Generic Data Modeling

Dan Collins

www.kalido.com

KALIDO

Agenda

- Brief History of Generic Data Modeling
- Basic Principles
- Differences from traditional modeling
- Example Implementation (Kalido)
  - > Logical Model
  - > Physical storage
- Q & A

Origins of Generic Modeling

- Originated in Europe in 1990's
- Developed under EPISTLE (European Process Industries STEP Technical Liaison Executive – STEP: Standard for the Exchange of Product Model Data – ISO10303)
- Clearly articulated in a white paper written in 1996 by Matthew West ("Developing High Quality Data Models" - http://www.matthew-west.org.uk/documents/princ03.pdf)
- The Core Model was adopted as ISO 15926
- Implemented by Shell International in several internal IT projects from 1995
- Managed by an application developed within Shell (code named Genie)

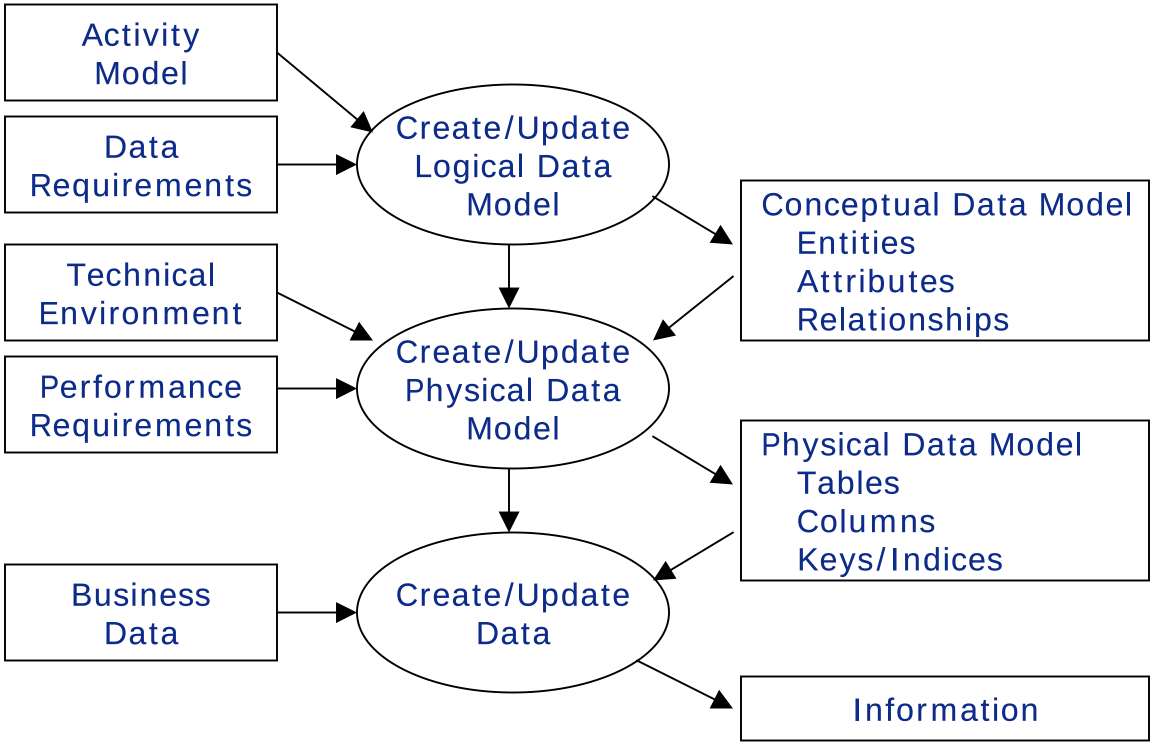
The problem

Poor data models had adversely impacted several projects.

The identified problems include:

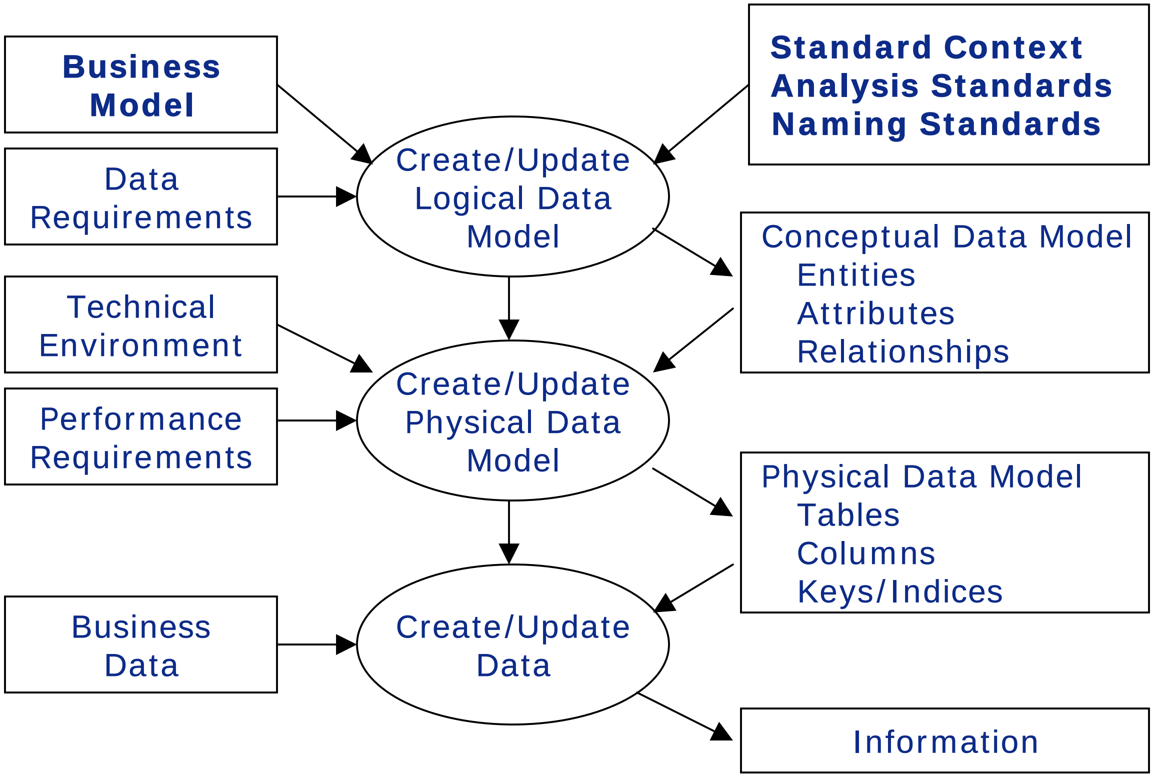
- Some business rules are fixed in the structure of a data model.
  - > Small changes in the way business is conducted lead to large changes in computer systems and interfaces.
- Entity types are often not identified, or incorrectly identified.
  - > Data, data structure, and functionality are replicated, together with the attendant costs of that duplication in development and maintenance.
- Data models for different systems are arbitrarily different.
  - > Complex interfaces are required between systems that share data.
- Data cannot be shared effectively.
  - > The structure and meaning of data has not been standardized, leading to multiple efforts to “reinvent the wheel.”

