Fact-Oriented Modeling in ERM and FCO-IM

Jan Pieter Zwart
Group Model-Based Information Systems
HAN University of Applied Sciences
HAN University of Applied Sciences

University
33,000 students, 2,500 staff
62 bachelor, 19 master programs in 4 faculties, 26 institutes
39 research groups

Faculty, Institute
Faculty of Engineering. Institute: ICA
Academy of Communication and Information Technology
7 bachelor programs
3 research groups
1,600 students, 120 teaching staff
2016: 650 first-year students
Dean: Ir. Ing. Peter Koburg
HAN University of Applied Sciences

Research group
Model-Based Information Systems (M-BIS)
Headed by: Prof. Dr. Stijn Hoppenbrouwers

Expertise:
- Development of model-based methods and techniques
- Collaborative modeling approaches
- Metadata management
- Business intelligence
- Courses, consultancy
HAN University of Applied Sciences

ICA: 7 programs for Bachelor of ICT, 4 years each.

In year 2:

Semester: Information Systems Engineering

Courses:
- Requirements
- Database Implementation
- Data Modeling and Relational Database Design

Fact-Based ERM: developed 2015-2016 for the last course.
HAN University of Applied Sciences

A few colleagues at the HAN:

Chris Scholten MSc, Senior Lecturer

Dineke Romeijn MSc, Lecturer

Marco Engelbart, MSc, Senior Lecturer
Fact Oriented Modeling: in ERM and FCO-IM
Fact Oriented Modeling: in ERM and FCO-IM
Fact Oriented Modeling: in ERM and FCO-IM
Killing three birds with one stone

Fact-Oriented Modeling: in ERM and FCO-IM
Catching three birds with one net
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling
- Problems in classic ER models
  - Only type level
  - No semantics
  - No method
- Verbalizing example facts helps modelers
- Method to draw up an ER model
- Better FOM technique: FCO-IM and CaseTalk
- Experiences and conclusion
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling
- Problems in classic ER models
  - Only type level
  - No semantics
  - No method
- Verbalizing example facts helps modelers
- Method to draw up an ER model
- Better FOM technique: FCO-IM and CaseTalk
- Experiences and conclusion
Fact-Based vs Attribute Modeling

Central point:

The fact–oriented/based perspective offers a valuable extra viewpoint to supplement the traditional entity / attribute viewpoint.
Fact-Based vs Attribute Modeling

Reservation Request Part: **Attribute/Entity** perspective:

- **Table:** models an **entity type:** a kind of thing in the UoD
- **Columns:** model **attributes:** properties of the entity type

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Dom: RRno</th>
<th>PK, NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Dom: Seqno</td>
<td>PK, NN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Dom: Perfno</td>
<td>NN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Dom: Number</td>
<td>NN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reservation Request</th>
<th>Res. Req. Part</th>
<th>Performance</th>
<th># Seats</th>
</tr>
</thead>
<tbody>
<tr>
<td>3456</td>
<td>1</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td>3456</td>
<td>2</td>
<td>277</td>
<td>6</td>
</tr>
<tr>
<td>5555</td>
<td>1</td>
<td>277</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Focus: the **atoms** of information, not the molecules
Fact-Based vs Attribute Modeling

Reservation Request Part: **Elementary Fact** perspective:

- Table contains **facts**: groups of attributes that belong together

  *Part 1 of res. req. 3456 concerns performance 256.*

  *Part 1 of res. req. 3456 claims 2 seats.*

<table>
<thead>
<tr>
<th>Reservation Request</th>
<th>Res. Req. Part</th>
<th>Performance</th>
<th># Seats</th>
</tr>
</thead>
<tbody>
<tr>
<td>3456</td>
<td>1</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td>3456</td>
<td>2</td>
<td>277</td>
<td>6</td>
</tr>
<tr>
<td>5555</td>
<td>1</td>
<td>277</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Focus: the **molecules** of information, not the atoms
Fact-Based vs Attribute Modeling

A few pros and cons of these perspectives:

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
<th>Elementary Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Easy for trivial properties</td>
<td>• No natural units of info</td>
<td>• Natural units of info</td>
</tr>
<tr>
<td>• Techniques widespread</td>
<td>• Impracticable for complex data structures</td>
<td>• Good for complex data str.</td>
</tr>
<tr>
<td>(ERM, UML, …)</td>
<td>• Semantics not included</td>
<td>• Semantics clear</td>
</tr>
<tr>
<td>• Many big software tools</td>
<td>• No good method</td>
<td>• Good methods</td>
</tr>
<tr>
<td></td>
<td>• Metamodel clumsy</td>
<td>• Metamodel simple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cons</td>
<td>• More elaborate</td>
<td>• More elaborate</td>
</tr>
<tr>
<td>• No natural units of info</td>
<td>• Techniques in niche</td>
<td>• Techniques in niche</td>
</tr>
<tr>
<td>• Impracticable for complex</td>
<td>• Semantics not included</td>
<td>(FCO-IM, ORM, CogNIAM)</td>
</tr>
<tr>
<td>data structures</td>
<td>• No good method</td>
<td>• Few supporting tools</td>
</tr>
<tr>
<td>• Semantics not included</td>
<td>• Metamodel clumsy</td>
<td>(CaseTalk, NORMA, …)</td>
</tr>
<tr>
<td>• No good method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Metamodel clumsy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling

- Problems in classic ER models
  - Only type level
  - No semantics
  - No method

- Verbalizing example facts helps modelers

- Method to draw up an ER model

- Better FOM technique: FCO-IM and CaseTalk

- Experiences and conclusion
Problems in classic ER models

Fact-oriented modeling aims to expand classic ERM with:

- Fact-based perspective
- Semantics
- Instance level
- Systematic technique

Here’s why:
Problems in classic ER models

Is this model correct?

At least some semantics is modeled: the meaning of the RT is clear

But Salary (per year? per month)?

What is Area? Size? Part of building?

Abstract model: difficult to check.
Types and instances

**Type level**

**EMPLOYEE**
- **Eno**<br>- **Ename**<br>- **Salary**

**ROOM**
- **Rno**<br>- **Area**

has a desk in

contains a desk of

**Instance level**

**Employee**
- E1, John
- E2, Lisa
- E45, John
- E68, Harry
- E55, Richard

**Room**
- 56
- 67
- 88
Semantics, types and instances

A classic ERM diagram shows only the type level

This suffices for simple everyday ETs and Atts (but many Atts are not simple at all)

However, for unfamiliar contexts and/or complex data structures this is not enough to understand the model

Adding the **semantics** (meaning) and **examples of instances** to the diagram can greatly help to validate the model (is it correct?)
Semantics, types and instances

Type level

Fact-Based ERM diagram with predicates and populations

Predicates: represents exactly one type of fact

Semantics and instance level

Workspace contains a desk of

EMPLOYEE

Eno <pi> EMPNO <M>
Ename NAME <M>
Salary MONEY <M>

has a desk in

ROOM

Rno <pi> ROOMNO <M>
Area AREA

Predicate: represents exactly one type of fact

Employee <Eno> is called <Ename>. Employee <Eno> earns a salary of € <Salary> per month.

E1
John 3000
E2
Lisa 5000
E45
John 2400
Semantics, types and instances

Type level

Semantics and instance level

Fact-Based ERM diagram with predicates and populations

Predicate: represents exactly one type of fact
Semantics, types and instances

Type level

Fact-Based ERM diagram with predicates and populations

Predicate: represents exactly one type of fact

Fact type:
either \(<\pi>+Att\) combination
or non-dependent RT

Population: concrete illustration
Substitute values into blanks

In practice: do this only
for unclear Atts and RTs
Types and instances, weak ET

Here is a simple example of a weak ET (only one <pi>+Att fact type is shown)

In complex data structures (like branching chains of weak ETs), a predicate and example population can clarify much

Note: a dependent RT cannot have a predicate or population
Problems in classic ER models

Three main problems with classic ERM:

- Only abstract type level is modeled
  Impossible to validate abstract model

- Semantics (of complete facts) not modeled
  Data Dictionary: absent, or only ET and Att.
  Semantics highly valued in practice

- No good modeling method
  Most textbooks show WHAT to model
  No textbook shows HOW to model

Here: attempt to solve all problems using verbalizations of concrete examples of facts
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling
- Problems in classic ER models
  - Only type level
  - No semantics
  - No method
- Verbalizing example facts helps modelers
- Method to draw up an ER model
- Better FOM technique: FCO-IM and CaseTalk
- Experiences and conclusion
Examples of verbalizations

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>Description</th>
<th>Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT P315</td>
<td>Update homepage Treasury Bank</td>
<td>InsEd</td>
</tr>
<tr>
<td>1</td>
<td>Elicit requirements</td>
<td>InsEd</td>
</tr>
<tr>
<td>2</td>
<td>Improve firewall</td>
<td>WndlA</td>
</tr>
<tr>
<td>3</td>
<td>Add new functionality</td>
<td>BsBg</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROJECT P422</td>
<td>Build DWH for OnlineHaberdashery</td>
<td>SmthE</td>
</tr>
<tr>
<td>1</td>
<td>Determine scope</td>
<td>HkstJa</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Employee InsEd manages project P315.

The description of subproject 2 of project P315 is: Improve firewall.

Subproject 2 of project P315 is led by employee WndlA.
Why use verbalizations of facts?

Verbalizations of elementary facts:

• Are on the concrete instance level
  Domain expert and modeler: common ground
  Validation by domain expert is easy

• Capture the semantics of the data
  Main issue in practice (>60% of design time)

• Are independent of modeling technique
  Do not change in model transformations:
  ORM, ERM, UML, Rel, …: same verbalizations

• Offer a valuable alternative viewpoint
  Natural units of information
How do verbalizations help a modeler?

Verbalizing concrete examples of facts:
• Makes the modeler understand the data
• Is done in constant dialogue between modeler and domein expert no ‘ivory tower’ modeling
• Enables an arcane abstract ER model to be built from familiar concrete facts
• Leads to a good and simple method to draw up an ERM diagram
• Enables easy validation of the model
• Enables adding semantics and examples to the diagram itself where convenient
Fact-Oriented Modeling (FOM)

• Fact-based vs Attribute-based modeling
• Problems in classic ER models
  • Only type level
  • No semantics
  • No method
• Verbalizing example facts helps modelers
• Method to draw up an ER model
• Better FOM technique: FCO-IM and CaseTalk
• Experiences and conclusion
Method

Plenty of ERM textbooks tell you WHAT to model

No ERM textbook tells you HOW to make a good model

Fact Oriented/Based Methods (FCO-IM, ORM, CogNIAM) have always provided a good method
Procedure to draw up an ERD

1. Collect concrete examples of facts
   - Use BPM as starting point
   - Make up examples if they don’t exist (yet)
2. Verbalize these examples
   - With domain expert. Result: fact expressions.
   - Make the meaning as clear as possible
3. Sort expressions into Fact Types (FTs)
   - Same kind of expression: same FT
   - Order FTs with most components last
4. Analyze each FT (2 segments) and add the results to the ERD

Steps 1 and 2 are not covered in this presentation.
Starting point

A process model shows data stores and flows: good sources of concrete examples of facts
Sorting fact expressions

Expressions of the same kind belong to a Fact Type.

Expressions of the same type have components: places where the text can vary.

FT4 has 2 components

FT6 has 3 components

FT4
Employee "InsEd" manages project "P315", "SmthE" "P422".

FT6
Subproject 1 of project "P315" starts on "20160201", "P315" "20160201", "P315" "20160208", "P422" "20160101".
Sorting fact expressions

Procedure for sorting:

- Place expressions of the same kind in the same Fact Type (FT)
  - 2 or 3 expressions per FT is enough
- Per FT: count the number of components
  - Component: place where text can vary
  - Highlight the components
- Order the FTs
  - FTs with the fewest components: first
  - FTs with the most components: last
Analyzing fact types

No matter how many components a FT has, it can have **only 2 segments**: groups of components that belong together (only 1 segment is also possible).

