Concepts & Principles of Conceptual Data Modeling

(Chapter 1 & 2)

Dr. Mustafa Jarrar

University of Birzeit
mjarrar@birzeit.edu
www.jarrar.info
Watch this lecture and download the slides from

http://jarrar-courses.blogspot.com/2015/01/dataandbusinessprocessmodelling.html

Some diagrams in this lecture are based on [1]
Concepts & Principles of Conceptual Data Modeling

- Part 1: what is Information Modeling/Engineering
  - Part 2: Information Modeling Approaches
  - Part 3: Introduction to Object Role Modeling (ORM)
  - Part 4: Information Levels
Do you like the design of this table?

<table>
<thead>
<tr>
<th>Movie</th>
<th>Year</th>
<th>Director</th>
<th>Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
<td>William Baldwin, Robert De Niro, Kurt Russell</td>
</tr>
<tr>
<td>Cosmology</td>
<td>1994</td>
<td>Terry Harding</td>
<td></td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
<td>Kevin Kostner, Mary McDonnell</td>
</tr>
</tbody>
</table>

- This table is an output report. It provides one way to view the data.
- Different movies may have the same title.
- Movie numbers are used to provide a simple identifier.
- Each cell (row--column slot) may contain many values.

**How can we design tables to store such facts?**
## Information Modeling – The need for good design

**A badly-designed table, why?**

### Movie

<table>
<thead>
<tr>
<th><strong>MovieName</strong></th>
<th><strong>Release Year</strong></th>
<th><strong>Director</strong></th>
<th><strong>Star</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Awakenings</td>
<td>1991</td>
<td>Penny Marshall</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Awakenings</td>
<td>1991</td>
<td>Penny Marshall</td>
<td>Robin Williams</td>
</tr>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
<td>William Baldwin</td>
</tr>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
<td>Kurt Russell</td>
</tr>
<tr>
<td>Cosmology</td>
<td>1994</td>
<td>Terry Harding</td>
<td></td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
<td>Kevin Kostner</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
<td>Mary McDonnell</td>
</tr>
</tbody>
</table>
Information Modeling – The need for good design

Do you like the design of this table?

<table>
<thead>
<tr>
<th>MovieName</th>
<th>Release Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awakenings</td>
<td>1991</td>
<td>Penny Marshall</td>
</tr>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
</tr>
<tr>
<td>Cosmology</td>
<td>1994</td>
<td>Terry Harding</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MovieName</th>
<th>Star</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awakenings</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Awakenings</td>
<td>Robin Williams</td>
</tr>
<tr>
<td>Backdraft</td>
<td>William Baldwin</td>
</tr>
<tr>
<td>Backdraft</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Backdraft</td>
<td>Kurt Russell</td>
</tr>
<tr>
<td>Cosmology</td>
<td></td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>Kevin Kostner</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>Mary McDonnell</td>
</tr>
</tbody>
</table>
### Information Modeling – The need for good design

Do you like the design of this table?

<table>
<thead>
<tr>
<th>MovieName</th>
<th>Release Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awakenings</td>
<td>1991</td>
<td>Penny Marshall</td>
</tr>
<tr>
<td>Backdraft</td>
<td>1991</td>
<td>Ron Howard</td>
</tr>
<tr>
<td>Cosmology</td>
<td></td>
<td>Terry Harding</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>1990</td>
<td>Mary McDonnell</td>
</tr>
</tbody>
</table>

**A relational database representation**

```
Movie ( movieName, releaseYr, director )

Starred ( movieName, star )
```
### Information Modeling – The need for good design

Do you like the design of this table?

- Information Modeling is both a science and an art.
- When supported by a good modeling approach, this design process is a stimulating and intellectually satisfying activity, with tangible benefits gained from the quality of the database applications produced.