The Modeling Process



The resulting data model is a “point in time” view of the business.

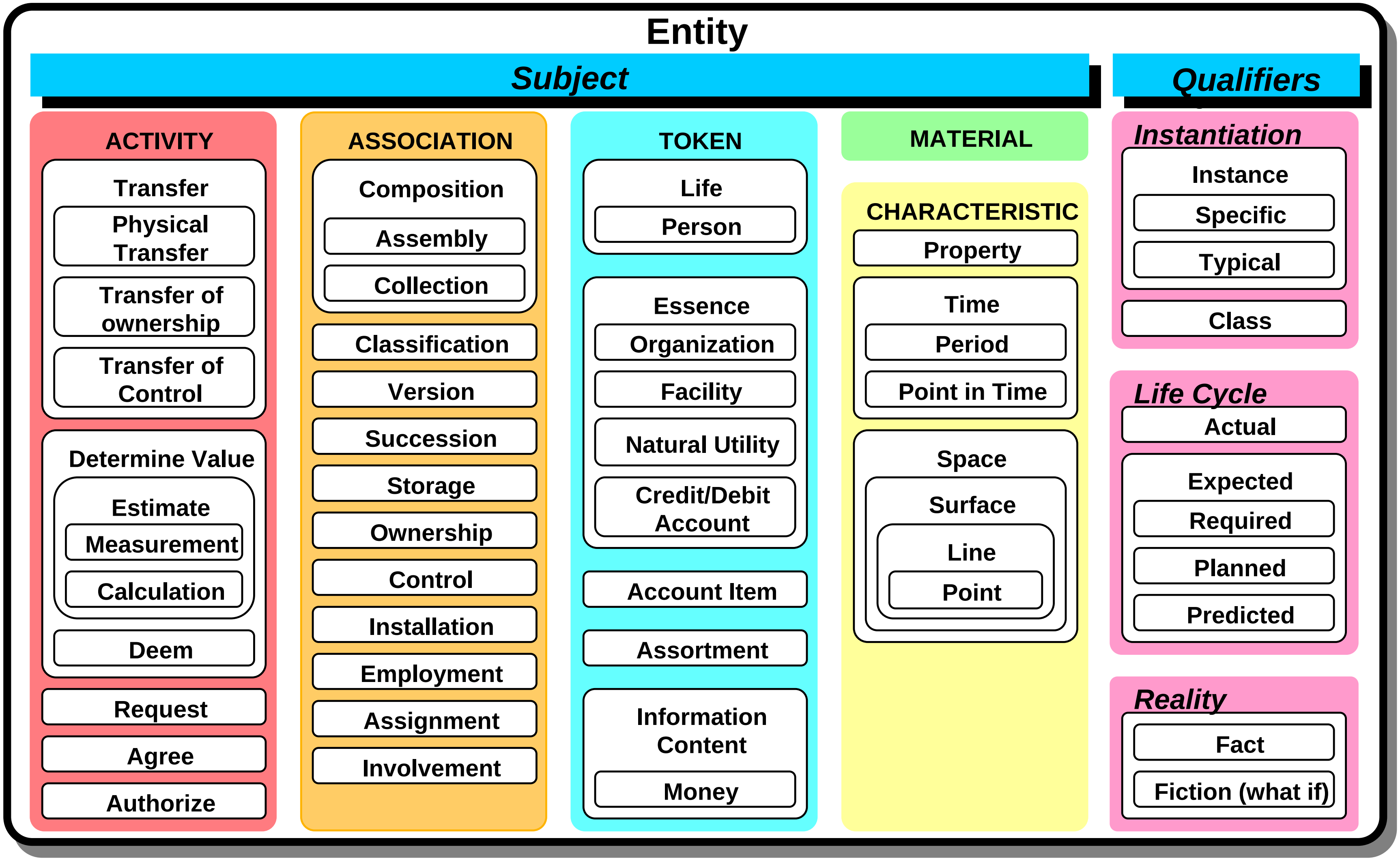
Change Resilient Models



The resulting data model is a “standardized” view of the business.

KALIDO

Core Model - Generic Entity Framework



Entity

Subject

ACTIVITY

- Transfer
  - Physical Transfer
  - Transfer of ownership
  - Transfer of Control
- Determine Value
  - Estimate
    - Measurement
    - Calculation
  - Deem
- Request
- Agree
- Authorize

ASSOCIATION

- Composition
  - Assembly
  - Collection
- Classification
- Version
- Succession
- Storage
- Ownership
- Control
- Installation
- Employment
- Assignment
- Involvement

TOKEN

- Life
  - Person
- Essence
  - Organization
  - Facility
  - Natural Utility
  - Credit/Debit Account
- Account Item
- Assortment
- Information Content
  - Money

MATERIAL

CHARACTERISTIC

- Property
- Time
  - Period
  - Point in Time
- Space
  - Surface
    - Line
      - Point

Qualifiers

Instantiation

- Instance
  - Specific
  - Typical
- Class

Life Cycle

- Actual
- Expected
  - Required
  - Planned
  - Predicted

Reality

- Fact
- Fiction (what if)

The 6 Rules of Generic Data Modeling

1. Candidate attributes should be treated as representing relationships to other entity types.
2. Entity types should represent, and be named after, the underlying nature of a thing, not the role it plays in a particular context.
3. Entities should have a local identifier within a database or exchange file. These should be artificial and managed to be unique. *Relationships should not be used as part of the **local** identifier.*
4. Activities, associations and event-effects should be represented by entity types (not relationships or attributes).
5. Relationships (in the entity/relationship sense) should only be used to express the involvement of entity types with activities or associations.
6. Entity types should be part of a sub-type/super-type hierarchy of generic entity types in order to define a universal context for the model.

Objects in the Business Data Model

Has employment contract

Has completed a degree course

Has a skill

Has a child

Person

Person

- Employee
- Graduate
- Profession
- Parent

Identifiers

- Employee number
- Student number
- Union code
- Social Security No

Attributes

- Colour of hair
- Gender
- IQ

Associations

- Employer
- Address
- Children

Any one Person can be any, one or none of the above.

