Object Role Modeling

A Better Way to do Data Modeling

deverest@csom.umn.edu
©Gordon C. Everest
Professor Emeritus, Carlson School of Management and College of Continuing Education University of Minnesota

NIAM/ORM Teaching Experiences

THE COURSE:
• Advanced Database Design & DBAdministration
• Taught NIAM modeling since 1975, but inadequate

NEEDED:
• Written Teaching Materials

• Data Modeling CASE Tool
  InfoDesigner from ServerWare, 1993
  InfoModeler from Asymetrix, 1994; then InfoModelers, Inc.
  VisioModeler from Visio, (Enterprise 5.0) acquired 1998, released 1999
  acquired by Microsoft, 1999/11.
  VisioEA, part of Visual Studio.net EA edition, released 2002/2
NIAM / ORM Data Modeling

What’s in the Name?

Nijssen --- Natural language
Information
Analysis
Method
or
Object
Role, Relationship
also: Binary-Relationship or just Binary

NOTE: Has no relationship with Object-Oriented ___.

Outline

Roadmap for this presentation:

1. Data Modeling
   – ER (most general) => Relational Modeling
   – Problems with Record-based modeling

2. Transitioning from ER/Relational to ORM

3. Modeling steps in ORM
   – Verbalize - elementary fact sentences
   – Symbolize - ORM data model diagram
   – Add Constraints
   – Abstracting a data model diagram... for presentation

4. Place of ORM in a Taxonomy of Data Modeling Schemes

5. Summary and Reprise
Logical Database Design
Objective, Principles, Benefits

OBJECTIVE of LOGICAL DATABASE DESIGN:

TO ACCURATELY AND COMPLETELY MODEL SELECTED PORTIONS OF THE REAL WORLD OF INTEREST TO A COMMUNITY OF USERS.

• USERS (COLLECTIVELY) WILL ALWAYS KNOW MORE ABOUT A DATA STRUCTURE THAN THE SYSTEM KNOWS, OR THAN COULD BE DEFINED TO THE SYSTEM.

• WHAT IS NOT FORMALY DEFINED TO THE SYSTEM, THE SYSTEM CANNOT MANAGE . . . THE USERS MUST!

• THEREFORE, NEED TO CAPTURE RICH SEMANTICS WITH COMPREHENSIVE DATA MODELING and DEFINITION, INCLUDING INTEGRITY CONSTRAINTS AND OPERATIONS.

Let the ‘system’ do it! Implications for a Tool!

Data Modeling Constructs

What to look for:

ENTITY

RELATIONSHIP

DOMAIN

ATTRIBUTE (Data Item)

IDENTIFIER

characteristics:
• Multiple / Exclusive
  (at most one)
• Optional / Required
  (at least one)
NULLable / or NOT
Data Modeling Constructs

Relative emphasis differentiates Data Modeling Schemes:

- **ER modeling** focuses on Entities and Relationships, de-emphasizing, even hiding Attributes.

- **Relational (restricted ER)** focuses on Entities and Attributes (single valued), relegating Relationships to Foreign Keys.

- **Object Role Modeling (ORM)** focuses on Objects/Domains and (Roles in) Relationships

Representing a M:N Relationship

Another Pattern: WATSONS, Ch.5, p.115.

- If you cannot store multiple Projects (or Project IDs) in an Employee record, or multiple Employees (or Employee IDs) in a Project record (as is the case in a Relational Database), then ... you must introduce an “Intersection Entity” between them to represent the Many-to-Many Relationship.

- The Intersection Entity also provides the place to store additional attributes of the relationship e.g., Hours Worked, Rate of Pay, ...

  *What is the IDentifier? Where store attributes if 1:M? What is the problem with this representation?*
Representing a Ternary Relationship

While we can develop a consistent notation for binary relationships, ternary relationships are a problem. Sample EER (Theory) notation:

- If one of the entities is single valued, is it really ternary? Or “attributed” binary?
- What lends uniqueness to each instance of the relationship?
- How to verbalize the relationship? Which order?
- How to represent Multiplicity / Exclusivity?
- How to represent Dependency? Must have all 3?