<table>
<thead>
<tr>
<th>Movie</th>
<th>Movie Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awakenings</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Backdraft</td>
<td>William Baldwin</td>
</tr>
<tr>
<td>Backdraft</td>
<td>Robert De Niro</td>
</tr>
<tr>
<td>Backdraft</td>
<td>Kurt Russell</td>
</tr>
<tr>
<td>Cosmology</td>
<td></td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>Kevin Kostner</td>
</tr>
<tr>
<td>Dances with wolves</td>
<td>Mary McDonnell</td>
</tr>
</tbody>
</table>
Information Modeling – The need for good design

Why a good design is important?
  – Consistency
  – Efficiency

What makes a good design good?
  – Correct
  – Complete
  – Efficient

• What skills you should have to be a good data engineer?
• What approaches exist to help you reach good models?
The application area being modeled is called the universe of discourse (UoD).

Building a good model requires a good understanding of the world we are modeling.

The main challenge is to describe the UoD clearly and precisely.

A person responsible for modeling the UoD is called a modeler.

We should consult with others who, at least collectively, understand the application domain —these people are called domain experts, subject matter experts, or UoD experts.

For implementation, it is important to represent information at the conceptual level—in concepts that people (molders and domain experts) find easy to work with.

This added flexibility also makes it easier to implement the same conceptual model in different ways, DB schema, XML schema, etc.
Concepts & Principles of Conceptual Data Modeling

- Part 1: what is Information Modeling
- Part 2: Information Modeling Approaches
  - Part 3: Introduction to Object Role Modeling (ORM)
  - Part 4: Information Levels
The main information modeling approaches are:

Entity-Relationship modeling (ER)

Object-oriented modeling (UML)

Fact-oriented modeling (ORM)
Modeling Approaches

Given simple data for room scheduling:

**ER-model**

**UML-model**

**ORM-model**
Entity-Relationship Modeling (ER)

- Introduced by Peter Chen in 1976, widely used approach for DB modeling.
- Pictures the world in terms of entities that have attributes and participate in relationships.
- Many different versions of ER (no standard ER notation). Different versions of ER may support different concepts and may use different symbols for the same concept.
- Relationships are depicted as named lines connecting entity types. Only binary relationships are allowed, and each half of the relationship is shown either as a solid line (mandatory) or broken line (optional). A fork or “crow’s foot” at one end of a relationship indicates that many instances of the entity type at that end may be associated (via that relationship) with the same entity instance at the other end of the relationship. The lack of a crow’s foot indicates that at most one entity instance at that end is associated with any given entity instance at the other end.
• UML class diagram are used to specify static data structures (OMG Standard).

• Encapsulates both data and behavior within objects.

• Pictures the world in terms of classes that have attributes and participate in associations. Ternary associations are allowed, see the diagram.

• UML allows constraints in braces or notes in whatever language you wish.

• Form example, \{P\} can be added to denote primary uniqueness and \{U1\} for an alternate uniqueness—these symbols are not standard and hence not portable. The uniqueness constraints on the ternary are captured by the two 0..1 (at most one) multiplicity constraints. The “*” means “0 or more”. Attributes are mandatory by default.
Fact-oriented Modeling (ORM)

- Introduced by Sjir Nijssen early 1970s, was called **NIAM**.
- Revised by Terry Halpin (late 1980s), and called:

  **Object-Role Modeling (ORM)**

- It views the world as **object-types playing roles**.
- **Object-types are ellipses (no attributes)**, and relations consists of **roles**.
- Not only **n-ary relations** are supported, but ORM supports also more than **15 types** of constrains graphically.
- ORM allows **verbalization** of diagrams.
- More **conceptual** than UML and ER.
- ORM is a **modeling approach**, not only a modeling language.
Concepts & Principles of Conceptual Data Modeling

- Part 1: what is Information Modeling
- Part 2: Information Modeling Approaches
- Part 3: Introduction to Object Role Modeling (ORM)
- Part 4: Information Levels
Object-Role Modeling (ORM)

