ifis
Institut für Informationssysteme
Technische Universität Braunschweig

Relational Database Systems I

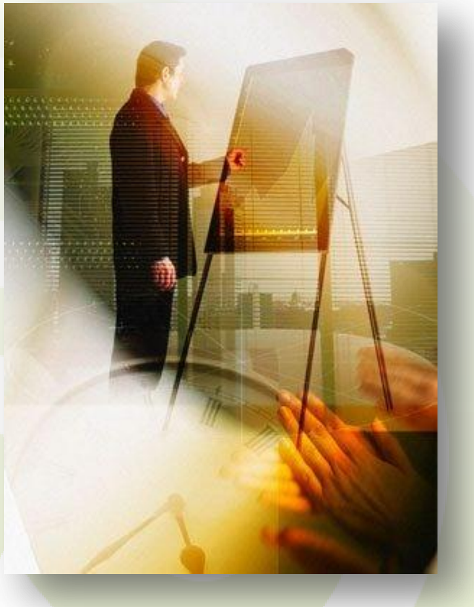
Wolf-Tilo Balke
Simon Barthel
Institut für Informationssysteme
Technische Universität Braunschweig
www.ifis.cs.tu-bs.de

4 Overview

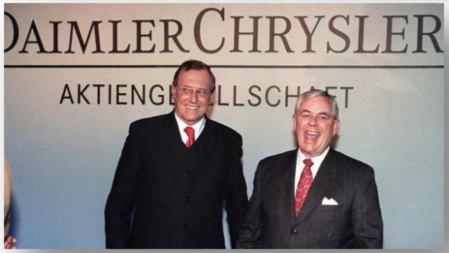
- View integration
- Resolving conceptual incompatibility
- Entity clustering for ER models
- Commercial dimension:
  The BEA story

4.1 Business Integration

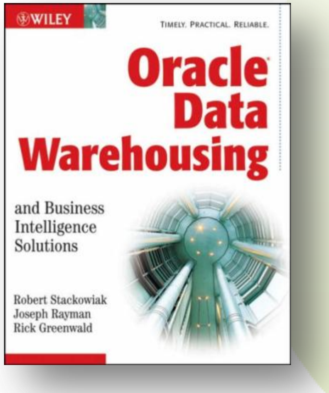
- Business currently is a world of M&A
  - Companies need to diversify/enhance their portfolio
  - But it is expensive to develop necessary applications
    - Knowledge gathering costs time
    - Will the output be worth it?
  - Idea: Rely on people who **are already** knowledgeable
    - Acquire small, specialized, and promising companies
    - Merge with big players in the field



4.1 Business Integration



- Examples
  - The Daimler–Chrysler merger
  - The Oracle–Sun merger
  - Oracle buys PeopleSoft, Siebel Systems, BEA Systems, …
    - Siebel Sales as CRM tool now part of Oracle's business intelligence suite

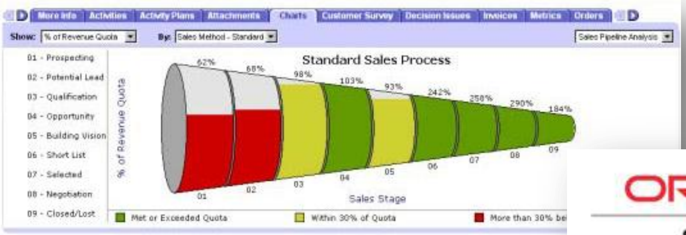


WILEY

TIMELY. PRACTICAL. RELIABLE.

Oracle Data Warehousing and Business Intelligence Solutions

Robert Stackowiak
Joseph Rayman
Rick Greenwald



ORACLE
SIEBEL
CUSTOMER RELATIONSHIP MANAGEMENT

4.1 Business Integration

- Merged (parts of) businesses are administrated by
  - Different specialized software systems?
  - One company-wide system?
- Usually there is an **historical evolution** of separate tools and programs
  - Accounting, sales & marketing, development, …
  - Based on individual requirements
- However, often a **unified view** is needed
  - Business intelligence? Warehousing…?!
    - Warehousing is also a **great lecture** at ifis…

4.1 View Integration

- Usually, there are **several conceptual schemas**
  - **Several designers** are part of the modeling process (modular software development)
  - **Different tasks** were modeled within the same organization (legacy systems)
  - **Several organizations** need to be integrated (business integration)

4.1 View Integration

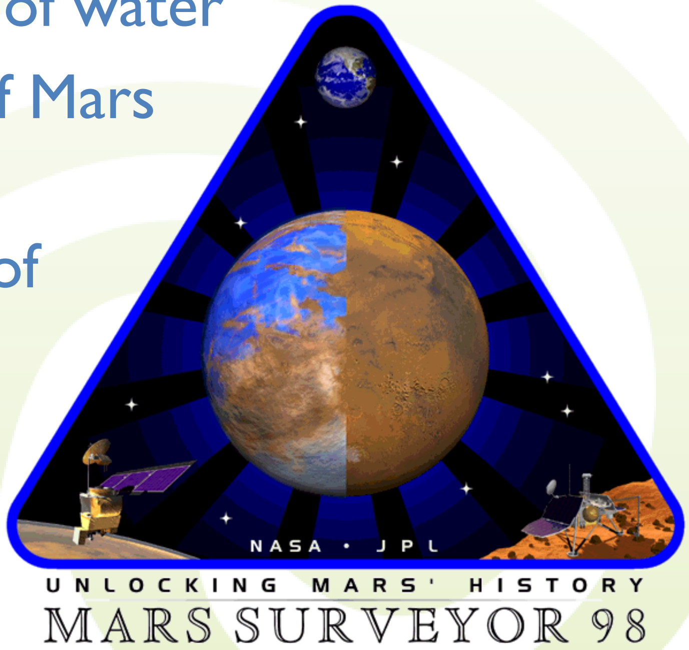
- **View Integration**
  - Several conceptual schemas need to be combined into a unified global schema
  - All differences in perspective and terminology have to be resolved
  - All redundancy has to be removed

- **But,… what happens, if you don't integrate?!**

4.1 The Mars Desaster

Detour

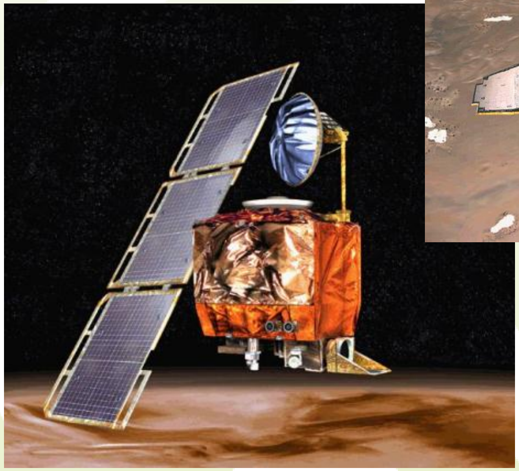
- Example: Big NASA project **Mars Surveyor**
  - The 1998 missions investigated “Volatiles and Climate History” on Mars
    - Characterization of climate change and its evolving impact on the distribution of water
    - Idea: explore the **polar ice caps** of Mars and see whether there is water ice
    - Planned were about three months of experiments on Mars

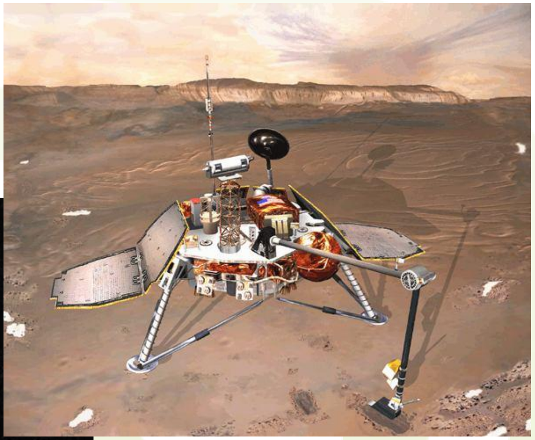


4.1 The Mars Desaster

Detour

- Two vehicles:
  Mars climate orbiter and **Mars polar lander**
  - The lander was supposed to probe the layers of ice and dust on the polar ice caps to investigate changes
  - The orbiter was built to monitor the daily weather and record changes in water vapor and dust in the atmosphere





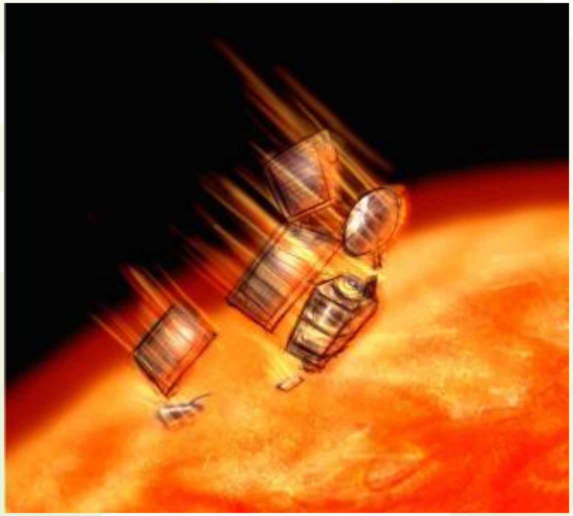
4.1 The Mars Desaster

Detour

- Catastrophic failure
  - The Mars climate orbiter approached Mars up to 57 km instead of 150 km, and was **destroyed** in the atmosphere on September 23, 1999
  - The Mars polar lander **crashed** during its attempted landing on Mars, December 3, 1999
  - **$327.6 million** in total for both
    - $193.1 million for spacecraft development
      $ 91.7 million for launch
      $ 42.8 million for mission operations

4.1 The Mars Desaster

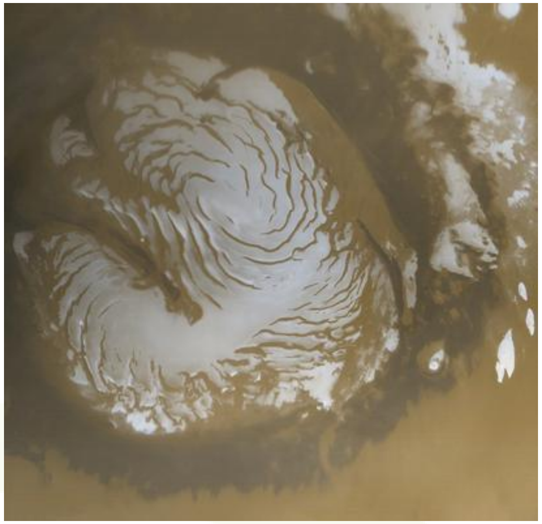
- Why did the **climate orbiter** come too close to Mars' atmosphere?
  - **Many organizations** were involved in the development
  - There was no global schema
    - Navigation software produced by Lockheed Martin used **non-metric** units (e.g., inches, feet, and pounds)
    - NASA uses **metric** units
    - A small correction of the course led to the fatally low orbit…



4.1 The Mars Desaster

Detour

- Happy ending!
  - The next try was the successful **2001 Mars Odyssey**
  - The measurements pointed to **water ice on Mars**
  - Confirmed by the Mars Express (ESA) in 2004
    - The polar caps consist of 85% carbon dioxide (CO_2) ice and 15% water ice



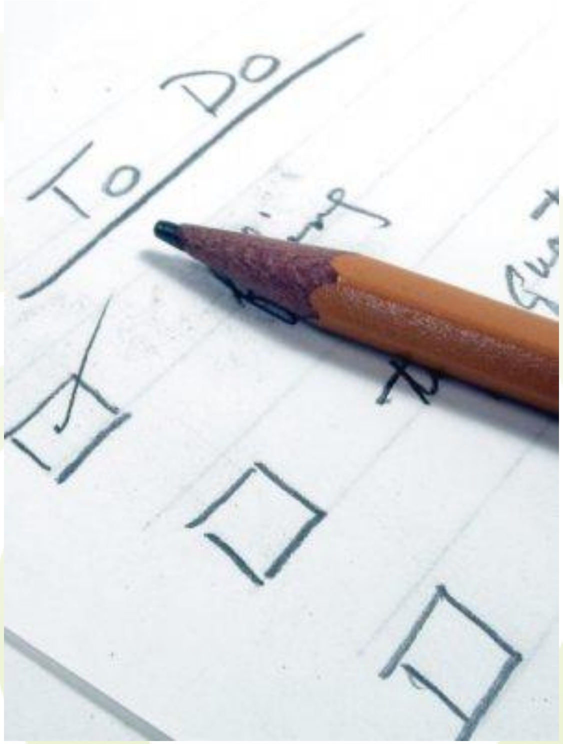
Mars · February 1995
HST · WFPC2
PR95-17 · ST ScI OPO · March 21, 1995
P. James (U.Toledo), S. Lee (U.CO), NASA

