Which Is Best?

DAMA Phoenix

September 12, 2013

Norman Daoust

Daoust Associates
## Introduction

- Daoust Associates, 2001
- Consultant, author, speaker
- Data modeling and database design
- Healthcare electronic data exchange using HL7 standard
- Requirements modeling
- Business analysis training
Introduction: Presentation Goals

- similarities and differences between entity relationship data modeling and XML modeling
- advantages and disadvantages of both entity relationship data models and XML models
- appropriate usages for data models and XML models
- criteria for determining which to create
Introduction: Presentation Outline

- Why should you care?
- Definitions
- Examples
- Comparison: XML model vs. entity relationship model
- Deriving XML models from an entity relationship model
- References
- Summary
Introduction: Disclaimers

- informal vs. dictionary definitions
- illustrative rather than proscriptive
- examples are deliberately incomplete and beyond reproach!
Why should you care?

- XML models are here to stay
- XML models are proliferating rapidly, frequently single-purposed and created in isolation
- XML models are frequently not created by data management professionals
- Data management professionals need to be at the forefront
**Definitions (1 of 2)**

- **ER model**: “a data model for describing a database in an abstract way” per Wikipedia; for describing data relevant for an organization
  - includes diagram(s) and associated text
- **instance**: a specific occurrence of an ER model or XML model
- **service (as in SOA)**: a unit of functionality packaged for convenient and consistent use
  - includes set of operations, each with an input and output message
Definitions (2 of 2)

- **XML model**: a specification for a set of XML documents
  - includes diagram(s) and associated text

- **XML model formats**: diagram/logical, grid/physical, text

- **XML document**: an instance of an XML model
Example

- Accident: motor vehicle
- XML model: diagram, grid, text view
- XML document: text view
- ER model (derived from XML model): diagram, text view
- ER model instance: diagram
- ER model (not derived from XML model): diagram
XML model: diagram representation
XML model: grid/physical representation
XML model: text representation, formatted (portion)
<xs:element name="Accident">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Vehicle" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Manufacturer"/>
                        <xs:element name="Model"/>
                        <xs:element name="ModelYear"/>
                        <xs:element name="Speed"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Driver"> <!-- child of Vehicle -->

<xs:complexType>
    <xs:sequence>
        <xs:element name="Name"/>
        <xs:element name="LicenseNumber"/>
        <xs:element name="IssuingState"/>
        <xs:element name="ExpirationDate"/>
    </xs:sequence>
</xs:complexType>

</xs:element>
<!-- child of Vehicle -->

<xsl:element name="Passenger" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>

</xs:complexType>

</xs:element> <!-- end of Vehicle -->
<xs:element name="Location">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PlaceSpecification"/>
      <xs:element name="City"/>
      <xs:element name="StateCode"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RoadCondition">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="clear"/>
      <xs:enumeration value="raining"/>
      <xs:enumeration value="snowing"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>

</xs:complexType>

</xs:element> <!-- end of Accident -->

</xs:schema>
XML model: instance (XML document)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)-->  
<Accident xsi:noNamespaceSchemaLocation="Accident.xsd" xmlns:xs1="http://www.w3.org/2001/XMLSchema-instance">
  <Vehicle>
    <Manufacturer>Ford</Manufacturer>
    <Model>Fusion</Model>
    <ModelYear>2012</ModelYear>
    <Speed>5 mph</Speed>
    <Driver>
      <Name>Bill Pawlowski</Name>
      <LicenseNumber>712345678</LicenseNumber>
      <IssuingState>AZ</IssuingState>
      <ExpirationDate>2014-02-28</ExpirationDate>
    </Driver>
  </Vehicle>
  
  <Vehicle>
    <Manufacturer>Toyota</Manufacturer>
    <Model>Prins</Model>
    <ModelYear>2011</ModelYear>
    <Speed>0 mph</Speed>
    <Driver>
      <Name>Laura Kostyo</Name>
      <LicenseNumber>587654321</LicenseNumber>
      <IssuingState>AZ</IssuingState>
      <ExpirationDate>2013-12-20</ExpirationDate>
    </Driver>
    