<table>
<thead>
<tr>
<th>Room (Nr)</th>
<th>Time</th>
<th>Activity Code</th>
<th>Activity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Mon 9am</td>
<td>VMC</td>
<td>VisioModeler Class</td>
</tr>
<tr>
<td>20</td>
<td>Tue 2pm</td>
<td>VMC</td>
<td>VisioModeler Class</td>
</tr>
<tr>
<td>33</td>
<td>Mon 9am</td>
<td>AQU</td>
<td>ActiveQuery Demo</td>
</tr>
<tr>
<td>33</td>
<td>Fri 5pm</td>
<td>SP</td>
<td>Staff party</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Diagram:

- Room (Nr) connected to Time (dh) with arrows indicating 'at' and 'is booked for'.
- Activity (code) connected to Activity Name with arrows indicating 'Has / refers to'.

<table>
<thead>
<tr>
<th>Room (Nr)</th>
<th>Time</th>
<th>Activity Code</th>
<th>Activity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Mon 9am</td>
<td>VMC</td>
<td>VisioModeler Class</td>
</tr>
<tr>
<td>20</td>
<td>Tue 2pm</td>
<td>VMC</td>
<td>VisioModeler Class</td>
</tr>
<tr>
<td>33</td>
<td>Mon 9am</td>
<td>AQU</td>
<td>ActiveQuery Demo</td>
</tr>
<tr>
<td>33</td>
<td>Fri 5pm</td>
<td>SP</td>
<td>Staff party</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Jarrar © 2015
Object-Role Modeling (ORM)

Representing information graphically

<table>
<thead>
<tr>
<th>Movie</th>
<th>Year</th>
<th>Director</th>
<th>Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosmology</td>
<td>1994</td>
<td>Terry Harding</td>
<td>William Baldwin, Robert De Niro, Kurt Russell</td>
</tr>
<tr>
<td>Dances with Wolves</td>
<td>1990</td>
<td>Kevin Kostner</td>
<td>Kevin Kostner, Mary McDonnell</td>
</tr>
</tbody>
</table>

- ReleasedIn
- DirectedBy/Directs
- StaredBy/
ORM is conceptual modeling language. ORM has an expressive graphical notation. ORM is designed for modeling DB schemes at the conceptual level. You build an ORM schema and then click a bottom to automatically generate a database.

Our goal in this course is to use ORM as general Conceptual Modeling language, rather than as database modeling language.

ORM can be used for modeling business rules, ontology, XML schemes, and others.
ORM Usage Scenarios

Originally

Database

Later

Ontology

XML Schema

Warehouse

Business Rules

Web (x)Forms

Requirements Engineering

Record my recipes!
Object-Role Modeling (ORM): Other Examples

- Book
- Has/Is-Of
- Title
- Witten-By/Writes
- Author
Object-Role Modeling (ORM): Other Examples

Has/Is-Of

1. Databases Systems
2. Intro to Java

Witten-By/Writes

1. J. Date
2. B. Hacker
2. A. Smith

Called Population
Object-Role Modeling (ORM): Other Examples
Object-Role Modeling (ORM) constructs

Called **Object Type** (or Concept, or Class)

**Relation**
Each part is called a **Role**

**Subtype** relation
Object-Role Modeling (ORM) constructs

Called **Fact Type**
Each fact type should be elementary.
Object-Role Modeling (ORM) constraints

**Uniqueness Constraint:**
Each Book must have at most one ISBN

**Uniqueness Constraint:**
Each ISBN must be Is-Of at most one Book

**Mandatory Constraint:**
Each Book must have at least one ISBN

**Uniqueness Constraint:**
It is possible that a Book is Written-By more than one Author, and vice versa
E-Government Ontology (in ORM)
Modeling Approaches (short discussion)

Which is more intuitive for modelers? For domain experts?