4.2 Resolving Incompatibilities

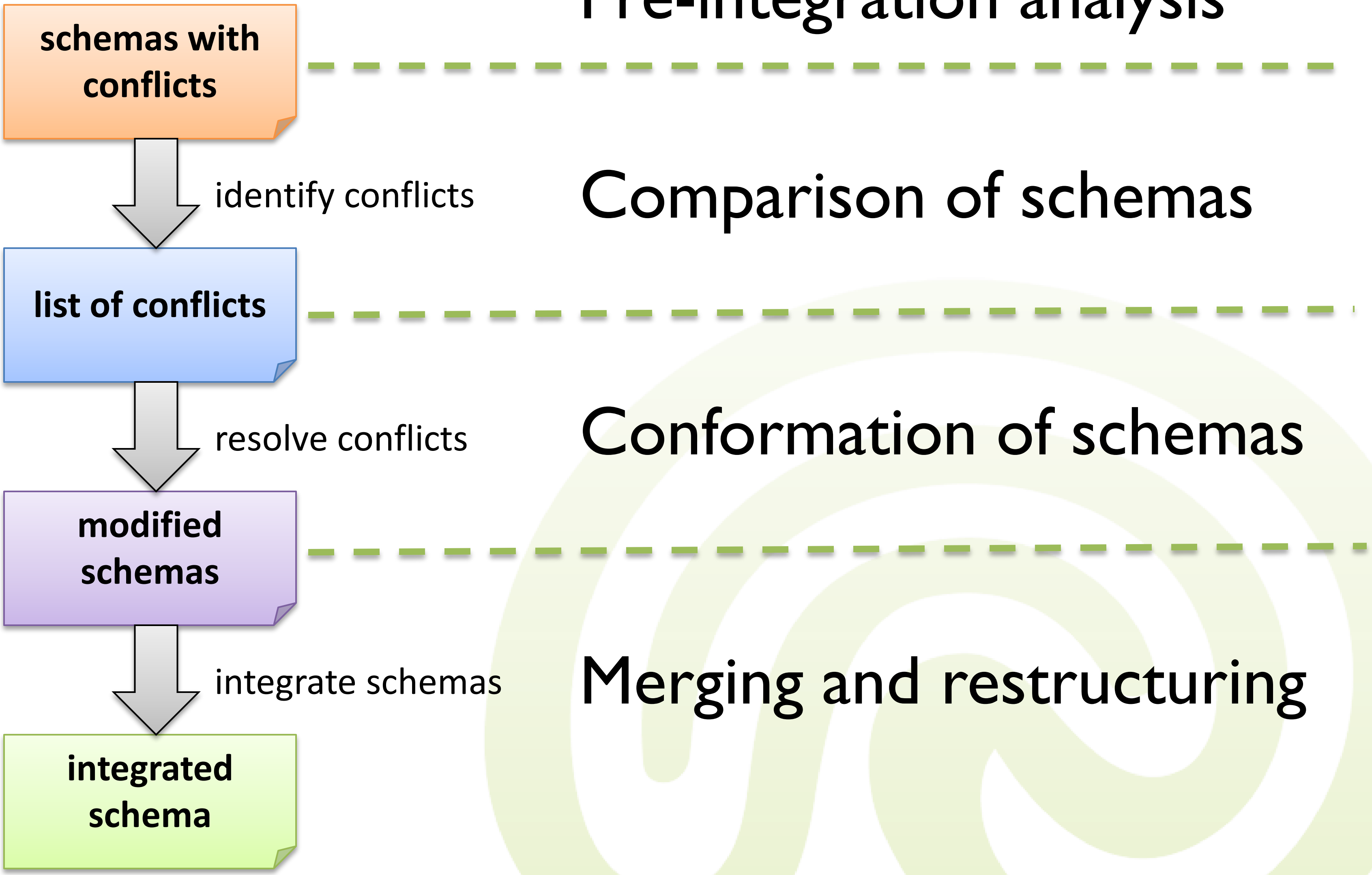
- **Schema diversity** occurs when different users develop their own understanding of the world
  - The same reality is not always modeled in the same way due to different information needs or workflows
- **Common principle** for schema integration
  - Identify the parts of the input schemas that represent the same reality
  - Unify their representations

4.2 Basic Steps

- There are **four basic steps** needed for conceptual schema integration
  1. Pre-integration analysis
  2. Comparison of schemas
  3. Conformation of schemas
  4. Merging and restructuring of schemas
- The integration process needs continual refinement and reevaluation



4.2 Schematic View



Pre-integration analysis

Comparison of schemas

Conformation of schemas

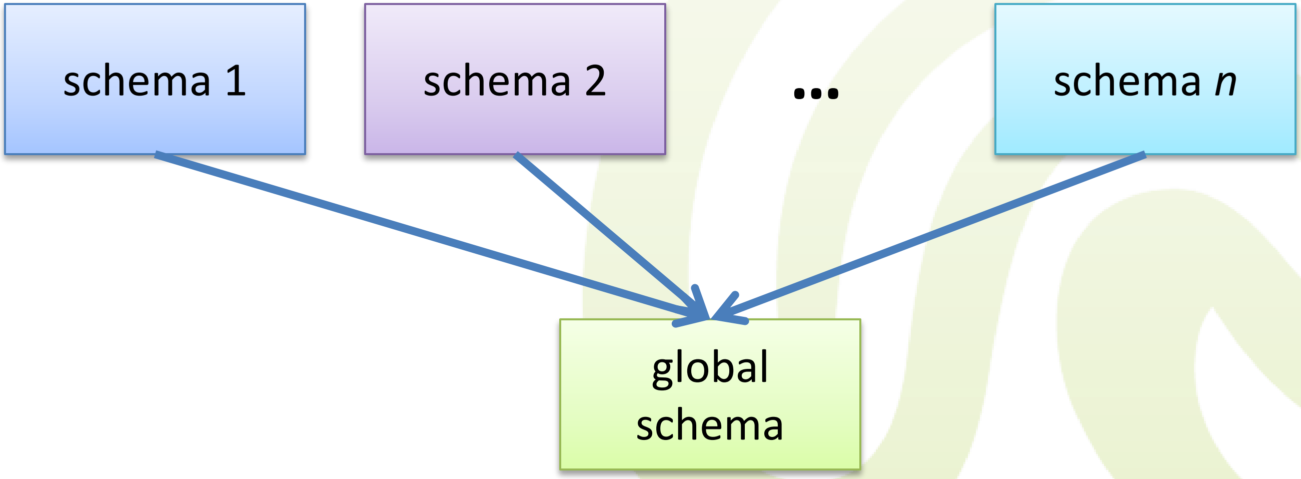
Merging and restructuring

4.2 Pre-integration Analysis

- **Pre-integration analysis** takes a close look on the individual conceptual schemas to decide for an **adequate integration strategy**
  - The larger the number of constructs, the more important is modularization
  - Is it really sensible/possible to integrate all schemas?

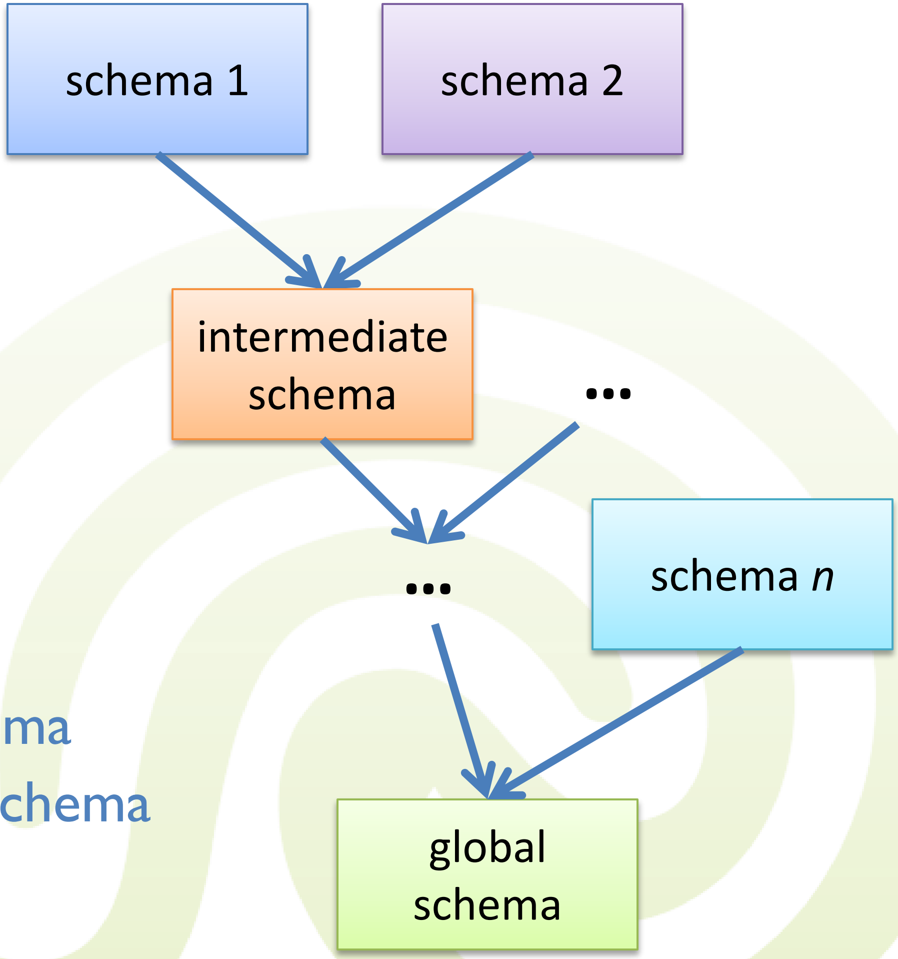
4.2 Pre-integration Analysis

- First, an **integration strategy** has to be chosen
- Schema integration can either be performed **many at a time**, …
  - Requires only one consistent merge
  - Conflict analysis from many schemas is difficult



4.2 Pre-integration Analysis

- … or can be performed **two at a time**
  - Results are step by step accumulated into a single schema
  - How to choose **the right order** of schema comparisons
    - The order can influence the final result
  - Selecting the first schema
    - Mixed strategy: skeleton schema
    - Otherwise: most important schema



4.2 Comparison of Schemas

- The **resolution of conflicts** needs a thorough comparison of schemas
  - General question: How do entities **correspond?**
  - **Naming conflicts** can be detected, e.g., by scanning the data dictionary
  - **Structural conflicts** regarding semantics have to be resolved:
    - Different cardinalities in relationships
    - Key conflicts
    - …

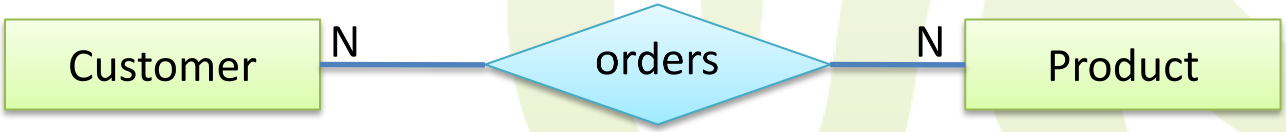
4.2 Comparison of Schemas

- The individual **perspective of the world** and the **level of abstraction** are major reasons for conceptual incompatibilities
  - Example: A **product** is a **unit of sale** for the marketing department, but consists of **parts** in the view of the engineering department

4.2 Comparison of Schemas

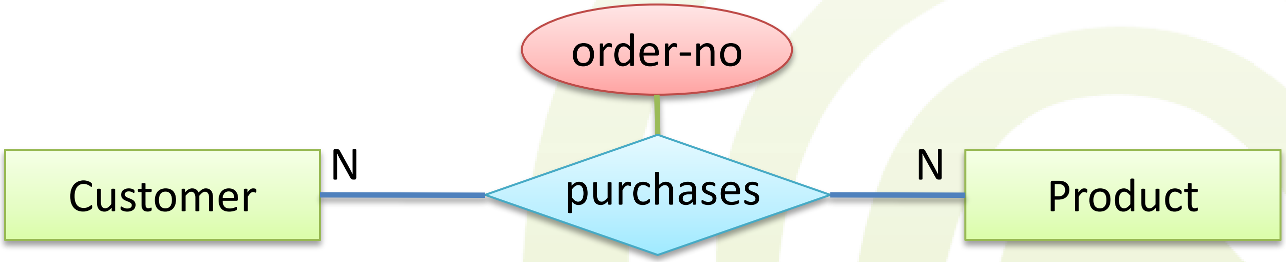
- The **level of abstraction** directly influences the schema design
- **Simple example:** A customer buys a product

- The **marketing view** focuses on how many people buy some product, e.g., for advertising
  - Only the characteristics of the customer and product and the connection are needed



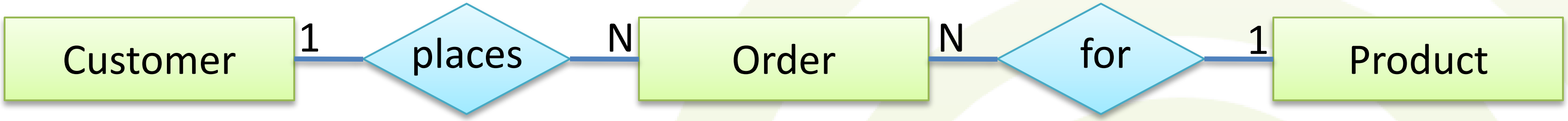
4.2 Comparison of Schemas

- The **accounting view** also needs the exact order number for identification of individual customer transactions
  - Focus is on the purchase, but individual orders have to be distinguished