Record-Based Modeling

Given two facts (conceptually):
- one about the city a person lives in
- another about the city a person works in

Assume:
- every person has to live and work in a city
- each person can live and work in only one city (at a time)
- not interested in anything more about persons or cities

Example (a fact instance):
- Gordon Everest lives in Roseville and works in Minneapolis

Diagram a conceptual data model
- to represent this information (a database to contain these facts)
Record-Based Data Model  Ex.2

- What is the entity and what is the attribute?
- Would it make any sense to say (to a novice layperson - a user):
  - CITY was an “attribute” of PERSON?
- Doing more than is necessary at the conceptual level

<table>
<thead>
<tr>
<th>PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersonID [key]</td>
</tr>
<tr>
<td>LiveCity</td>
</tr>
<tr>
<td>WorkCity</td>
</tr>
</tbody>
</table>

- cannot have CITY and CITY as attributes of PERSON
- column/attribute name reflects “entity + role”
- CITY as an entity/object is lost (not its own table)
- what if there is a CITY where no one lives or works
- some add concept of a DOMAIN

Object-Role Model  Ex.3

FORML language statements:
- PERSON lives in CITY
- Every PERSON lives in some CITY
- Each PERSON lives in at most one CITY
- ... for works in

© Gordon C. Everest, All rights reserved.
Record-Based Modeling

Ex.4

for an additional fact:

- A PERSON makes sales calls in multiple CITIES

DIAGRAM the extended conceptual data model

- can you add an attribute "SalesCallCities" to PERSON?

FLAT Record-Based Modeling is even worse:

- create a new table SALESCALLS with a compound key
  – Is this a real entity in the conceptual view?

EXTEND THE OBJECT-ROLE DATA MODEL

Object-Role (ORM) Data Modeling

THE ESSENTIAL DIFFERENCE:

- Three main constructs ..rolled into.. Two main constructs

Record-based modeling:

NIAM/ORM modeling:

- ENTITY
- Attribute
- Relationship

What to call it?

- ENTITY
- OBJECT
- DOMAIN
- ATTRIBUTE

Role in relationship

© Gordon C. Everest, All rights reserved.
**Record-based Design**

**WHAT DOES THIS “RECORD” REPRESENT?**

**ENTITY:**

X A B C

Design minimal "records" with *at most one* non-key domain.
Remedy for Normal form violations is Decomposition.
This is the ultimate end of Record Decomposition.

Now what do these “records” represent?
Perhaps Codd was right in naming it a __________!

Avoids spurious associations, e.g., A – B ...
Could there be any violations of normal forms?

What about representing the entity X? or any domain?

What if A is related to other “entities”?

---

**ER / Record-based Modeling**

**VALUE DOMAIN**

**VALUE DOMAIN**

**VALUE DOMAIN**

**VALUE DOMAIN**

... roles

**TABLE:**

X A B C D

CLUSTERING of ATTRIBUTES into RECORDS/RELATIONS
– NOT a necessary or desirable first step
– gets us into trouble: if too much, must decompose to normalize

© Gordon C. Everest, All rights reserved.
Transform Record-based (ER) Design

To really represent the entity domains, objects (entities) have "attributes" (descriptors) by playing roles in relationships with other entities.

Transforming ERel to ORM

ADD foreign keys (redundant). NOTE: Duplicate fields - C, P, I, O.

Break out all entities and attributes into separate objects. ADD relationships (presuming above to be fully normalized).

NOTE: All object domains are shown only once!
NOTE: All functional dependencies are now explicitly shown!
Data Modeling in ORM

Try it yourself, starting with =>
Modify / extend the diagram with these semantics:

For each X:
1. A is REQUIRED
2. A is UNIQUE for all X
3. A is MULTIVALUED
4. A is INDEPENDENT
   i.e., can be ORPHAN

Also:
5. B is FUNCTIONALLY
   DEPENDENT on A
6. B and C are RELATED
7. D is an ATTRIBUTE of A

NOTE:
• No "Attributes"
• No 'table think'
• No Foreign Keys
• No Normalization
• Focus on Object Domains
• Think all Relationships

Data Modeling Terminology

O-R
("conceptual")

E-R
("logical")

