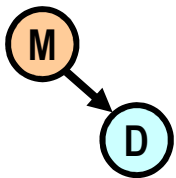




# XForms

---

- W3C Standard for web-forms processing
- Allows web-forms to load and save complex XML data with many repeating sub-structures
- Works very well with REST-type interfaces
- Bundled with XML databases (eXist and MarkLogic)
- Large library of sample applications



# Sample XForms

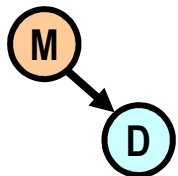
systems user: dan [login](#) [logout](#)

[Home](#) > [Glossary Manager](#)

## Update Glossary Term

Glossary Editor - 0.08 - File: /exist/rest/db/crossflo/apps/glossary/data/8.xml

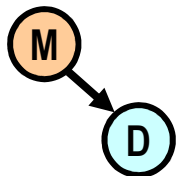
**Glossary Term**  
**Term Name:**  \*  
**Acronym:**   
**Synonym Set ID:**   
**Definition:**  \*  
**Select Categories**  
**See Also (non-synonym):**    
  
**Broader Term:**   
**Definition Sources**  
**Definition Source:**



# Requirements Editor

Code  
Table  
Selection  
Lists

Repeating  
Elements



## Edit Requirement

ID: 16

Name:  \* ⓘ

Description:  ⓘ

Parent ID:  ⓘ

Project ID:  ⓘ

Priority Code:  ⓘ

Status:  ⓘ

Testability:  ⓘ

Notes:

### Classifiers:

Classifier:  ⓘ

### Tags:

Tag:

# Page Components

The screenshot shows a web page from the Minnesota Historical Society. The page is titled "View Requirement 16" and contains a list of requirement details. The page is annotated with labels and arrows indicating its components:

- Header:** The top section of the page, containing the Minnesota Historical Society logo and navigation links.
- Breadcrumb:** The navigation path "MHS Home Apps Requirements Manager > List Requirements" located below the header.
- Content:** The main body of the page, containing the requirement details for "View Requirement 16".
- Footer:** The bottom section of the page, containing the text "Last Updated: January 2010".
- Edit Controls:** The section at the bottom left of the content area, containing links "Edit Item" and "Delete Item".

The page is 950 pixels wide, as indicated by the blue double-headed arrow at the bottom.

**Requirement Details:**

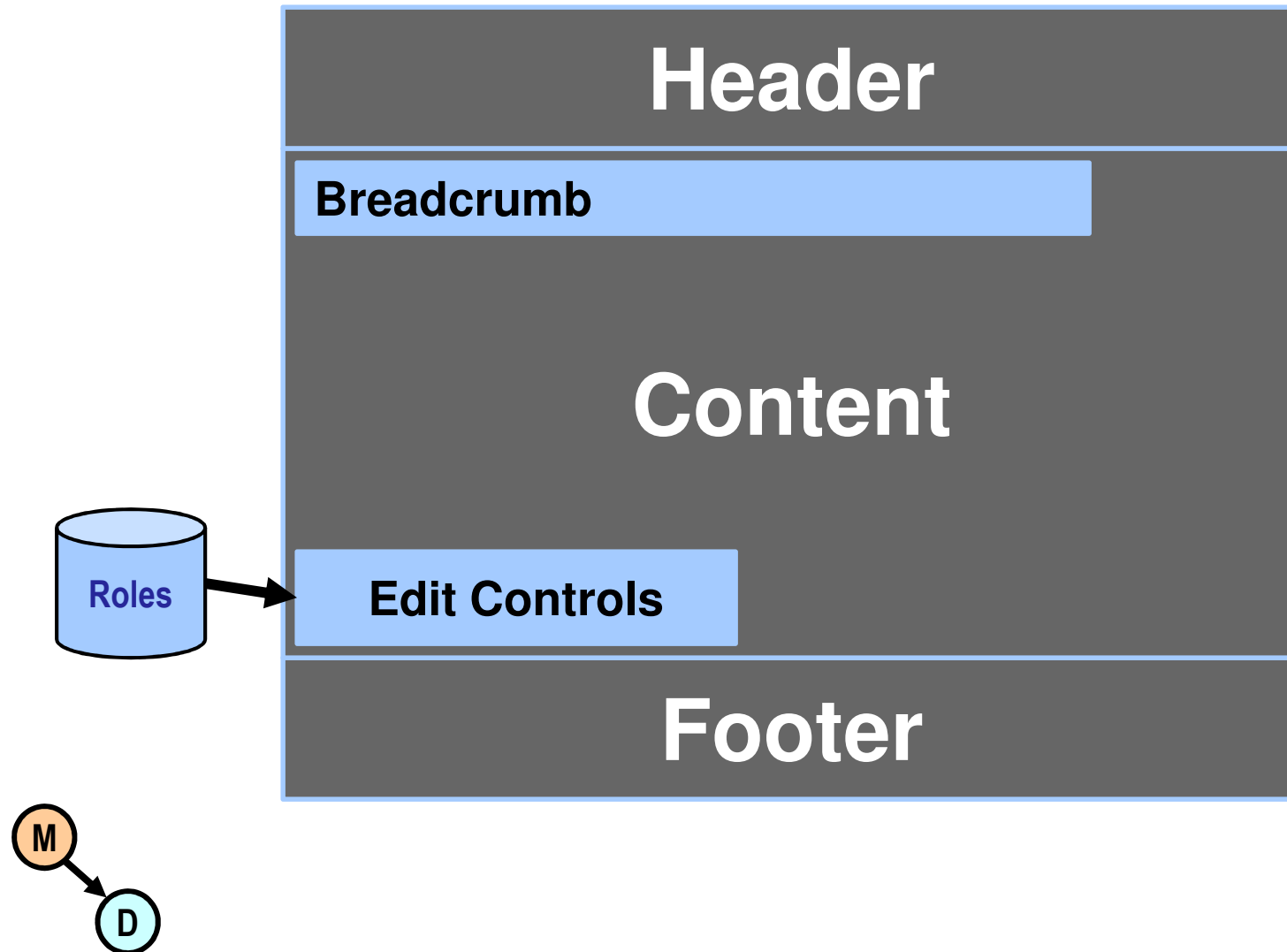
- ID: 16
- Parent ID: 3
- Name Short Name: Import XML Data
- Description: The system must be able to import well-formed XML data.
- Name Short Name: Import XML Data
- Priority: critical
- Completion Status: approved
- Approved By: robert.horton@mnhs.org
- Approval Date: 2009-09-04
- Notes:
- Classifiers: architecturally-significant
- Tags:
- Testable: unknown
- XML: </db/cust/mhs/apps/requirements/data/16.xml>

[Edit Item](#) [Delete Item](#)

Last Updated: January 2010

# Page Assembler Function

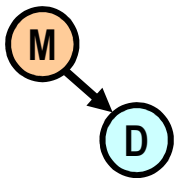
---



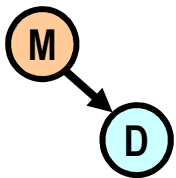
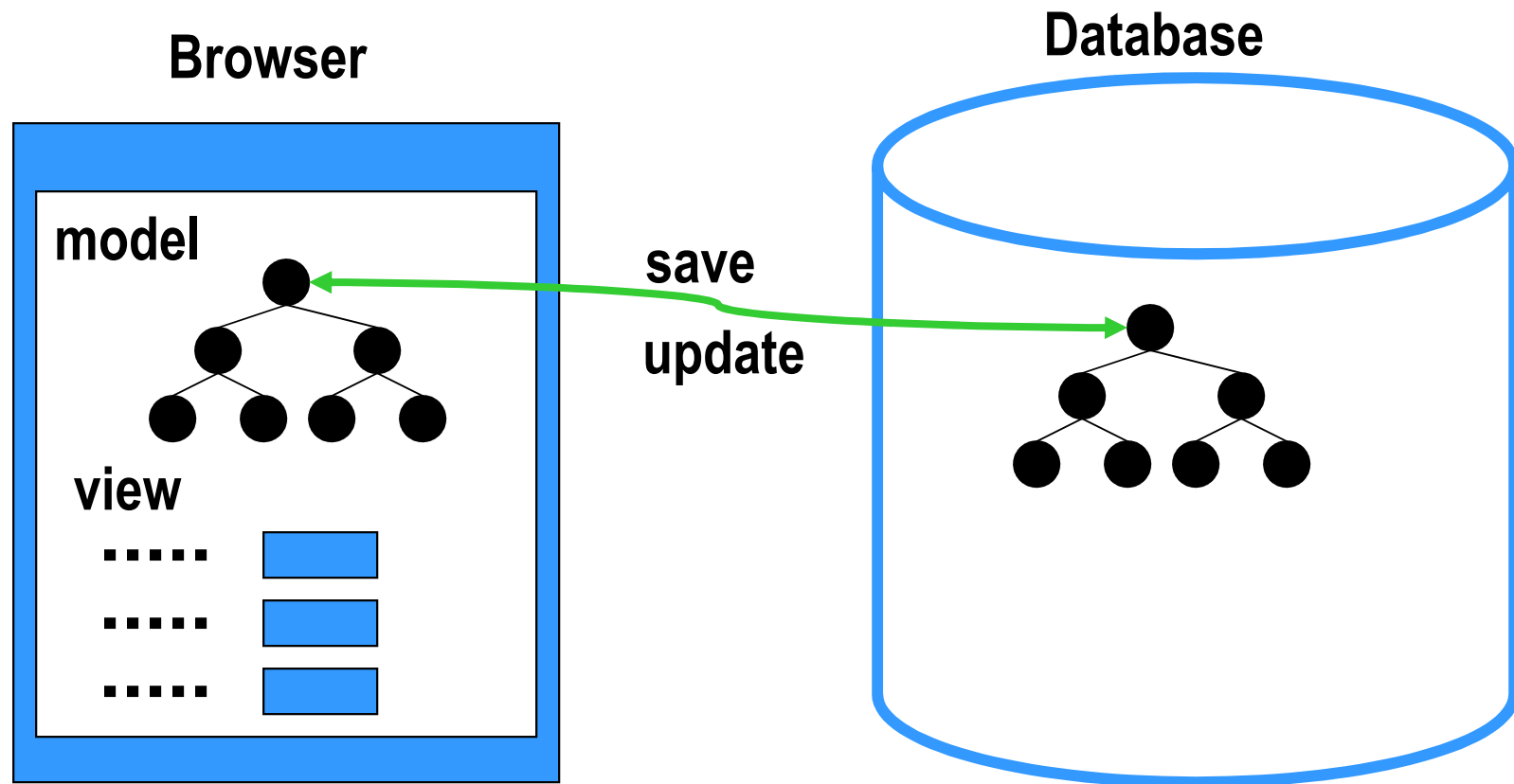
# Style Module