    <DateTime>2013-09-12T12:15:00</DateTime>
    <Location>
      <PlaceSpecification>9200 E Pima Center Parkway</PlaceSpecification>
      <City>Scottsdale</City>
      <StateCode>AZ</StateCode>
    </Location>
    <WeatherCondition>raining</WeatherCondition>
  </Vehicle>
</Accident>
```
ER model: derived from XML model

class Accident-MotorVehicle-fromXML

Accident: Motor Vehicle
data model diagram (UML class model notation)
derived from XML model

- accident
  - weather condition
  - datetime
  - involved in vehicle
  - operator of
  - occupied by
  - occurred at location

- location
  - place specification
    - city
    - state code

- vehicle
  - manufacturer
  - model
  - speed

- driver
  - name
  - license number
  - issuing state
  - expiration date

- passenger
  - name

«enumeration»
weather condition
- clear
- raining
- snowing
## ER model: text representation (portion)

### accident

#### Columns

<table>
<thead>
<tr>
<th>PK</th>
<th>Name</th>
<th>Type</th>
<th>Not Null</th>
<th>Unique</th>
<th>Len</th>
<th>Prec</th>
<th>Scale</th>
<th>Init</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>weather</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>datetime</td>
<td>datetime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Relationships

<table>
<thead>
<tr>
<th>Columns</th>
<th>Association</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1..?</td>
<td>accident.occurred at</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>location.location for</td>
<td></td>
</tr>
<tr>
<td>1..?</td>
<td>accident.involves</td>
<td></td>
</tr>
<tr>
<td>1..*</td>
<td>vehicle.involved in</td>
<td></td>
</tr>
</tbody>
</table>

### driver

**Database:** Java, **Stereotype:** , **Package:** Accident-MotorVehicle-fromXML  
**Detail:** Created on 9/1/2013, Last modified on 9/1/2013.

#### Columns

<table>
<thead>
<tr>
<th>PK</th>
<th>Name</th>
<th>Type</th>
<th>Not Null</th>
<th>Unique</th>
<th>Len</th>
<th>Prec</th>
<th>Scale</th>
<th>Init</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>name</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>license number</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>issuing state</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>expiration date</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ER model: instance diagram

- **imaginary :accident**
  - datetime = 2013-09-12T12:15:00
  - weather condition = raining

- **Bill’s :vehicle**
  - manufacturer = Ford
  - model = Fusion
  - speed = 5 mph

- **Bill :driver**
  - name = Bill Pawlowski
  - license number = S12345678
  - issuing state = AZ
  - expiration date = 2014-02-28

- **Western Heritage Insurance Co pkg lot :location**
  - place specification = 9200 E Pima Center Parkway
  - city = Scottsdale
  - state code = AZ

- **Laura’s :vehicle**
  - manufacturer = Toyota
  - model = Prius
  - speed = 0 mph

- **Laura :driver**
  - name = Laura Kostyo
  - license number = S87654321
  - issuing state = AZ
  - expiration date = 2013-12-20
ER model: not derived from XML model

class Accident-MotorVehicle

Accident
- datetime
- weather condition
- involves 1..* VehicleInvolve
- occurs at place of 1
- place of 1

VehicleInvolve
- speed 1
- involvement in 1

Vehicle
- manufacturer name
- model name
- model year
- includes 0..*

Driver
- name
- drivers license number
- drivers license expiration date
- drivers license state name

Passenger
- name

Location
- place specification
- city
- state code
- includes 0..*

Location
- place of

Driver
- involves 1..* Location

Passenger
- involves 0..* Location

## Comparisons: XML vs. ER Model

- graphical format
- text format
- target audience
- datatypes
- adoption
- acceptance
- tool support
- advantages and disadvantages
- criteria for selection
- appropriateness for database, message, and services design
## Graphical Format: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A/D</strong>: several relatively standard formats: Entity Relationship (ER)/Barker Notation, IDEF1X Notation, UML (primary and foreign keys illustrated in different ways in different tools)</td>
<td><strong>D</strong>: each vendor is different</td>
</tr>
<tr>
<td><strong>A</strong>: ability to illustrate relationships in both directions</td>
<td><strong>D</strong>: no ability to illustrate relationships in both directions</td>
</tr>
<tr>
<td></td>
<td><strong>D</strong>: hierarchical only</td>
</tr>
<tr>
<td>ER</td>
<td>XML</td>
</tr>
<tr>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>D: each tool vendor may have their own default format</td>
<td>A: standard XML schema definition from W3C industry standards organization</td>
</tr>
</tbody>
</table>
### target audience: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>data modelers, DBAs, software developers, business subject matter experts</td>
<td>software developers, business subject matter experts</td>
</tr>
</tbody>
</table>
## datatypes: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>D: no standard datatypes (except DBMS-specific for physical data models; ISO/IEC 11404, Information technology — General-Purpose Datatypes (GPD), for programming languages and software interfaces, seldom used in data models)</td>
<td>A: standard XML schema datatypes</td>
</tr>
</tbody>
</table>
### Adoption: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>D: niche, only by data modelers</td>
<td>A: widespread, becoming ubiquitous</td>
</tr>
<tr>
<td></td>
<td>ER</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>acceptance:</td>
<td>widespread in data management community, not widely used in software development community</td>
</tr>
<tr>
<td>XML vs. ER Model</td>
<td></td>
</tr>
</tbody>
</table>
## tool support: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diagram Creation:</strong> D: limited</td>
<td><strong>Diagram Creation:</strong> D: limited</td>
</tr>
<tr>
<td><strong>Text Creation:</strong> D: limited/not</td>
<td><strong>Text Creation:</strong> A: widespread</td>
</tr>
<tr>
<td>applicable since no standard format</td>
<td>(any text editor)</td>
</tr>
</tbody>
</table>

## advantages and disadvantages: XML vs. ER Model

<table>
<thead>
<tr>
<th>ER</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: normalization rules</td>
<td>A: widespread tool support for reading XML instance formats (every web browser and text editor can display an XML document)</td>
</tr>
<tr>
<td>D: no standard format for representing instances/data contents (except UML object diagrams) (spreadsheets, text are non-standard)</td>
<td>A: defines serialization format in XML for instances and electronic data exchange</td>
</tr>
<tr>
<td>D: no specified serialization format for instances (although one can be derived from the model)</td>
<td>D: no standard datatypes</td>
</tr>
<tr>
<td>D: no standard datatypes</td>
<td>A: standard datatypes in XML schema</td>
</tr>
</tbody>
</table>
instance example: ER (UML object diagram)

object Accident-MotorVehicle-ObjectDiagram

Accident: Motor Vehicle
data model instance diagram (UML object model notation)

imaginary :accident
datetime = 2013-09-12T12:15:00
weather condition = raining

Bill's :vehicle
manufacturer = Ford
model = Fusion
speed = 5 mph

Bill :driver
name = Bill Pawlowski
license number = S12345678
issuing state = AZ
expiration date = 2014-02-28

Western Heritage Insurance Co pkg lot :location
place specification = 9200 E Pima Center Parkway
city = Scottsdale
state code = AZ

Laura's :vehicle
manufacturer = Toyota
model = Prius
speed = 0 mph

Laura :driver
name = Laura Kostyo
license number = S87654321
issuing state = AZ
expiration date = 2013-12-20
instance example: XML document

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)>-

<Vehicle>
    <Manufacturer>Ford</Manufacturer>
    <Model>Fusion</Model>
    <ModelYear>2012</ModelYear>
    <Speed>5 mph</Speed>
    <Driver>
        <Name>Bill Pawlowski</Name>
        <LicenseNumber>S12345678</LicenseNumber>
        <IssuingState>AZ</IssuingState>
        <ExpirationDate>2014-02-28</ExpirationDate>
    </Driver>
</Vehicle>

<Vehicle>
    <Manufacturer>Toyota</Manufacturer>
    <Model>Prin</Model>
    <ModelYear>2011</ModelYear>
    <Speed>0 mph</Speed>
    <Driver>
        <Name>Laura Kostyo</Name>
        <LicenseNumber>S87654321</LicenseNumber>
        <IssuingState>AZ</IssuingState>
        <ExpirationDate>2013-12-20</ExpirationDate>
    </Driver>
</Vehicle>

<Date/time>2013-09-12T12:15:00</Date/time>
<Location>
    <PlaceSpecification>9200 E Pima Center Parkway</PlaceSpecification>
    <City>Scottsdale</City>
    <StateCode>AZ</StateCode>
</Location>
<WeatherCondition>raining</WeatherCondition>
```