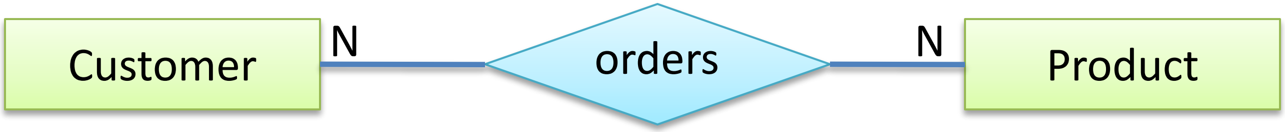
4.2 Comparison of Schemas

- The **sales view** needs all individual order details, e.g., for troubleshooting or CRM
  – Focus is on orders
  (which provide the basis for purchase contracts)

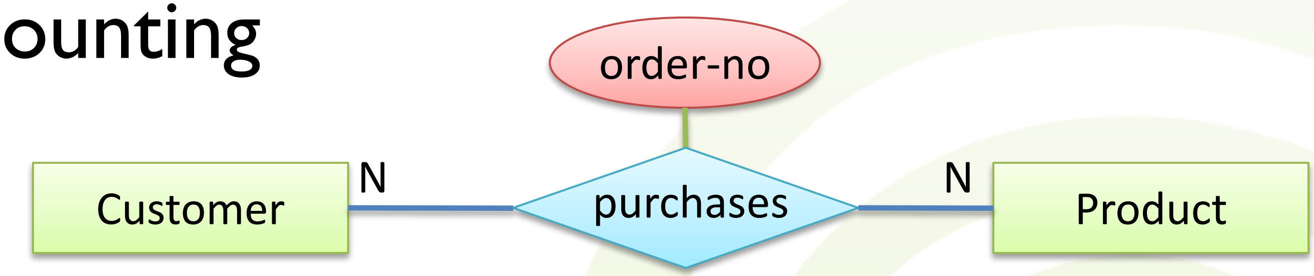


4.2 Comparison of Schemas

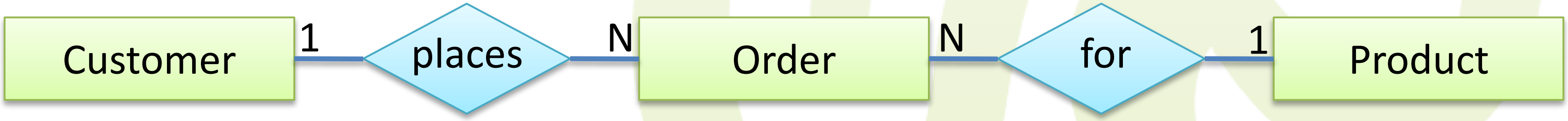
- Marketing



- Accounting



- Sales

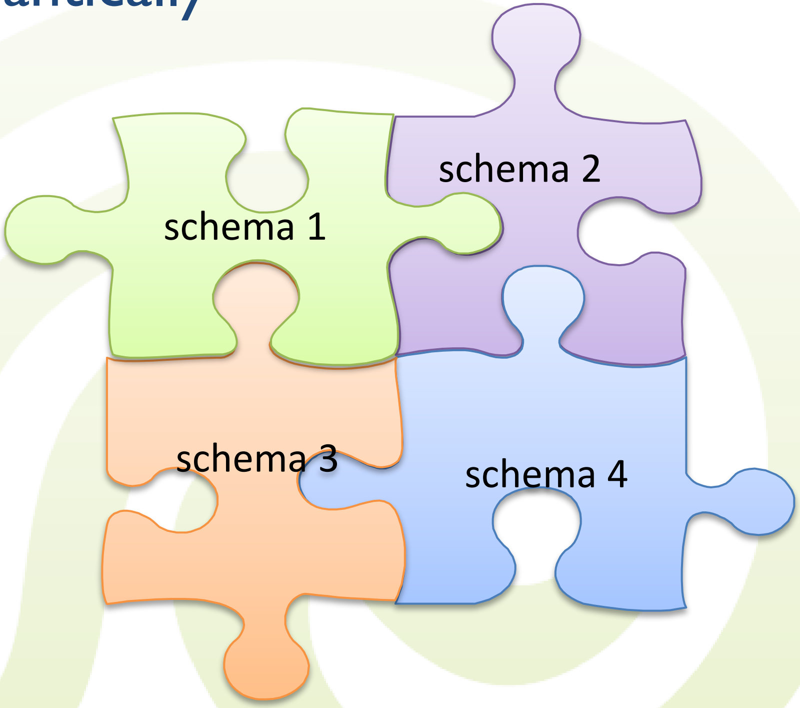


4.2 Comparison of Schemas

- Different user groups use different names to refer to the same entities (differing terminology)
  - **Synonyms:** Two terms for the same entity
  - **Homonym:** The same term for different entities
- **Rule of thumb:** eliminate synonyms, rename homonyms!

4.2 Conformation of Schemas

- The main goal is to make schemas **compatible** for integration
- Conformation usually needs **manual** interaction
  - Conflicts need to be resolved semantically
  - Rename entities/attributes
  - Convert differing types, e.g., convert an entity to an attribute or a relationship
  - Align cardinalities/functionalities
  - Align different data types



4.2 Conformation of Schemas

- Besides renaming and type conversions, **abstraction** can be useful
  - Generalization and aggregation allows to create new supertypes or subtypes
- Also **assertions and constraints** must be generalized or distributed among the type hierachy
  - For example, checking accounts and saving accounts are both types of accounts, but may differ with respect to the minimum balance constraint

4.2 Conformation of Schemas

- **Example:** Resolving differing terminology
- **Homonyms:** Schema 1 and 2 both contain the term "jaguar," but mean different entities
  - Rename to **"jaguar_car"** and **"jaguar_animal"**
- **Synonyms:** Schema 1 contains the term "jaguar," whereas schema 2 contains the term "panther"
  - Global schema should model "**panther" is_a "jaguar"**
  - Constraint on the black color should be added

4.2 Merging and Restructuring

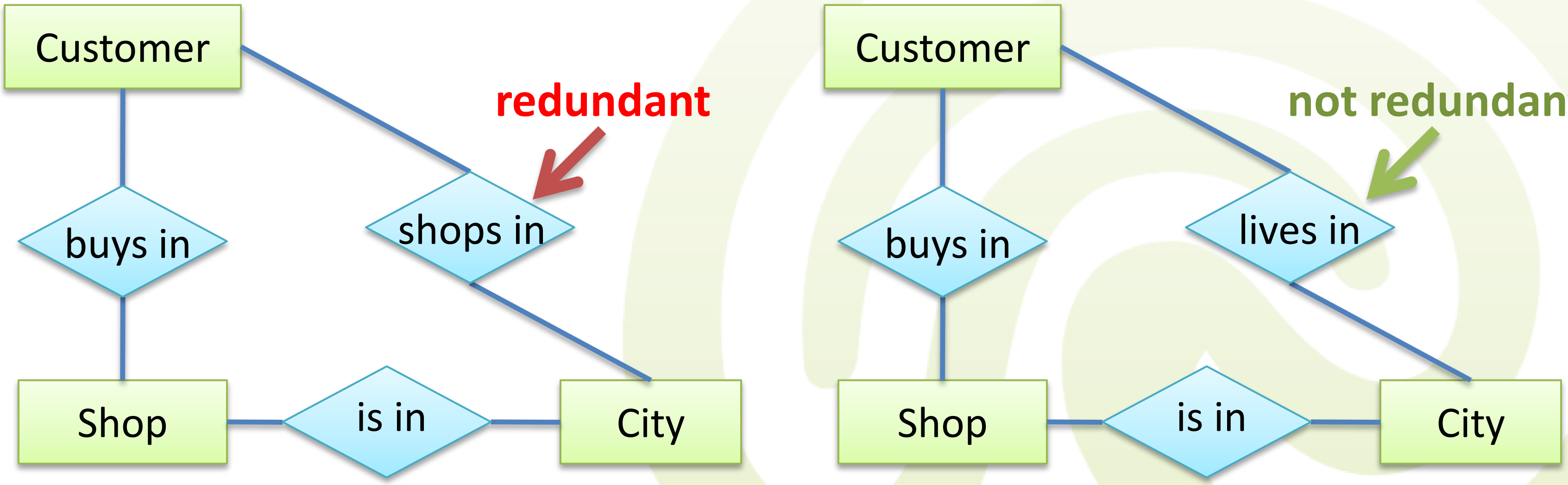
- How to merge schemas into a **global schema**?
  - Copy all distinct **entities** from the individual schemas
  - Apply renaming, overlapping entity integration, abstraction, attribute type conversions, etc.
  - Put in the distinct **relationships** from all schemas
  - Again use renaming, cardinality/functionality conversions, etc.
  - Restructure the resulting global schema

4.2 Merging and Restructuring

- The final **restructuring of the schema** is driven by the goal of completeness, minimality, and ease of understanding
  - **Completeness** mandates that all concepts in the global schema appear "semantically intact"
    - All different concepts of every individual schema are also part of the global schema
    - For each concept, there are no missing attributes, no constraints that cannot be met by all members of a type, etc.

4.2 Merging and Restructuring

- **Minimality** enforces to remove all redundant concepts from the global schema
  - For example, overlapping entities or redundant relationships
  - Often, the question of minimality can only be decided semantically



Customer

buys in

shops in

Shop

is in

City

redundant

Customer

buys in

lives in

Shop

is in

City

not redundant

4.2 Merging and Restructuring

- **Ease of understanding** means that the global schema makes sense to the users
  - In particular, abstraction and fine granular levels for entities can be very confusing
  - Example: Subtype entities have to be clearly distinguishable, and should have only attributes that are not inherited from the supertype

Example: View Integration

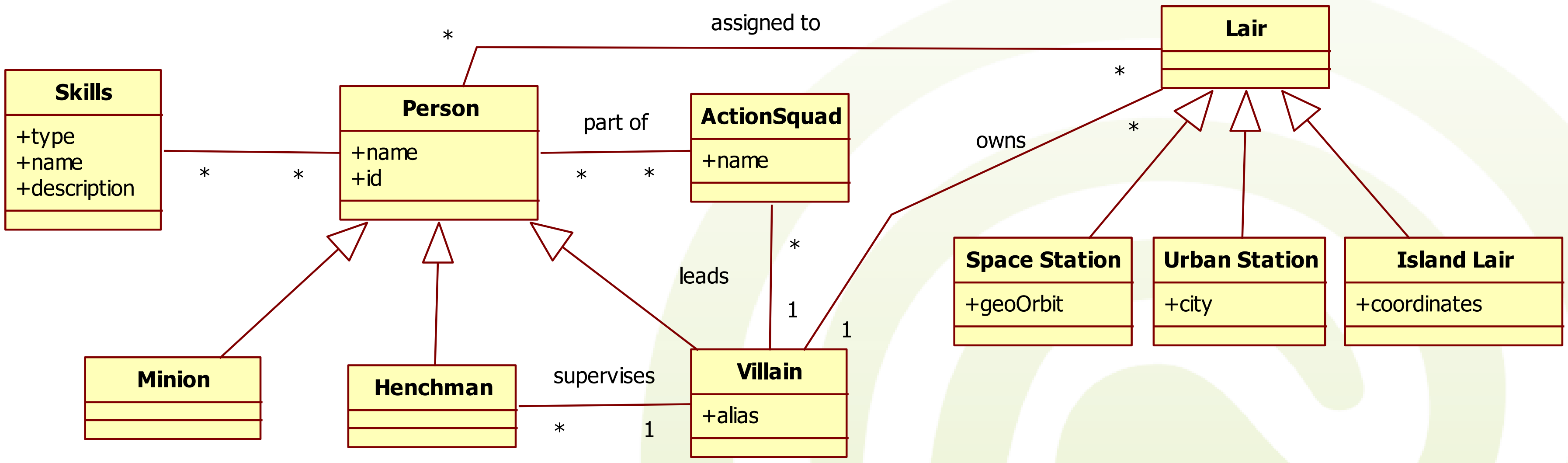
Detour

- **Doomsday Legion (DDL)**
  - Cooperation of villains from all over the world striving for global domination
  - Channeling resources, staff, experience and power for reaching their goals
  - Centralized and coordinated management of all shared assets
    - Lairs
    - Personnel
    - Assault squads
    - …