There are only two possibilities for the 2 segments:
- One segment concerns an ET, the other segment concerns an Att of this ET
- Both segments concern ETs, with a mutual RT

There is only one possibility for a FT with 1 segment:
- The segment concerns an ET

Analyzing fact types is:
  determining which segments there are, and which ETs, Atts and RTs are involved.

The cases with 2 segments are treated in slides 9-18.

The cases with 1 segment are treated in slides 19-21.
Analyzing fact types

The procedure to analyze FTs will be illustrated using the following four FTs:

FT1:
The family name of student S17 is Johansen.
   T66    Robberts.

FT2:
The course SQL is taught by Tmina.
   ERM    Ttigo.

FT3:
The exam of the course SQL on 14/1/2016 is held in room R67.
   ERM    25/2/2016 45a.

FT4:
Student T66 scored a mark of 85 for the exam of SQL on 14/1/2016.
   S17    47    ERM    25/2/2016.

All modeling decisions are discussed with domain experts.
Analyzing fact types: FT1 (ET+Att)

Two components.
Segments underlined.
Segments: ET + Att.
Meaningful names.

Identifier of STUDENT: S17 and T66 are student numbers, which are called ‘Studno’ according to the domain expert.

ERD
The <pi> and <M> were checked with the domain experts. Domains for the Atts were specified also.

For each ET: establish its <pi> (if Att: always <M>)

FT1
The family name of student S17 is Johansen.
" " " " T66 " Robberts.
ET STUDENT Att Family_name

Predicate: The family name of student <Studno> is <Family_name>.

<table>
<thead>
<tr>
<th>STUDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studno</td>
</tr>
<tr>
<td>Family_name</td>
</tr>
<tr>
<td>STUDNO</td>
</tr>
<tr>
<td>NAME</td>
</tr>
</tbody>
</table>
Rules for analyzing FTs

- Mark 2 segments (or 1), and decide on ET + Att or ET + ET (if 1 segment: ET).
- If you find a new ET: determine its ID (primary identifier)
- Give the complete predicate
- Determine <M> for new Atts
Analyzing fact types: FT2 (ET+ET)

Two components.
Segments: ET + ET
Meaningful names
Identifier determined

RT: explicit notation with ET-names needed in general

ERD
All constraints, domains and cardinalities were determined with the domain experts
Rules for analyzing FTs

• Mark 2 segments (or 1), and decide on ET + Att or ET + ET (if 1 segment: ET).

• If you find a new ET: determine its ID (primary identifier)

• In the ET + ET case: add a non-dependent RT

• Give the complete predicate
• Determine <M> for new Atts
• Determine cardinalities for new RTs
Analyzing fact types: FT3 (weak ET)

FT3:
The exam of the course SQL on 14/1/2016 is held in room R67.  

ET EXAM
ID: ET COURSE + Att Date
MATCH

Old ET: MATCH

ID contains ET: EXAM is weak ET:
add RT with dependency

RT COURSE_of_EXAM between EXAM(dependent) and COURSE

Predicate: The exam of the course <Course_code> on <Date> is held in room <Room>.

Could also be ET, if Atts for rooms were to be recorded, or a domain list would be convenient.
Rules for analyzing FTs

- Mark 2 segments (or 1), and decide on ET + Att or ET + ET (if 1 segment: ET).
- If you find an old ET: MATCH
- If you find a new ET: determine its ID (primary identifier)
- If this ID contains an ET: add a dependent RT to it
- In the ET + ET case: add a non-dependent RT
- Give the complete predicate
- Determine <M> for new Atts
- Determine cardinalities for new RTs
Analyzing fact types: FT4 (Complex)

FT4:
Student T66 scored a mark of 85 for the exam of SQL on 14/1/2016.
" S17 " " " " 47 " " " " ERM " 25/2/2016.

Predicate: Student <Studno> scored a mark of <Mark>
for the exam of <Course_code> on <Date>.