COBOL/DBTG
"NETWORK"
("physical" implementation)

RELATIONAL

OBJECT
FACT
SENTENCE

ENTITY (TYPE)
ATTRIBUTE
INSTANCE
IDENTIFIER
PREDICATE
RELATIONSHIP
CONSTRAINT

RECORD TYPE
DATA ITEM
(RECORD)
IDENTIFIER
RELATION,
or TABLE
COLUMN,
or FIELD
ROW,
or TUPLE
KEY
FOREIGN KEY
CONSTRAINT
ORM - Digging a little deeper

THE PROCESS: Familiarize > Verbalize > Symbolize

• Gather evidence from User domain
  – look at forms, reports, files
  – what users say; what users do

• Transform into Elementary Facts

• Discover Objects (nouns) and Relationships (verbs)

• Document findings in:
  – Elementary fact sentences
  – ORM Data Model Diagram

Elementary Fact Sentences

ENGLISH GRAMMAR - Structure of a Sentence:

```
SUBJECT + PREDICATE [ + OBJECT ]
   ↓                  ↓
   NOUN       VERB     NOUN
```

**Elementary Fact** -- cannot be decomposed into pieces which collectively provide the same information as the original fact.

George runs. => UNARY
George runs to the store. => BINARY
George likes to run.

NOT:
George likes to run and jump.
George does not like to run. ... CLOSED WORLD ASSUMPTION
George and Mary like to run. ... together!
If George runs, then Mary runs.
All people who run are happy! ...HANDLED WITH ROLE SET CONSTRAINTS
Symbolize: ORM Constructs

- **OBJECT** (ENTITY, CONCEPT) - noun … in an ellipse
- **PREDICATE** (RELATIONSHIP) - verb … in a box
  = role name(s).
  – one or more roles: unary, binary, ternary, +++

Elementary Binary Fact Sentence (schema):

```
PERSON  works in  employs  DEPARTMENT
```

Verbalization: (both ways)

“PERSON works in DEPARTMENT”
“DEPARTMENT employs PERSON”

Exercise – ORM Diagrams

Find the illegal ORM Diagrams; explain why:
**Object Reference Mode**

Non-Lexical Object Types (NOLOTS):

- **PERSON** → *works in* employs **DEPARTMENT**

  bridge

  name

  **PERSON** (name) → *works in* employs **DEPARTMENT** (code)

*Shorthand (simplifies the diagram):*
- add a REFERENCE MODE to each Non-Lexical Object Type

*Now, every Object has a lexical handle.*

---

**Adding ORM Constraints**

- **PERSON** → *works in* employs **DEPARTMENT**

  - Verbalization:
    - "PERSON works in DEPARTMENT"
    - "DEPARTMENT employs PERSON"

  - DEPENDENCY (REQUIRED or MANDATORY):
    - "PERSON must work[s] in some DEPARTMENT"

  - EXCLUSIVITY (UNIQUENESS):
    - "PERSON works in at most one DEPARTMENT"
Uniqueness Constraint

- to represent the exclusivity / multiplicity characteristic of a relationship (ER focuses on the Entities)
- ORM focuses on the Elementary Fact Sentence, so... construct a ‘fact table’ of representative instances:

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peterson</td>
<td>2000</td>
</tr>
<tr>
<td>Lynn</td>
<td>2000</td>
</tr>
<tr>
<td>Carr</td>
<td>2100</td>
</tr>
<tr>
<td>Callagan</td>
<td>2100</td>
</tr>
<tr>
<td>Guttman</td>
<td>2110</td>
</tr>
</tbody>
</table>

- Put a line across the role(s) that make the predicate unique:

An EMPLOYEE can participate in the relationship with DEPARTMENT at most once.

Uniqueness Constraints - Exercises

- The keys for certain fact types are as shown. On this basis, which of these fact types are definitely splitable? – Halpin, §4.5, p.151

(a) ![Diagram](a.png)  (b) ![Diagram](b.png)
(c) ![Diagram](c.png)  (d) ![Diagram](d.png)
(e) ![Diagram](e.png)  (f) ![Diagram](f.png)
(g) ![Diagram](g.png)  (h) ![Diagram](h.png)

RULE:

- Okay?
- Why not?
Mandatory Role Constraint

- the Optional / Dependent characteristic of a relationship
- also called: - total role constraint (other notations: \( \forall \ T \))
- exhaustibility

- every EMPLOYEE must work for some DEPARTMENT.