---

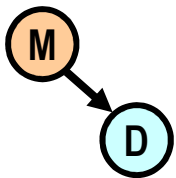
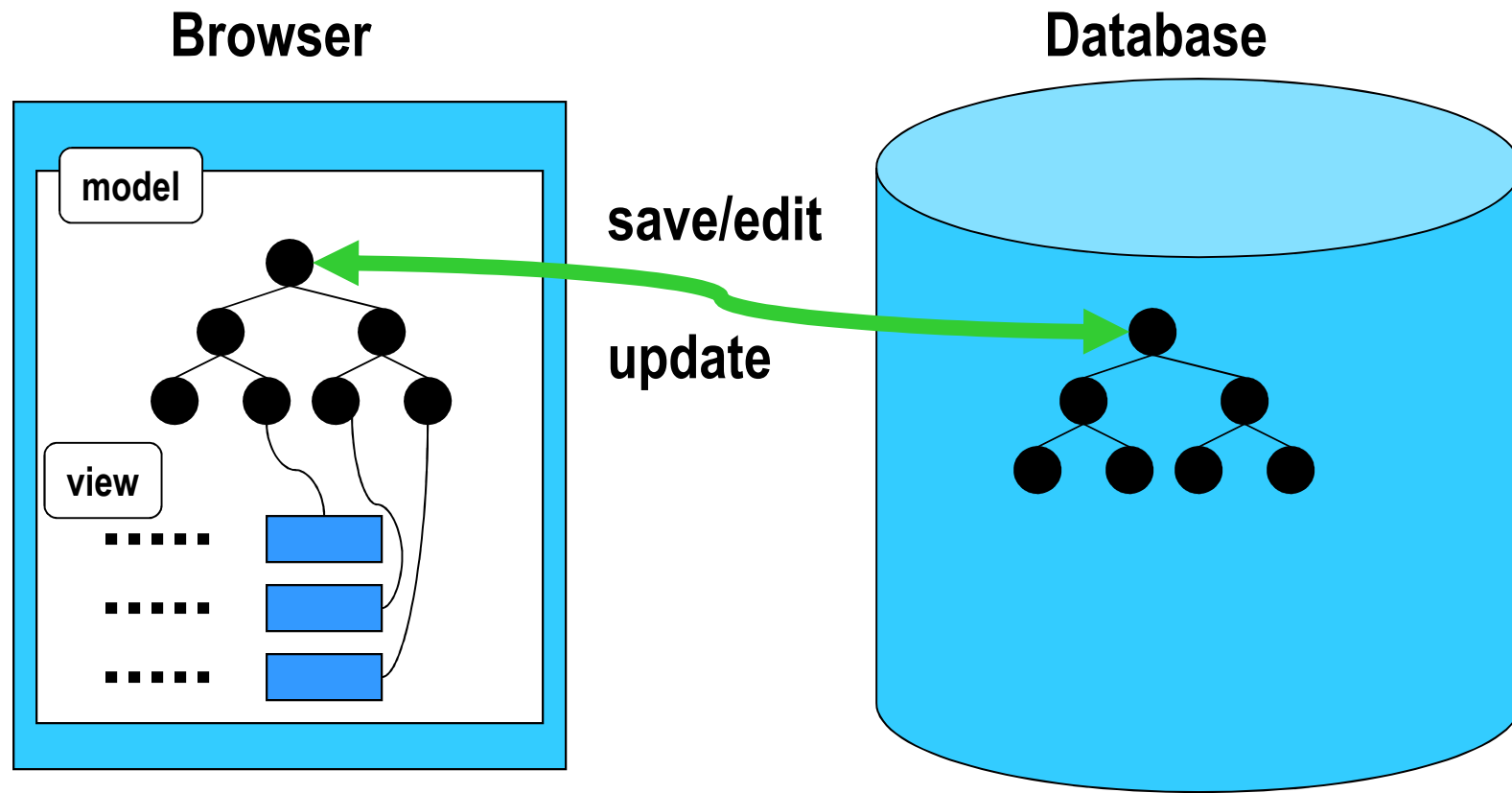
- Each non-content region of the page is generated by a server-side XQuery function
- Users can change a single function and the entire site will be updated
- Functions are dynamic and can take into account the page function



# XML Stored in XForms Model

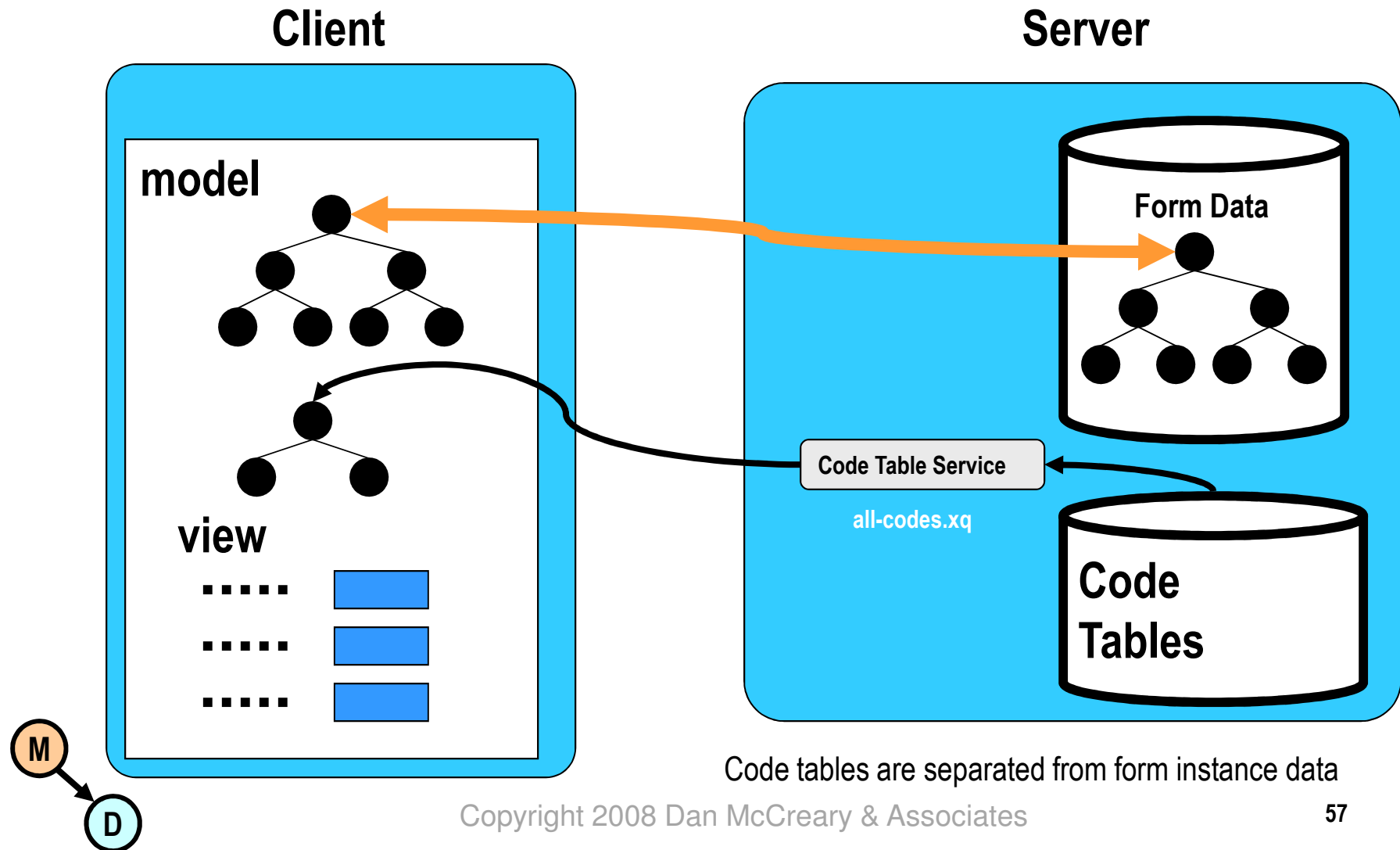


# XRX Core Process

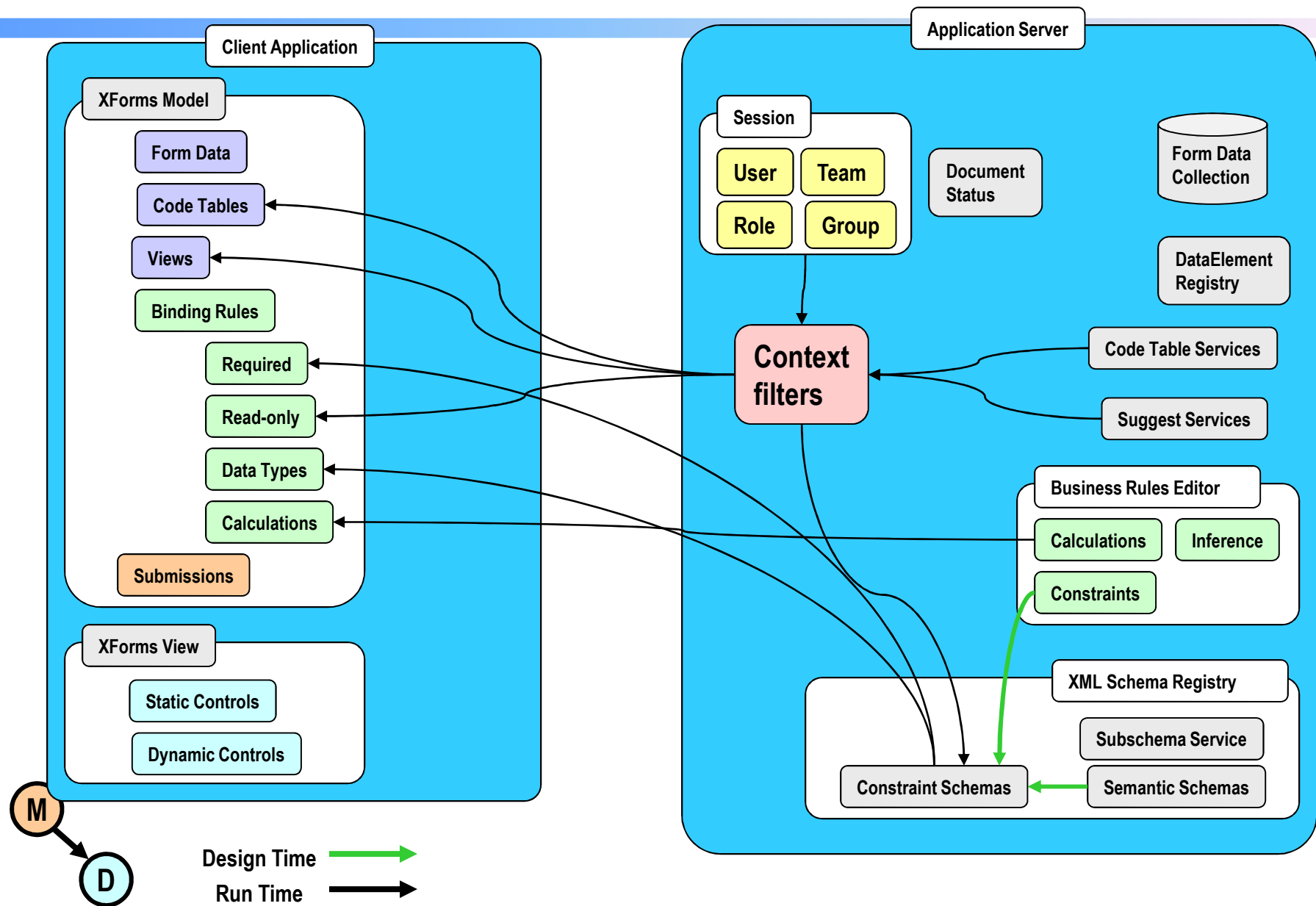




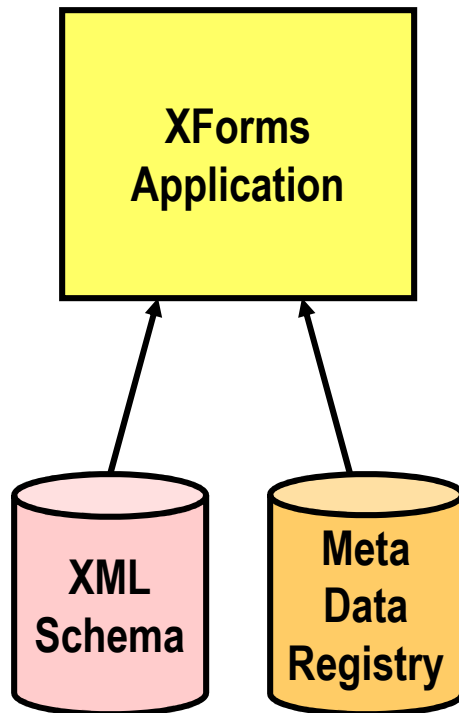
# Code Table Services



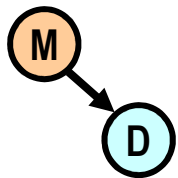
# XRX Dynamic Forms Generation



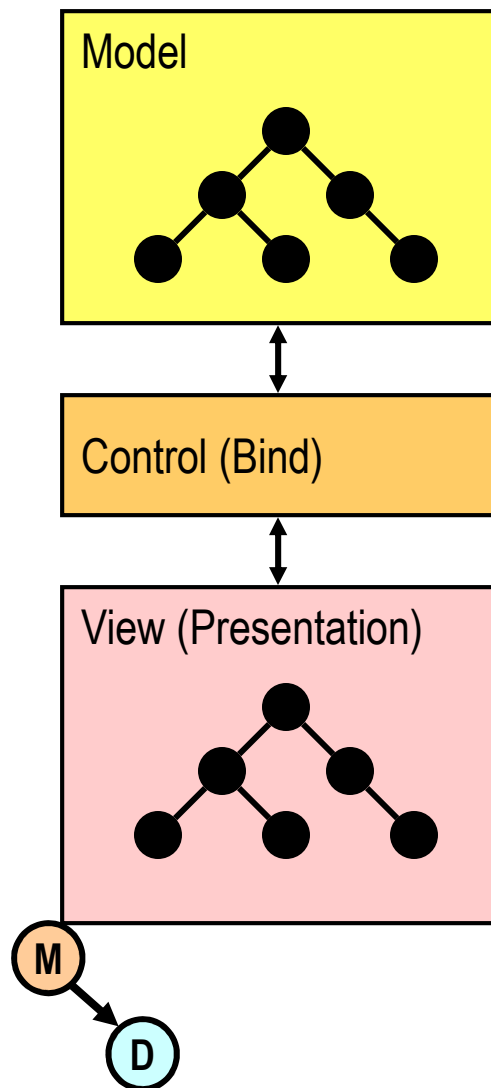
# Model Driven



- XForms enables the developer to reuse business rules encapsulated in XML Schemas (xsd) and XML Transforms (xslt)
- XForms reduces duplication and ensures that a change in the underlying business logic does not require rewriting in another language