Example: View Integration

Detour

- Doomsday Legion schema



Example: View Intgration

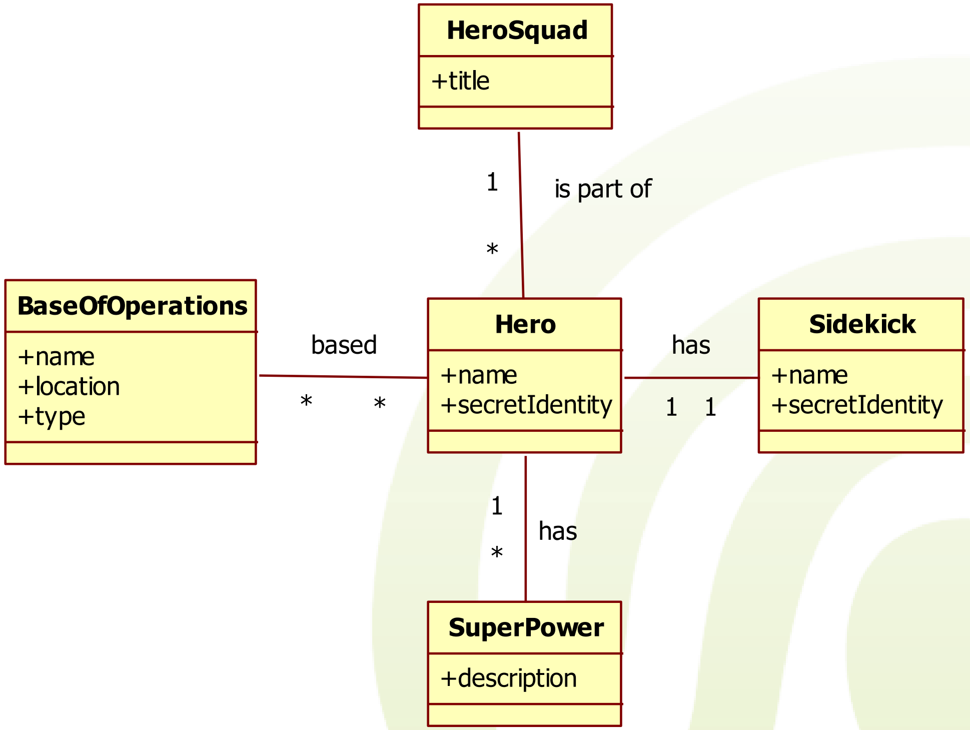
Detour

- **Justice League (JL)**
  - Federation of super-powered heroes fighting against global crime and villainousness
    - In particular: Opposing the Doomsday Legion
  - Central management of joint operations and resources

Example: View Integration

Detour

- **Justice League schema**



Example: View Integration

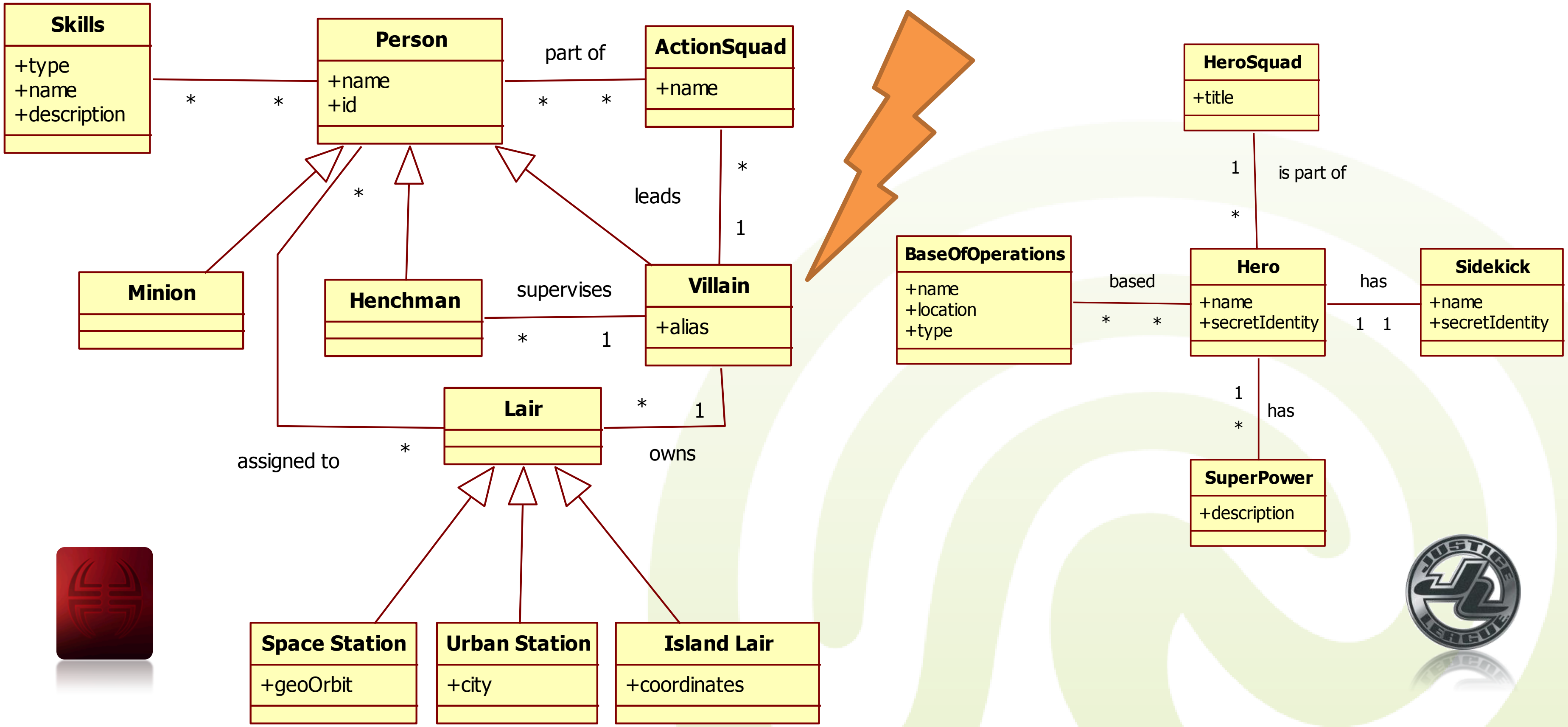
Detour

- "And then, strange and evil **aliens invade earth** without any obvious reason"
  - Justice League wants to save earth (that's what they do)
  - Doomsday Legion wants to save earth (without people, global domination is no fun)
  - Great Idea: **Join Forces**
    - "**D**efenders **o**f **t**he **E**arth"
  - Great Problem: Joining large organizations is **not that easy**
    - Beside the problem of ignoring old hatred, the data **schemas need to be integrated** for central mission control and planning

Example: View Integration

Detour

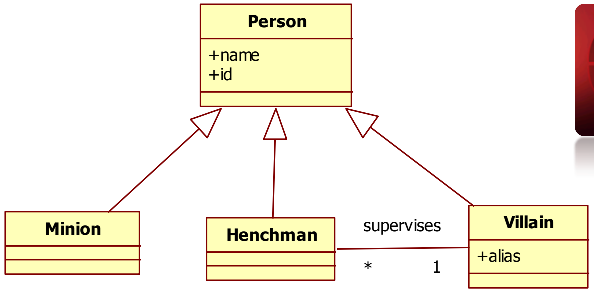
- How to integrate?

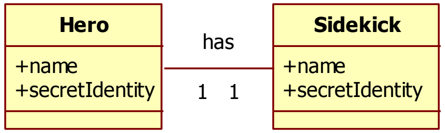


Example: View Integration

Detour

- Integrating the person models



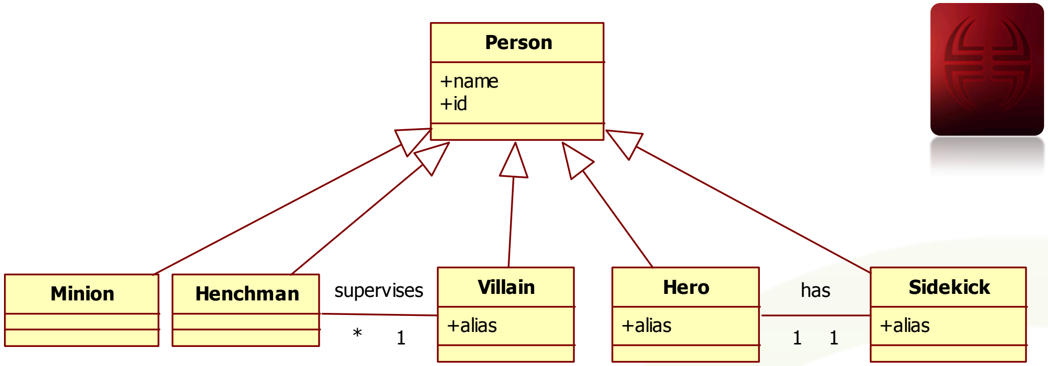


  - Different structure
    - DDL more general → Merge JL into DDL
      - Generalize Hero and Sidekick into Person
    - But: **Attribute Homonyms**!
      - DDL uses the real name of "*name*," the villain identity is "*alias*"
      - JL puts real name into "*secret identity*" and hero name into "*name*"
      - That is: (*name*: "Victor von Doom", *alias* "Dr. Doom")
        (*name*: "Invisible Woman", *secret identity* "Susan Storm")
      - Attributes need to be renamed and transformed correctly

Example: View Integration

Detour

- Integrating the person models

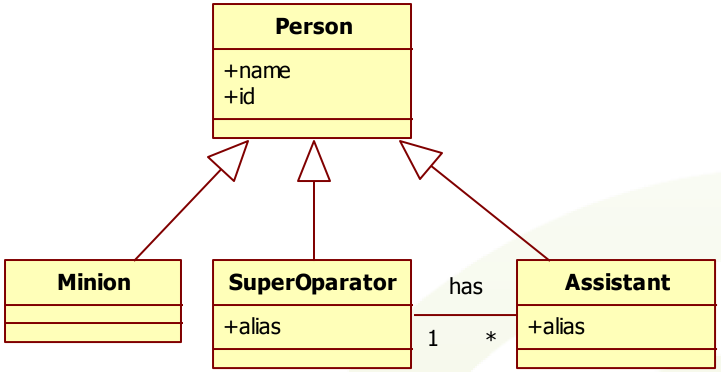


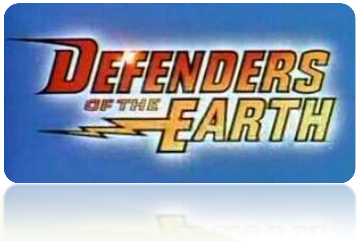
- **Semantic consolidation**
  - Hero and Villains are should be treated the same
    - Both are highly skilled and powerful super members of DotE
    - Merge classes into *SuperOperator* class
  - Sidekicks and henchmen are close *assistants* of an operator
    - Heroes usually only have one sidekick
    - Use more general 1:n association to also capture henchmen

Example: View Integration

Detour

- Integrating the person models

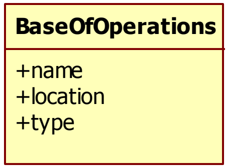




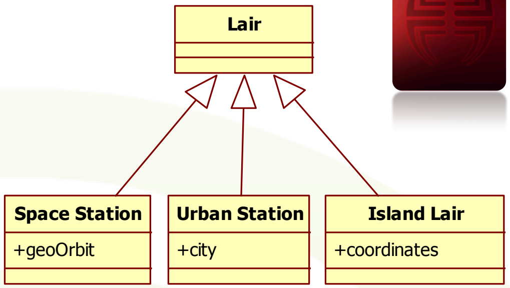
  - Contains heroes and villains, as well as their respective sidekicks or henchmen
  - Heroes and sidekicks get an additional id

Example: View Integration

Detour



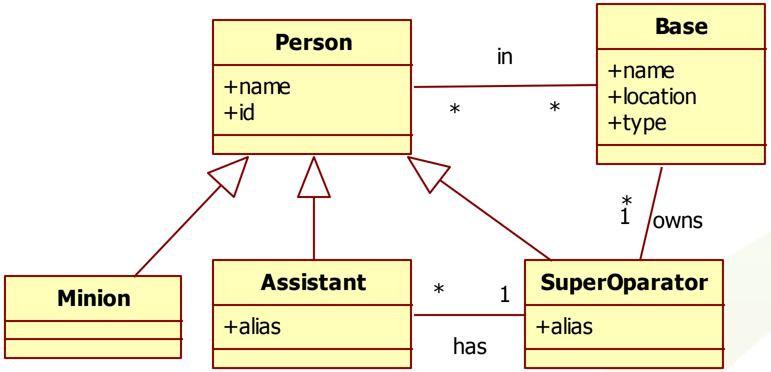
- Integrating bases
  - Villains only have 3 types of bases, explicitly modeled
  - Heroes may have any kind of base, given by the type attribute
  - Two solutions
    - **Merge DDL into JL**
      - *geoOrbit*, *city*, and *coordinates* become *location*
      - *type* is given by subclass
      - Only possible in a lossless fashion because subtypes don't have additional attributes
    - Merge JL into DDL
      - Depending on type, a base is assigned to one of the subclasses
      - 4th subclass necessary for all other types (could also be merged into superclass)