Disjunctive mandatory:

External Disjunctive Mandatory:
- every EMPLOYEE must work for some DEPARTMENT.

Ternary++ Relationship
Nested ‘Objectified’ Predicate/Fact

What is wrong with this?
Must have a Grade

What about this?
CONTRIBUTION:
DATE  AMOUNT  FUND

In "Table Think":
what is the first question you ask?

ORM gives an explicit means to analyze and record higher order relationships.
Value Set Constraint

On the Population of an Object, defined by:

- **Enumeration** - { M, F }
- **Range** (if an ordering) – { 1 ... 10 }
- **Pattern of Characters** –
  - e.g. \{ a15 \} = 15 alpha characters
  - e.g. \{ d6.d2 \} = up to 6 digits followed by 2 digits after a decimal point
- **Reference Entity** (Table)
  to contain all possible values of the Entity, even if not used elsewhere in the database.
  - Declare the Entity as ‘Independent’
  to allow for orphans, e.g., ...

Frequency Constraints

- **Role Frequency**
  - Limits the number of times an object can play a role; or the number of times a role (or role combination) can appear in a fact table.
  - Place on the role(s) of the Predicate:
    - an ‘n’ or a range
    - optionally, with comparator operators ( <, >, …)
    - or a Range - (min … max)

- **Object Cardinality** (indirect in VisioEA thru Value Set)
  - Limits the size of an Object Population

Not in Record-based modeling schemes!
Role Population Constraints

- Applies at the type level, on whole populations
- Only consider if both roles are optional
- Only apply when roles are on the same entity type population.

May also apply to the tuple populations (both roles in predicate) or to ternary predicates.

Equality:
\[ R_1 \iff R_2 \]
\[ \text{pop}(R_1) = \text{pop}(R_2) \]

Subset:
\[ \text{If } R_2, \text{ then } R_1 \]
\[ \text{pop}(R_2) \leq \text{pop}(R_1) \]

Exclusion:
\[ \text{If } R_2, \text{ then } -R_1 \text{ and vv.} \]
\[ \text{pop}(R_1) \land \text{pop}(R_2) = \text{NULL} \]

Not in Record-based modeling schemes!

Ring Constraints

on a Reflexive Relationship
(Ring Fact Type, Homogeneous Predicate)

Examples:
- PERSON ----- parent of | child of ----- PERSON (M:M)
- PERSON ----- mother of | child of ----- PERSON (1:M)
- PERSON ----- husband of | wife of ----- PERSON (roles, NOW or EVER)
- DEPARTMENT ---- reports to | oversees ---- DEPARTMENT

ACyclic
InTransitive
WHICH ALSO IMPLIES
ASymmetric
IRreflexive

All possible relationship combinations are allowed unless Ring Constraints are declared.

Not in Record-based modeling schemes!
The Many Faces of Databases

- Object-Oriented
- CODASYL Network
- Hierarchical File (COBOL)
- Flat File (FORTRAN)
- Multi-Dimensional
- Snowflake
- Relational
- No File

What do all these have in common? Ridiculously simple.

Logical Database Structures

Stages of Data Modeling

Start at the highest Conceptual Level!

CONCEPTUAL

- ORM
  - Objects
  - Obj. ID’s
  - Roles/Relships
  - (Fnl. Dep)
  - Constraints
  - Sub/SupTypes
  - NO clustering
  - NO “attributes”

ER

- Attrs in Records
- MultiValued,
- Nested - - - - - - -
- Ternaries - - - - - -
- M:N - - - - - - - - -
- Normalized (2,3,4)
- Relationships - - -
- w/attributes

RELATIONAL

- Flat (1NF)
- Binary only
- 1:Many only
- Primary Keys
- Foreign Keys

PHYSICAL

- Implementation in/for a DBMS
- Denormalize (for performance)
  + triggers, stored procedures