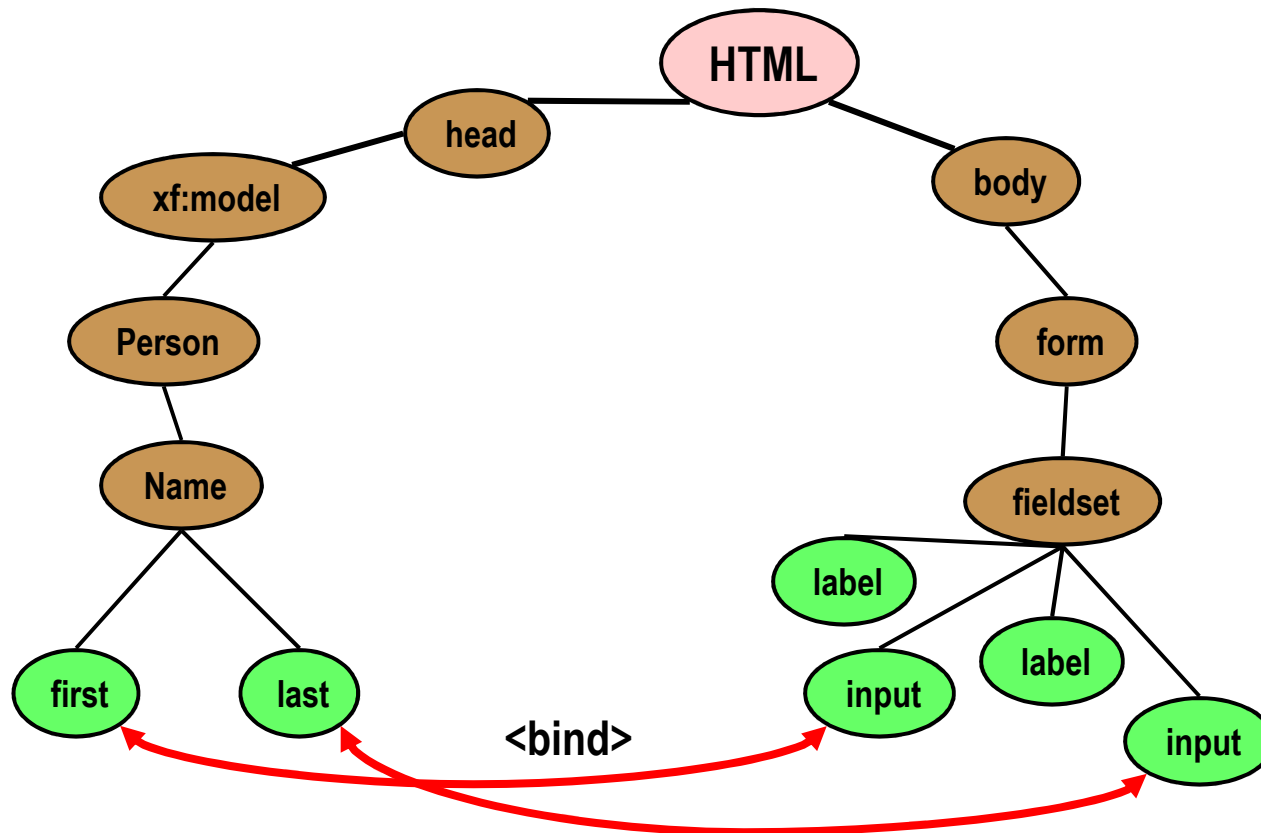


# View and Model are Trees

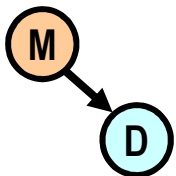


- The view is a tree of a presentation data element
- Models are comprised of one or more trees
- XForms supplies the control layer that moves data elements to and from the model
- Users don't have to worry about moving things to and from the screen

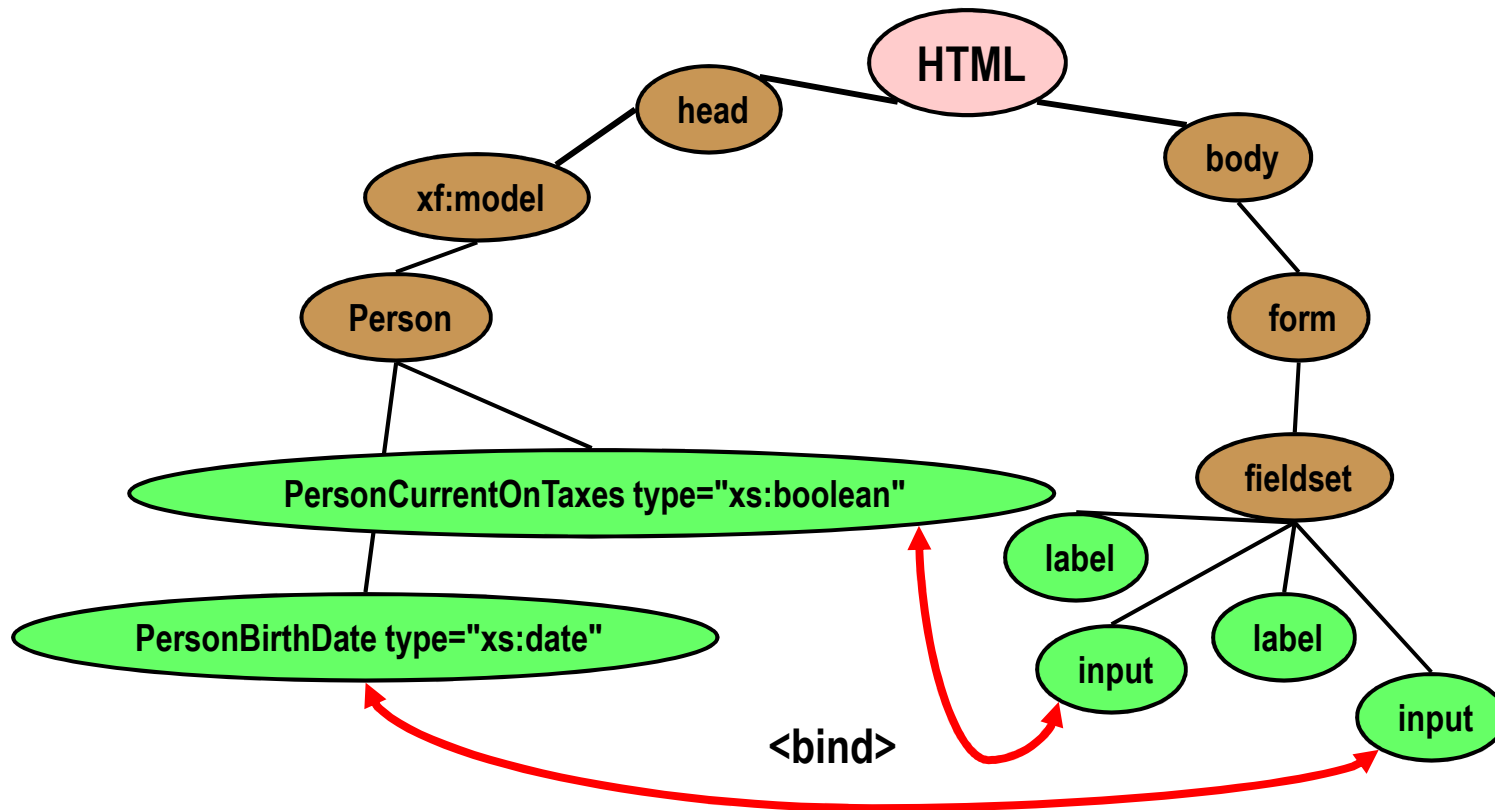
# Models and View Are Linked with "Bind"



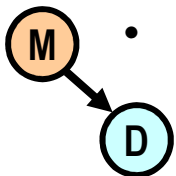
- Both the model and the views are trees of data elements



# Just “Do The Right Thing”



- Data types from the model just do the right thing
- Boolean variables become checkboxes
- Dates have date selectors



# Example of Automatic UI Generation

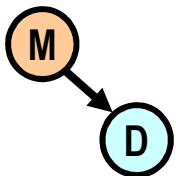
Boolean:

☐

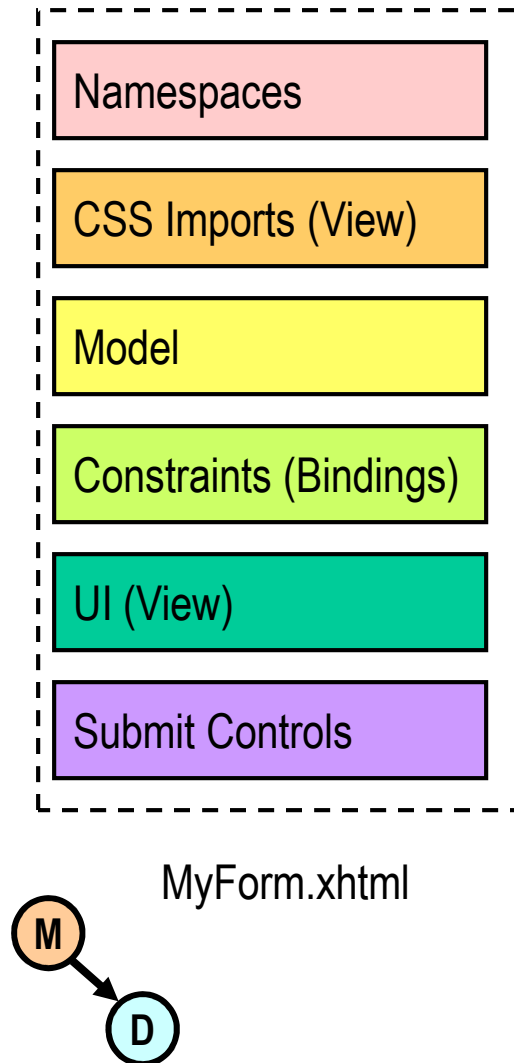
Enter a Date:



- All true/false data types (xs:boolean) automatically become a checkbox
- All dates (xs:date) have a date selector to the right of the date field
- All codes can be selected from lists



# Structure of a XForms File



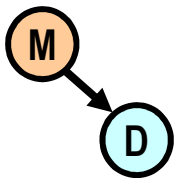
- XForms tags are just XML tags imbedded in a standard XHTML file with a different namespace
- Most HTML form tags are exactly the same but some attributes have been promoted to be full elements



# REST

---

- REpresentation State Transfer
- Create applications based on well designed URLs
- Take advantage of web caching
- Migrate toward Resource-Oriented Computing (ROC)
- REST evangelists: RESTifarians

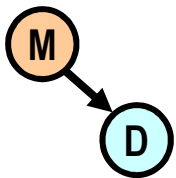


# Five RESTFull Friends

---

1. In-resident memory cache in your browser
2. Your local hard drive cache
3. Your local enterprise cache
4. The cache on the web server farm
5. The cache on the database

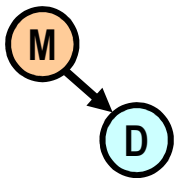
Please make sure to check with your RESTfull friends **BEFORE** you bother the database.