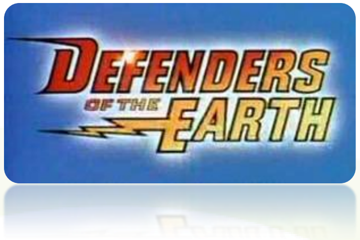


Example: View Integration

Detour

- Integrating bases

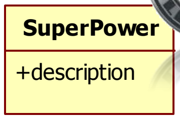


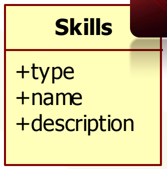


  - Only villains owned lairs; no information of ownership for former hero bases

Example: View Integration

- Integrating skills and powers
  - JL only stores super powers
  - DDL stores all skills (including super powers)
    - More general
  - Merge JL into DDL
    - All old justice league super powers become skills of the *type* “super power”
    - *name* is either null or manually completed
    - No information on skills of sidekicks

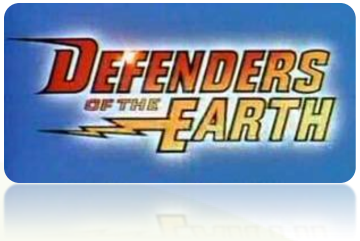


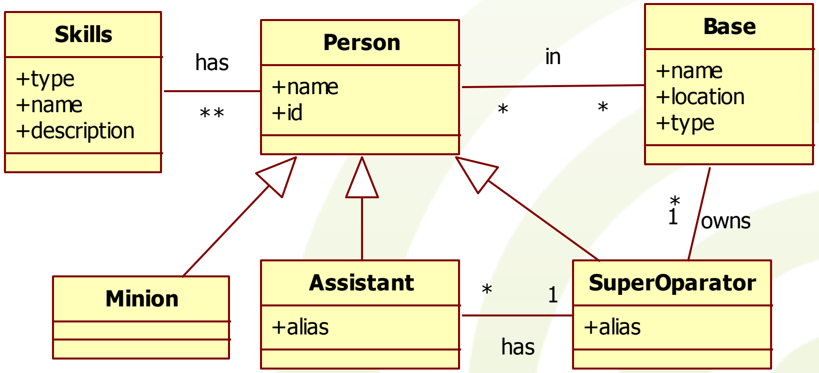


Example: View Integration

Detour

- Integrated schema





4.2 Outlook

- View integration is a **semantic process**
  - This usually means a lot of **manual work**
  - Computers can support the process by **matching** some (parts of) schemas
- There have been some approaches towards **(semi-)automatic matching** of schemas
  - Matching is a complex process and usually only focuses on simple constructs like
    “Are two entities semantically equivalent?”
  - The result is still rather error-prone…

4.2 Outlook

- **Basic methods** (that can of course be mixed freely)
  - **Label-based matching**
    - For each label in one schema, consider all labels of the other schema and every time gauge their semantic similarity
  - **Instance-based matching**
    - Looking at the instances (of entities or relationships) one can e.g., find correlations between attributes:
      “Are there duplicate tuples?” or
      “Are the data distributions in their respective domains similar?”
  - **Structure-based matching**
    - Abstracting from the actual labels, only the structure of the schema is evaluated, e.g., regarding element types, depths in hierarchies, number and type of relationships, etc.

4.2 Outlook

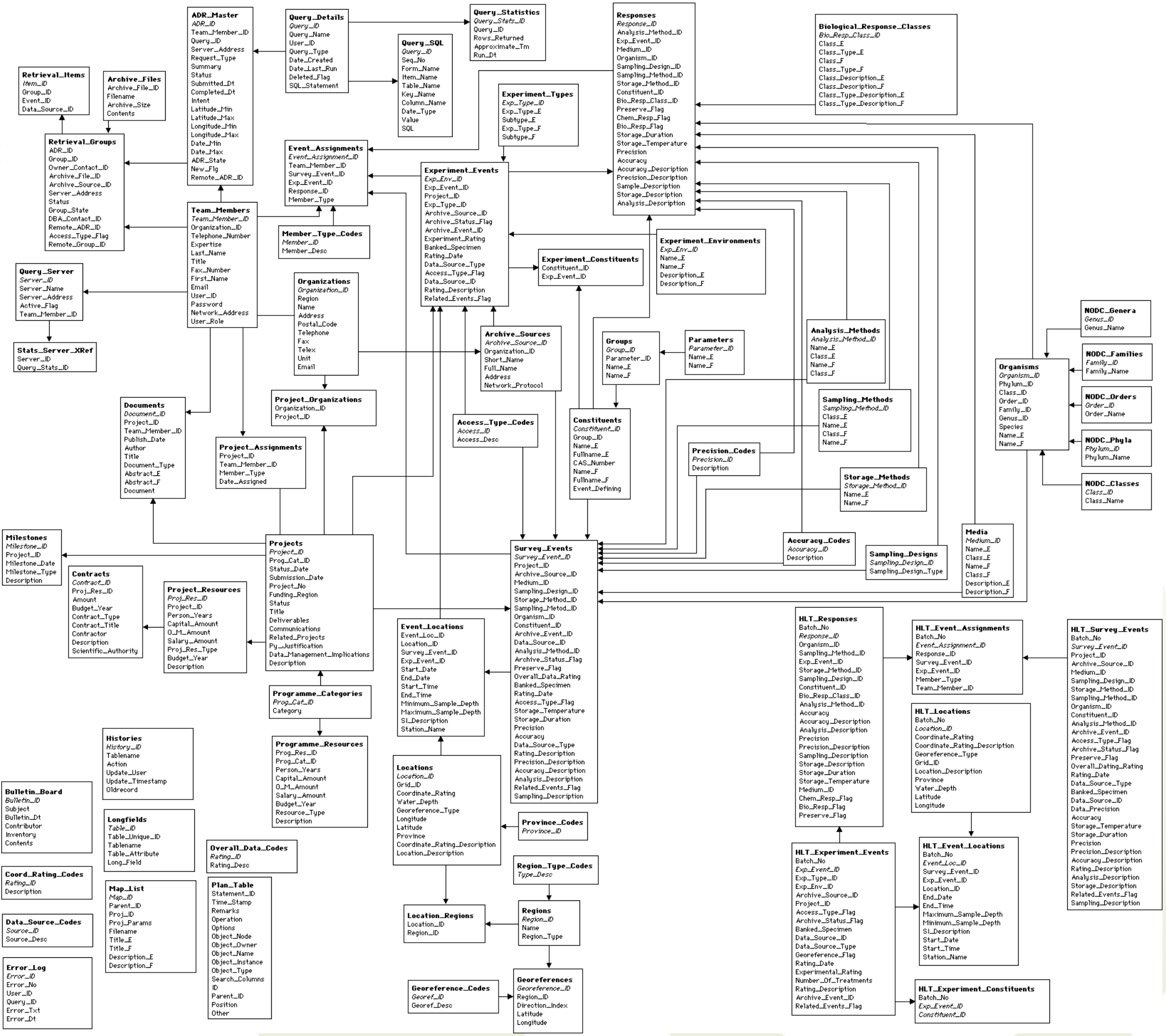
- Sometimes schema integration is **query-driven**
  - The integration is only needed in order to query several different information sources having different schemas
- In that case only a **schema mapping** is needed
  - Basically the mapping is a **list of correspondences** between equivalent entities or relationships of heterogeneous schemas
  - The query can then be **translated** for each different schema using the mapping
  - The mapping can be derived manually or automatically from a respective matching

4.3 Entity Clustering

- **Sample schema integration result**

  National Contaminants Information System (NCIS)

- © Fisheries and Oceans, Canada



ADR_Master
ADR_ID
Team_Member_ID
Query_ID
Server_Address
Request_Type
Summary
Status
Submitted_Dt
Completed_Dt
Intent
Latitude_Min
Latitude_Max
Longitude_Min
Longitude_Max
Date_Min
Date_Max
ADR_State
New_Flg
Remote_ADR_ID

Query_Details
Query_ID
Query_Name
User_ID
Query_Type
Date_Created
Date_Last_Run
Deleted_Flag
SQL_Statement

Query_Statistics
Query_Stats_ID
Query_ID
Rows_Returned
Approximate_Tm
Run_Dt

Query_SQL
Query_ID
Seq_No
Form_Name
Item_Name
Table_Name
Key_Name
Column_Name
Date_Type
Value
SQL

Responses
Response_ID
Analysis_Method_ID
Exp_Event_ID
Medium_ID
Organism_ID
Sampling_Design_ID
Sampling_Method_ID
Storage_Method_ID
Constituent_ID
Bio_Resp_Class_ID
Preserve_Flag
Chem_Resp_Flag
Bio_Resp_Flag
Storage_Duration
Storage_Temperature
Precision
Accuracy
Accuracy_Description
Precision_Description
Sample_Description
Storage_Description
Analysis_Description

Biological_Response_Classes
Bio_Resp_Class_ID
Class_E
Class_Type_E
Class_F
Class_Type_F
Class_Description_E
Class_Description_F
Class_Type_Description_E
Class_Type_Description_F

Retrieval_Items
Item_ID
Group_ID
Event_ID
Data_Source_ID

Archive_Files
Archive_File_ID
Filename
Archive_Size
Contents

Retrieval_Groups
ADR_ID
Group_ID
Owner_Contact_ID
Archive_File_ID
Archive_Source_ID
Server_Address
Status
Group_State
DBA_Contact_ID
Remote_ADR_ID
Access_Type_Flag
Remote_Group_ID

Experiment_Types
Exp_Type_ID
Exp_Type_E
Subtype_E
Exp_Type_F
Subtype_F

Event_Assignments
Event_Assignment_ID
Team_Member_ID
Survey_Event_ID
Exp_Event_ID
Response_ID
Member_Type

Experiment_Events
Exp_Env_ID
Exp_Event_ID
Project_ID
Exp_Type_ID
Archive_Source_ID
Archive_Status_Flag
Archive_Event_ID
Experiment_Rating
Banked_Specimen
Rating_Date
Data_Source_Type
Access_Type_Flag
Data_Source_ID
Rating_Description
Related_Events_Flag

Team_Members
Team_Member_ID
Organization_ID
Telephone_Number
Expertise
Last_Name
Title
Fax_Number
First_Name
Email
User_ID
Password
Network_Address
User_Role

Member_Type_Codes
Member_ID
Member_Desc

Experiment_Environments
Exp_Env_ID
Name_E
Name_F
Description_E
Description_F

Experiment_Constituents
Constituent_ID
Exp_Event_ID

Query_Server
Server_ID
Server_Name
Server_Address
Active_Flag
Team_Member_ID

Organizations
Organization_ID
Region
Name
Address
Postal_Code
Telephone
Fax
Telex
Unit
Email

NODC_Genera
Genus_ID
Genus_Name

Analysis_Methods
Analysis_Method_ID
Name_E
Class_E
Name_F
Class_F

Archive_Sources
Archive_Source_ID
Organization_ID
Short_Name
Full_Name
Address
Network_Protocol

Groups
Group_ID
Parameter_ID
Name_E
Name_F

Parameters
Parameter_ID
Name_E
Name_F

Stats_Server_XRef
Server_ID
Query_Stats_ID

Organisms
Organism_ID
Phylum_ID
Class_ID
Order_ID
Family_ID
Genus_ID
Species
Name_E
Name_F

NODC_Families
Family_ID
Family_Name

Project_Organizations
Organization_ID
Project_ID

Sampling_Methods
Sampling_Method_ID
Class_E
Name_E
Class_F
Name_F

NODC_Orders
Order_ID
Order_Name

Documents
Document_ID
Project_ID
Team_Member_ID
Publish_Date
Author
Title
Document_Type
Abstract_E
Abstract_F
Document

Access_Type_Codes
Access_ID
Access_Desc

Constituents
Constituent_ID
Group_ID
Name_E
Fullname_E
CAS_Number
Name_F
Fullname_F
Event_Defining

NODC_Phyla
Phylum_ID
Phylum_Name

Project_Assignments
Project_ID
Team_Member_ID
Member_Type
Date_Assigned

Precision_Codes
Precision_ID
Description

Storage_Methods
Storage_Method_ID
Name_E
Name_F

NODC_Classes
Class_ID
Class_Name

Milestones
Milestone_ID
Project_ID
Milestone_Date
Milestone_Type
Description

Projects
Project_ID
Prog_Cat_ID
Status_Date
Submission_Date
Project_No
Funding_Region
Status
Title
Deliverables
Communications
Related_Projects
Py_Justification
Data_Management_Implications
Description

Survey_Events
Survey_Event_ID
Project_ID
Archive_Source_ID
Medium_ID
Sampling_Design_ID
Storage_Method_ID
Sampling_Metod_ID
Organism_ID
Constituent_ID
Archive_Event_ID
Data_Source_ID
Analysis_Method_ID
Archive_Status_Flag
Preserve_Flag
Overall_Data_Rating
Banked_Specimen
Rating_Date
Access_Type_Flag
Storage_Temperature
Storage_Duration
Precision
Accuracy
Data_Source_Type
Rating_Description
Precision_Description
Accuracy_Description
Analysis_Description
Related_Events_Flag
Sampling_Description

Accuracy_Codes
Accuracy_ID
Description

Sampling_Designs
Sampling_Design_ID
Sampling_Design_Type

Media
Medium_ID
Name_E
Class_E
Name_F
Class_F
Description_E
Description_F

Contracts
Contract_ID
Proj_Res_ID
Amount
Budget_Year
Contract_Type
Contract_Title
Contractor
Description
Scientific_Authority