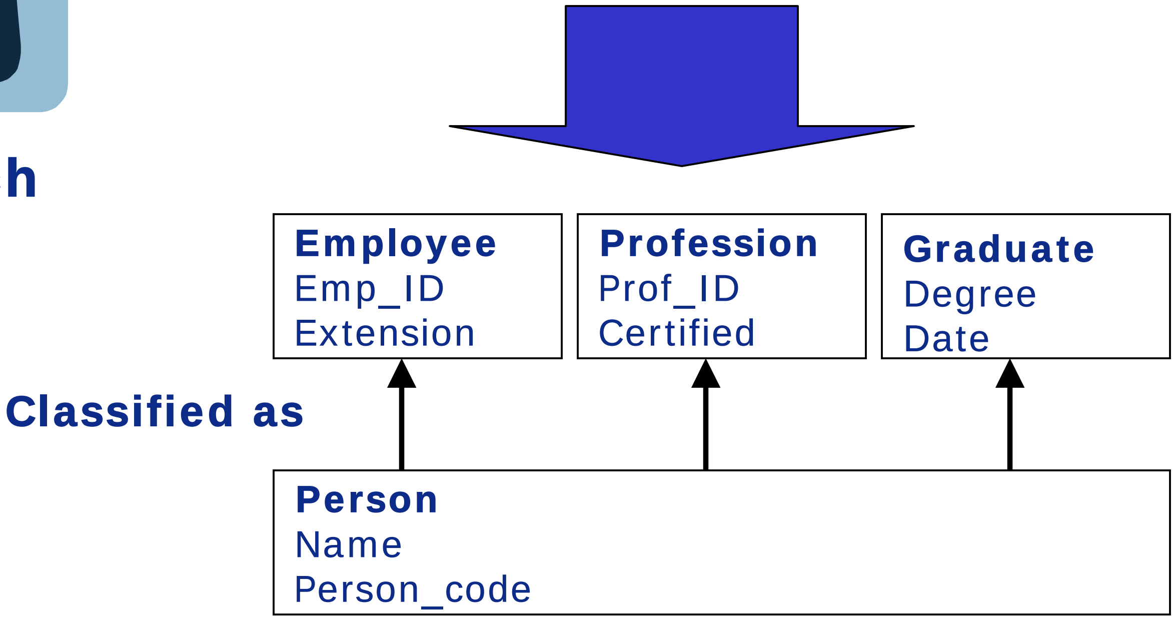
Model the “natural” entity

Sam Hirsch

Employee
- Employee ID
- Name
- Extension

Profession
- Profession ID
- Name
- Certified

Graduate
- Name
- Degree
- Date



Employee
Emp_ID
Extension

Profession
Prof_ID
Certified

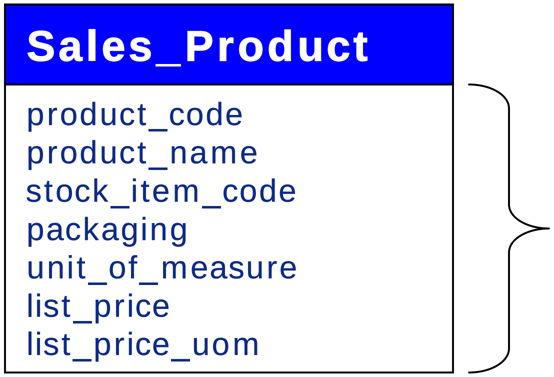
Graduate
Degree
Date

Classified as

Person
Name
Person_code

Difference from traditional data models

A typical data model entity:

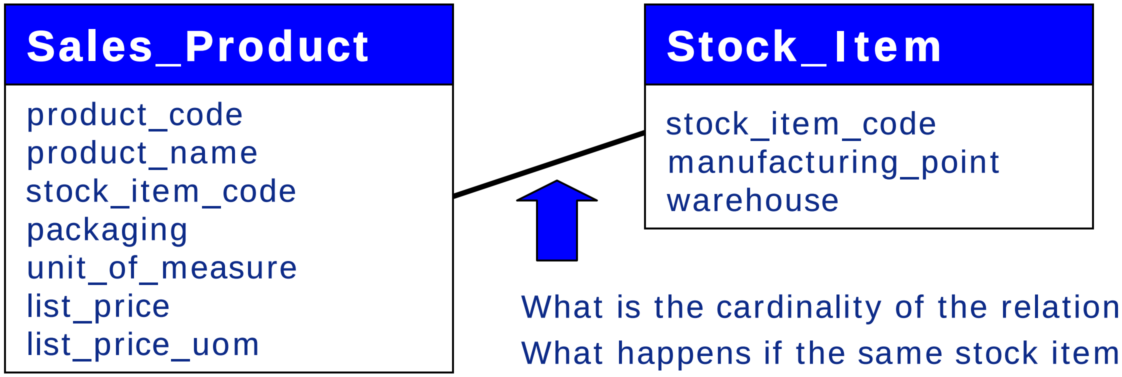


Sales_Product

product_code
product_name
stock_item_code
packaging
unit_of_measure
list_price
list_price_uom

Reflects a particular view of the thing, not the thing itself.

A typical relationship:



Sales_Product

product_code
product_name
stock_item_code
packaging
unit_of_measure
list_price
list_price_uom