Old ETs STUDENT and EXAM present. Mark: attribute (property) of an exam participation. So other three components must be one ET.

ID contains 2 old ETs: 2 MATCHes

For each ET in the ID: add a dependent RT
Analyzing fact types: Complete ERD

Conceptual Data Model
Model: Students, exams and marks
Package: 
Diagram: ERM part 2
Author: J.P. Zwart Date: 11-8-2016
Version: 2

Student Studno scored a mark of Mark for the exam of Course_code on Date. 

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.

T66 S17
85 47
SQL ERM
14/1/2016 25/2/2016

Student Studno scored a mark of Mark for the exam of Course_code on Date.
Rules for analyzing FTs

• Mark 2 segments (or 1), and decide on ET + Att or ET + ET (if 1 segment: ET).
• If you find an old ET: MATCH
• If you find a new ET: determine its ID (primary identifier)
• If this ID contains an ET: add a dependent RT to it
• In the ET + ET case: add a non-dependent RT
• Give the complete predicate
• Determine <M> for new Atts
• Determine cardinalities for new RTs
• Add predicates and populations to the diagram to make the meaning of the fact types more clear
Examples of FTs with one segment

Example 1: Domain list

Such verbalizations might be given for domain lists (departments in an organization, wards in a hospital, towns in a country, ...).

Domain lists prevent typos, save users time and effort, and are easily updated by the DB admin.

There is a course ERM.

"""""""" SQL.

Only one component, only one segment possible. This must then be an ET.

There is a course ERM.

"""""""" SQL.

ET COURSE

ID: Att Course_code

Predicate: There is a course <Course_code>.

<table>
<thead>
<tr>
<th>COURSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course_code  &lt;pi&gt;  C_CODE  &lt;M&gt;</td>
</tr>
</tbody>
</table>
Examples of FTs with one segment

Example 2: Empty weak ET

Student S17 has enrolled for the course ERM.

```
  "   T66   " "     "     "   "   SQL.
```

Two components, only 1 segment chosen: must be ET.

Student S17 has enrolled for the course ERM.

```
  "   T66   " "     "     "   "   SQL.
```

ET ENROLLMENT
ID: ET STUDENT + ET COURSE
MATCH MATCH

RT R_STUDENT_in_ENROLLMENT between ENROLLMENT(dependent) and STUDENT

RT R_COURSE_in_ENROLLMENT between ENROLLMENT(dependent) and COURSE

Predicate: Student <Studno> has enrolled for the course <Course_code>.

Suppose you know that enrollments have attributes of their own (date, status, …). Then you don’t want to treat this as an ET+ET case: it will result in a Many-to-Many RT.

Instead, an empty ET for the future Atts is desired.
Examples of FTs with one segment

Example 2: Empty weak ET

Note:

• Attributes for ENROLLMENT can be easily added: when analyzing a verbalization like:

  The status of student S17’s enrollment in the course ERM is: Pending.

  the ET ENROLLMENT is old, so MATCH will do.

• The rules given in slide 18 also capture the one-segment cases.
Practical recommendations

• Always work exclusively from concrete examples of facts.
• Always verbalize these facts carefully, with the possible exception of widely known simple attributes, but don’t be too sloppy!
• Add predicates and/or example populations for
  ➢ all unclear non-dependent RTs
  ➢ all unclear <pi>+Att fact types
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling
- Problems in classic ER models
  - Only type level
  - No semantics
  - No method
- Verbalizing example facts helps modelers
- Method to draw up an ER model
- Better FOM technique: FCO-IM and CaseTalk
- Experiences and conclusion
FOM technique: FCO-IM

Fact Oriented Modeling with FCO-IM

Published 2015
FOM technique: FCO-IM
Tool: CaseTalk

Marco Wobben
FOM technique: FCO-IM

- FCO-IM uses the same principles
  - Focus: complete elementary facts
  - Model is built by analyzing verbalizations of example facts
- Method more fully worked out
  - Verbalizations incorporated
  - Many constraint types included
    (uniqueness, totality, cardinality, subset, …)
FCO-IM model (in CaseTalk)
FCO-IM model (in CaseTalk)