Appropriateness

- for database design
- for message design
- for services
appropriateness for database design

- **XML**
  - A: appropriate only for design of XML databases

- **ER**
  - A: appropriate for design of relational databases, less appropriate for design of XML databases
appropriateness for message design

- **XML**
  - **A:** appropriate for design of messages in XML format

- **ER**
  - **D:** not directly appropriate, but message design can be systematically derived from an ER model
## Appropriateness for Services

<table>
<thead>
<tr>
<th>XML</th>
<th>A: appropriate for specification of input, output, error messages of each operation (assuming XML message format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER</td>
<td>D: not appropriate for specification of input, output, error messages of each operation, however their specification can be derived from an ER model</td>
</tr>
</tbody>
</table>
criteria for selection: XML or ER Model

- If goal is relational database design, ER model is best
- If goal is XML database design, XML model is best, ER model is possible
- If goal is message design, XML model is appropriate for XML messages, message design can be systematically derived from an ER model
- If goal is determining a service and its operations, neither is appropriate
- If goal is defining operation input and output messages for a service, XML model is appropriate for XML messages, message design can be systematically derived from an ER model
Deriving XML Message Format from ER Model

1. determine message category (e.g., request/response, query/query response, notification) and triggering event (if applicable) and message type (message structure)

2. determine starting entity (typically explicit in business purpose, message type, or triggering event)

3. select applicable attributes of the entity

4. sequentially select all applicable associations and their target entities and attributes (and sequentially select those entities’ applicable associations and target entity and attributes, and …)
Derive from ER model: example

- list of future hotel reservations for a guest
1. determine message category (e.g., request/response, query/query response, notification) and triggering event (if applicable) and message type (message structure)