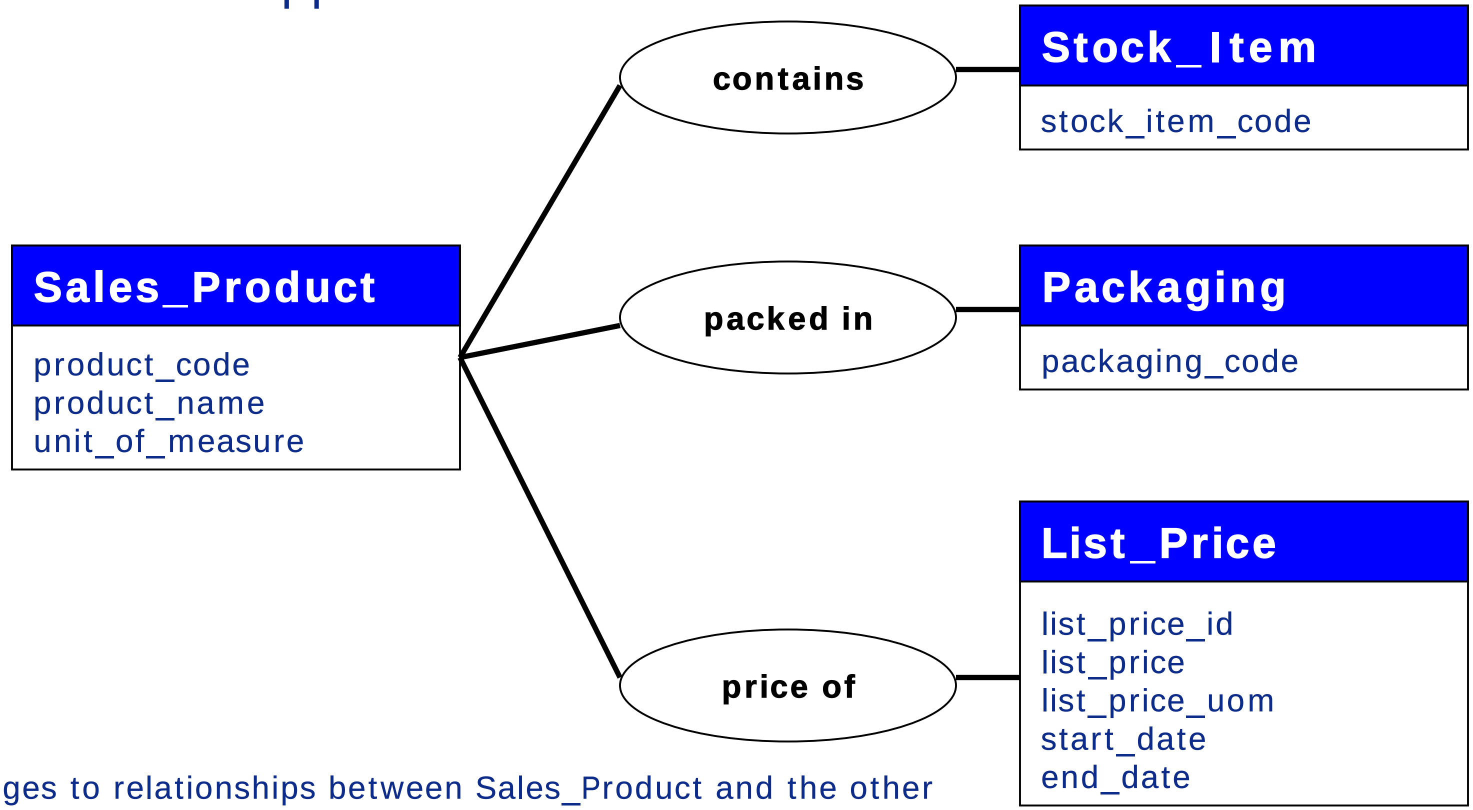
Stock_Item

stock_item_code
manufacturing_point
warehouse

What is the cardinality of the relationship?
What happens if the same stock item is sold under a new name?
What happens if the product is re-designed (a new stock item)?

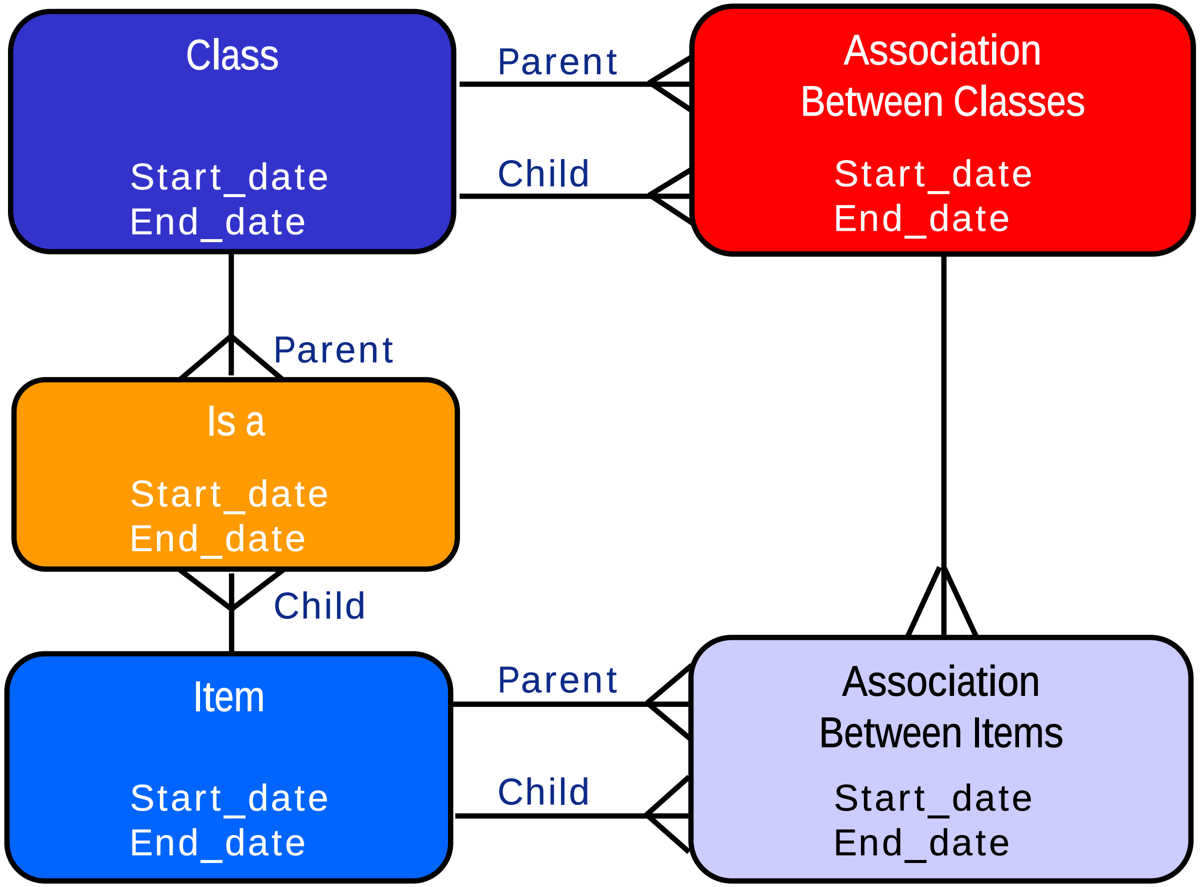
Difference from traditional models

The Generic Approach:

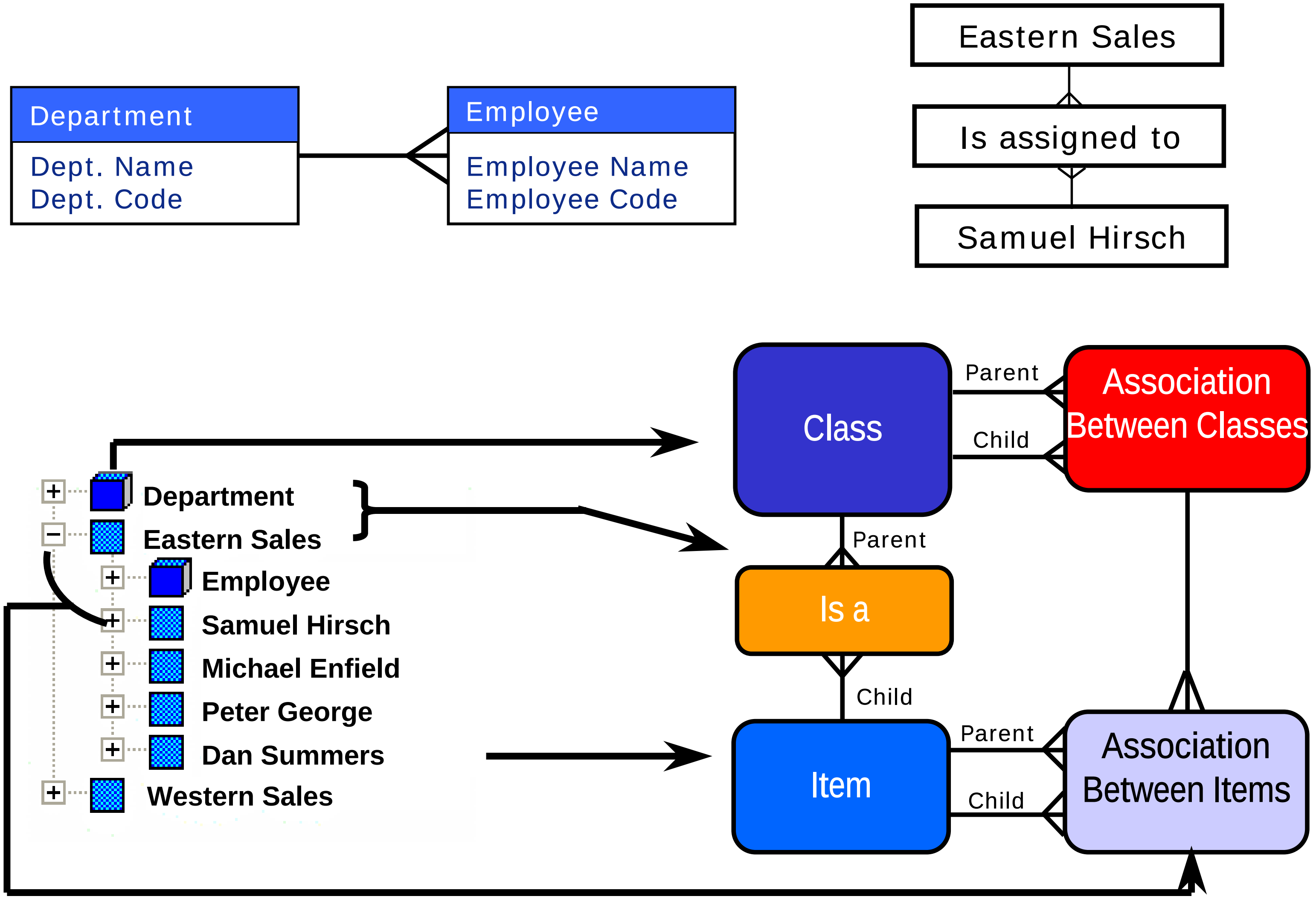


Changes to relationships between Sales_Product and the other entities are represented by new instances of the **association entities**.

Implementation - Logical Model



Implementation Example



Department

Dept. Name
Dept. Code

Employee

Employee Name
Employee Code

Eastern Sales

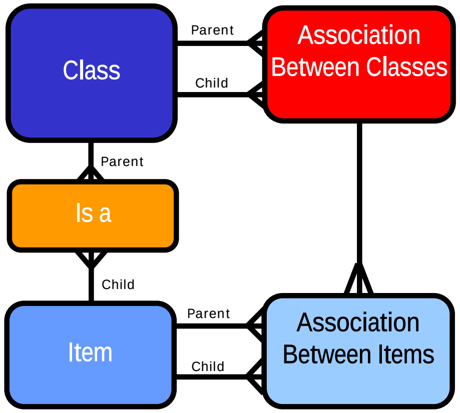
Is assigned to

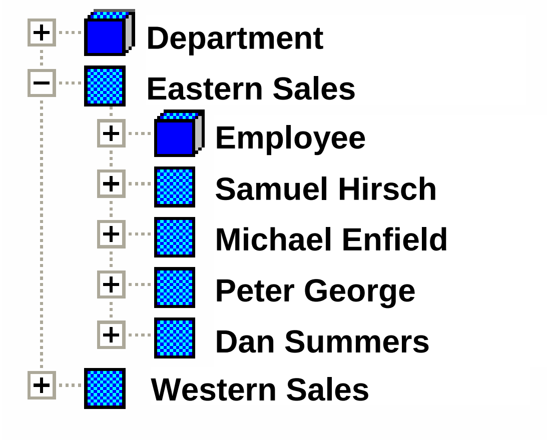
Samuel Hirsch

KALIDO

Implementation Example – Data Storage

Object ID	Entity	*Name*	Type	Parent	Child	Start Date	End Date
1001	Object	*Department*	Class	-	-	1/1/1753	12/31/9999
1002	Object	*Employee*	Class	-	-	1/1/1753	12/31/9999
1003	Assoc. Type	*Assigned to*	ABC	1001	1002	1/1/1753	12/31/9999
1004	Object	*Eastern Sales*	Item	-	-	1/1/1753	12/31/9999
1005	Object	*Samuel Hirsch*	Item	-	-	1/1/1753	12/31/9999
1006	Assoc.	-	Is a	1001	1004	1/1/1753	12/31/9999
1007	Assoc.	-	Is a	1002	1005	1/1/1753	12/31/9999
1008	Assoc.	-	1003	1004	1005	1/1/1753	12/31/9999

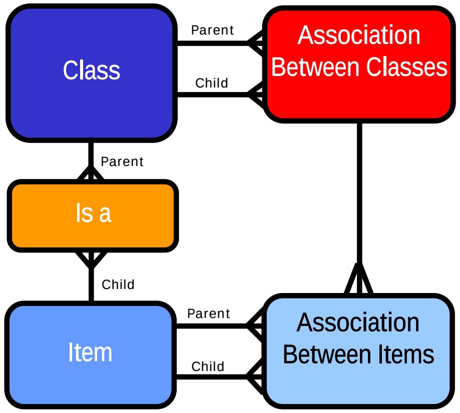


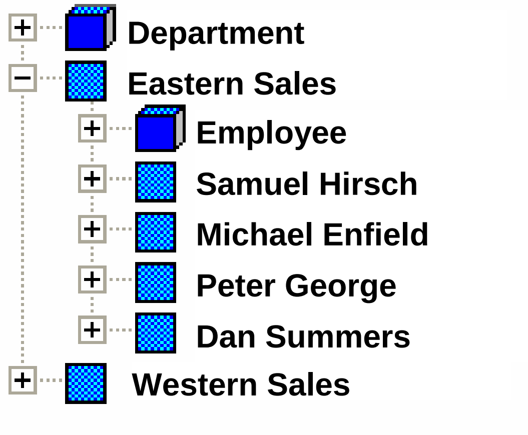


- Department
- Eastern Sales
  - Employee
  - Samuel Hirsch
  - Michael Enfield
  - Peter George
  - Dan Summers
- Western Sales