Project_Resources
Proj_Res_ID
Project_ID
Person_Years
Capital_Amount
O_M_Amount
Salary_Amount
Proj_Res_Type
Budget_Year
Description

Event_Locations
Event_Loc_ID
Location_ID
Survey_Event_ID
Exp_Event_ID
Start_Date
End_Date
Start_Time
End_Time
Minimum_Sample_Depth
Maximum_Sample_Depth
SI_Description
Station_Name

HLT_Responses
Batch_No
Response_ID
Organism_ID
Sampling_Method_ID
Exp_Event_ID
Storage_Method_ID
Sampling_Design_ID
Constituent_ID
Bio_Resp_Class_ID
Analysis_Method_ID
Accuracy
Accuracy_Description
Analysis_Description
Precision
Precision_Description
Sampling_Description
Storage_Description
Storage_Duration
Storage_Temperature
Medium_ID
Chem_Resp_Flag
Bio_Resp_Flag
Preserve_Flag

HLT_Event_Assignments
Batch_No
Event_Assignment_ID
Response_ID
Survey_Event_ID
Exp_Event_ID
Member_Type
Team_Member_ID

HLT_Survey_Events
Batch_No
Survey_Event_ID
Project_ID
Archive_Source_ID
Medium_ID
Sampling_Design_ID
Storage_Method_ID
Sampling_Method_ID
Organism_ID
Constituent_ID
Analysis_Method_ID
Archive_Event_ID
Access_Type_Flag
Archive_Status_Flag
Preserve_Flag
Overall_Dating_Rating
Rating_Date
Data_Source_Type
Banked_Specimen
Data_Source_ID
Data_Precision
Accuracy
Storage_Temperature
Storage_Duration
Precision
Precision_Description
Accuracy_Description
Rating_Description
Analysis_Description
Storage_Description
Related_Events_Flag
Sampling_Description

Programme_Categories
Prog_Cat_ID
Category

HLT_Locations
Batch_No
Location_ID
Coordinate_Rating
Coordinate_Rating_Description
Georeference_Type
Grid_ID
Location_Description
Province
Water_Depth
Latitude
Longitude

Histories
History_ID
Tablename
Action
Update_User
Update_Timestamp
Oldrecord

Programme_Resources
Prog_Res_ID
Prog_Cat_ID
Person_Years
Capital_Amount
O_M_Amount
Salary_Amount
Budget_Year
Resource_Type
Description

Locations
Location_ID
Grid_ID
Coordinate_Rating
Water_Depth
Georeference_Type
Longitude
Latitude
Province
Coordinate_Rating_Description
Location_Description

Bulletin_Board
Bulletin_ID
Subject
Bulletin_Dt
Contributor
Inventory
Contents

Longfields
Table_ID
Table_Unique_ID
Tablename
Table_Attribute
Long_Field

Province_Codes
Province_ID

Overall_Data_Codes
Rating_ID
Rating_Desc

HLT_Event_Locations
Batch_No
Event_Loc_ID
Survey_Event_ID
Exp_Event_ID
Location_ID
End_Date
End_Time
Maximum_Sample_Depth
Minimum_Sample_Depth
SI_Description
Start_Date
Start_Time
Station_Name

HLT_Experiment_Events
Batch_No
Exp_Event_ID
Exp_Type_ID
Exp_Env_ID
Archive_Source_ID
Project_ID
Access_Type_Flag
Archive_Status_Flag
Banked_Specimen
Data_Source_ID
Data_Source_Type
Georeference_Flag
Rating_Date
Experimental_Rating
Number_Of_Treatments
Rating_Description
Archive_Event_ID
Related_Events_Flag

Coord_Rating_Codes
Rating_ID
Description

Map_List
Map_ID
Parent_ID
Proj_ID
Proj_Params
Filename
Title_E
Title_F
Description_E
Description_F

Plan_Table
Statement_ID
Time_Stamp
Remarks
Operation
Options
Object_Node
Object_Owner
Object_Name
Object_Instance
Object_Type
Search_Columns
ID
Parent_ID
Position
Other

Region_Type_Codes
Type_Desc

Data_Source_Codes
Source_ID
Source_Desc

Location_Regions
Location_ID
Region_ID

Regions
Region_ID
Name
Region_Type

Error_Log
Error_ID
Error_No
User_ID
Query_ID
Error_Txt
Error_Dt

Georeference_Codes
Georef_ID
Georef_Desc

Georeferences
Georeference_ID
Region_ID
Direction_Index
Latitude
Longitude

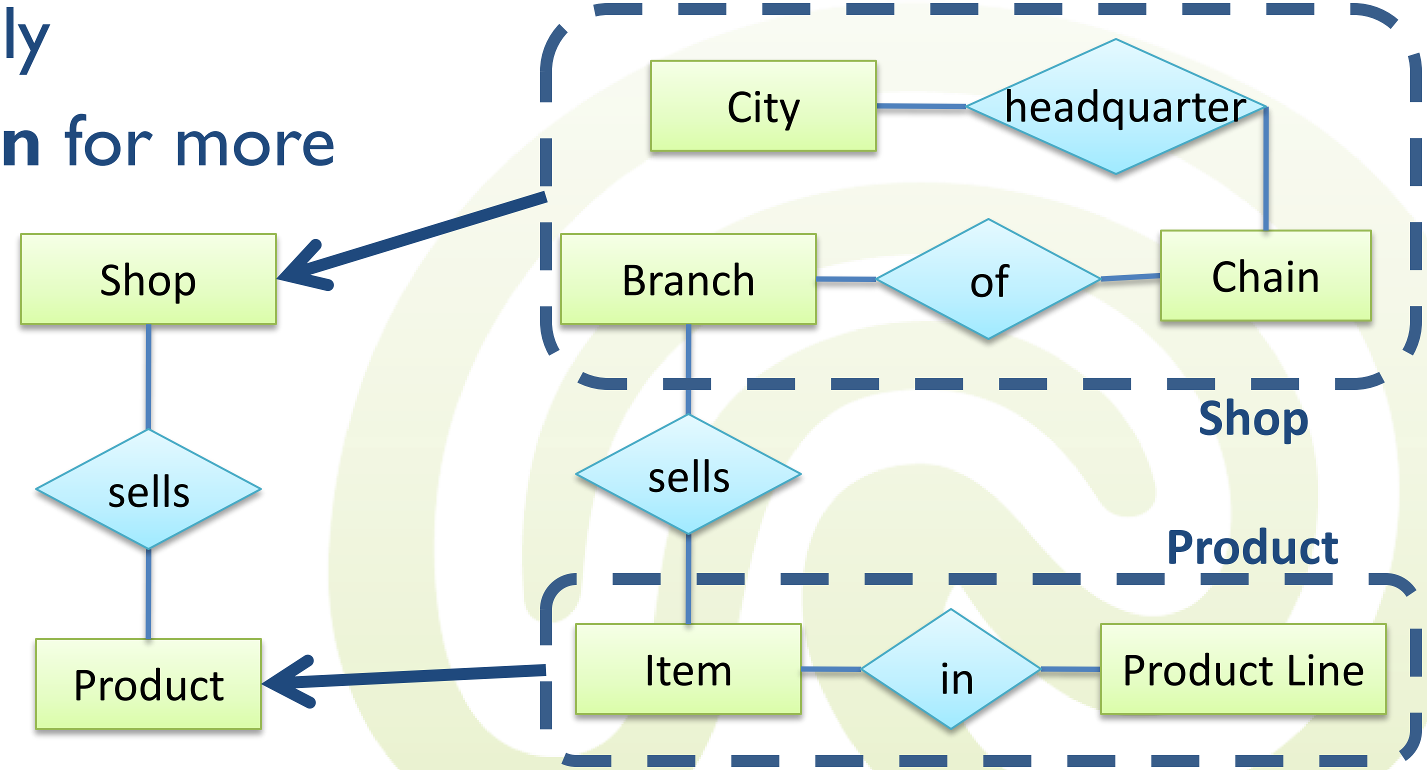
HLT_Experiment_Constituents
Batch_No
Exp_Event_ID
Constituent_ID

4.3 Entity Clustering

- When multiple schemas are merged, global schema can become **very large**
  - Many different entities and relationships between them
    - E.g., global view of a company with all its dependencies
  - But some parts from different views are entirely independent
    - E.g., accounting does not need technical specifications of products
- **Idea:** Cluster **semantically coherent parts** and abstract from their actual entities in the global schema

4.3 Entity Clustering

- Abstracting complex units allow showing the entire model on a **single sheet of paper**
  - Easy to get an overview and easier to integrate units separately
  - **Zoom in** for more details

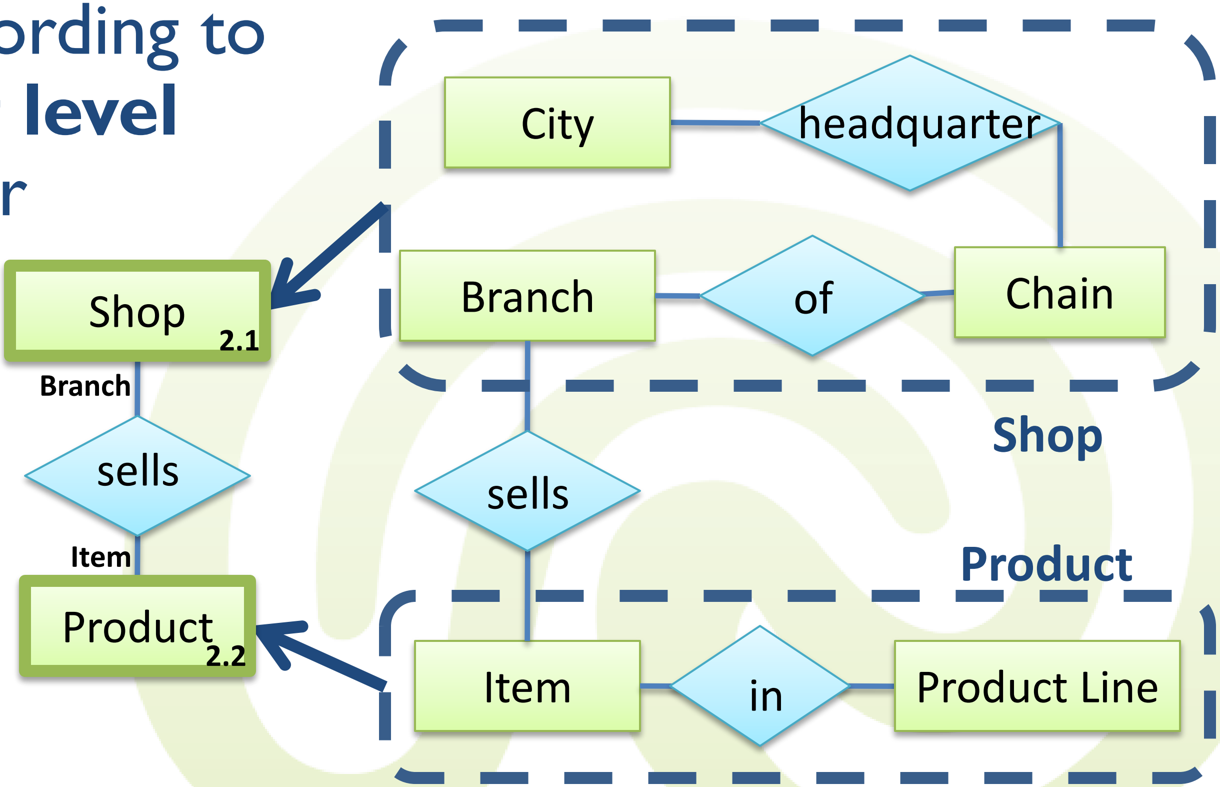


4.3 Clustering Concepts

- **Grouping** is an operation that combines entities and their relationships to form higher-level constructs
  - Groups are called **entity clusters**
  - Can also be performed **hierarchically** from the entire database (root entity cluster) over several levels down to the individual entities
    - All original entities are on clustering level 1

4.3 Clustering Concepts

- Usually, **entity clusters** are depicted similar to normal entities in ER diagrams
  - By a **dark-bordered** box
  - Numbered according to the **clustering level** and an identifier
  - Interfaces for **inter-cluster** relationships have to be annotated

