Mapping Entities to Objects

Francois C. Cartier

Senior Consultant – e-Modelers, Inc.

DAMA - San Francisco Bay Area Chapter
Wednesday January 9, 2008

Key Notions

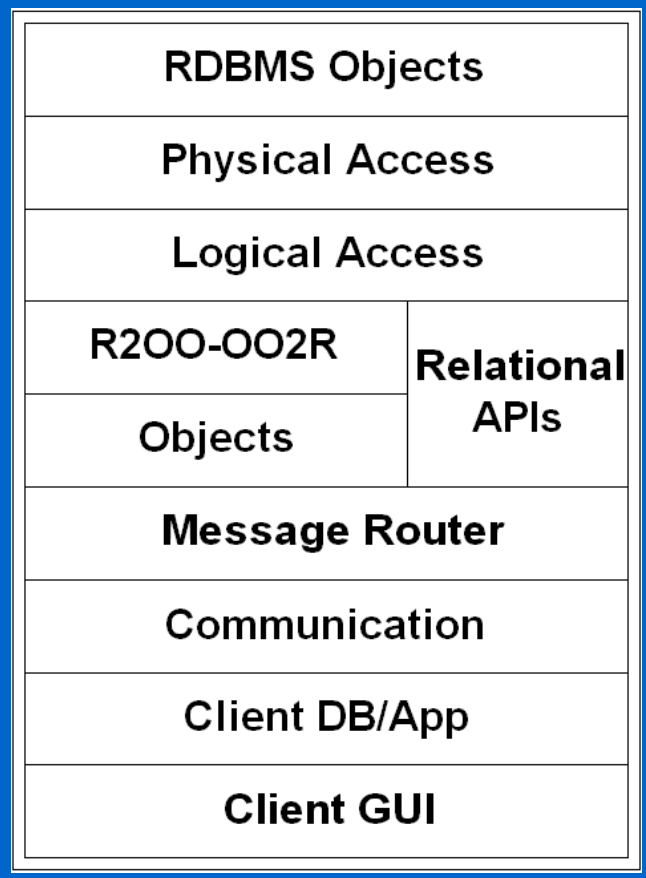
- **The context: ROOLSA**
- **Entity Types**
- **Entity Key Dimensions**
- **Generate from the Modeling Tool (directly or indirectly)**

ROOLSA

- **Relational Object Oriented System Architecture**
- **Nine Layers from DBMS to UI**
- **Layers = Encapsulation**
- **Data Driven Functionality**

The Nine Layers of ROOLSA

- 1. Tables, Views…
- 2. SQL, S/P, Triggers…
- 3. Logical Ops
- 4. Translation
- 5 & 6. Methods
- 7. Asynch queues
- 8. Client or Web App
- 9. GUI or Browser



Entity Types

- **R-OO mapping at the Logical level**
- **R model MUST be 4th NF or higher for clear entity typing – denormalization adds substantial complexity**
- **R to OO: not a 1 to 1 relationship**
- **Business object vs service object**
- **FAR-ACME method**

The FAR-ACME method

- **Fundamental Entity**
- **Associative Entity**
- **Reference Entity**
- **Attributive Entity**
  - Characteristic Entity
  - Multi-Value Entity
  - Exception Entity

The Fundamental Entity

- **One to one with a business object**
- **Contains only intrinsic attributes**
- **Home to the unique identifier (not OID!)**
- **Carries the OID (OID is not the PK!)**

The Associative Entity

- **Represents a M-M relationship**
- **If it is not a Dependent Business Object (e.g. an Order), then:**
- **One to one with a Service Object**
- **Services parent objects**
- **UNLESS it represents a rule! (association between reference entities)**

The Reference Entity

- **Data driven validation rule implemented through RI**
- **Substitution by code**
- **Special cases: Types, Rules, and Roles**
- **Reference+: control flags and recursiveness**
- **May evolve into a fundamental entity**

The Attributive Entity

- **Part of the same Business/Service Object as the parent entity**
- **Comes in many flavors**
- **Results often in O-O attribute lists and matrices**

The Characteristic Entity

- **An Attributive Entity flavor**
- **PK's only FK is from it's parent Fundamental Entity**
- **Represents variable set of object characteristics**
- **The "Entity In Time" carries the Business Object's status**

The Multi-Value Entity

- **Another attributive entity flavor**
- **Contains attributes that can have simultaneously more than one value for one Fundamental, or Associative, parent instance**
- **If these attributes are not intrinsic, a time attribute has to be part of the PK**

The Exception Entity

- **Another Attributive Entity flavor**
- **Contains the attributes of a fundamental, associative, or characteristic entity**
  - that have a value on rare occasions, or
  - for which sensitivity requires segregation

Subtypes and Supertypes

- **Subtyping = Subclassing**
- **Entity Types of a Supertype and its Subtypes are the same, regardless of whether the subtyping is exclusive (type) or non-exclusive (role)**
- **Relational model does not prevent an entity from having multiple supertypes**

Entity Dimensions

- **What are entity dimensions?**
- **The overall guiding paradigm**
- **A multi-dimensional example**
- **How does it apply to Objects?**
- **Another kind of “Dimension”**

What are dimensions?

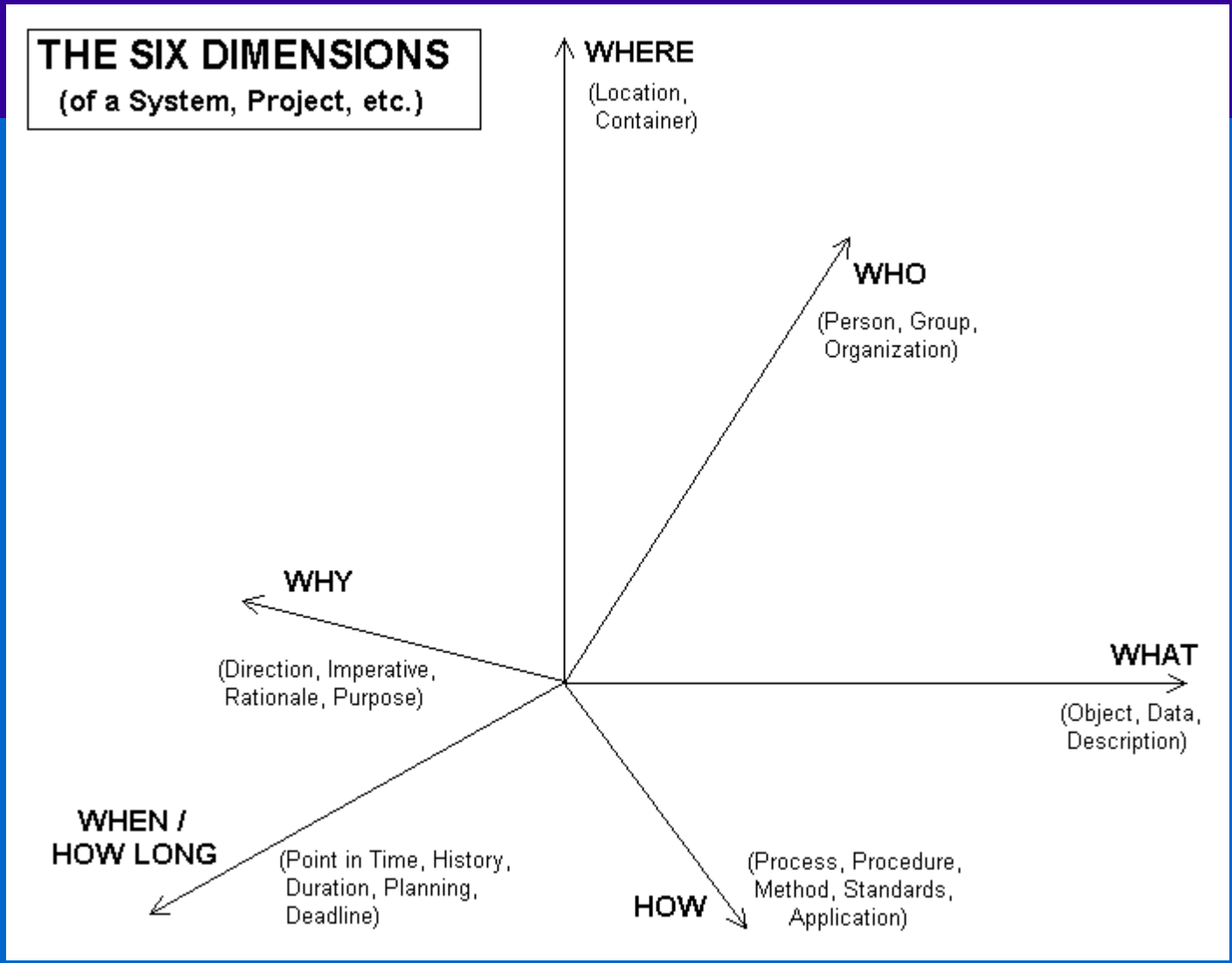
- **A general definition**
- **The 2nd dimension: Time**
- **The 3rd dimension: Location**
- **The 4th dimension: Data Source**
- **The 5th dimension: Method**
- **The 6th dimension: Purpose**
- **… and a half: Value**

What is a dimension?