# Shallow REST vs. Deep REST

---

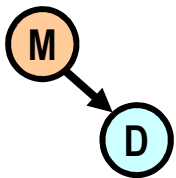
- You can start taking advantage of ReST by just doing well thought-out URL design
- To take advantage of deep ReST you must consider the subtleties of the HTTP protocol
  - GET vs POST vs PUT
  - DELETE



# Benefits of REST

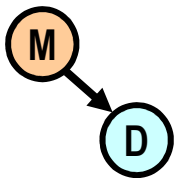
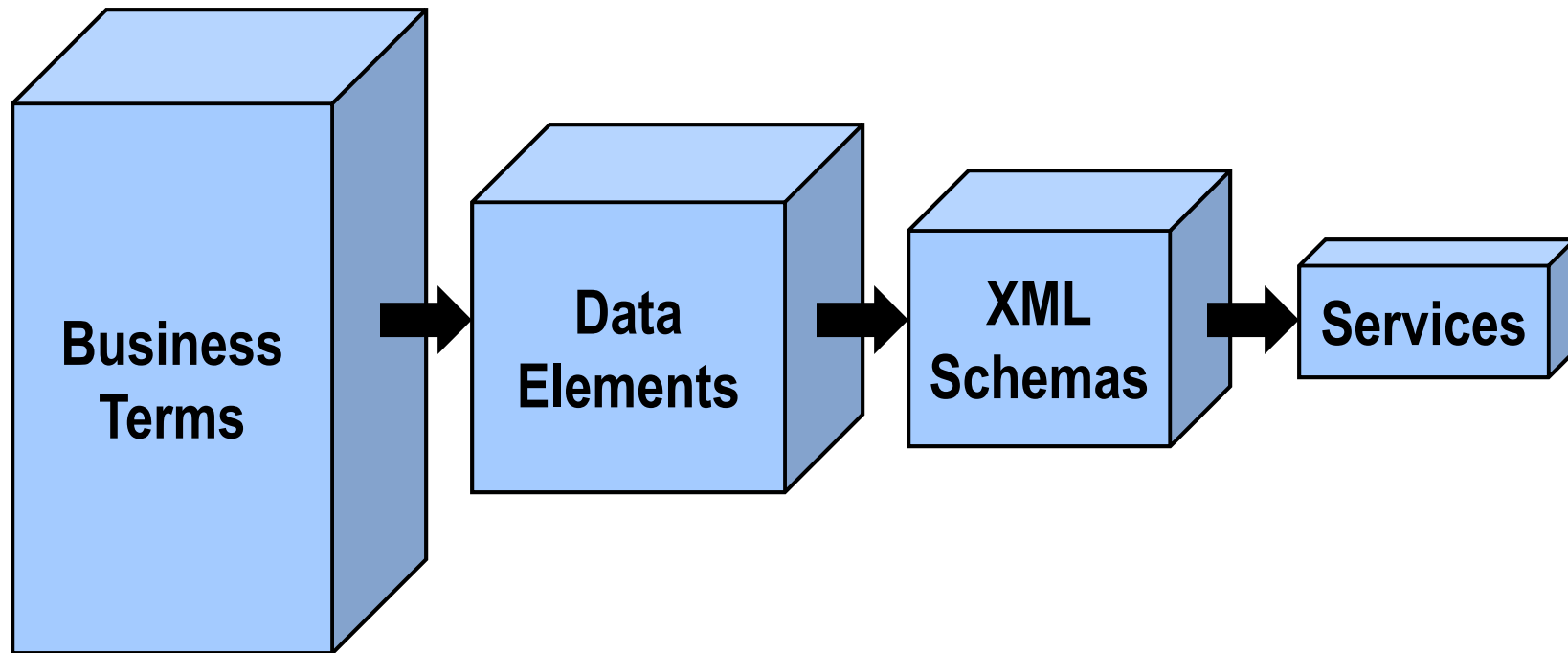
---

- Provides improved response time
- Reduced server load
- Improves server scalability
- Requires less client-side software
- Depends less on vendor dependencies
- Promotes discovery
- Provides better long-term compatibility
- Better and evolvability

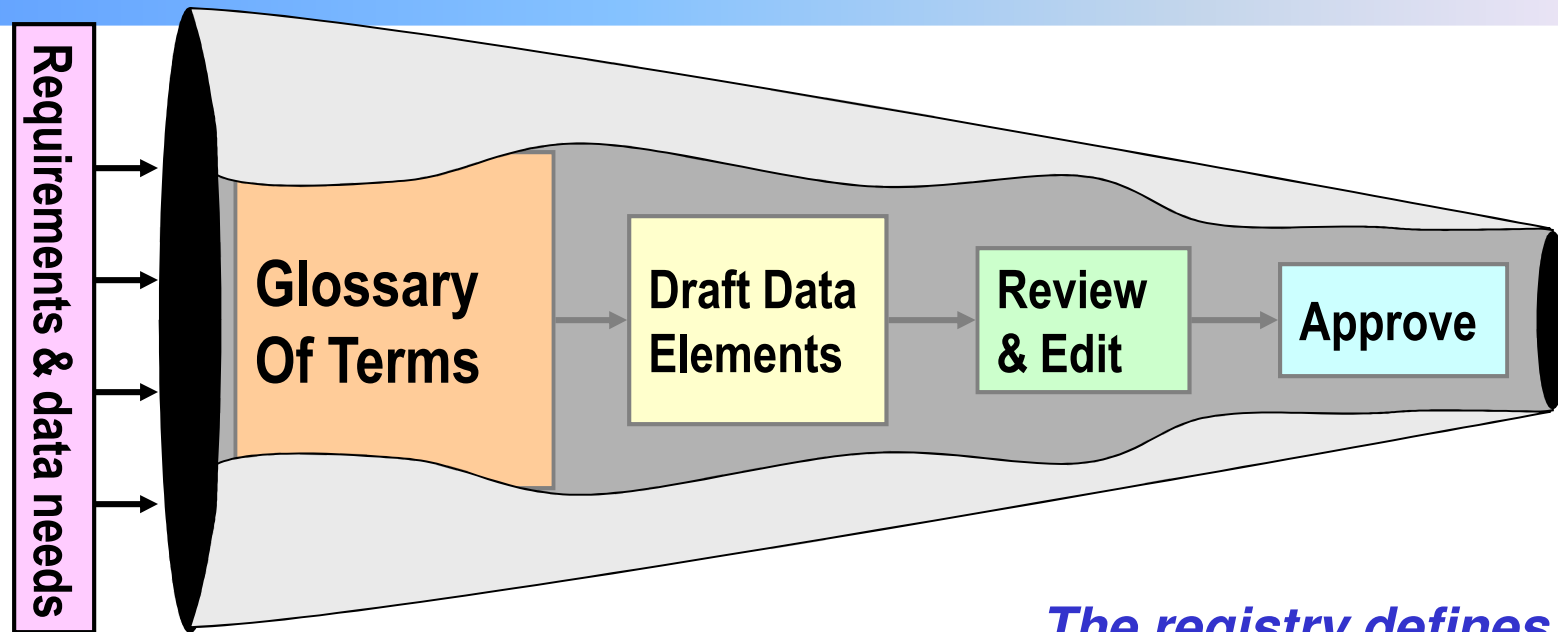


# Terms to Services

---



# Metadata Registry Workflow Funnel



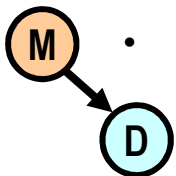
## Create the Registry

- Define your glossary and data elements
- Review & make changes
- Approve & publish by stakeholders

## Use the Registry

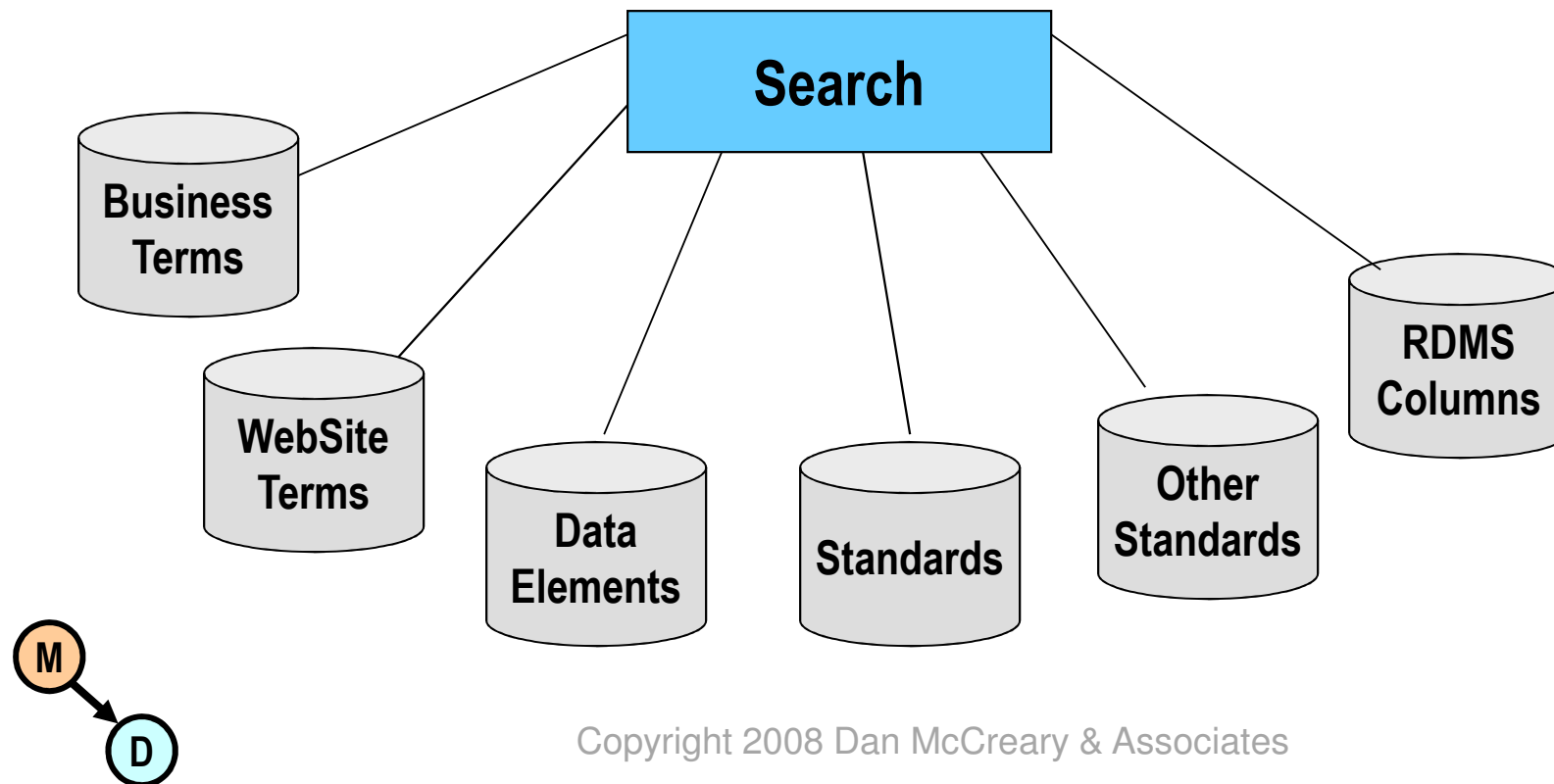
- Generate data schemas (XML) by selecting and organizing data elements
- Add new items to the registry as needs change

*The registry defines the data we exchange and keeps our need for code changes to a minimum*