Managing Change

Object ID	Entity	*Name*	Type	Parent	Child	Start Date	End Date
1001	Object	*Department*	Class	-	-	1/1/1753	12/31/9999
1002	Object	*Employee*	Class	-	-	1/1/1753	12/31/9999
1003	Assoc. Type	*Assigned to*	ABC	1001	1002	1/1/1753	12/31/9999
1004	Object	*Eastern Sales*	Item	-	-	1/1/1753	12/31/9999
1005	Object	*Samuel Hirsch*	Item	-	-	1/1/1753	12/31/9999
1006	Assoc.	-	Is a	1001	1004	1/1/1753	12/31/9999
1007	Assoc.	-	Is a	1002	1005	1/1/1753	12/31/9999
1008	Assoc.	-	1003	1004	1005	1/1/1753	1/1/2004
1009	Object	*Central Sales*	Item	-	-	1/1/2004	12/31/9999
1010	Assoc.	-	Is a	1001	1009	1/1/2004	12/31/9999
1012	Assoc.	-	1003	1009	1005	1/1/2004	12/31/9999

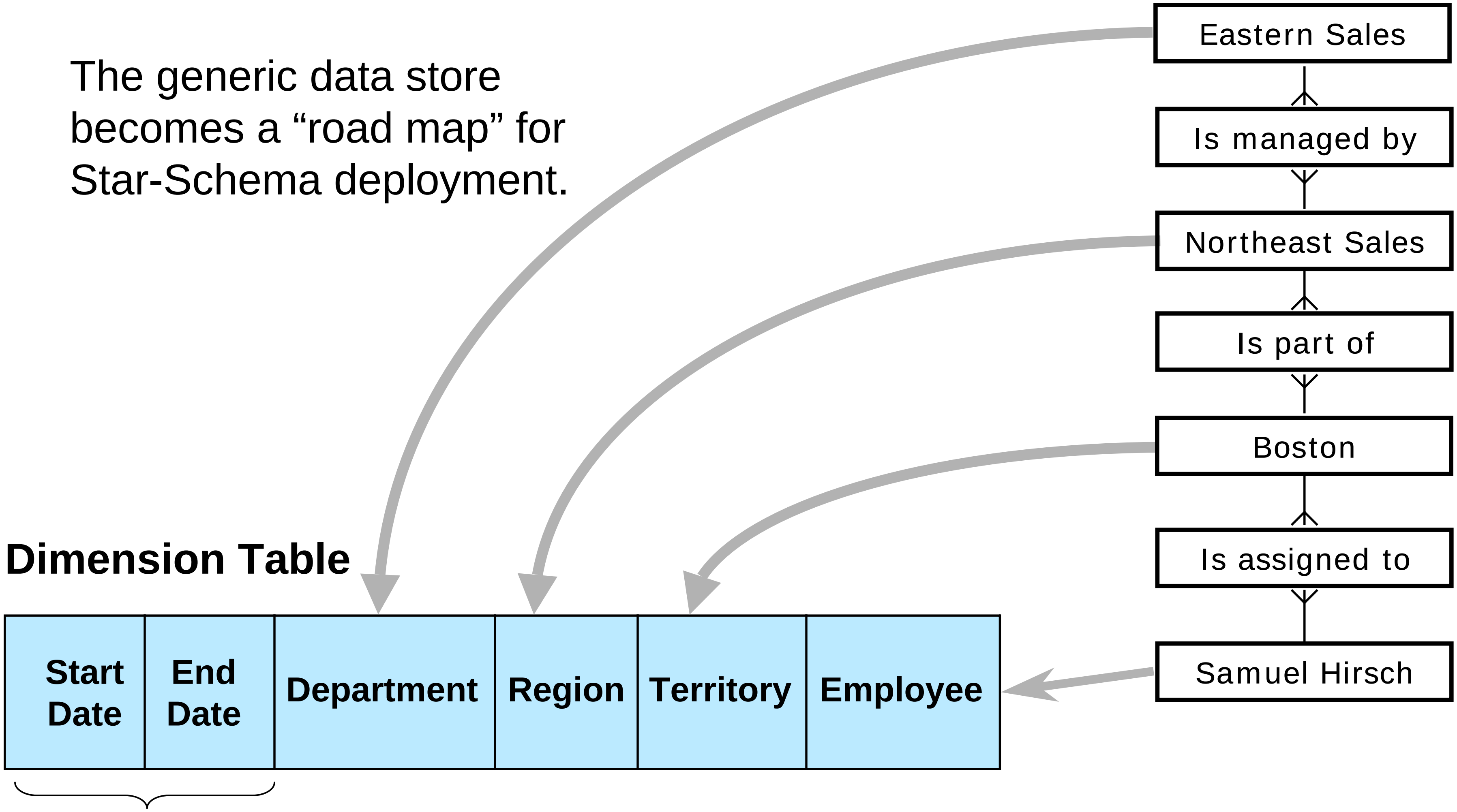




Dan Summers
Western Sales

Reporting – Automated Star Schemas

The generic data store becomes a "road map" for Star-Schema deployment.