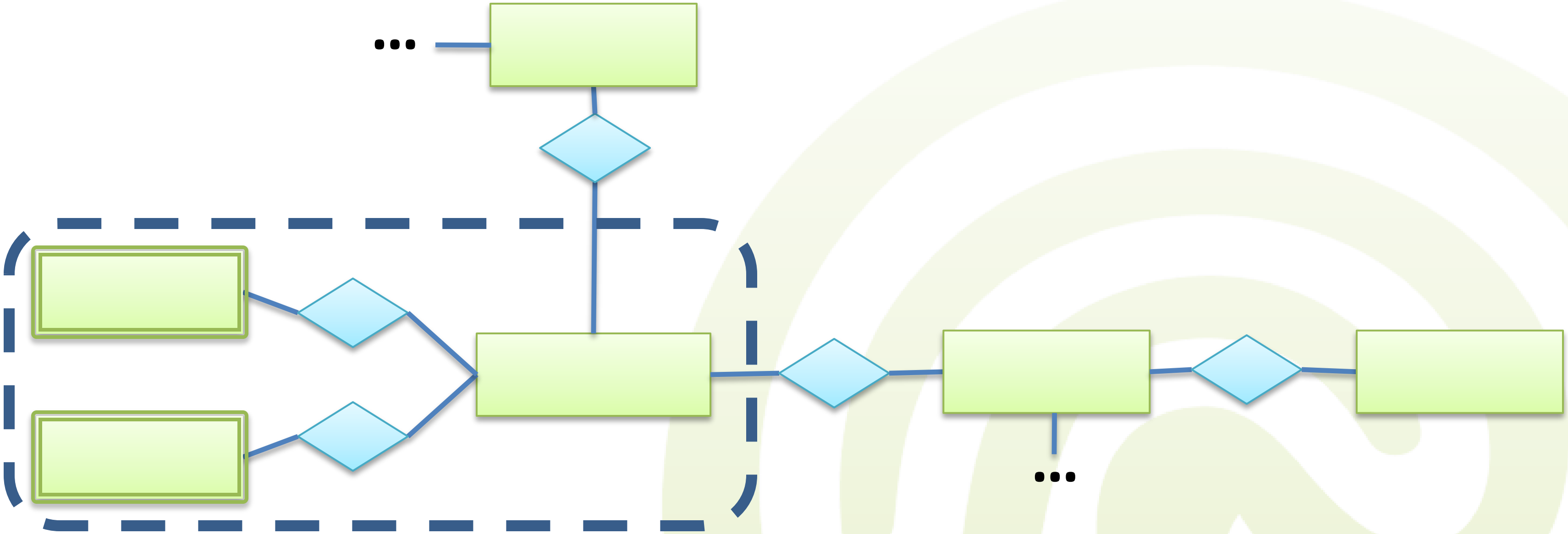


4.3 Grouping Operations

- **Grouping operations** are the fundamental components of entity clustering
  - All operations are **heuristic** in nature
- **Often occurring** operations are
  - Dominance grouping
  - Abstraction grouping
  - Constraint grouping
  - Relationship grouping
- They can be applied **recursively** or in a variety of combinations to produce higher level clusters

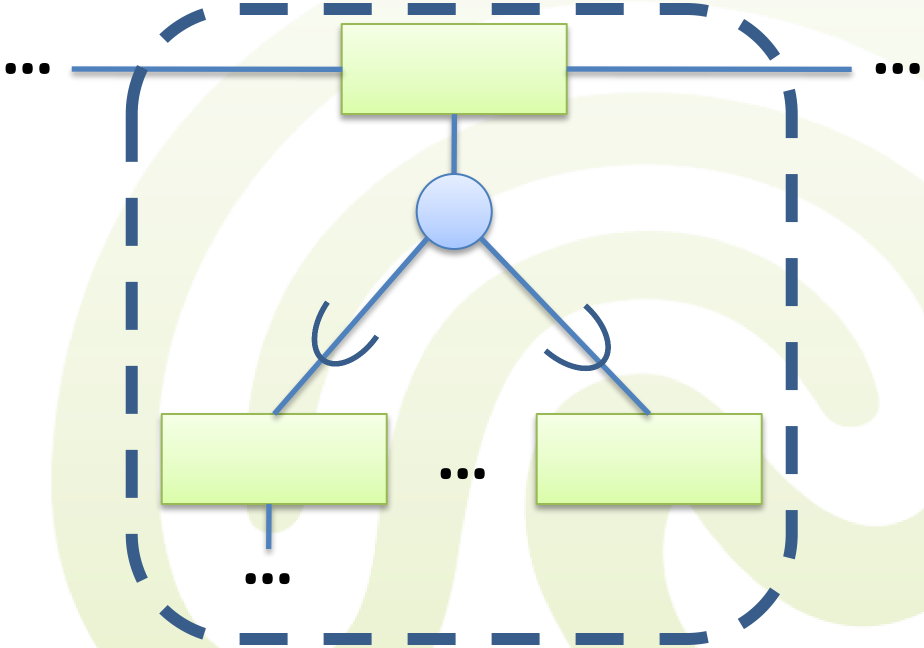
4.3 Grouping Operations

- **Dominance grouping** focuses on semantically dominant entities in the ER diagrams
  - Hubs for **otherwise unconnected** or **weak** entities



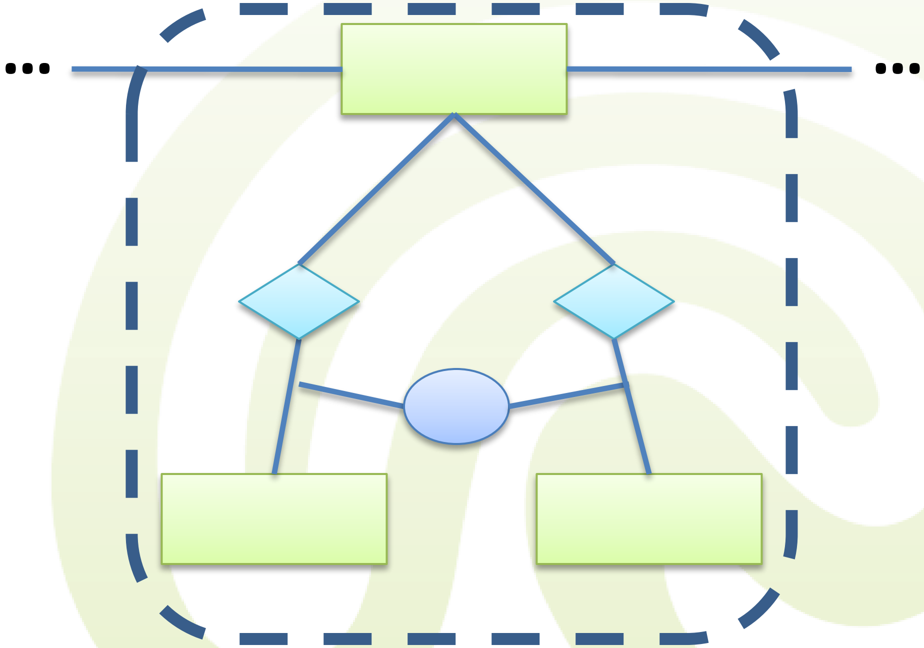
4.3 Grouping Operations

- **Abstraction grouping** clusters entities of a specific super-type
  - – Especially helpful, if sub-classses have no individual relationships



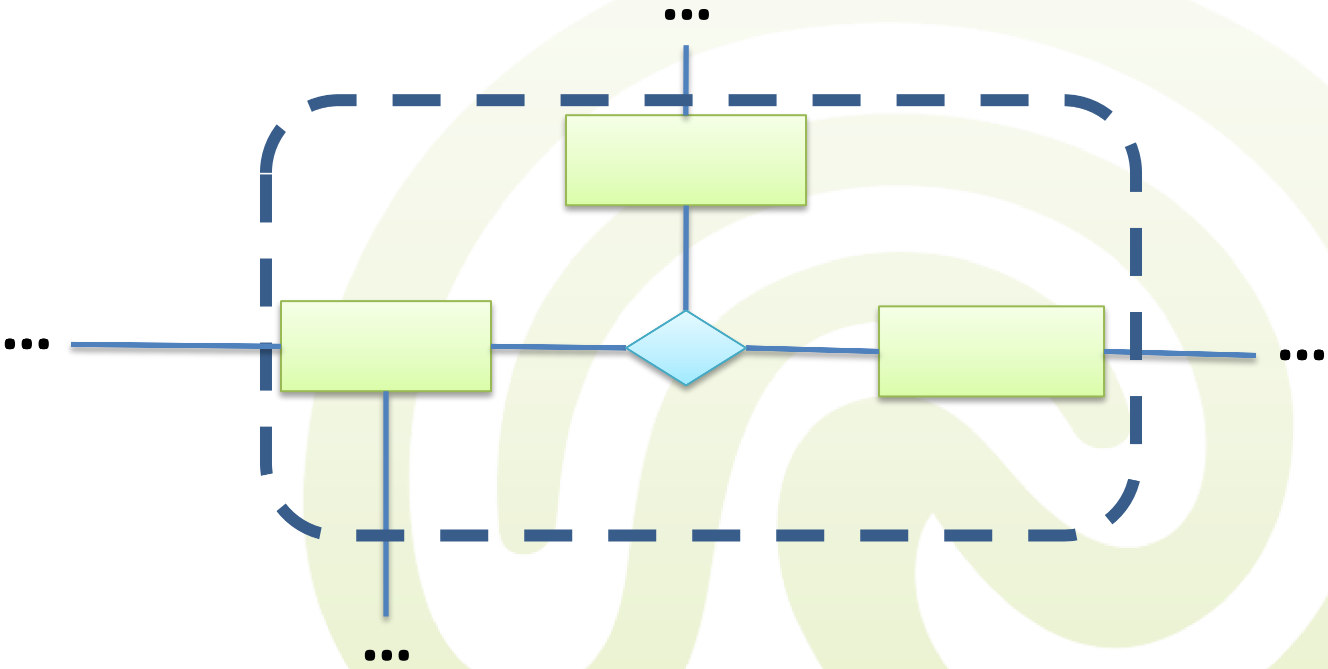
4.3 Grouping Operations

- **Constraint grouping** clusters entities related by the same constraint
  - E.g., integrity constraints such as XOR constraints



4.3 Grouping Operations

- **Relationship grouping** focuses on ternary or higher-degree relationships
  - The relationship is represented **as a whole**



4.3 Clustering Technique

- Identify all major functional areas and subareas in a top down analysis
  - Functional areas are often defined during the requirement analysis as important organizational units (e.g., HR or R&D) or business activities
  - Usually there will be a certain degree of overlap, for example employee data will be administrated by HR, but may also be needed in other departments

4.3 Clustering Technique

- The actual clustering has **four steps**
  - Define points of grouping within each functional area
    - Locate **dominant entities**, consider **abstraction**, find ***n*-ary** or **constrained relationships**, etc.
    - If such points do not exist, consider grouping the entire area
  - Form entity clusters
    - Use the **basic grouping operations** on elementary entities and their relationships to form higher level clusters
    - Since entities might belong to several clusters, **define priorities** like "always prefer abstraction grouping," "avoid crossing boundaries of functional areas," or "leave entities ungrouped, if they belong to two or more groups at the same level of precedence"

4.3 Clustering Technique

– Form higher level entity clusters
  - Apply the grouping operations **recursively** to any combination of elementary entities and entity clusters
  - Stop, if the diagram’s complexity is **sufficiently low**: This defines the root entity cluster

– Validate the cluster diagram
  - Check for **consistency** of the interfaces (relationships) between entities or entity clusters at each level of the diagram
  - **Verify the meaning** of each level with the intended users

4.4 The BEA Story

Detour

- What happens, if you don't integrate properly?
  - Think about the Mars disaster…
- What happens, if you integrate?
  - Well, your processes are improved and you become more efficient…
- What happens, if you **help others** to integrate?
  - Short version: you found a company, get insanely rich and are finally bought by Oracle for 8.5 billion USD in 2008

4.4 The BEA Story

Detour

- **BEA Systems Inc.**
  - Founded in 1995 in San José, CA, USA
  - Before Oracle's takeover, the company had more than 4000 employees and about **one billion** in revenues
  - **Product lines**
    - Tuxedo for **distributed transaction processing** (1995)
    - WebLogic provides a **J2EE enterprise infrastructure** (1998)
    - AquaLogic provides a **service-oriented infrastructure** (2005)
  - **Acquisitions** of some companies specializing in middleware and business process management
    - WebLogic (1998), SolarMetric (2005), Plumtree Software (2006), Fuego (2006), …

4.4 The BEA Story

Detour

DekaBank Finanzgruppe

- What are they actually doing?
  - **Case study:**
    - The **DekaBank Group** is the central asset manager of the Sparkasse Financial Group managing funds of around 90 billion EUR
    - The Bank wanted an **access layer** to central data sources so that all data for the portfolio structure is available for fund management
    - In 2006 DekaBank deployed the **BEA AquaLogic Data Services Platform,** which models the central data uniformly in a technical context and provides these business objects to local applications in real time

4.4 The BEA Story

Detour

- Two **Challenges**
  - Consolidation of various pieces of information from numerous channels
  - Provide the information in different formats such that local applications can further process the data
- Finally, after a lot of integration, data is presented to the outside via a **standard access layer** in real time
- **Duration:** about five month
- **Costs:** ???

4.4 The BEA Story

Detour

- **BEA AquaLogic Data Services**
  - Special Feature: **easy-to-use modeling**
    - "In an **SOA environment,** a data model must be flexible so that it can represent any **complex entity** and rich enough to provide information about **data structure, relationships**, and services to read or update"
    - "Data services are illustrated in **model diagrams** and can easily be shared with others in the enterprise for greater data consistency and reuse."
    - "Mappings and transformations can be designed in an easy-to-use **GUI tool** using a library of over 200 functions. For complex mappings and transformations, architects and developers can bypass the GUI tool and use an XQuery source code editor to define or edit services."