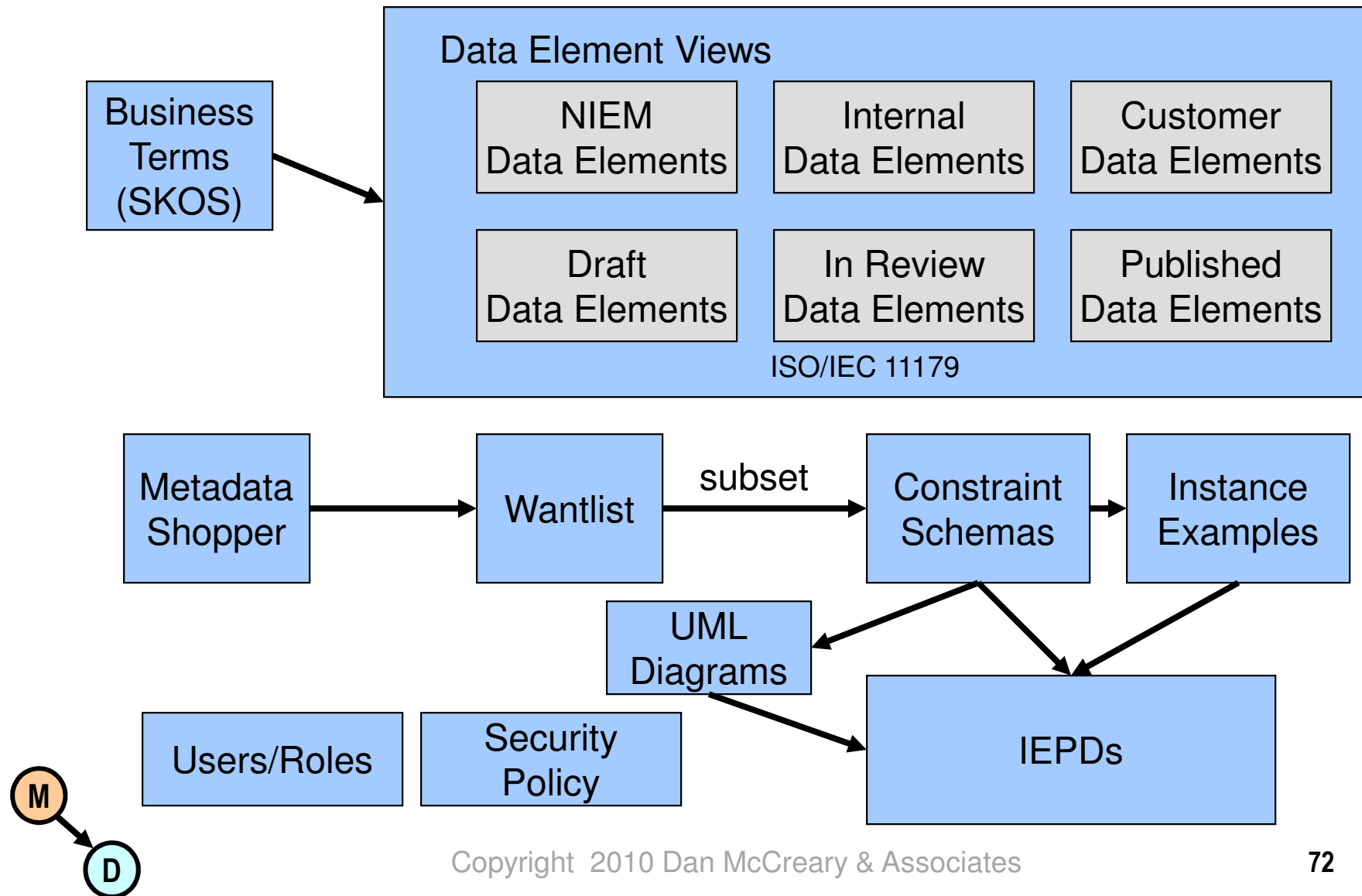


# Federated Search

- **Federation:** When many different sources can return search results from a single search





# Sample Data Flows





# Application Modularity

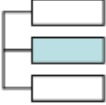



















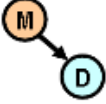




**Metadata Manager**

Search:

---

## Metadata Registry Demo Applications

[Icons](#) [List](#) [Details](#)

 Data Element Manager	 FAQ Manager	 Glossary Manager	 Namespace Manager	 OWL File Manager
 Pattern Registry	 Product Manager	 Project Manager	 Report Manager	 Requirements Manager
 Rule Manager	 Sales Pipeline Manager	 XML Schema Manager	 Global Search	 Metadata Shopper
 Site Map	 Services	 Subscription Manager	 Task Manager	 Taxonomy Manager
 Template	 Training	 Use Case Manager	 User Manager	 Wantlist Manager

Copyright 2009 Dan McCreary and Associates.  
All rights reserved.

# Financial Institution

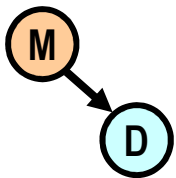
Welcome, mdr Site Map | Help | Feedback

## Metadata Management System

### Main Menu

 Glossaries	 Taxonomies	 Data Elements
 RDBMS Domains	 Vendors	 Meta Shopper
 Wantlists	 XML Schemas	 Services
 Capabilities	 Standards	 Data Stores
 Search	 Reports	 Rules
 Tasks	 Admin Tools	 Labs


    

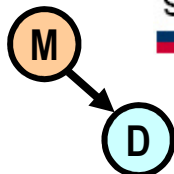


# Federal Integrator


[Icons](#) [List](#) [Details](#) App Count=25 Per Row=5 [Static Main Menu](#) apps-base = /db/crossflo/apps

## Metadata Registry Main Menu

 Glossary Manager	 NIEM 2.0	 Data Element Manager	 Report Manager	 Global Search
 Metadata Shopper	 Wantlist Manager	 XML Schema Manager	 IEPD Manager	 Services
 Task Manager	 Site Map	 Namespace Manager	 FAQ Manager	 Use Case Manager
 Training	 NDEX	 Requirements Manager	 Template	 LEXS
 Subscription Manager	 Integration Product	 User Manager	 ESB Registry	 Rule Manager




















# Minnesota Historical Society

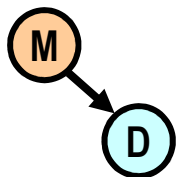
 Minnesota Historical Society

Icons [List](#) [Details](#) App Count=17 apps-base = /db/cust/mhs/apps

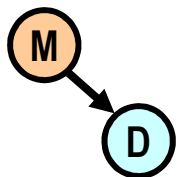
## MHS Applications

 <a href="#">Requirements Manager</a>	 <a href="#">Glossary Manager</a>	 <a href="#">FAQ Manager</a>	 <a href="#">Search Across Bill Collections</a>	 <a href="#">MN Bills</a>
 <a href="#">User Story Manager</a>	 <a href="#">Dublin Core</a>	 <a href="#">Search Minnesota Apps</a>	 <a href="#">Tree Editor Demo</a>	 <a href="#">User Manager</a>
 <a href="#">CA Bills</a>	 <a href="#">Stress Test Tool</a>	 <a href="#">Check Sum</a>	 <a href="#">IL Bills</a>	 <a href="#">Uploader</a>
 <a href="#">Index Advisor</a>	 <a href="#">Template</a>			

Last Updated: January 2010



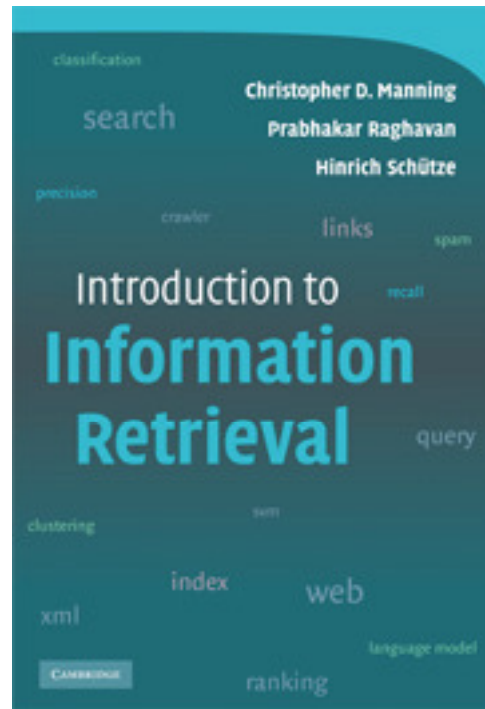
# Metadata Shopping Tools

A screenshot of a web application titled "NIEM Schema Search". It features a search bar with a dropdown menu set to "Property" and a text input field containing "first name". A "Search" button is to the right of the input field. Below the search bar is a link that says "Show Advanced" with a small downward arrow. Underneath is a section titled "Add Selected to WantList" containing a "Placement:" label and three radio buttons: "type" (which is selected), "independent component", and "Add" button. At the bottom, there is a list of three items, each with a checkbox and a text label:   
- ☐ [j:PersonNameInitialsText \(u:TextType\)](#)  
- ☐ [u:PersonGivenName \(u:PersonNameTextType\)](#)  
- ☐ [u:personNameInitialIndicator \(xsd:boolean\)](#)

- You don't need to know about 100,000 SKUs to purchase 10 items from a grocery store
- Sub-schema generation tools give you exactly what you need and nothing more

See <http://niem.gtri.gatech.edu/iepd-sstg/SSGT-SearchSubmit.do>

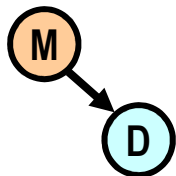
# Information Retrieval Textbook



## Introduction to Information Retrieval

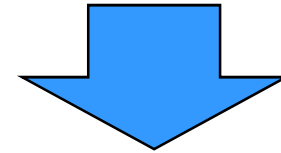
by Christopher D. Manning,  
Prabhakar Raghavan and  
Hinrich Schütze

Cambridge University  
Press, 2008



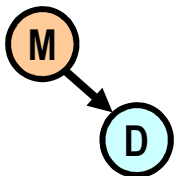
<http://nlp.stanford.edu/IR-book/information-retrieval-book.html>

# Table 10.1



	RDB search	unstructured retrieval	structured retrieval
objects	records	unstructured documents	trees with text at leaves
model	relational model	vector space & others	?
main data structure	table	inverted index	?
queries	SQL	free text queries	?

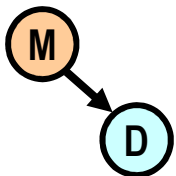
XML - Table 10.1 and structured information retrieval. SQLRDB (relational database) search, unstructured information retrieval



# Table 10.1 - Revised

	RDB search	unstructured retrieval	structured retrieval
objects	records	unstructured documents	trees with text at leaves
model	relational model	vector space & others	<b>XML hierarchy</b>
main data structure	table	inverted index	<b>trees with node-ids for document ids</b>
queries	SQL	free text queries	<b>XQuery fulltext</b>

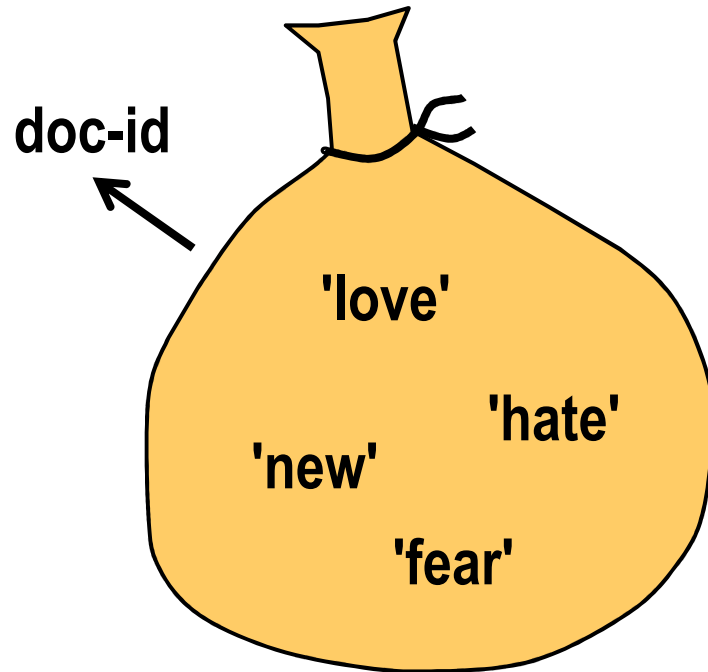
XML - Table 10.1 and structured information retrieval. SQLRDB (relational database) search, unstructured information retrieval