- **A column in the Zachmann framework**
- **It arrays information**
- **Dimension attribute(s) in PK**
- **Dimension attribute may be an FK**

Paradigm: The Six Dimensions

- **Answers the "Who?, What?, Where?, When?, Why?, How?" questions**
- **Can be applied to anything (a system, a project, etc.)**
- **Each dimension can be re-applied n times (the nature of the attribution or relationship is different each time)**


THE SIX DIMENSIONS
(of a System, Project, etc.)
WHERE
(Location, Container)
WHO
(Person, Group, Organization)
WHY
(Direction, Imperative, Rationale, Purpose)
WHAT
(Object, Data, Description)
WHEN / HOW LONG
(Point in Time, History, Duration, Planning, Deadline)
HOW
(Process, Procedure, Method, Standards, Application)

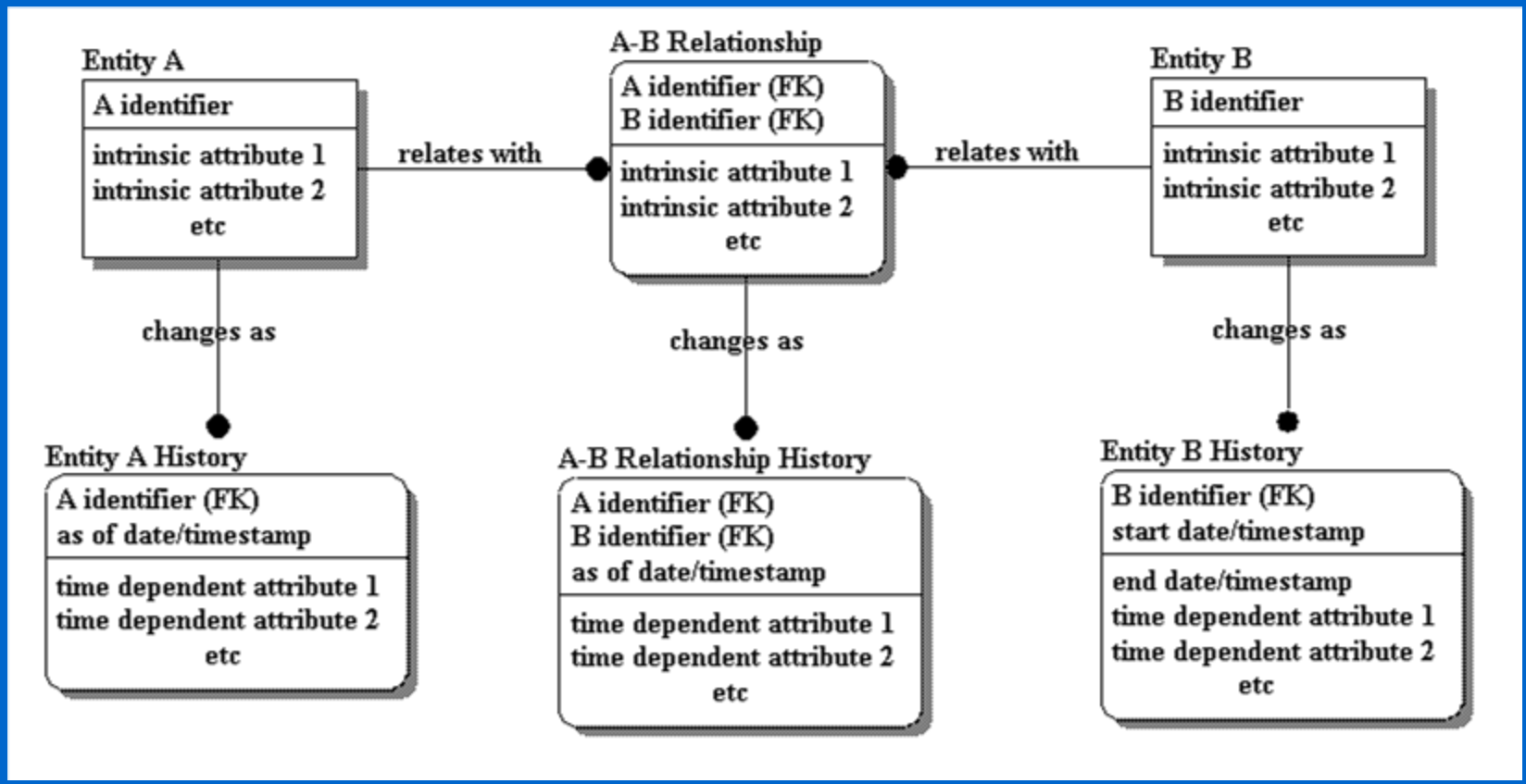
The 2nd Dimension: Time/History

- **Wait: what was the 1st dimension again?**
- **Intrinsic Attributes:**

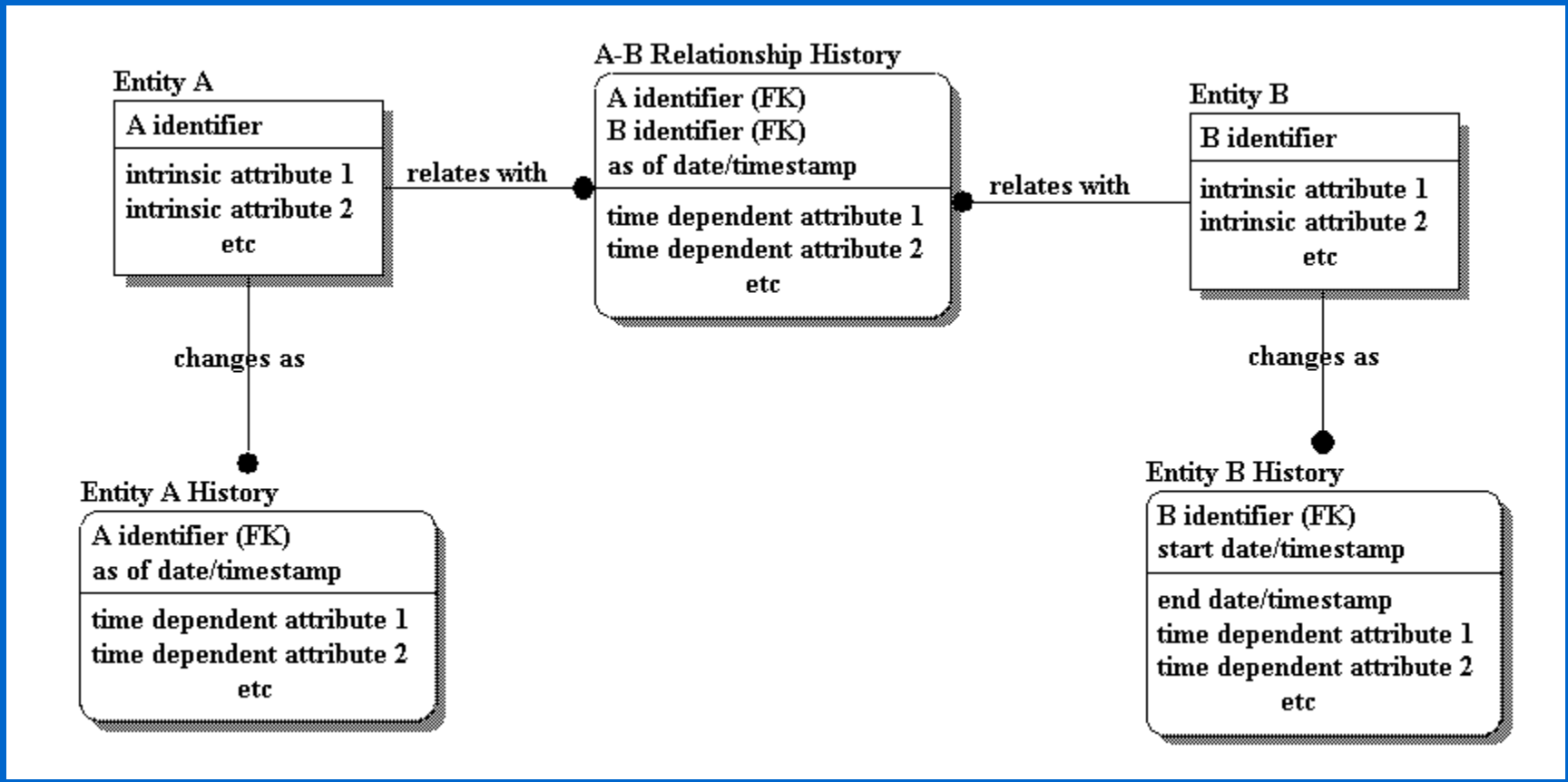
  values are defined once for an entity instance; once set, they never change throughout the entity instance's lifetime
- **Includes the past as well as the future**
- **Time-dependent Attributes:**

  values may change during the entity instance's lifetime

Time Dimensioning - History



History - A Common Simplification



Use “as of date/timestamp” when:

- **There are no gaps between intervals**

OR

- **The date/timestamp corresponds to a snapshot (state/values between snapshots is not known)**

Use “start date/timestamp” and “end date/timestamp” when:

- **Gaps between time intervals may exist,**

AND

- **The state/values are considered current till the end date or the start date of the next row.**

History - A Dilemma: New Values or Deltas?

New Values:

- **easy to query**
- **maintenance process is complex if time intervals can be inserted prior to existing ones and unrelated time dependent attributes are kept in the same table**

History - A Dilemma: New Values or Deltas?

Deltas:

- **queries have to recalculate current values each time**
- **maintenance process is simplified**

History - Another Dilemma: Inclusive or Exclusive?

As of date: inclusive or exclusive?

Start date / end date

- **inclusive/inclusive?**
- **inclusive/exclusive?**
- **exclusive/inclusive?**
- **exclusive/exclusive?**

History - Another Dilemma: Inclusive or Exclusive?