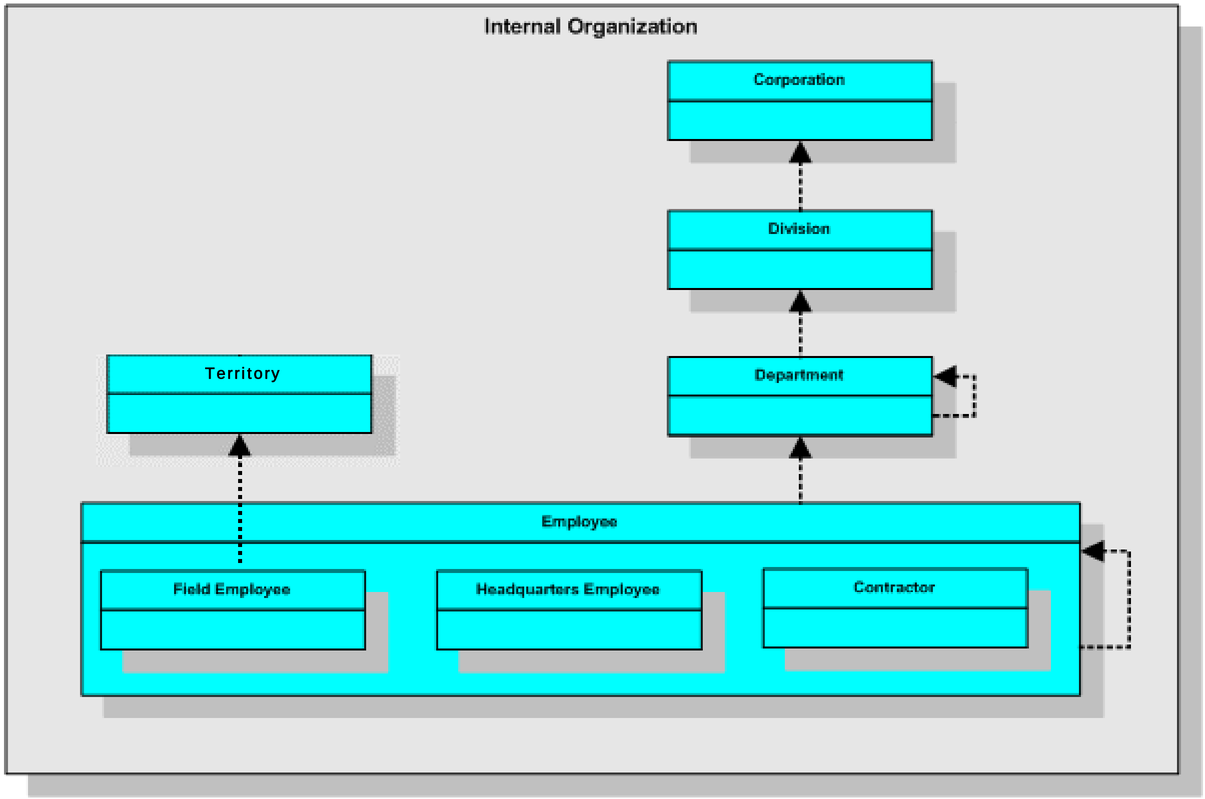


Dimension Table

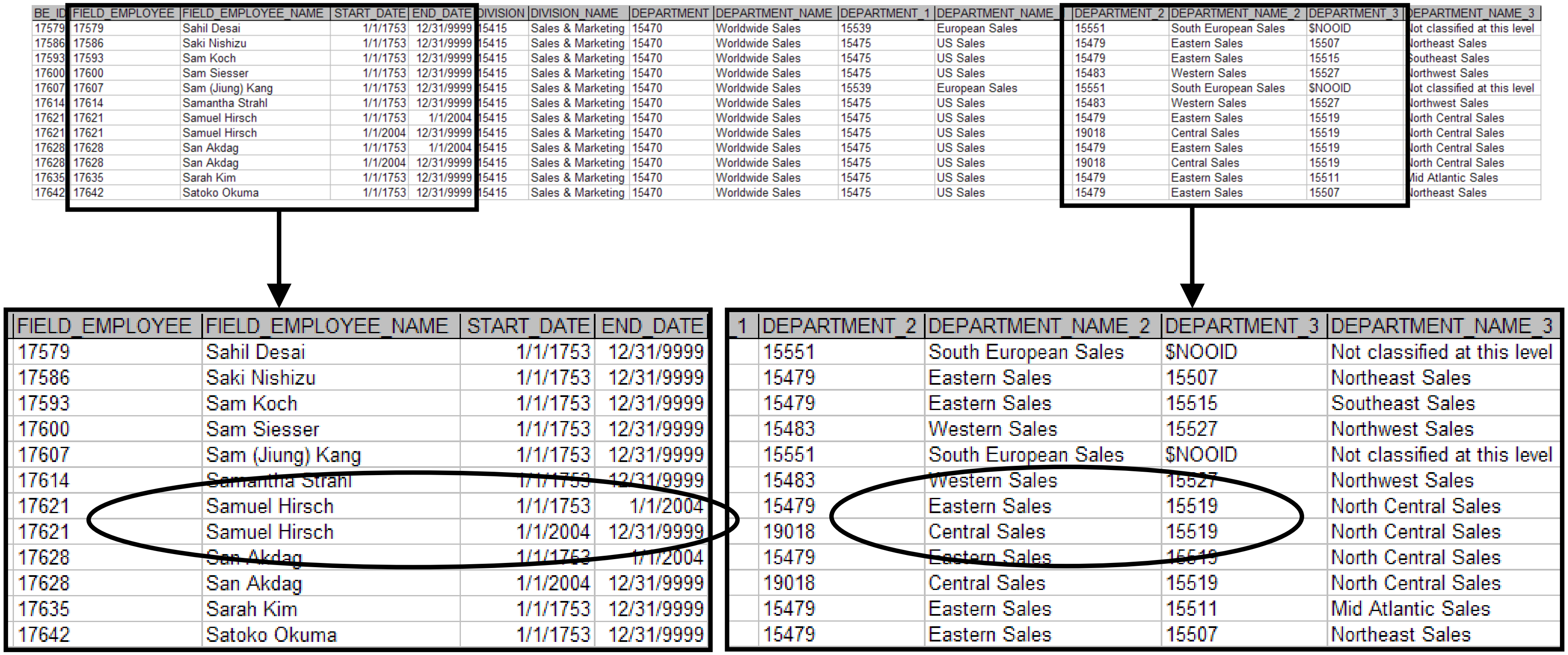
Start Date	End Date	Department	Region	Territory	Employee

Supports Type 1, 2 or 3 slowing changing dimensions, automatically.

Organizational Example



Dimension Table Contents

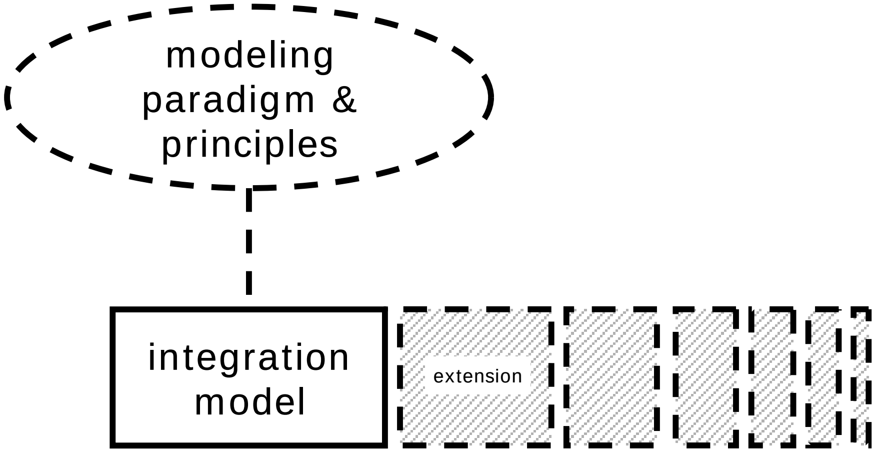


BE_ID	FIELD_EMPLOYEE	FIELD_EMPLOYEE_NAME	START_DATE	END_DATE	DIVISION	DIVISION_NAME	DEPARTMENT	DEPARTMENT_NAME	DEPARTMENT_1	DEPARTMENT_NAME_	DEPARTMENT_2	DEPARTMENT_NAME_2	DEPARTMENT_3	DEPARTMENT_NAME_3
17579	17579	Sahil Desai	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15539	European Sales	15551	South European Sales	$NOOID	Not classified at this level
17586	17586	Saki Nishizu	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15507	Northeast Sales
17593	17593	Sam Koch	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15515	Southeast Sales
17600	17600	Sam Siesser	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15483	Western Sales	15527	Northwest Sales
17607	17607	Sam (Jiung) Kang	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15539	European Sales	15551	South European Sales	$NOOID	Not classified at this level
17614	17614	Samantha Strahl	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15483	Western Sales	15527	Northwest Sales
17621	17621	Samuel Hirsch	1/1/1753	1/1/2004	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15519	North Central Sales
17621	17621	Samuel Hirsch	1/1/2004	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	19018	Central Sales	15519	North Central Sales
17628	17628	San Akdag	1/1/1753	1/1/2004	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15519	North Central Sales
17628	17628	San Akdag	1/1/2004	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	19018	Central Sales	15519	North Central Sales
17635	17635	Sarah Kim	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15511	Mid Atlantic Sales
17642	17642	Satoko Okuma	1/1/1753	12/31/9999	15415	Sales & Marketing	15470	Worldwide Sales	15475	US Sales	15479	Eastern Sales	15507	Northeast Sales