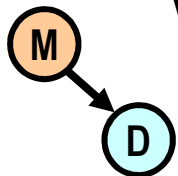


# Two Models

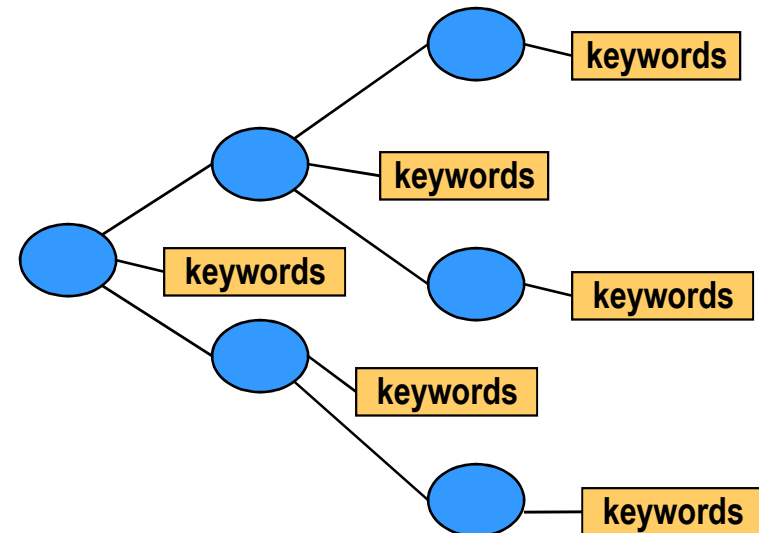
## "Bag of Words"



- All keywords in a single container
- Only count frequencies are stored with each word

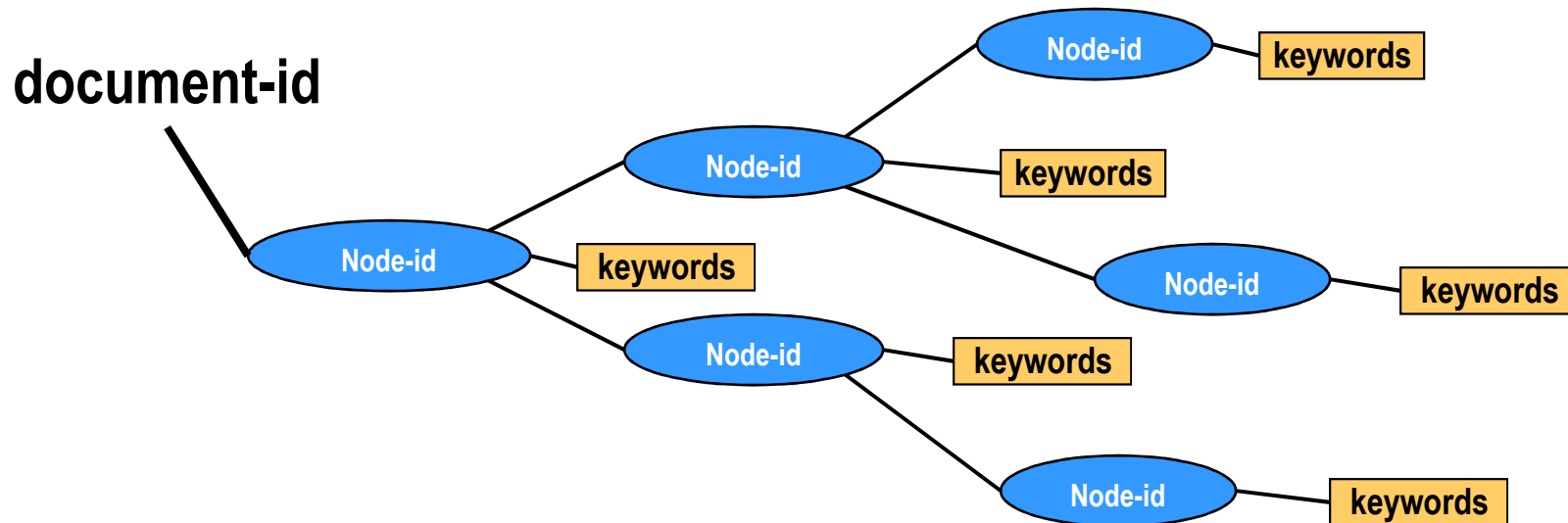


## "Retained Structure"

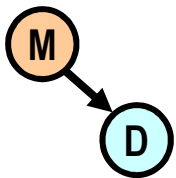


- Keywords associated with each sub-document component

# Keywords and Node IDs



- Keywords in the reverse index are now associated with the **node-id** in every document



# Search is a REST Service

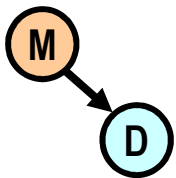
- Every search form is a “wrapper” of a REST web service
- You can call the web service from any browser or any other web service
- Results can be either HTML (for humans) or XML for remote systems

Example:

`http://mdr.example.com/search/search.xq?q=birth&output=xml`

The diagram illustrates the components of the URL `http://mdr.example.com/search/search.xq?q=birth&output=xml`. Blue brackets and arrows identify the following parts:

- Search Services Collection**: A bracket under `search/search.xq`.
- Search Query**: An arrow pointing to the `q=` part of `q=birth`.
- Search Parameter q=Query**: An arrow pointing to the `birth` value of `q=birth`.
- Output Format**: A bracket under `output=xml`.



# Global Search

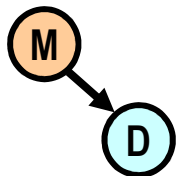


[MHS Home](#) [Apps](#) [Search Minnesota Apps](#)

## Search Results

**Search results for:** search

1. [Glossary: Federated Search](#)
2. [Glossary: Search Precision](#)
3. [Glossary: Boolean Search](#)
4. [Fags: What is precision in an information retrieval system?](#)
5. [Requirements: Source Customizable Search Rank](#)
6. [Glossary: Search Recall](#)
7. [Requirements: Search by Date Range](#)
8. [Fags: What is recall in an information retrieval system?](#)
9. [Glossary: Ranked Search](#)
10. [Glossary: XQuery Search Template](#)
11. [Requirements: Indexes](#)
12. [Requirements: Search multiple documents](#)
13. [Requirements: Search with XML Element](#)
14. [Glossary: Zone Ranking](#)
15. [Requirements: Searches must cross XML tag boundaries](#)
16. [Requirements: Search Score](#)
17. [Requirements: Index Imported Documents](#)
18. [Fags: How was this web site constructed?](#)
19. [Requirements: Support CSS](#)



# Complex Search

Metadata Registry > Glossary Manager

## Advanced Business Terms Search

This will search all Terms with a Term Name that meets the following criteria:

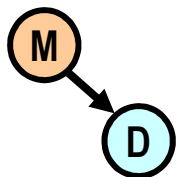
Term Name:

☒ Exact Match  
☐ Starts with  
☐ Anywhere in Term Name

Excluded Terms Names with (Filter):

[Return to Simple Search](#)

- Exact Match
- Starts with
- Anywhere
- Filters
  - Removed results



# Internal vs. External Terms

## Internal Data Standards

## External Data Standards

[Metadata Registry Home](#) > [Federated Search Home](#)

### Federated Search Results for "Risk":

#### Internal references:

**Glossary Term** [Credit for Sale](#) CFS is an adjustment system on Appleton SYSA. Contains the Credit for Sale Product Plan Category table.

**Glossary Term** [Commission](#) An amount of money paid to compensate a sales producer.

**Glossary Term** [Target Premium](#) The minimum premium to issue a UL or VUL contract. Note that this definition is significantly different than the Corporate Glossary definition which is as follows: The premium that the contract owner pays in the first five contract years. It is also the target premium on which the FR's full first-year commission is based. Premiums in excess of the target receive a lower commission rate. See the Corporate Dictionary for the full definition.

**Glossary Term** [Subsequent Selling Transaction](#) A financial transaction against an insurance policy that occurs subsequent to the original issue date.

**Glossary Term** [Guaranteed Lifetime Withdrawal Benefit](#) The GLWB is the fastest growing guaranteed living benefit in the market today.

**Glossary Term** [Future Purchase Option](#) A supplemental benefit that is provided by some disability income contracts and LTC; it gives the insured the right to increase the contract's benefit amount in accordance with increases in the insured's earnings usually without providing evidence of insurability.

**Glossary Term** [Protection Product](#) Life and health products, including but not

#### External references:

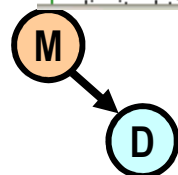
**ACORD** [RiskToleranceCode](#) This field is used to determine suitability of the investment options made in a sale based upon a customer's risk tolerance. Each fund has a risk tolerance and it is based upon Lipper classifications.

**ACORD** [NetAmtAtRisk](#) The net amount of risk for this coverage.

**ACORD** [TotalRiskAmt](#) The amount of insurance underwritten for the primary insured. This includes the basic face amount plus the amounts of any riders and benefits being applied for.

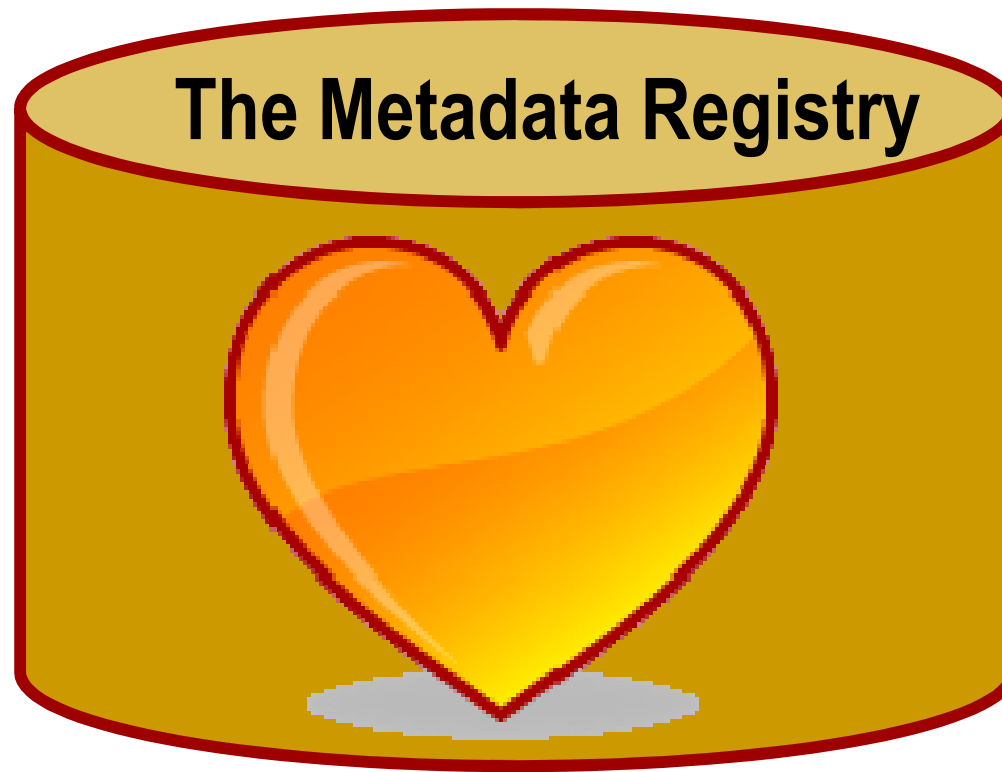
**ACORD** [ReinsuranceInd](#) Indicates that the risk of this policy is shared in whole or in part by another carrier. TRUE if risk is shared, FALSE if not.

**InsWeb** [Amount At Risk](#) The difference between the face amount of a Whole Life Insurance contract and the cash value which it has built up. The net amount at risk declines throughout the life of the contract, while the policy reserve increases along with the cash value. It is the amount the insurer would have to draw from its own funds rather than the policy reserve were the contract to become a death claim. (LI)

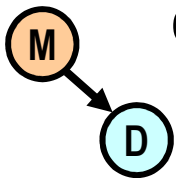


# The Heart of the Enterprise

---



A metadata registry is a central location in an organization where metadata definitions are stored and maintained in a controlled method.