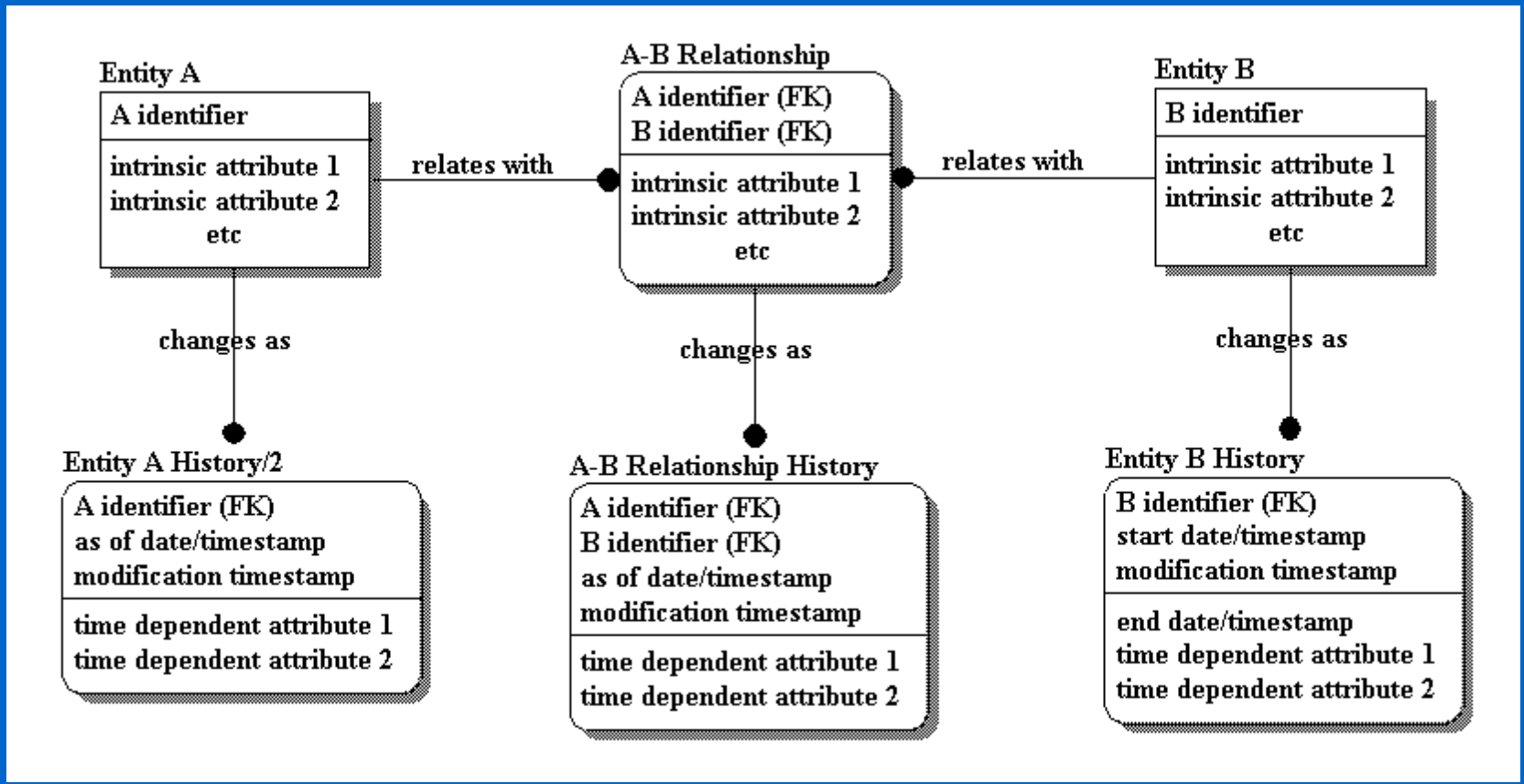
Hint:

- **The scheme used should be consistent between as of dates / timestamps and start dates / end dates throughout the data model and the data bases.**

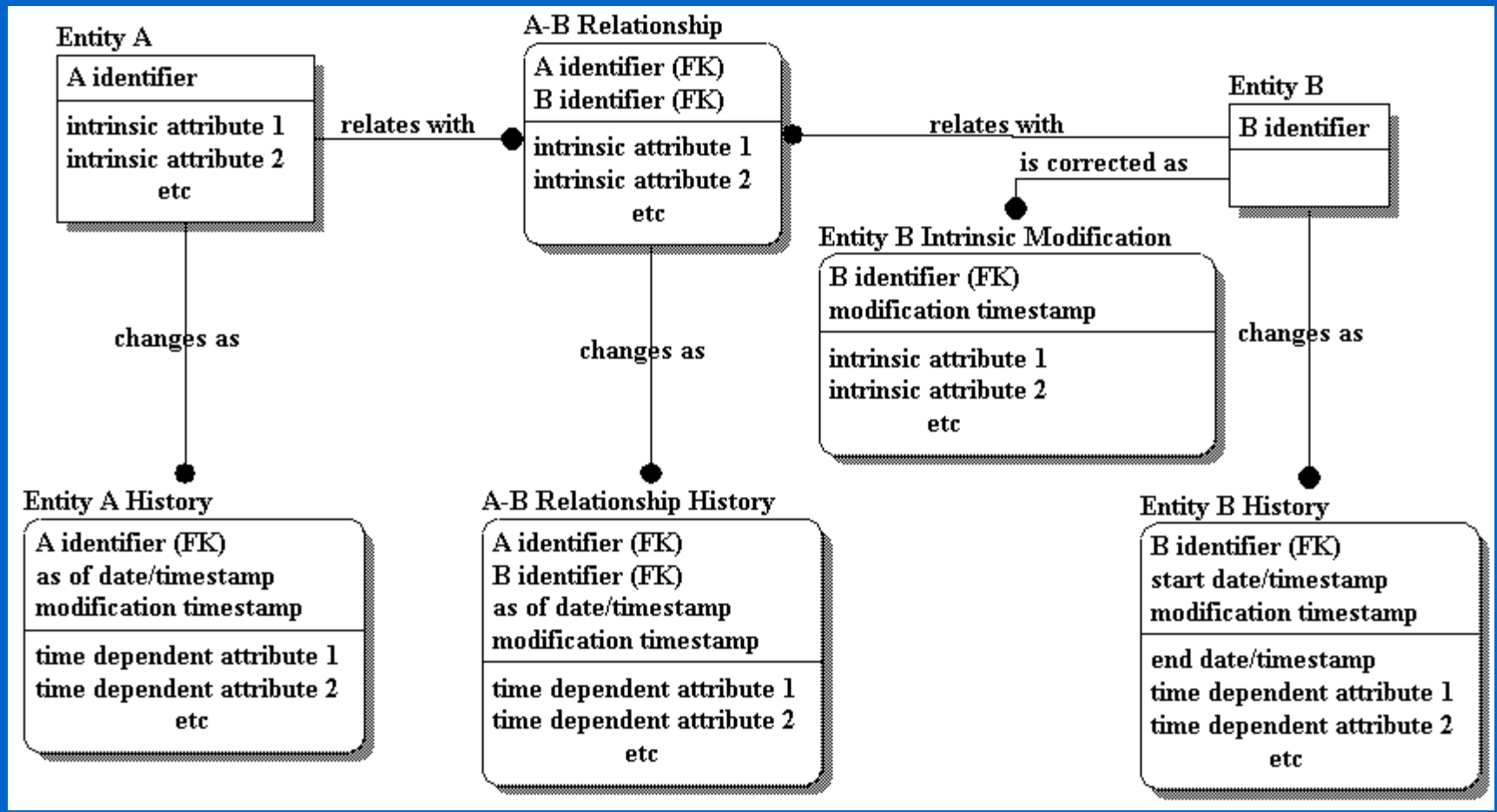
The Other 2nd Dimension: History Maintenance

- **“This bill shows a payment of $13.25 received last month; actually the check amount is $1,325.00!”**
- **“This stock’s closing price of 4 days ago was misquoted; it should have been $43.65 per share, not $46.35!”**
- **Keeps track of corrections to data**