**ER-model**
- Nice for picturing to DB-schemes

**UML-model**
- Close to the way programmers think

**ORM-model**
- Suitable for general conceptual modeling, not only DB schemes
Part 1: what is Information Modeling

Part 2: Information Modeling Approaches

Part 3: Introduction to Object Role Modeling (ORM)

Part 4: Information Levels
Information Levels (Data Modeling Viewpoint)

Conceptual Level
- What kind of facts/concepts we need, and how they are related.
- Conceptual models are designed for clear communication, especially between modelers and domain experts.

Logical Level
- Abstract data structures
- Same conceptual schema can be mapped into several logical structures

Physical Level
- The physical storage and access structures used in a system (indexes, file clustering, etc.).
- Same Logical schema can be stored in different ways
Information Levels (Data Modeling Viewpoint)

**Conceptual Level**
- What kind of facts/concepts we need, and how they are related.
- Conceptual models are designed for clear communication, especially between modelers and domain experts.

**Logical Level**
- Abstract data structures
- Same conceptual schema can be mapped into several logical structures

**Physical Level**
- The physical storage and access structures used in a system (indexes, file clustering, etc.).
- Same Logical schema can be stored in different ways

**Ontological Level**
- Concerned with the meaning, in the real world.
- Same meaning (/intentions) can be conceptualized in different ways.

**Linguistic Level**
- Concerned with the terms used to lexicalize the meaning.
- Same meaning can be lexicalized in different languages.

- Abstract data structures
- Same conceptual schema can be mapped into several logical structures
## Knowledge Levels (from philosophy viewpoint)

<table>
<thead>
<tr>
<th>Level</th>
<th>Primitives</th>
<th>Interpretation</th>
<th>Main feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linguistic</td>
<td>Linguistic terms</td>
<td>Subjective</td>
<td>Language dependence</td>
</tr>
<tr>
<td>Conceptual</td>
<td>Conceptual relations</td>
<td>Subjective</td>
<td>Conceptualization</td>
</tr>
<tr>
<td>Ontological</td>
<td>Ontological relations</td>
<td>Constrained</td>
<td>Meaning</td>
</tr>
<tr>
<td>Epistemological</td>
<td>Structuring relations</td>
<td>Arbitrary</td>
<td>Structure</td>
</tr>
<tr>
<td>Logical</td>
<td>Predicates, functions</td>
<td>Arbitrary</td>
<td>Formalization</td>
</tr>
</tbody>
</table>

➢ Will be discussed later
Information Levels (Data Modeling Viewpoint)

Conceptual Level
- What kind of facts/concepts we need, and how they are related.
- Conceptual models are designed for clear communication, especially between modelers and domain experts.

Ontological Level
- Concerned with the meaning, in the real world.
- Same meaning (intentions) can be conceptualized in different ways.

Logical Level
- Abstract data structures
- Same conceptual schema
- ORM is the most suitable language for conceptual modeling (not only conceptual data modeling). That is, it allows modelers to think more conceptually and be more independent from the logical level.
- ORM is also being used as ontology modeling language, business rules and requirements specification, XML-schema modeling, etc. (not only DB modeling)

Linguistic Level
- Concerned with the terms used to lexicalize the meaning.
- Same meaning can be lexicalized in different languages.

Physical Level
- The physical storage and access structures used in a system (indexes, file clustering, etc.).
- Same Logical schema can be stored in different ways.
ORM Tools (feel free to use any tool in the course)

➔ Microsoft Visio

➔ Other Tools:

Microsoft VisioModeler (Free but not supported anymore)
- you need replace a DLL to run it in Win7.

NORMA
- downloadable from SourceForge or http://www.ormfoundation.org/files/folders/norma_the_software/default.aspx.
- Free and open source (but you need Visual Studio 2005 or 2008 to run it).
- Supports ORM2

DogmaModeler
- downloadable from http://www.jarrar.info/Dogmamodeler/
- Free and open source (prototype status)
- Designed as Ontology modeling tool (Norma and VisioModeler are database tools)
- Will be required later in the course.
References