Fraunhofer Institut Autonome Intelligente Systeme

Grouping and Aggregation in the Concept-Oriented Data Model

Alexandr Savinov

Fraunhofer Institute for Autonomous Intelligent Systems

Knowledge Discovery Team

Germany

savinov@conceptoriented.com

Fraunhofer Institut Autonome Intelligente Systeme

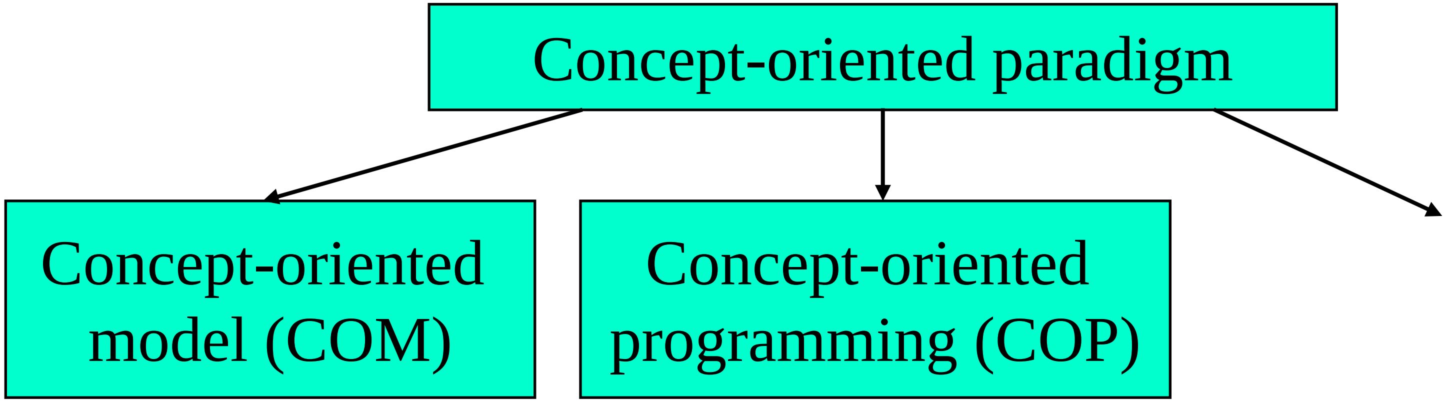
Outline

- Introduction
- Physical and Logical Structures
- Model Dimensionality
- Projection and De-projection
- Multidimensional Analysis
- Conclusions

Introduction

Concept-oriented paradigm

- **Duality:** any element is a collection of other elements and a combination of other elements, for example:
  - references vs. properties
  - entity modeling vs. identity modeling
- **Order:** order of elements determines most of syntactic and semantic properties
- **Representation and access (RA)** is the main concern.