Time Dimensioning - History Maintenance



Time Dimensioning - Entity Maintenance



Aspects of the 2nd Dimension

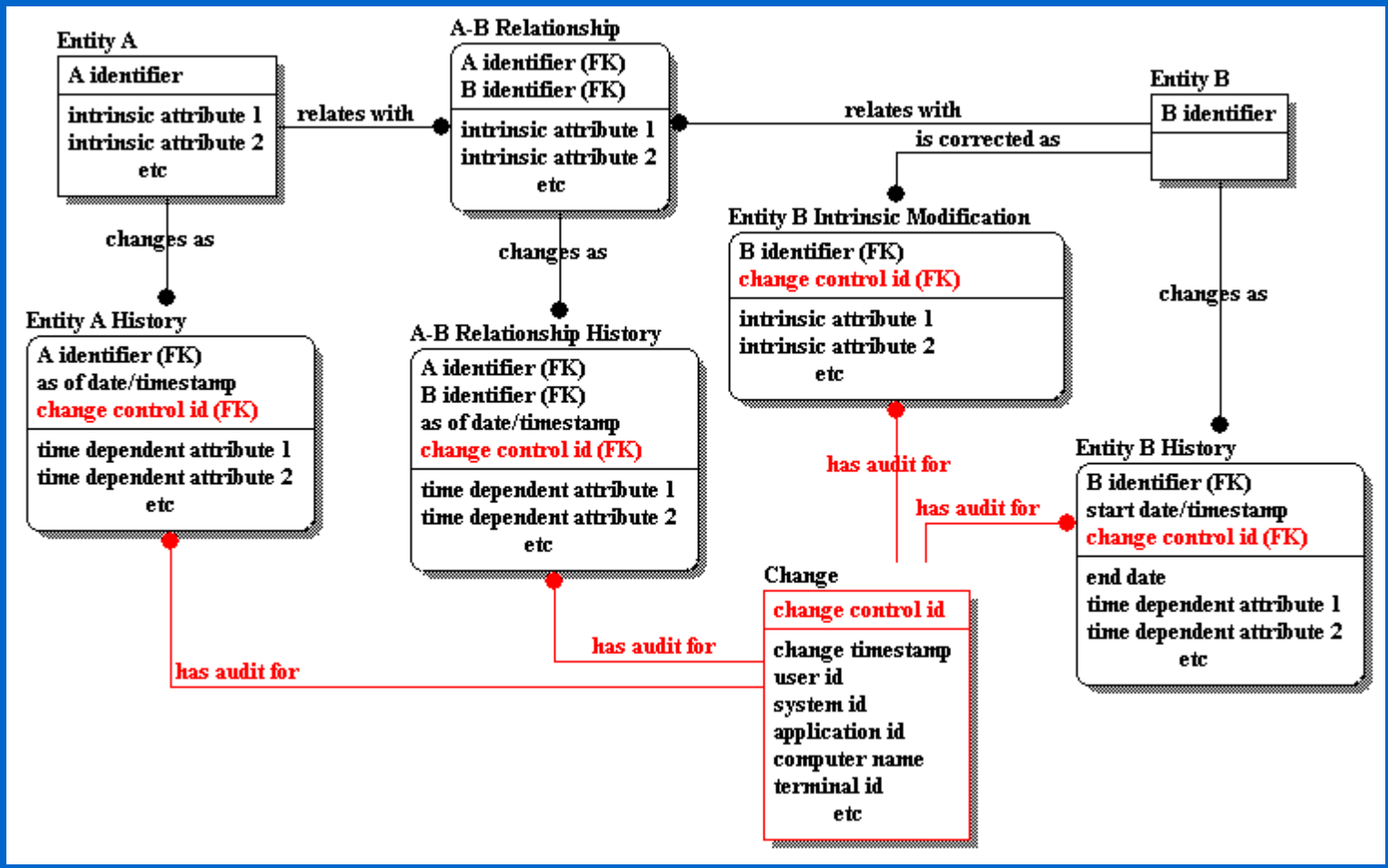
Shadowing

- **Before/after change image**

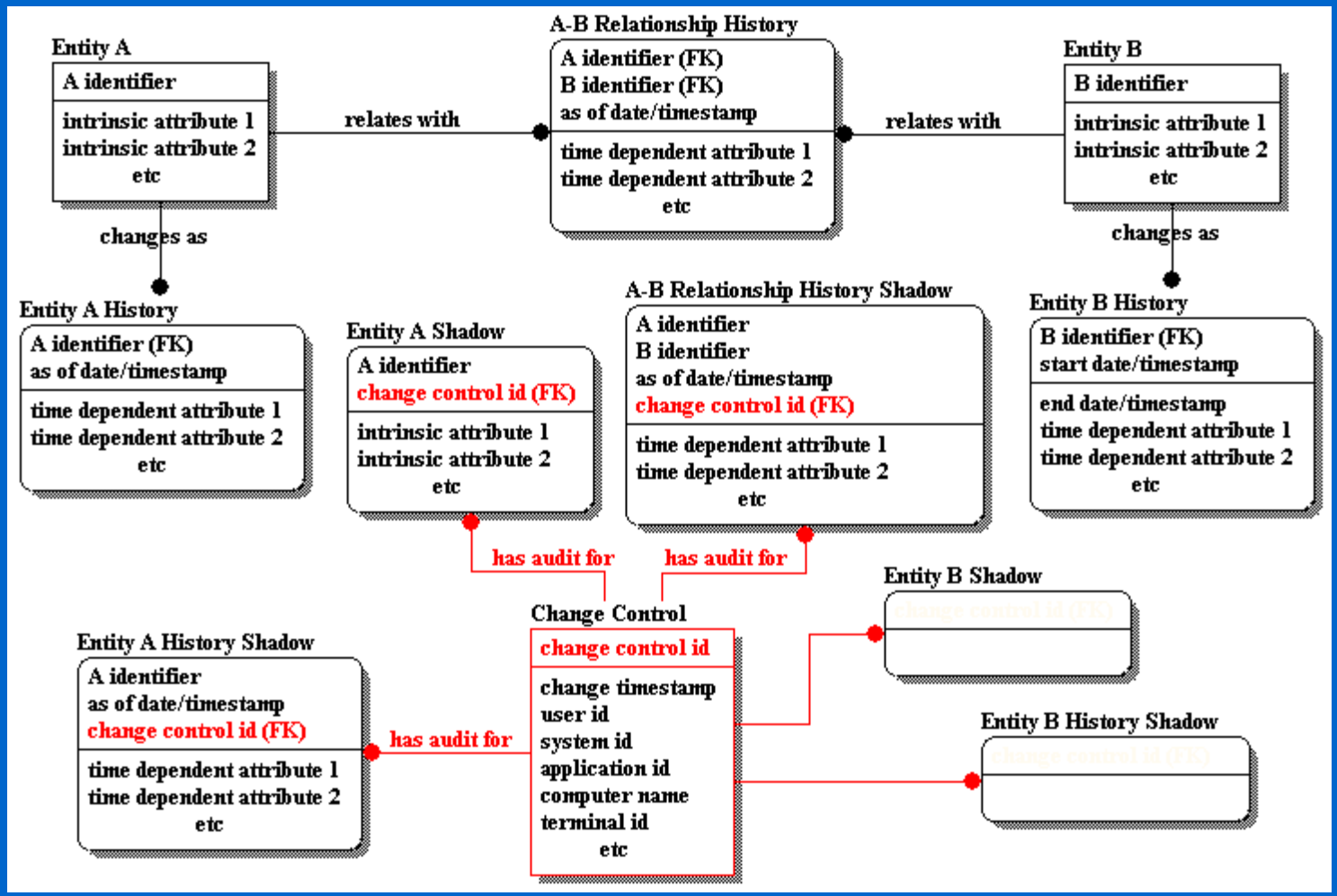
Change Control

- **How is it done?**
- **How to hide it?**

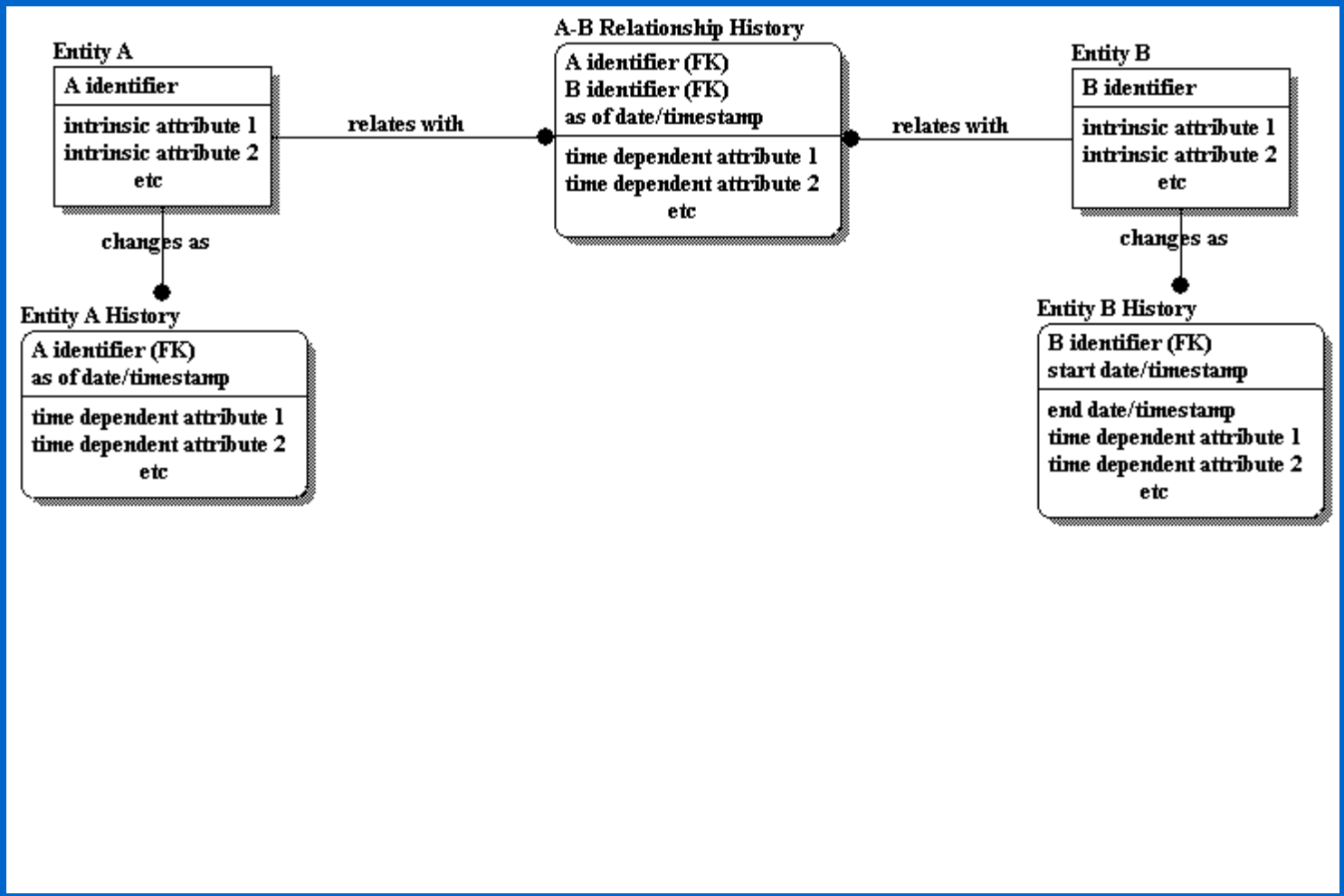
Time Dimensioning - Change Control



Entity Shadowing with Change Control



Entity Shadowing with Change Control



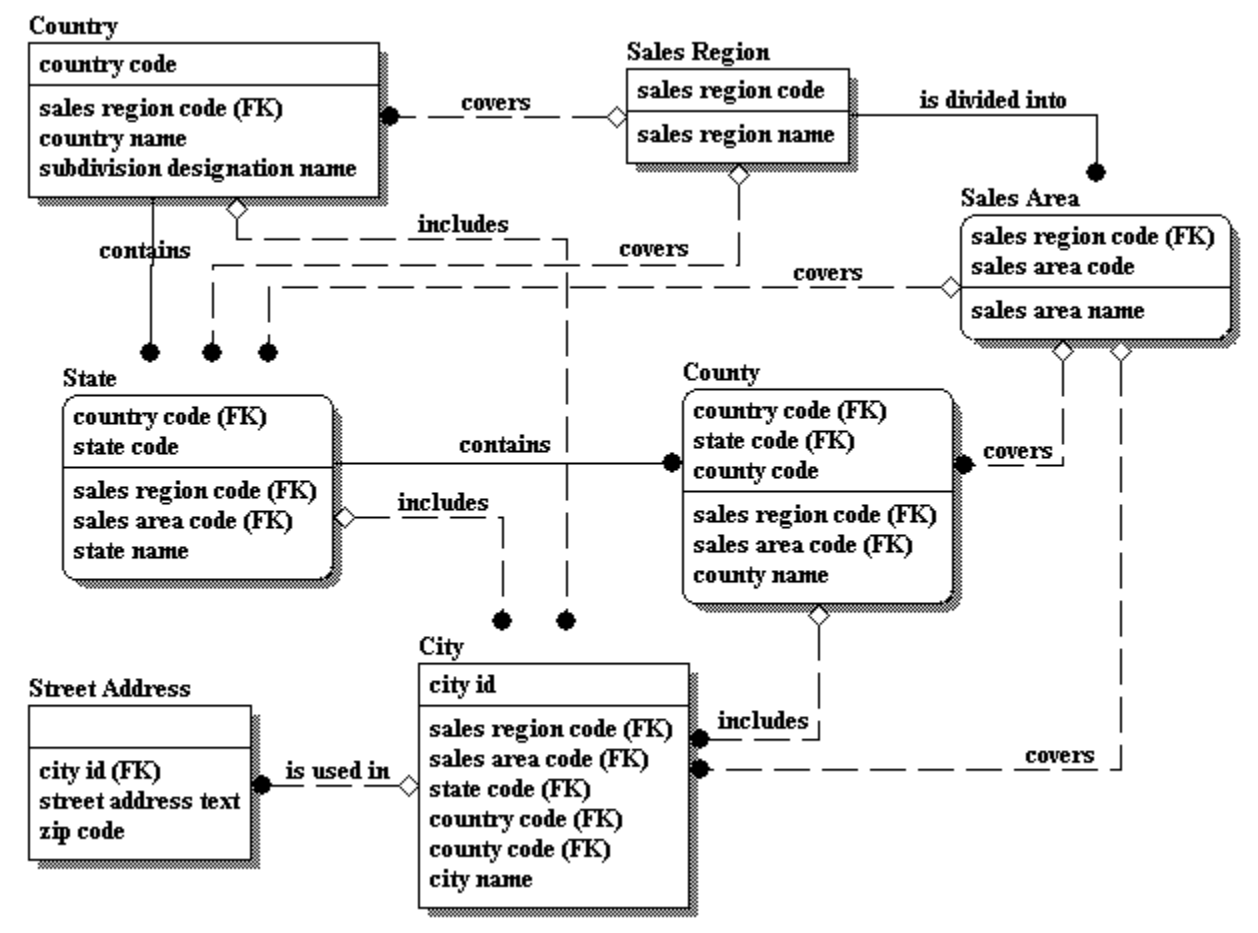
Aspects of the 2nd Dimension (cont.)