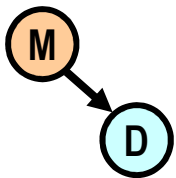


[http://en.wikipedia.org/wiki/Metadata\\_registry](http://en.wikipedia.org/wiki/Metadata_registry)

# Dan's Promise to Every BA

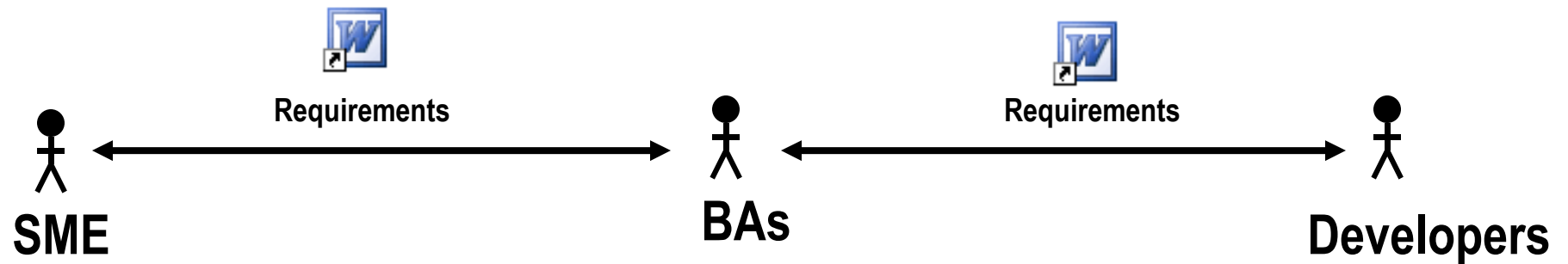
---

- If you are...
  - somewhat familiar with HTML and SQL
  - willing to "know your data"
  - willing to spend around 40 hours in training
  - able to use open source software
- Then...
  - You can build and maintain your own metadata registry



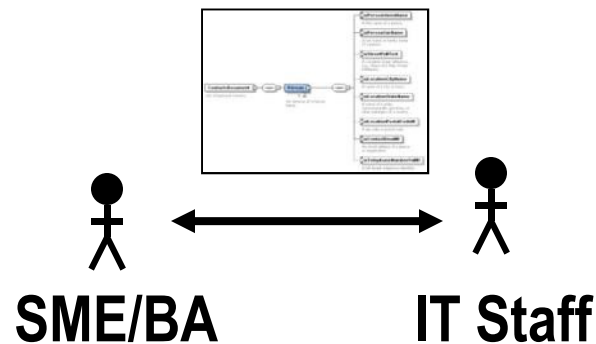


# Change Where the Line is Drawn

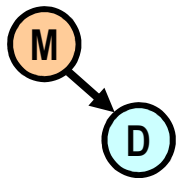


VS.

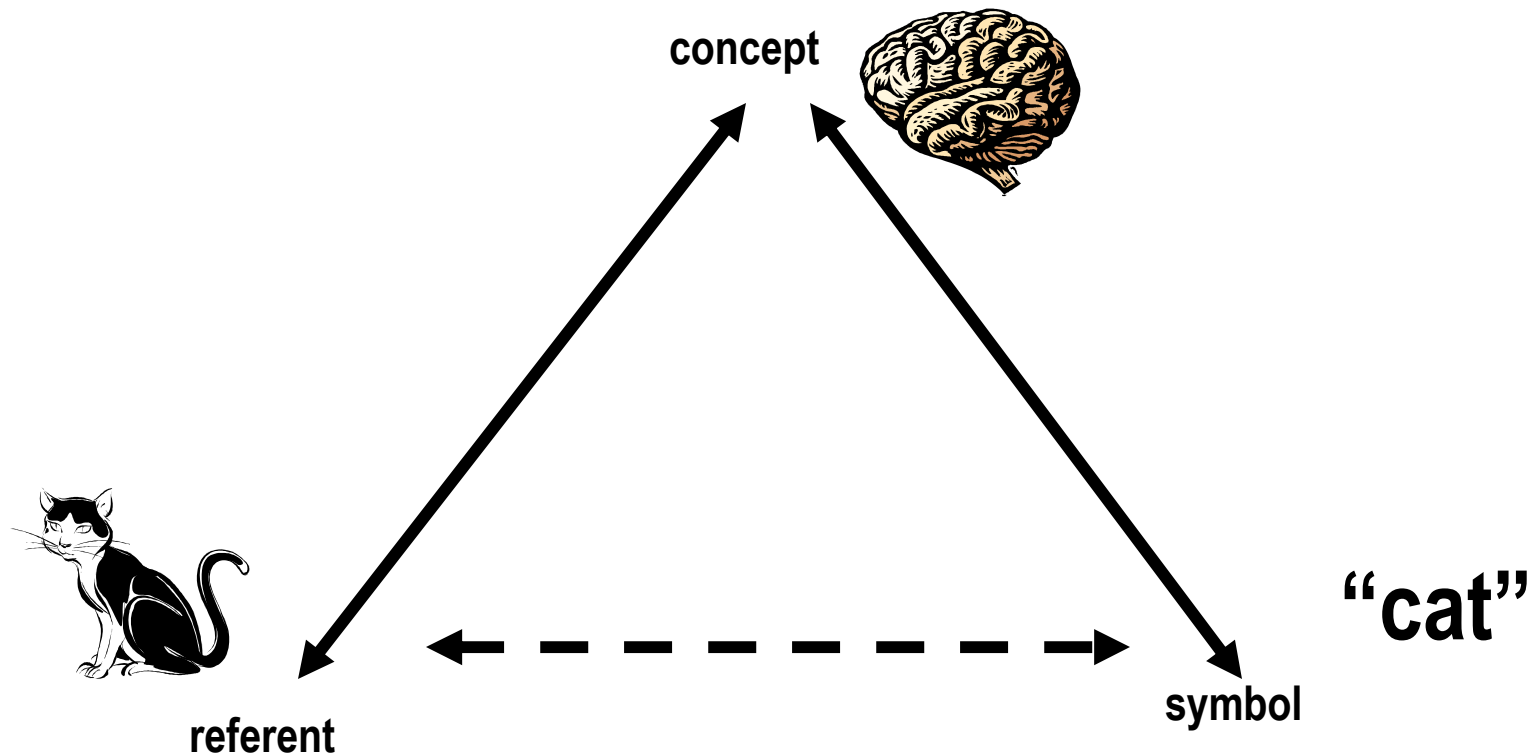
Graphical Requirements and Specifications



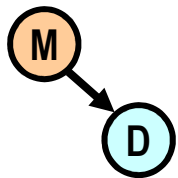
Shorten the “distance” between the business unit and the IT staff



# Semantic Triangle

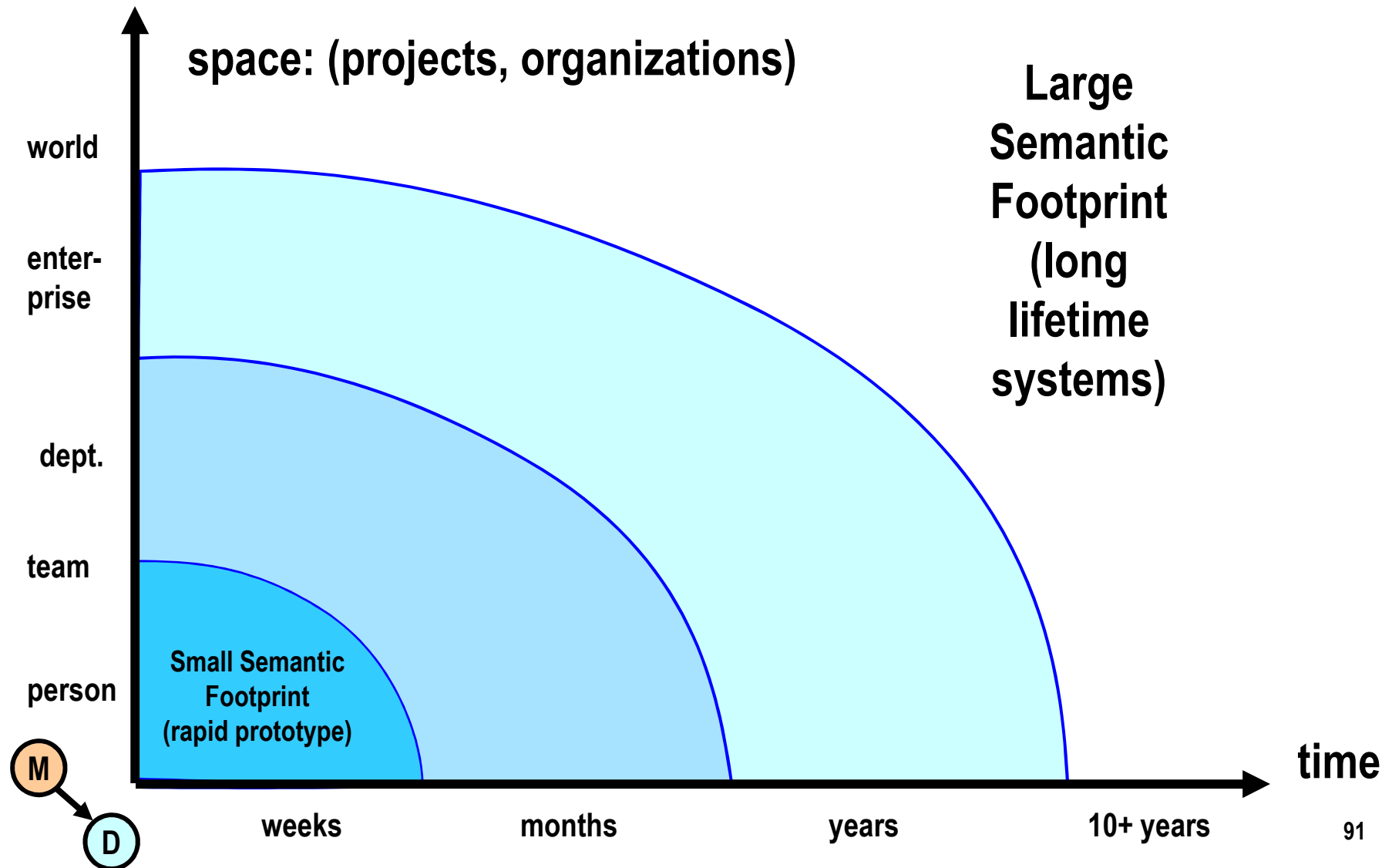


- Symbols can only link to referents through concepts
- You can not link directly from a symbol to a referent