4.4 The BEA Story

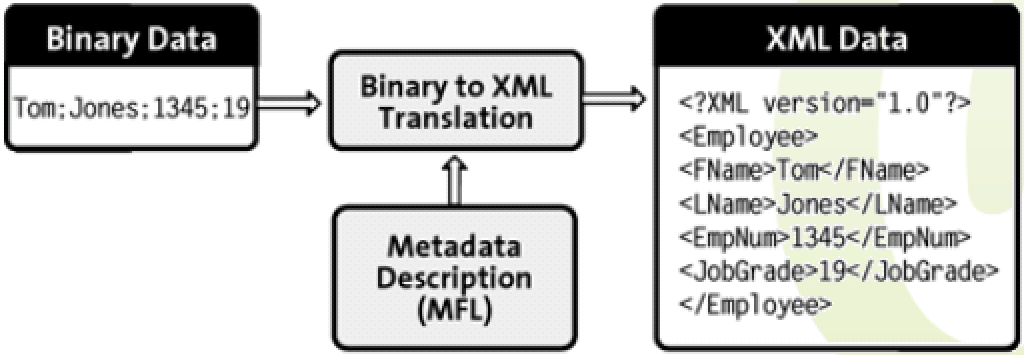
Detour

- What tools are actually given to support integration?
  - Data Translation Tool
    - Transforms binary data into XML
    - Transforms XML to binary data
  - Data Transformation Tool
    - Transforms an XML to another XML
  - Idea
    - Transform data to application specific XML
      → Transform to other application's XML or general schema
      → Transform back to binary
    - Note: the integration work still has to be done **manually**

4.4 The BEA Story

Detour

- Data Translation
  - Metadata Description Language describes how **binary data** (files, DB query results, etc.) is mapped to XML
  - BEA provides **editor** and execution **engine** for MFL
  - **Customer** creates MFL
    - Or alternatively: **Highly paid BEA consultants**



Binary Data

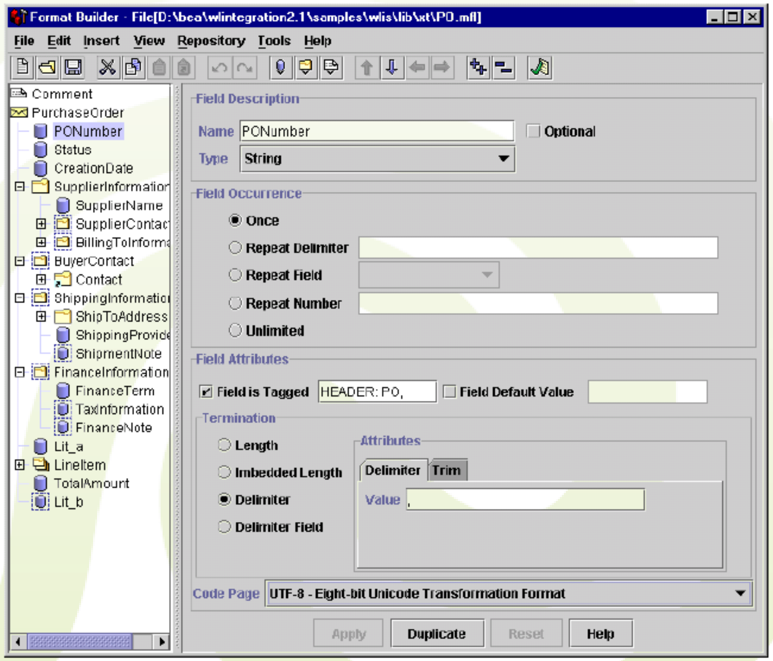
Tom:Jones:1345:19

Binary to XML Translation

Metadata Description (MFL)

XML Data

```
<?XML version="1.0"?>
<Employee>
<FName>Tom</FName>
<LName>Jones</LName>
<EmpNum>1345</EmpNum>
<JobGrade>19</JobGrade>
</Employee>
```



Format Builder - File[D:\bea\wlintegration2.1\samples\wlis\lib\xt\PO.mfl]

Field Description

Name PONumber ☐ Optional

Type String

Field Occurrence

● Once

○ Repeat Delimiter

○ Repeat Field

○ Repeat Number

○ Unlimited

Field Attributes

☑ Field is Tagged HEADER: PO, ☐ Field Default Value

Termination

○ Length

○ Imbedded Length

● Delimiter

○ Delimiter Field

Attributes

Delimiter Trim

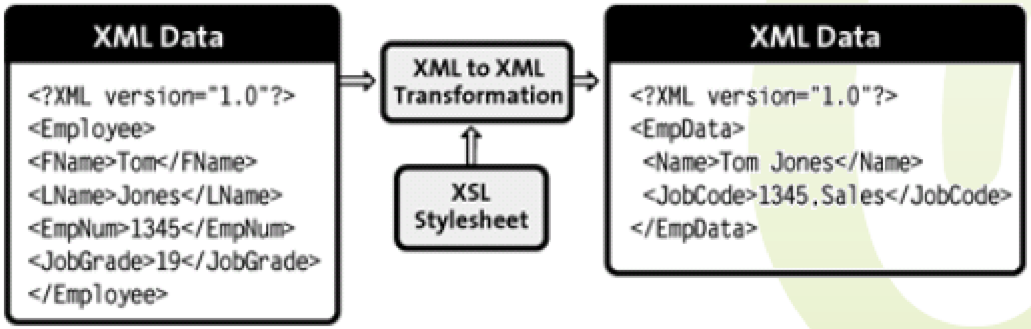
Value ,

Code Page UTF-8 - Eight-bit Unicode Transformation Format

4.4 The BEA Story

Detour

- Data Transformation
  - XSL (eXtensible Stylesheet Language, a W3C standard…) is used to transform XML to different XML
  - BEA provides **editor** and execution **engine** for XSL
    - Both are rather…simple?
  - **Customer** manually creates XSL
    - Or alternatively:
      Highly paid BEA consultants



XML Data

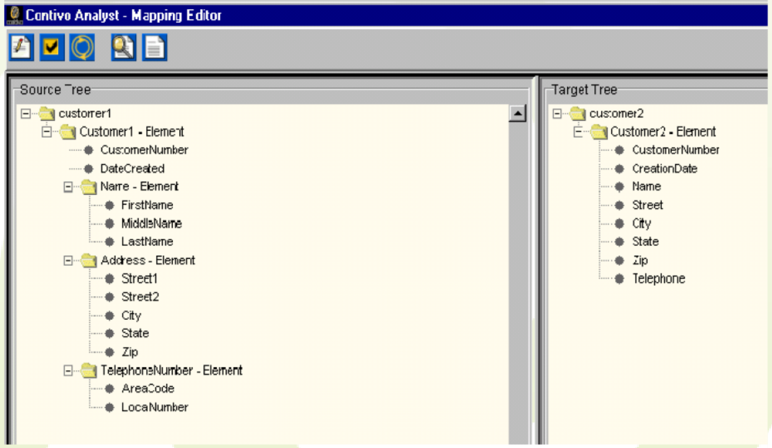
```
<?XML version="1.0"?>
<Employee>
<FName>Tom</FName>
<LName>Jones</LName>
<EmpNum>1345</EmpNum>
<JobGrade>19</JobGrade>
</Employee>
```

XML to XML Transformation

XSL Stylesheet

XML Data

```
<?XML version="1.0"?>
<EmpData>
 <Name>Tom Jones</Name>
 <JobCode>1345,Sales</JobCode>
</EmpData>
```



Contivo Analyst - Mapping Editor

Source Tree

- customer1
  - Customer1 - Element
    - CustomerNumber
    - DateCreated
    - Name - Element
      - FirstName
      - MiddleName
      - LastName
    - Address - Element
      - Street1
      - Street2
      - City
      - State
      - Zip
    - TelephoneNumber - Element
      - AreaCode
      - LocalNumber

Target Tree

- customer2
  - Customer2 - Element
    - CustomerNumber
    - CreationDate
    - Name
    - Street
    - City
    - State
    - Zip
    - Telephone

4.4 The BEA Story

Detour

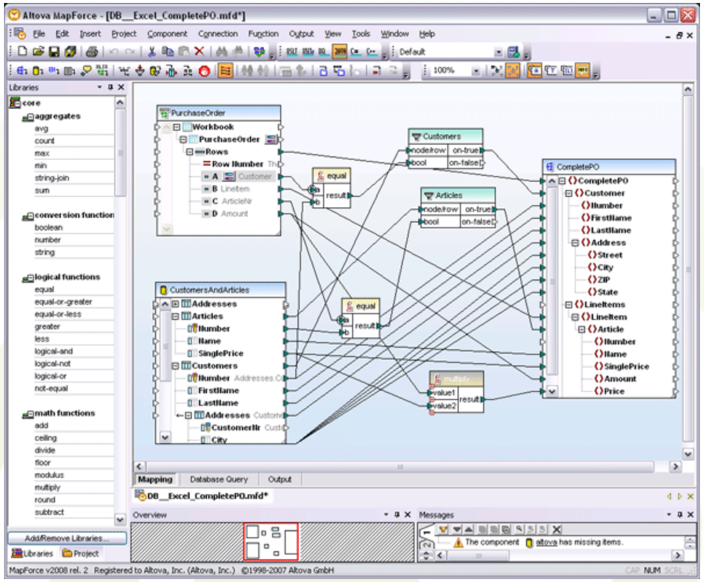
- "I can't afford expensive BEA consultants and the AquaLogic Integration Suite, what now??"
  - Do it completely **yourself**
    - Most used technologies can be found as open source projects (data mappers, XSL engines, XSL editors, etc.)
  - Do it **yourself** with **specialized tools**
    - Many companies and open source projects are specialized in developing data integration and transformation tools
      - CloverETL
      - Altova MapForce
      - BusinessObjects Data Integrator
      - etc…

4.4 The BEA Story

Detour

- **Altova MapForce**
  - Same idea than BEA Integrator
    - Also based on XSLT and a data description language
  - Editors for binary/DB to XML mapping
  - Editor for XSL transformation
  - Automatic generation of data sources, web-services, and transformation modules in Java, C#, C++

ALTOVA® mapforce® 2008



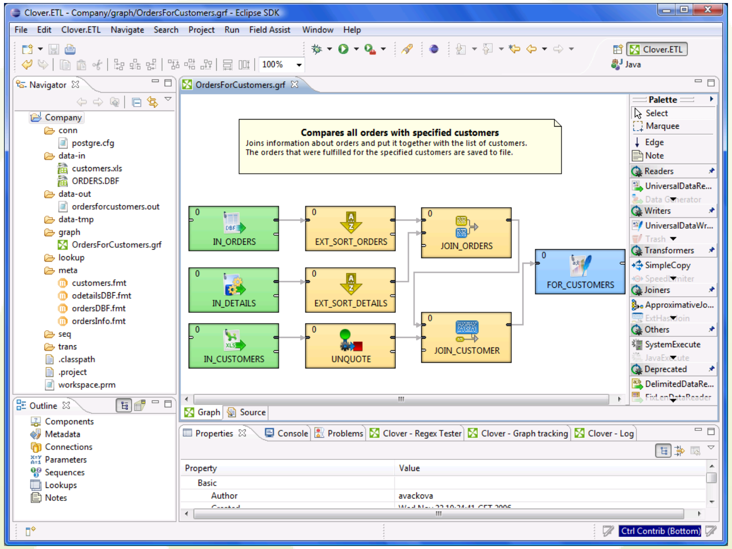
The component altova has missing items.

MapForce v2008 rel. 2 Registered to Altova, Inc. (Altova, Inc.) ©1998-2007 Altova GmbH

4.4 The BEA Story

Detour

- **CloverETL**
  - Based on own ETL transformation language
  - Core tools are open source
    - Server and GUI tools are sold under commercial license
  - Can read data from any database
  - (Visually designed) ETL Script converts data into other data
    - XML
    - DB with different schema
    - etc

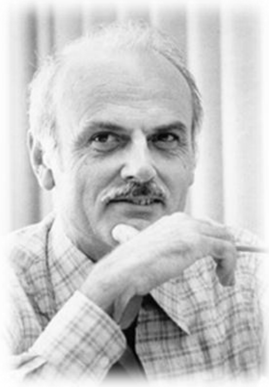


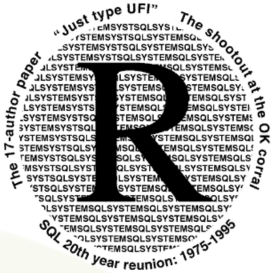
Compares all orders with specified customers
Joins information about orders and put it together with the list of customers.
The orders that were fulfilled for the specified customers are saved to file.

Property	Value
Basic	
Author	avackova

Next Week

- Basic set theory
- Relational data model
- Transformation from ER
- Integrity Constraints
- From Theory to Practice




“Just type UFI”
The shootout at the OK corral
The 17-author paper
SQL 20th year reunion: 1975-1995
R