- **Time and time again, those who don’t plan for it end wishing they had.**
- **Data Warehouse: Can’t have one without it.**

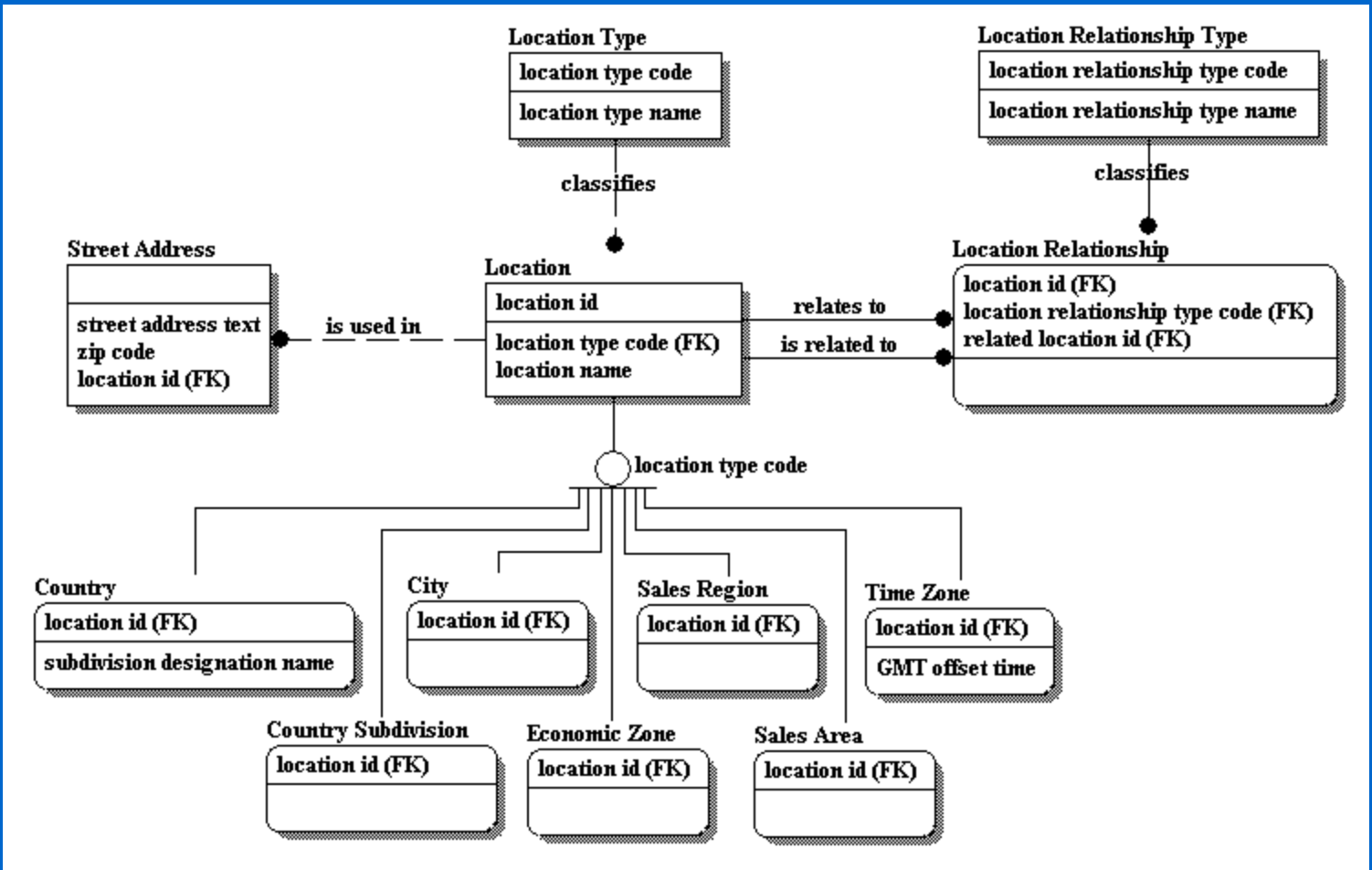
The 3rd Dimension: Location

- **Geographic locations**
- **Spatial locations (Absolute, Relative)**
- **Addresses (Street, Electronic, Net)**
- **Functional Locations (Exchanges, Fabs)**
- **Component Locations (Connectivity)**
- **Virtual Locations**

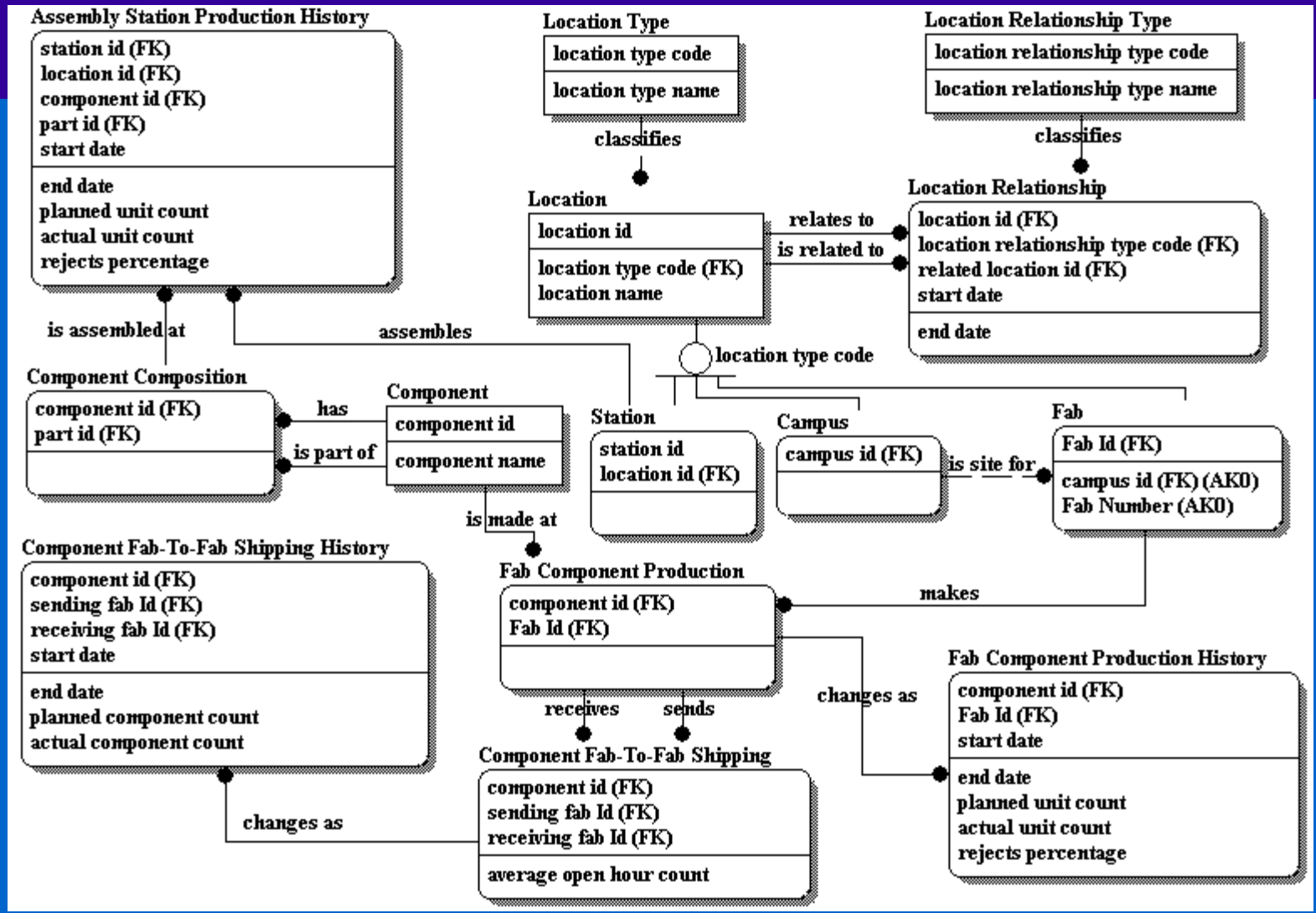
What is wrong with this picture?