- message category: **query response**
- triggering event: **not applicable**
- message type: **future hotel reservations for a guest**
Derive from ER model: ER model

class HotelReservation

- guest
  - name
  - email address
- hotel reservation
  - arrival date
  - checkout date
  - average daily rate
- credit card
  - type
  - number
  - expiration date
- hotel
  - name
  - web address
  - phone number

Hotel Reservation data model (UML class model notation)
Derive from ER model, process: model path

- guest (A)
  - hotel reservation (A.1)
    - hotel (A.1.a)
    - credit card (A.1.b)
      - (guest: no) (A.1.b.1)
    - (credit card: no) (A2)
Derive from ER model, process: 2 of 7

- 2. determine starting entity (typically explicit in business purpose, message type, or triggering event)

- starting entity: guest (A)
3. select applicable attributes of the entity

attributes of guest: name, email address (A)
4. sequentially select all applicable associations and their target entities and attributes (and sequentially select those entities’ applicable associations and target entity and attributes, and …)

- first association from guest: holds one-to-many hotel reservation

- target entity: hotel reservation (A.1)

- target entity attributes: arrival date, checkout date, average daily rate

- target entity (hotel reservation) first association: is for one hotel (A.1.a)
Derive from ER model, process: 5 of 7

- target entity (hotel reservation) first association: is for one hotel (A.1.a)

- target entity: hotel (A.1.a)

- target entity attributes: name, phone number, web address

- target entity (hotel) first association: no additional associations
target entity (hotel reservation) second association: reserved by one credit card

target entity: credit card (A.1.b)

target entity attributes: number (other attributes not applicable)

target entity (credit card) first association: guest (A.1.b.1): not applicable, don’t select the association
second association from guest: holder of one-to-many credit card (A.2): not applicable, don’t select the association
Derive from ER model: ER model

- Hotel Reservation
  - holds
  - guest
    - name
    - email address
    - holder of
  - hotel reservation
    - arrival date
    - checkout date
    - average daily rate
    - reserved by
      - 0..*
      - 1
    - holder of
      - 1..*
      - 1
    - reserved by
      - 0..*
      - 1
  - credit card
    - type
    - number
    - expiration date
  - hotel
    - name
    - web address
    - phone number
XML schema from ER model

Generated by XMLSpy

www.altova.com

<?xml version="1.0" encoding="UTF-8"?>
<!-- Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)-->
<GuestHotelReservation xmlns:noNamespaceSchemaLocation="HotelReservation.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Guest>
    <Name>Norman Daoust</Name>
    <EmailAddress>NormanD@DaoustAssociates.com</EmailAddress>
  </Guest>
  <HotelReservation>
    <ArrivalDate>2013-09-11</ArrivalDate>
    <CheckOutDate>2013-09-11</CheckOutDate>
    <AverageDailyRate>$69</AverageDailyRate>
  </HotelReservation>
  <Hotel>
    <Name>Best Western</Name>
    <WebAddress>http://www.innsuites.com/phoenix-biltmorescottsdale-suites.php</WebAddress>
    <PhoneNumber>602-997-8285</PhoneNumber>
  </Hotel>
  <CreditCard>
    <Number>1234567890123456</Number>
  </CreditCard>
</GuestHotelReservation>
Advantages of deriving XML models/messages from ER model

- insures holistic view by looking at relationships in both directions, not just one; that may eliminate analysis blind spots
- typically results in increased consistency and reuse
- typically results in higher quality XML models
Disadvantages of deriving XML models/messages from ER model

- takes more time
- you’ll ask questions that people may not have thought about and don’t have answers to
- requires thinking about possible future uses
References

- www.w3.org/TR/#tr_XML_Schema, XML schema specification
- en.wikipedia.org/wiki/XML_Schema_Editor, XML schema tools
- *Succeeding with SOA*, Addison Wesley, 2007, Paul C. Brown
Summary

➢ To understand a business or business area, create an ER model

➢ To design a relational database to provide the data for a business or business area, create an ER model

➢ To design an XML data exchange format, create an XML model, preferably derived from an ER model

➢ To design a set of consistent data exchange formats across the enterprise, derive the data exchange formats from an ER model
Presentation Outline

- Why should you care?
- Definitions
- Examples
- Comparison: XML model vs. entity relationship model
- Deriving XML models from an entity relationship model
- References
- Summary
Presentation Goals

- similarities and differences between entity relationship data modeling and XML modeling
- advantages and disadvantages of both entity relationship data models and XML models
- appropriate usages for data models and XML models
- criteria for determining which to create
Quote

“To a person with a hammer, every problem looks like a nail.”

Daoust Associates corollary: “A modeler using multiple tools can create useful and high quality deliverables for their organization.”
Thanks!

Norman Daoust
Daoust Associates
www.daoustassociates.com
NormanD@DaoustAssociates.com
(617) 491-7424