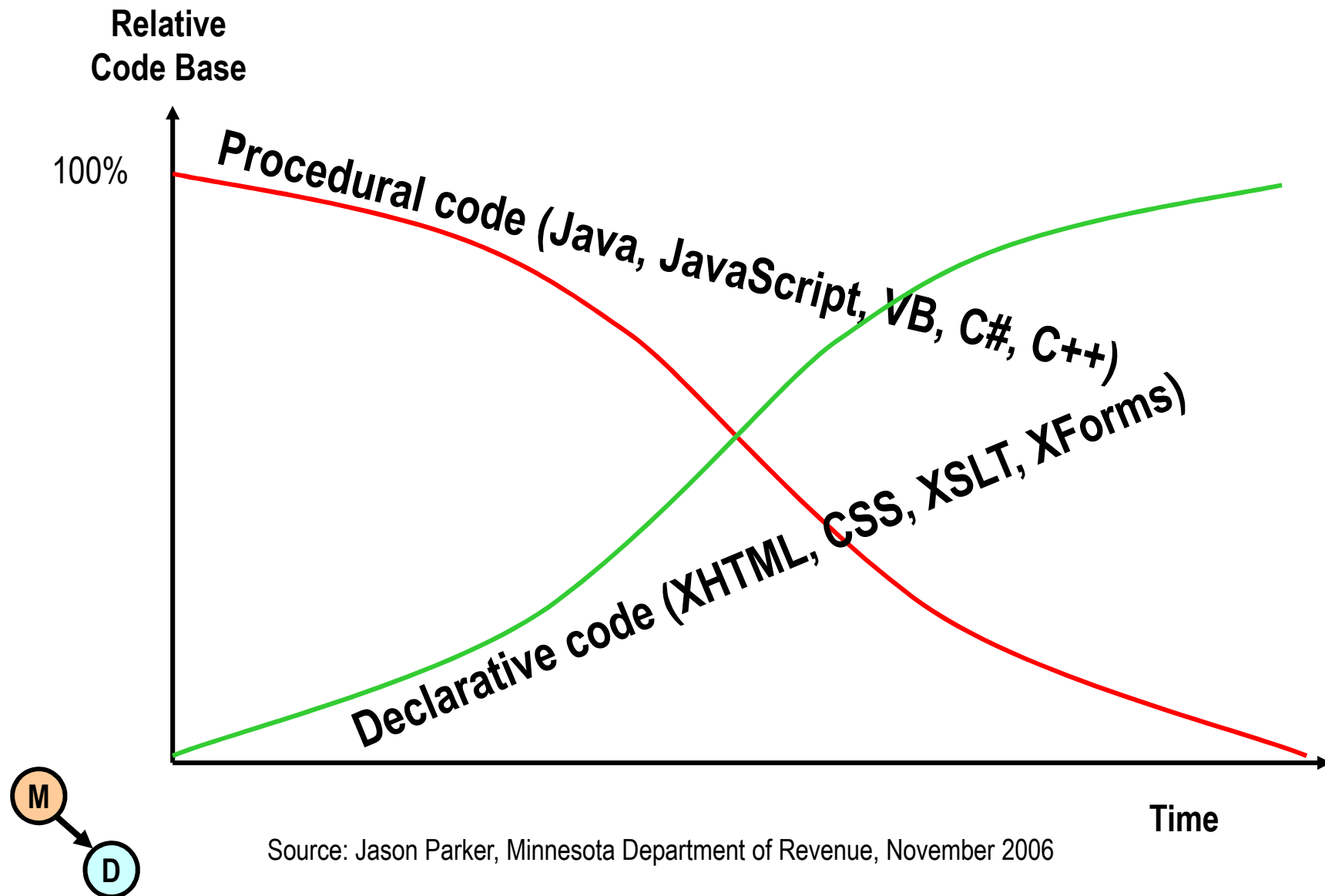


Wikipedia: Semiotic triangle

# Semantic Precision in Space and Time



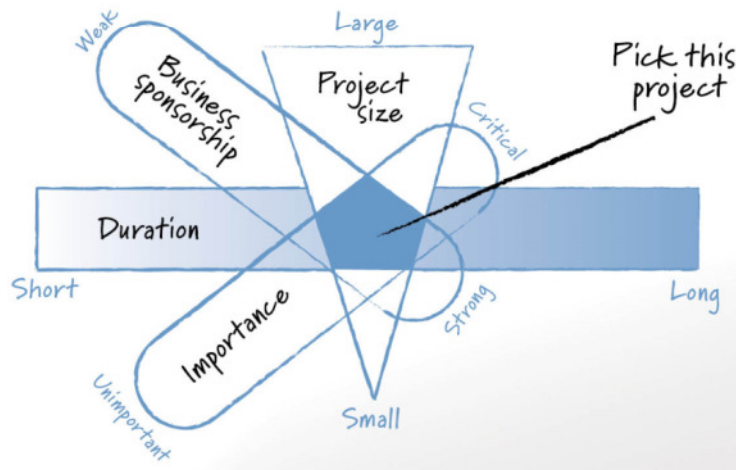
# Parker Projection



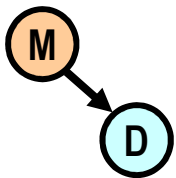
# Incoming!



# Selecting a Pilot Project



- The "Goldilocks Pilot Project Strategy"
- Not too big, not too small, just the right size
- Duration
- Sponsorship
- Importance
- Skills
- Mentorship

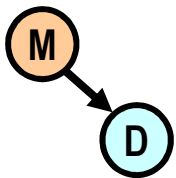


# Find A Community...

---



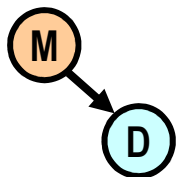
**eXist Meeting Prague March 12<sup>th</sup>, 2010**



# Challenges

---

- Minimal local talent with XQuery
- XForms performance issues for large forms (over 100 fields per form)
  - User smaller forms
- Role-based access control at the collection level

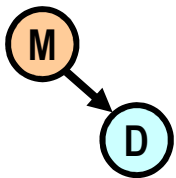




# Words of Caution

---

- Only use "latest stable" releases
  - Currently eXist 1.4
- Backup your system
- Put critical transactions in at least two places (transaction logs)
- Avoid long-running transactions
- Use locking to avoid missing updates

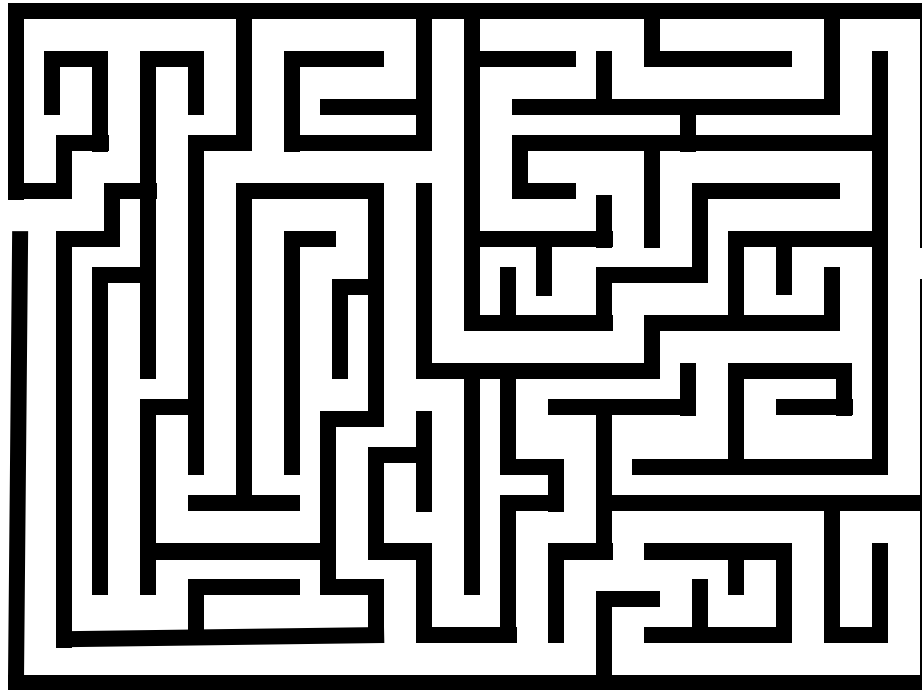


# Using the Wrong Architecture

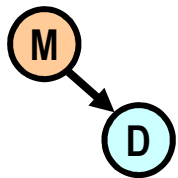
---



Start



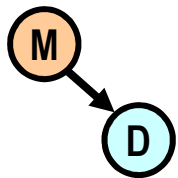
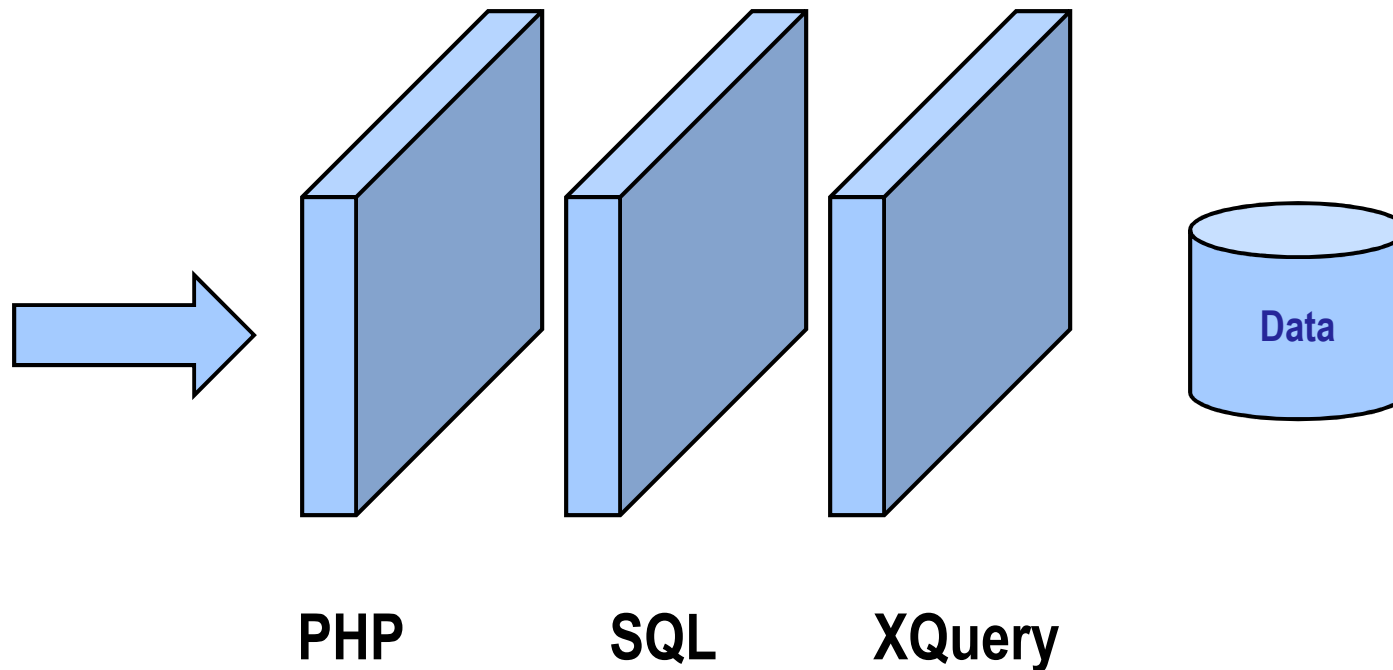
Finish



Credit: Isaac Homeland – MN Office of the Revisor

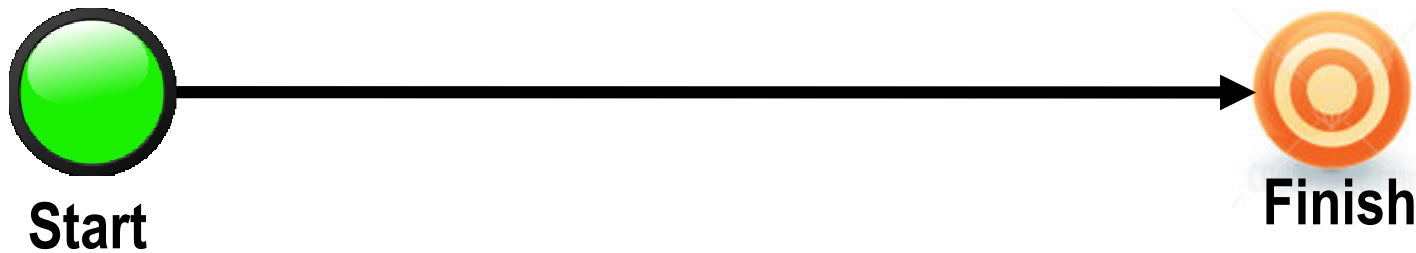
# The Problem with Layers...

*It's a nightmare trying to write XQuery within SQL within PHP...*

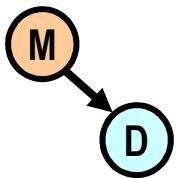


# Using the Right Architecture

---



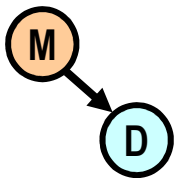
**Find ways to remove barriers to empowering the non programmers on your team.**



# Six "S"s of Metadata Registries

---

1. **S**emantics
2. **S**earch
3. **S**tandards
4. **S**ervices
5. **S**olutions that are Customized
6. **S**uper - BA

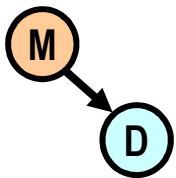


# If You Give a Kid a Hammer...

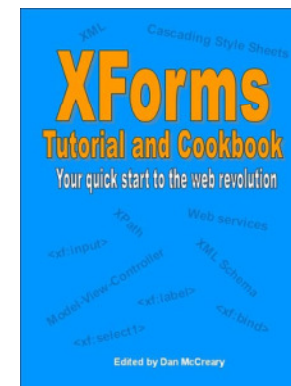
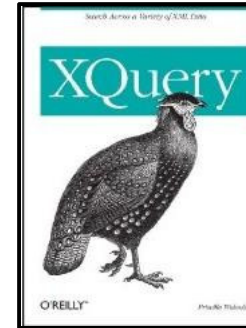
...the whole world becomes a nail



- People solve problems using familiar tools
- People develop specific *Cognitive Styles*\* based on training and experience
- What are we teaching the next generation of developers?



\* Source: Shoshana Zuboff: In the Age of the Smart Machine (1988)



A directed graph with two nodes, M and D. Node M is an orange circle with a black border, and node D is a light blue circle with a black border. A black arrow points from node M to node D.

103

# Questions?

---

Dan McCreary  
President  
Dan McCreary & Associates  
[dan@danmccreary.com](mailto:dan@danmccreary.com)  
(952) 931-9198