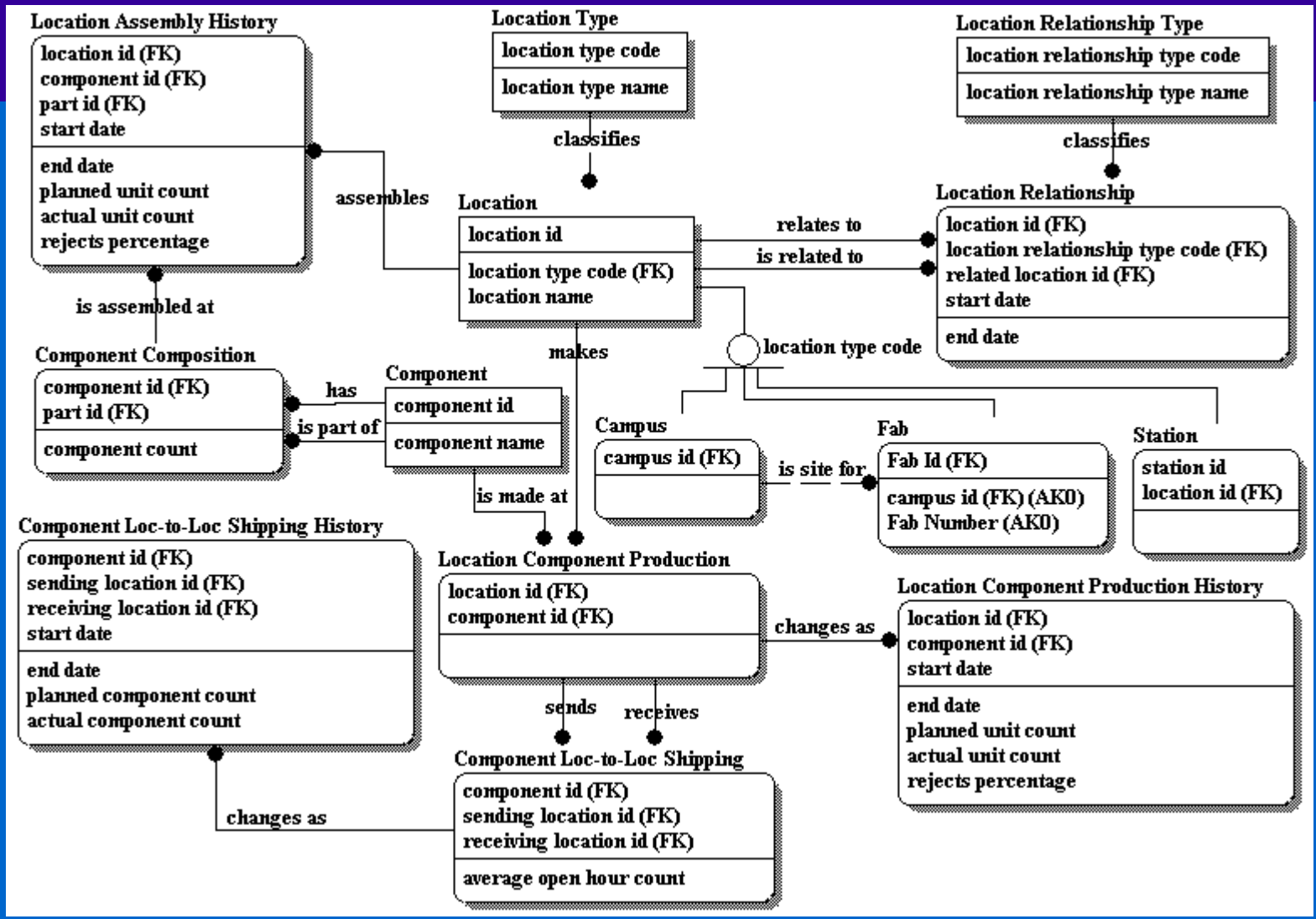
Location: An Open-Ended Structure



In Theory: Structure with Business Rules



In Practice: Open-Ended is Better



Notions About Locations

- **A non-punctual location is a container**

  If a location is not expressed in terms of coordinates (scalar, polar, etc.), it is an entity (or object) which implicitly represents a set of coordinates within which other entities (or objects) may reside at a point in time.

Notions About Locations (cont.)

- **Location vs. Government**

  For territorial or geographic “entities”, the distinction between the location entity and the entities representing the population set, or the organization(s) governing the resources tied to the location, should be preserved.

Notions About Locations (cont.)

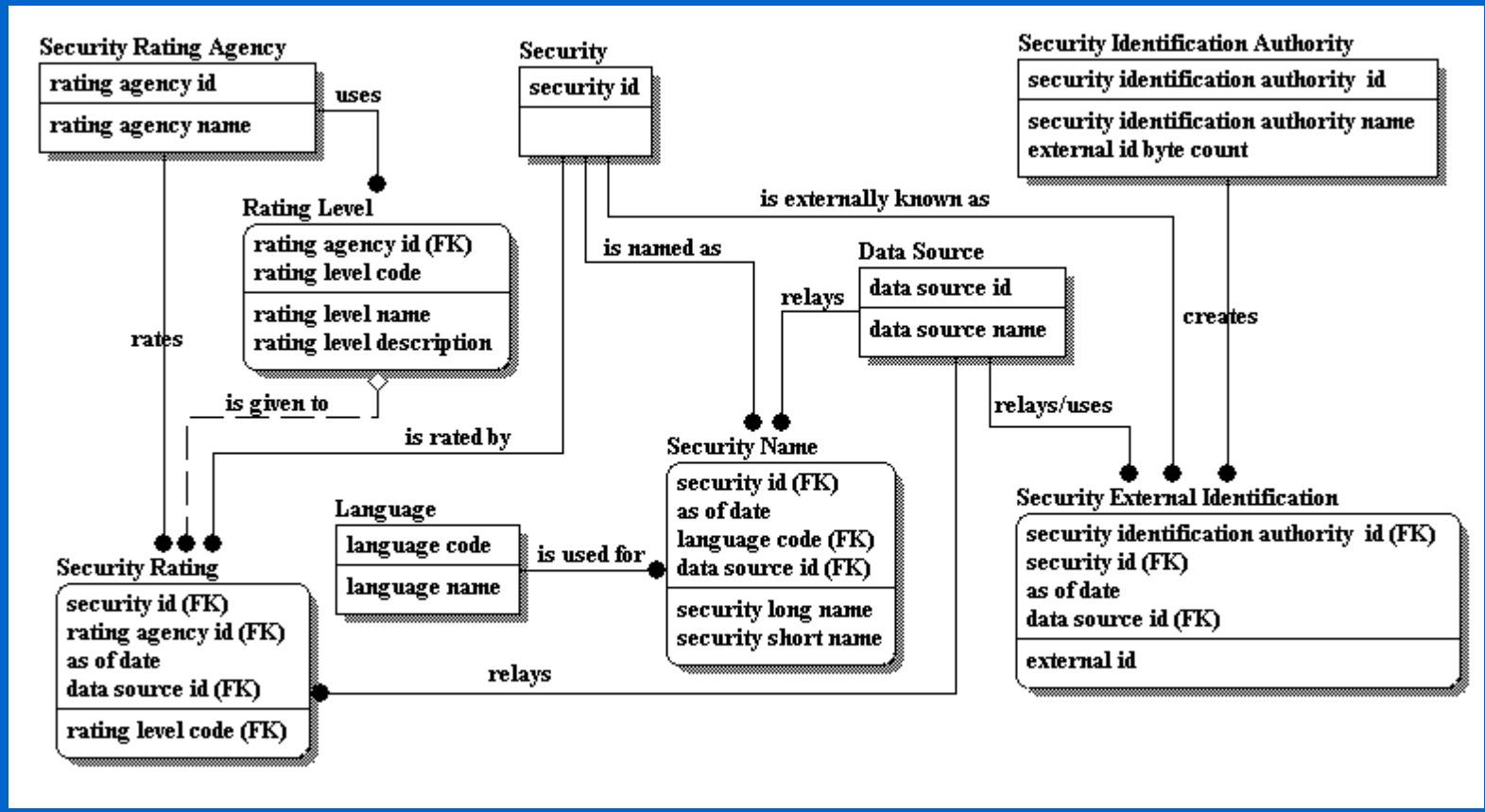
- **Location vs. Government - an example**

  The location entity instance named "France" should relate to, and not be merged with, the political entity instance called "The Republic of France", and the economic entity instance called "French Market".

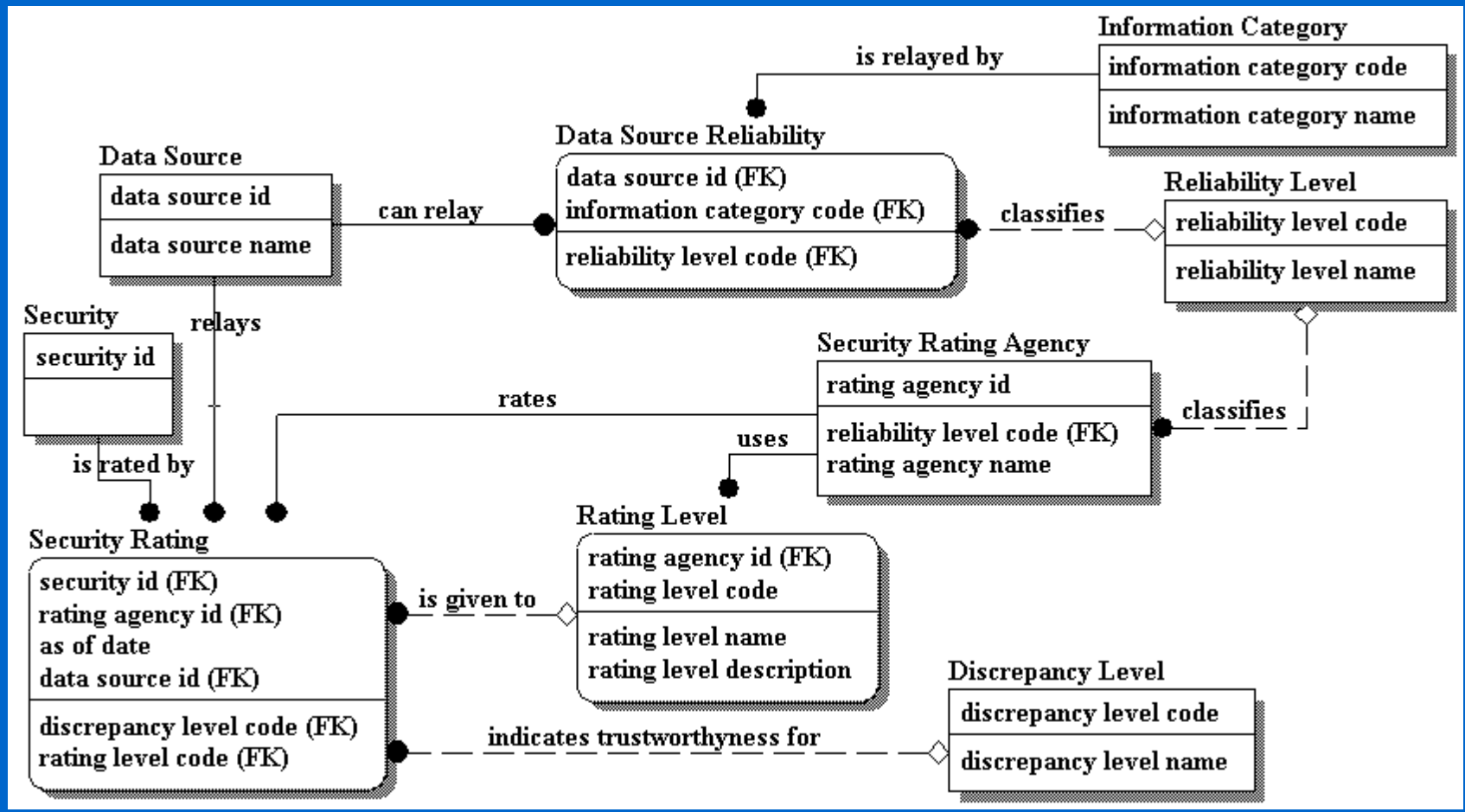
The 4th Dimension: Data Source

- **Programs or systems**
- **Users**
- **External Data Providers/Services**
- **Data Generators (Markets, Banks, etc.)**