DATABASE

SCHEMA

USER

Domain Knowledge

Start at the highest Conceptual Level!
Problems with ER Modeling - Summary

- Cannot capture the "conceptual" view directly, Must mentally map to the "logical" (record-based) view by clustering Attributes into Entity records/tables.
  - Modeler must a priori choose whether Entity or Attribute
  - Too much clustering; attributes in the wrong place
  - Ignores (or presumes normalized) intra-record structure (that is, relationships between/among Attributes)
    - creates (implies) spurious inter-attribute relationships
- Human modeler is responsible for normalization remedy is always record decomposition
- Must choose unique names
  - for attributes in a record; for spurious new "entities"
  - column names = domain + role; lose object domains
- Modeling / Processing dilemma:
  - Complete representation of an entity object - more clustering
  - Full normalization (1NF) – decomposition, more fragmentation
- Indirect representation of M:N relationships
- Difficulty representing Ternary relationships
- Stability of the query language (SQL)

All are solved in ORM!

Why NIAM/OR Modeling?

- roots in both LOGIC & LINGUISTICS
- based on one modeling construct: the fact sentence
  - more expressive, understandable - diagrams & verbalization
- more, richer semantics in the diagram (than E-R, EER, IDEF1X)
- capture and represent all functional dependencies => avoids normalization problems with record-based modeling
- diagrams can be populated with actual data samples
- more stable under changes to the application domain
- abstraction levels equivalent to E-R modeling
- better meets criteria for good data modeling
- organizations that switched wouldn’t go back to E-R
- direction of Standards (SUMM, UDM, Express, OMG-Busn Rules...)
- now supported with a viable PC-based CASE tool
What's Done in ORM?

that is different than in ER/Relational modeling:

- Uses two constructs, OBJECT and ROLE/Relationship, rather than three as in ER: Entity, Attribute, Relationship
- Elementary FACT SENTENCE is the basic construct
  \[
  \text{OBJECT} = \text{Subject, Object (noun)} \\
  \text{RELATIONSHIP} = \text{Predicate (verb)}
  \]
  hence can directly verbalize a data model, including all the constraints in the model diagram
- Represents all DOMAINS directly (Entity, Value)
- Explicitly represents all Relationships including all Functional Dependencies.
Resulting Benefits of ORM

Benefits over all record-based modeling schemes:
(ER, EER, IDEF1X, IE, Barker, Relational, SOM, OO, UML, ...)

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>No need to determine <em>a priori</em> if something is an entity or an attribute (i.e., will have its own Table)</td>
<td></td>
</tr>
<tr>
<td>Avoids the traps of &quot;Table Think&quot;</td>
<td></td>
</tr>
<tr>
<td>Direct, unambiguous representation of higher-order relationships (ternary, ++)</td>
<td></td>
</tr>
<tr>
<td>Can represent more, richer semantics and constraints in a data model diagram</td>
<td></td>
</tr>
<tr>
<td>Can automatically generate the relational data model</td>
<td></td>
</tr>
<tr>
<td>Generates a fully normalized relational data model</td>
<td></td>
</tr>
</tbody>
</table>

NO disadvantages for "conceptual" data modeling
(ER, record-based, high-level "logical" data modeling)

Why Not NIAM?

If NIAM/OR modeling is superior why is it not more popular and widely used; why the lack of support?

- Few academics involved
- CDC kept it proprietary - in Europe and USA
- Lack of strong vendor support
- Too detailed; bottom up design - NOT!
- Lack of support for mental chunking and Levels of Abstraction - NOT!
  - ENTITIES, ATTRIBUTES, VALUES (LOTS), RELATIONSHIPS are all represented the same way. Diagrams should give visual prominence in proportion to semantic importance.
Resources on ORM

**BOOK:**

**Database DESIGN TOOL(s):**
- 'NORMA' - ORM2 tool is under development
  - improved notation

**WEB SITES:**
- [www.orm.net](http://www.orm.net) = Halpin's web site

**COURSES:**
- IDSc 4431 (for U of MN CSOM Undergrads)
- INet 4131 (U of MN College of Continuing Education)
  - Online version is under development

© Gordon C. Everest, *All rights reserved.*