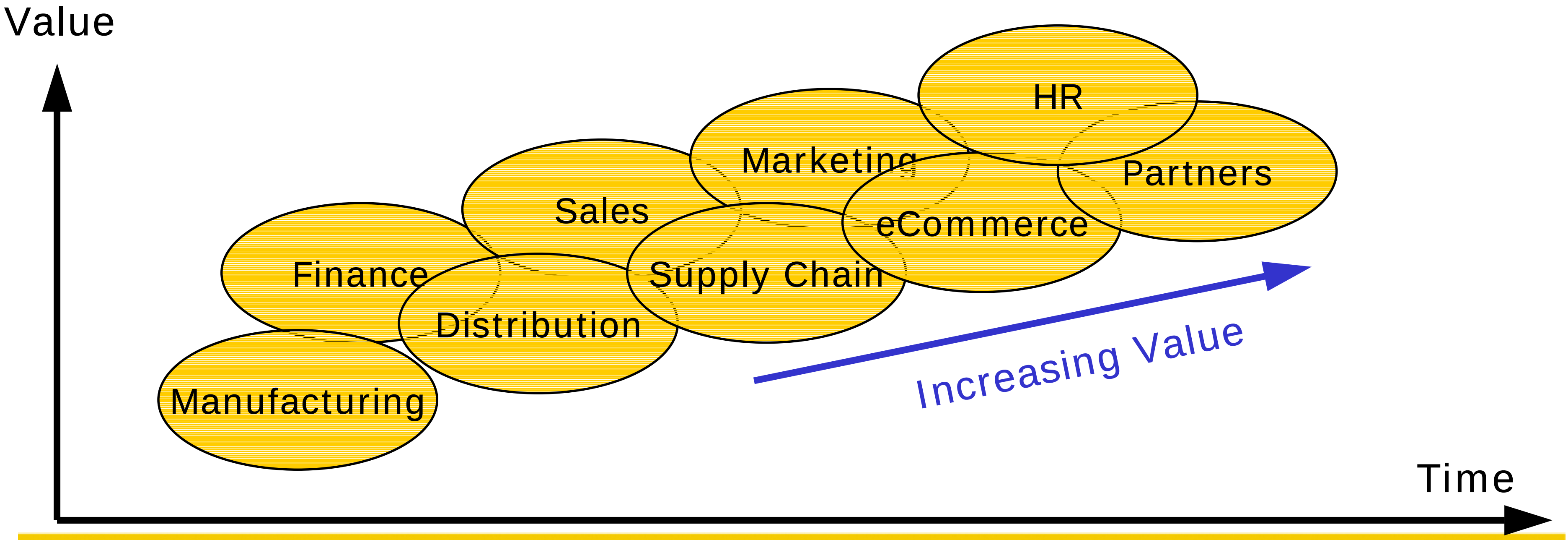
FIELD_EMPLOYEE	FIELD_EMPLOYEE_NAME	START_DATE	END_DATE
17579	Sahil Desai	1/1/1753	12/31/9999
17586	Saki Nishizu	1/1/1753	12/31/9999
17593	Sam Koch	1/1/1753	12/31/9999
17600	Sam Siesser	1/1/1753	12/31/9999
17607	Sam (Jiung) Kang	1/1/1753	12/31/9999
17614	Samantha Strahl	1/1/1753	12/31/9999
17621	Samuel Hirsch	1/1/1753	1/1/2004
17621	Samuel Hirsch	1/1/2004	12/31/9999
17628	San Akdag	1/1/1753	1/1/2004
17628	San Akdag	1/1/2004	12/31/9999
17635	Sarah Kim	1/1/1753	12/31/9999
17642	Satoko Okuma	1/1/1753	12/31/9999

1	DEPARTMENT_2	DEPARTMENT_NAME_2	DEPARTMENT_3	DEPARTMENT_NAME_3
	15551	South European Sales	$NOOID	Not classified at this level
	15479	Eastern Sales	15507	Northeast Sales
	15479	Eastern Sales	15515	Southeast Sales
	15483	Western Sales	15527	Northwest Sales
	15551	South European Sales	$NOOID	Not classified at this level
	15483	Western Sales	15527	Northwest Sales
	15479	Eastern Sales	15519	North Central Sales
	19018	Central Sales	15519	North Central Sales
	15479	Eastern Sales	15519	North Central Sales
	19018	Central Sales	15519	North Central Sales
	15479	Eastern Sales	15511	Mid Atlantic Sales
	15479	Eastern Sales	15507	Northeast Sales

An iterative approach to the enterprise model



- The generic approach allows iterative development
- The model can be evolved over time
- Based on the principles, the extensions get simpler with each extra iteration
- Eventually, new subject areas are nearly completely supported by the existing model



The 6 Rules of Generic Data Modeling

☑ 1. Candidate attributes should be treated as representing relationships to other entity types.

2. Entity types should represent, and be named after, the underlying nature of a thing, not the role it plays in a particular context.

☑ 3. Entities should have a local identifier within a database or exchange file. These should be artificial and managed to be unique. *Relationships should not be used as part of the* ***local*** *identifier.*

☑ 4. Activities, associations and event-effects should be represented by entity types (not relationships or attributes).

☑ 5. Relationships (in the entity/relationship sense) should only be used to express the involvement of entity types with activities or associations.

☑ 6. Entity types should be part of a sub-type/super-type hierarchy of generic entity types in order to define a universal context for the model.

Summary

- Generic Data Modeling treats associations between things as things themselves.
- Always suspect attributes as being things – they are, more often than not, reflective of a particular view of the thing.
- The result is a business data model – a representation of the whole business not just a view of data specific to a business function.
- Generic modeling can be supported by a data storage system that is also generic.
- The combination of generic modeling and generic storage provides a data infrastructure that is highly adaptive to change.
- Data driven applications can automate master data and warehouse management.

Generic Data Modeling

Questions?