Data Source vs. Data Generator



Data Source/Generator Reliability



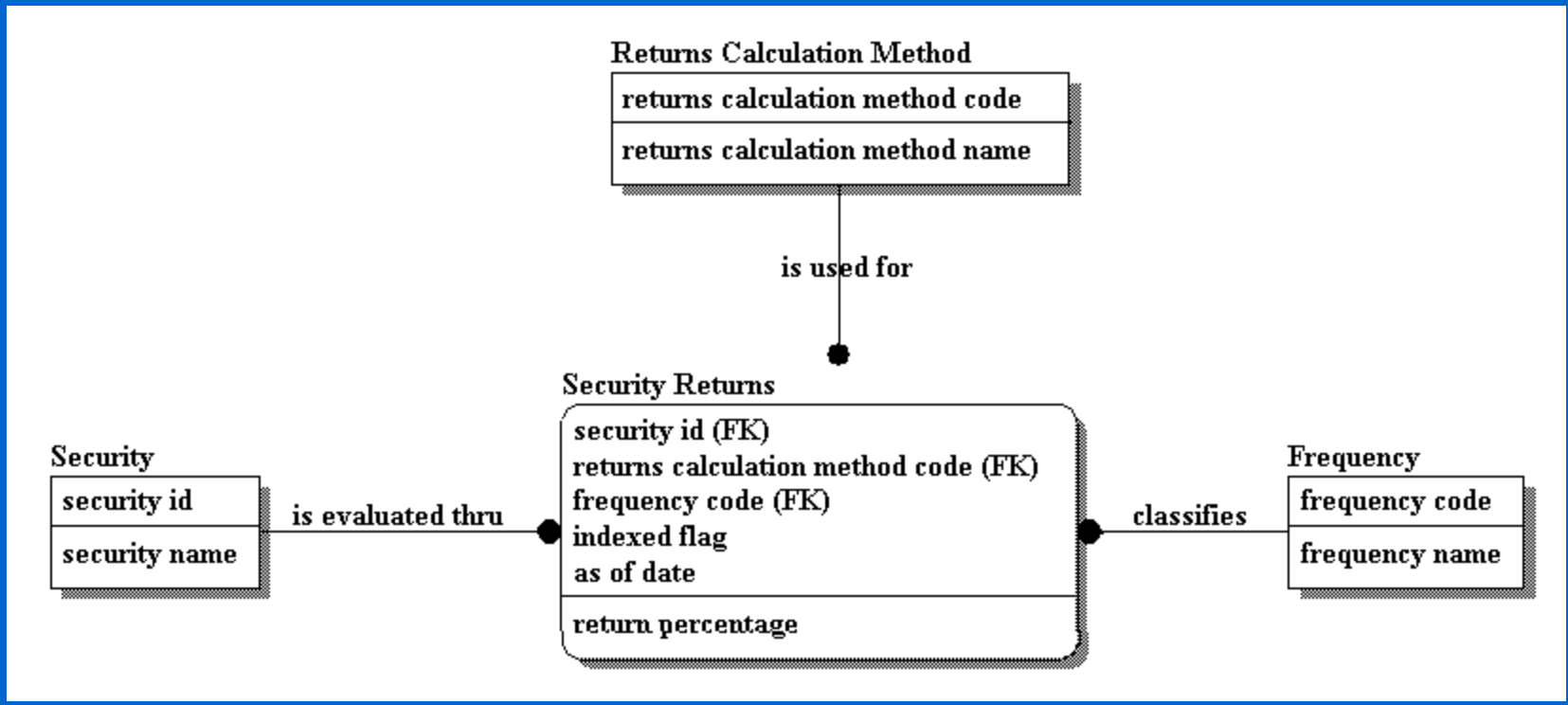
The Information Flow from Data Generators and Providers

- **Can be documented thru a DFD tool**
- **The original reliability is not necessarily preserved (cases of manual retyping**
- **Vendor typifying codes do not necessarily map well**
- **Vendors do not necessarily uniquely identify chunks of (the same) information**

The 5th Dimension: Method

- **Calculation/valuation method**
- **Selection criterion**
- **Choice of parameters**
- **What-if scenarios**

Calculation Methods / Selection Criteria



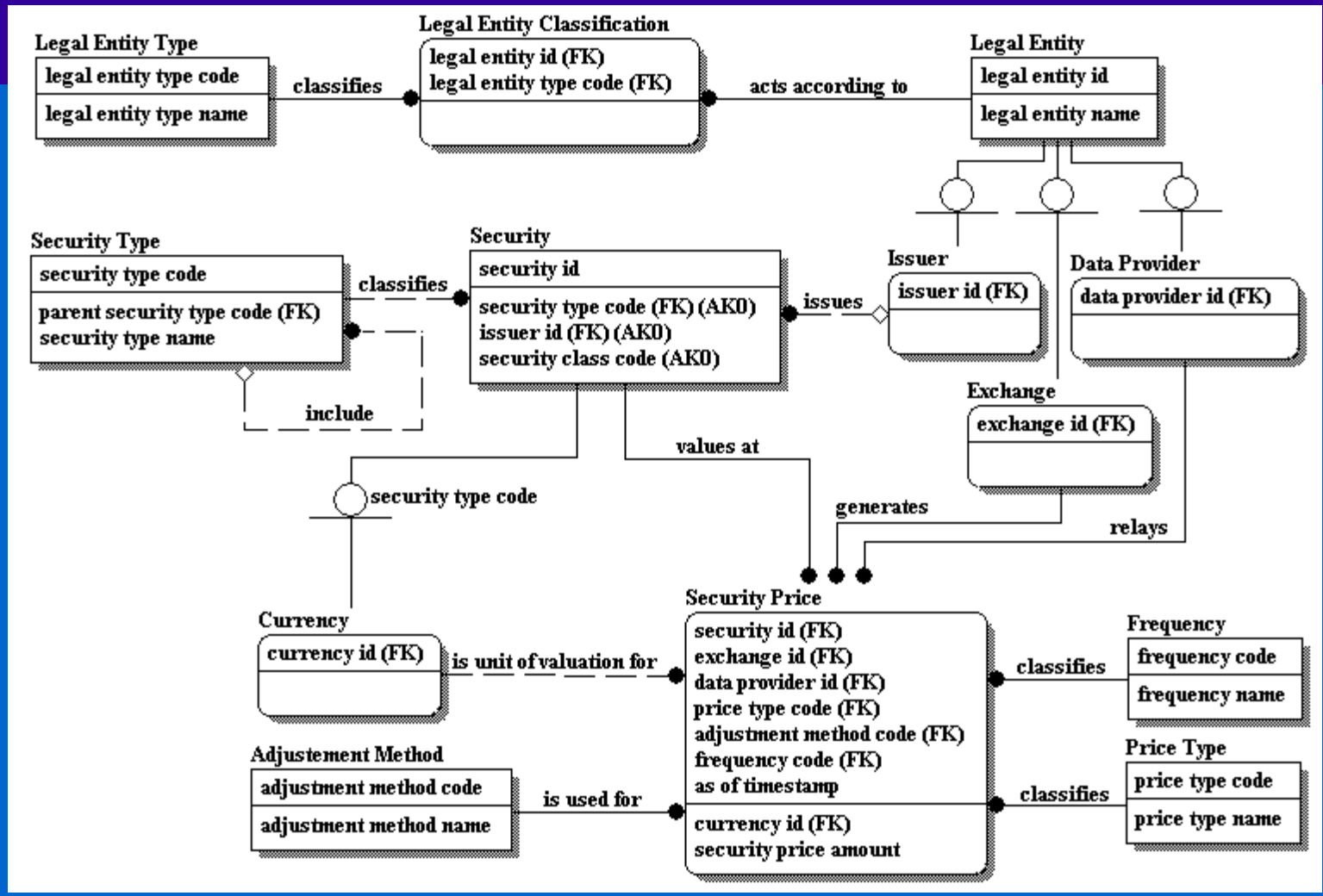
The 6th Dimension: Purpose

- **Strategy component**
- **What-if goal**

Cost/Value: A 7th Dimension?

- **For a system or project: definitely, yes (ignore it at your own risk!)**
- **Often a passive resultant from the projection of the entity/object six-dimensional “vector” on the cost/value “axis”**
- **Cost/Value: for a model, no (not in PK)**

Multi-Dimensioning: Security Prices



Cost/Value: A 7th Dimension? (cont.)