Type level
Instance level
Verbalizations

O2: 'the course <6>'
1: SQL
2: ERM

O3: '<7>'
1: Ttigo

Course_teacher
F2: '<4> is taught by <5>'
1: SQL Ttigo
2: ERM Ttigo

Course_code
O2: 'the course <6>'

Teacher_code
O3: '<7>'

Course_code
O2: 'the course <6>'

<4> is taught by <5>
the course <6> is taught by <7>
the course ERM is taught by Ttigo
FOM technique: FCO-IM
Tool: CaseTalk

• **Automatic transformation of FCO-IM model into**
  • ERM data model
  • UML class diagrams
  • Relational database schema
  • DWH Star Schema
  • Data Vault
  • ...

• **Script generation**
  • Several RDBMS platforms
ERM model (derived in CaseTalk)

Student
- Studno
- Family_name

Exam Participation - Student

Exam Participation
- Mark

Exam
- Date
- Room

Course
- Course_code

Teacher
- Teacher_code

Course_teacher

Exam - Course

Participation - Exam

Derivation case studies in CaseTalk
ERM model (derived in CaseTalk)

- Entity type
- Attributes
- Relationship type

Shown or hidden ad libitum:

- Predicate
- Example facts

**STUDENT**

<table>
<thead>
<tr>
<th>Studno</th>
<th>Family_name</th>
</tr>
</thead>
</table>

**EXAM_PARTICIPATION**

<table>
<thead>
<tr>
<th>Mark</th>
</tr>
</thead>
</table>
| F4: Student <Studno> scored a mark of <Mark> for the exam of the course <Course_code> on <Date>.
| F4: Student T66 scored a mark of 85 for the exam of the course SQL on 14/1/2016.
| F4: Student S17 scored a mark of 47 for the exam of the course ERM on 25/2/2016. |
Relational schema (derived in CaseTalk)

STUDENT

Student
Family_name

EXAM_PARTICIPATION - STUDENT

EXAM_PARTICIPATION

Course_code
Date
Student
Mark

F4: Student <Student> scored a mark of <Mark> for the exam of the course <Course_code> on <Date>.

F4: Student T65 scored a mark of 85 for the exam of the course SQL on 14/1/2016.

F4: Student S17 scored a mark of 47 for the exam of the course ERM on 25/2/2016.

TEACHER

Teacher_code

Course_teacher

Course_teacher - TEACHER

Course_code

Course_code

Teacher_code

EXAM

Date
Course_code
Room

EXAM_PARTICIPATION - EXAM

EXAM - COURSE

COURSE

Course_code
Fact-Oriented Modeling (FOM)

- Fact-based vs Attribute-based modeling
- Problems in classic ER models
  - Only type level
  - No semantics
  - No method
- Verbalizing example facts helps modelers
- Method to draw up an ER model
- Better FOM technique: FCO-IM and CaseTalk
- Experiences and conclusion
Experience with this approach in class

- Procedure: can be taught and practised well in class
- Case studies (hospital, music theater, travel agency):
  - Students were only allowed to continue if the verbalizations were approved by the domain expert (teacher)
  - Verbalizing takes time
  - Students understand UoD better:
    - Less jumping to (wrong) conclusions, and misunderstandings corrected quickly
    - Excellent way to solve semantic issues
      - Analyzing and drawing up the ERM diagram was easy after this
- Students: appreciate the ‘best of both worlds’ approach
  - For trivial attributes: why the fuss?
  - More difficult modeling: benefit is acknowledged
Conclusions

• Fact-Based viewpoint: valuable additional perspective
  – Exactly one complete fact (natural unit of information)
  – vs Entity Type (cluster of facts) or Attribute (fact fragment)

• Verbalizations of elementary facts can be used to
  – Supplement a classic ER model where convenient with instance-level examples to add clarification by illustration
  – Supplement a classic ER model where convenient with elementary fact predicates to add semantics
  – Draw up an ER model using a systematic easy-to-learn procedure telling you how to do so
Catching three birds with one net

Thank you for your attention