- **Cost/value range: for a model, maybe**
  - **- can be part of PK, but**
  - **- involves redundant data (calculated), unless**
  - **- it is part of the expression of a business rule, but then**
  - **- it hides as part of the 5th dimension**

Multi-Dimensioning Objects

- **Multi-dimensioning objects is applying the pertinent dimensioning to attributes/characteristics**
- **Multi-dimensioning the attribute of an object is transforming the attribute into an attribute list corresponding to an n-dimensional matrix**

An Object Attribute List Matrix

SecurityPriceList ::=

(PriceAmount, PriceCurrency) xch(0),dprv(0),ptyp(0),adjm(0),frq(0),tms(0) ,

(PriceAmount, PriceCurrency) xch(0),dprv(0),ptyp(0),adjm(0),frq(0),tms(1) ,

. . .,

(PriceAmount, PriceCurrency) xch(0),dprv(0),ptyp(0),adjm(0),frq(0),tms(n) ,

(PriceAmount, PriceCurrency) xch(0),dprv(0),ptyp(0),adjm(0),frq(1),tms(0) ,

(PriceAmount, PriceCurrency) xch(0),dprv(0),ptyp(0),adjm(0),frq(1),tms(1) ,

. . .,

(PriceAmount, PriceCurrency) xch(i),dprv(j),ptyp(k),adjm(l),frq(m),tms(n)

An Object Attribute List Matrix - legend

Where:

- **xch = echange market (location/source)**
- **dprv = data provider**
- **ptyp = process type**
- **adjm = adjustment method (par, shr out)**
- **frq = frequency (of valuation)**
- **tms = timestamp (of valuation)**

Multi-Dimensioning Objects (cont.)

- **Each attribute may be dimensioned differently**
- **Each matrix is an object**

ROOLSA 2nd Layer: Physical Ops

- **Physical Operations: Create, Read, Update, Delete**
- **Targets a table with its Time-Dimensioned Attributive dependent tables**
- **Insures Subtype/Supertype integrity**
- **Knows which column goes with which table or audit-hiding view**
- **Each C/U/D request is one LUW**

ROOLSA 3rd Layer: Logical Ops (1)

- **Numerous Logical Operators**
- **One Logical Operation can include many Physical Operation requests**
- **Operations on Fundamental Entities and associated non-Time Dimension attributive entities**
- **Each logical request is one LUW and manages change control**

ROOLSA 3rd Layer: Logical Ops (2)

List of Basic Logical Operators:

- **Add**
- **Correct**
- **Delete**
- **List**
- **Modify**
- **Read**
- **{De|Pro}mote (Change State)**
- **Undo**
- **Verify**

ROOLSA Logical vs. Physical Ops

No equivalence. Examples:

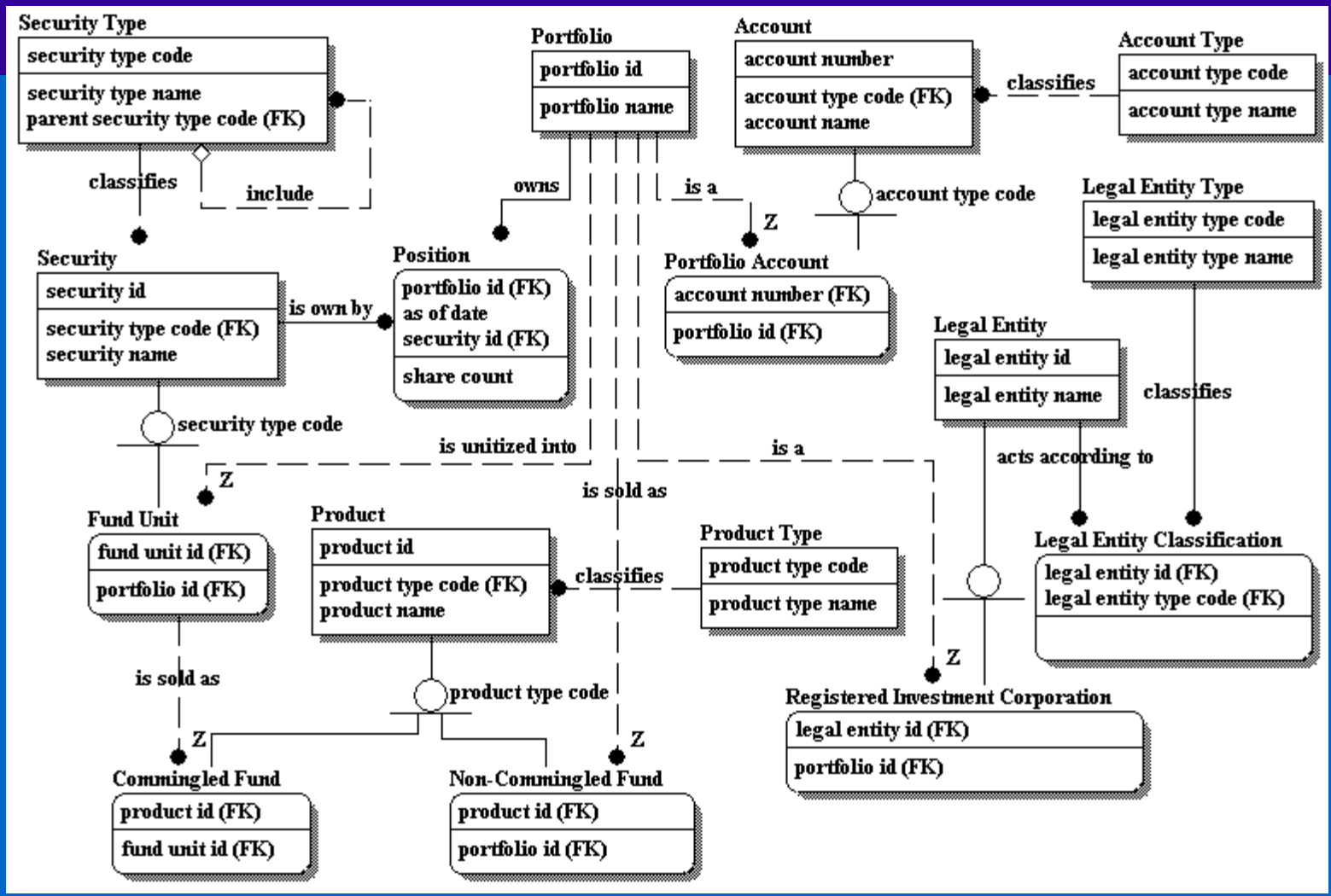
- **Logical Delete = Physical Insert(s)**
- **Logical Undo = Physical Inserts**
- **Logical Modify = Physical Insert(s)**
- **Logical Correct = Physical Update(s)**
- **Logical Verify = Distributed Physical Select(s) [+ any Physical Ops]**

ROOLSA 4th Layer: Translation

- **The Translation layer implements the O-O/R-R/O-O Mapping**
- **O-O to R: Method → Logical Op(s)**
- **R-R to O: Logical Op(s) results → Method(s)**
- **A Logical Operation result may cause an event trigger on a non-requesting Method**

Multi-Faced Objects

Another Dimension
to the definition of
“Multi-Dimensional
Entities and Objects”?


Security Type
security type code
security type name
parent security type code (FK)
classifies
include
Portfolio
portfolio id
portfolio name
owns
is a
Z
Account
account number
account type code (FK)
account name
classifies
Account Type
account type code
account type name
account type code
Legal Entity Type
legal entity type code
legal entity type name
Security
security id
security type code (FK)
security name
is own by
Position
portfolio id (FK)
as of date
security id (FK)
share count
Portfolio Account
account number (FK)
portfolio id (FK)
Legal Entity
legal entity id
legal entity name
classifies
security type code
is unitized into
is a
Z
is sold as
acts according to
Fund Unit
fund unit id (FK)
portfolio id (FK)
Product
product id
product type code (FK)
product name
classifies
Product Type
product type code
product type name
Legal Entity Classification
legal entity id (FK)
legal entity type code (FK)
is sold as
product type code
Z
Registered Investment Corporation
legal entity id (FK)
portfolio id (FK)
Commingled Fund
product id (FK)
fund unit id (FK)
Non-Commingled Fund
product id (FK)
portfolio id (FK)

Multi-Faced Objects (cont.)

The answer is: no!

- **The object can act as another object**
- **The question the object is asked is “who are you?”**
- **Multi-faced objects are therefore a reality coming from dimensioning from the 1st dimension**

Multi-Faced Objects vs. Class Hierarchy

- **Multi-faced objects are hard to model**
- **Cumbersome container-containee technique has to be used to obviate the shortcomings of hierarchies**
- **Requires multiple inheritance**
- **Just as RDBMSs outclassed HDBMSs, RCOODBMS will outclass HCOODBMS**

Generate from Modeling Tool

- **Create an Entity UDP called “Type”**
- **Create a Script using**
  %ForEachTable() {
  %If(%==(%EntityProp(Type), Fundamental)) { … }}
  to document your business objects
  by listing non-associative and non-fundamental children entities

Generate from Modeling Tool (2)

- **Create a Script using**
  %ForEachTable() {
  %If(%==(%EntityProp(Type), Associative)) { … }}
  to document your service objects

Generate from Modeling Tool (3)

- **Create a Script using**
  %ForEachTable() {
  %If(%==(%EntityProp(Type), Reference)) { … }}
  to generate reference table insert statements from key validation rule
  (validation rule changed to logical only afterward)

Generate from Modeling Tool (4)

- **Physical only audit columns**
- **Physical only status codes**
- **Physical only delete flags**
- **Audit columns maintenance triggers**
- **Use CC to reverse engineer the generated DDL back into the model**
- **Once used, deactivate the script**

Generate from Modeling Tool (5)

Physical Ops Layer stored procedures:

- **Automated documentation**
- **Standardized cookie-cutter code**
- **Data transfer code & SQL based on metadata contained in data model**
- **Standardized return data stream & return codes**
- **Generate from one script & customize**

Generate from Modeling Tool (6)

Logical Ops Layer code (e.g. Java):

- **Automated documentation**
- **Standardized base code**
- **Data transform, change control & Physical Ops calls code based on metadata contained in data model**
- **Standardized return data stream & return codes**
- **Generate from one script & customize**

Generate from Modeling Tool (6)

O-O/R Translation Layer code (e.g. Java):

- **Automated documentation**
- **Standardized base code**
- **Data transform, synchronization & Logical Ops calls code based on metadata contained in data model**
- **Standardized return data stream & return codes**
- **Generate from one script & customize**

Generate from Modeling Tool (7)

O-O Layer code (e.g. Java):

- **Automated documentation**
- **Standardized base methods (SOA)**
- **Standardized skeleton code per method**
- **Level 4 module calls code based on metadata contained in data model**
- **Standardized return messages**
- **Generate from one script & add extra business rules logic and local variables**

For in-depth details about ROOLSA and about automated metadata-based code generation tools and techniques

- **Please contact e-Modelers at:**
- **Nicholas.Khabbaz@emodelers.com**
- **Ingrid.Hunt@emodelers.com**
- **Francois.Cartier@emodelers.com**