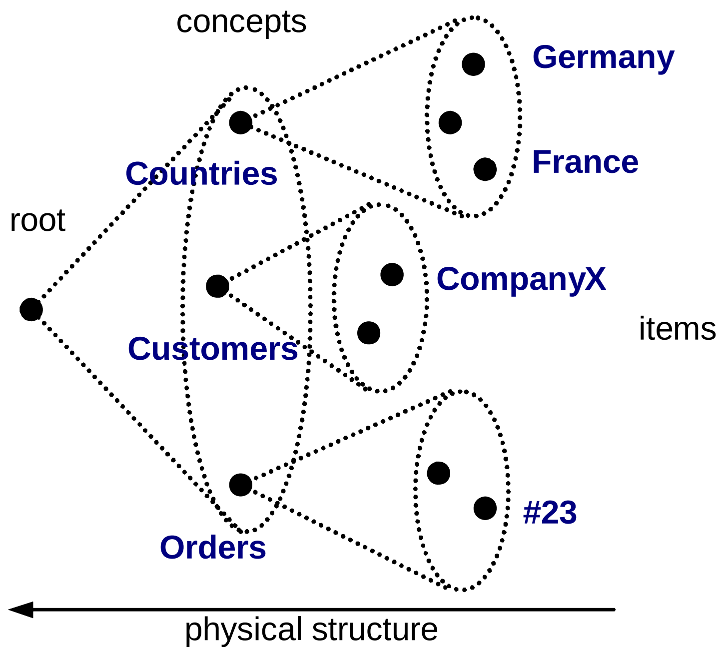


Physical and Logical

Physical structure

Fraunhofer Institut Autonome Intelligente Systeme

- At physical level an element of the model is a collection of other elements
- Physical structure is used for representation and access
- Physical structure is used to implement reference
- Physical structure is hierarchical where each element has only one parent

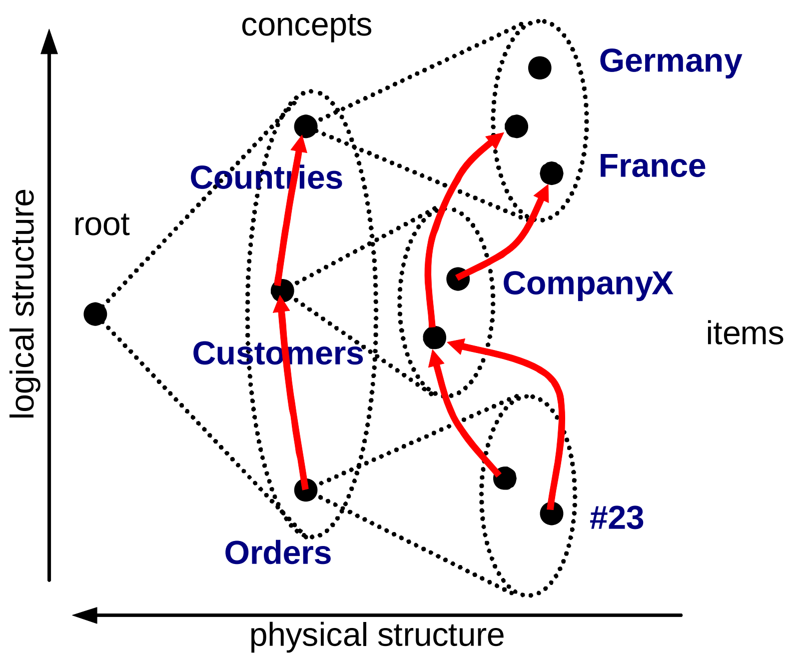


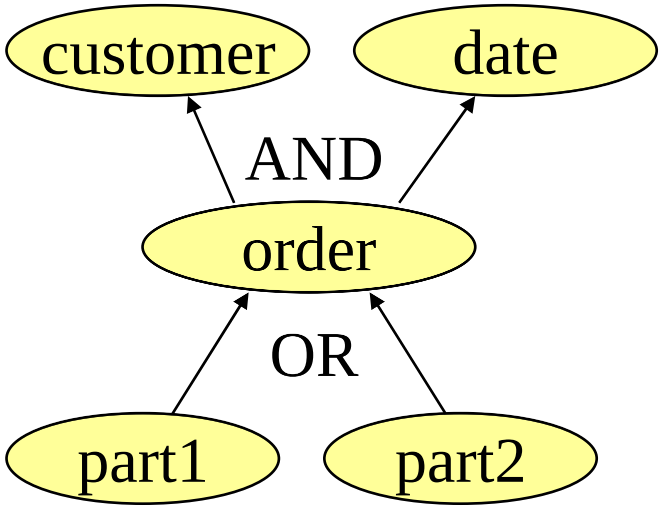
Physical and Logical

Logical structure

Fraunhofer Institut Autonome Intelligente Systeme
AIS

- Each element is a combination of other elements (by reference)
- Logical structure is used to represent data semantics (properties)
- Logical collection is a dual combination
- Each element has many parents and many children



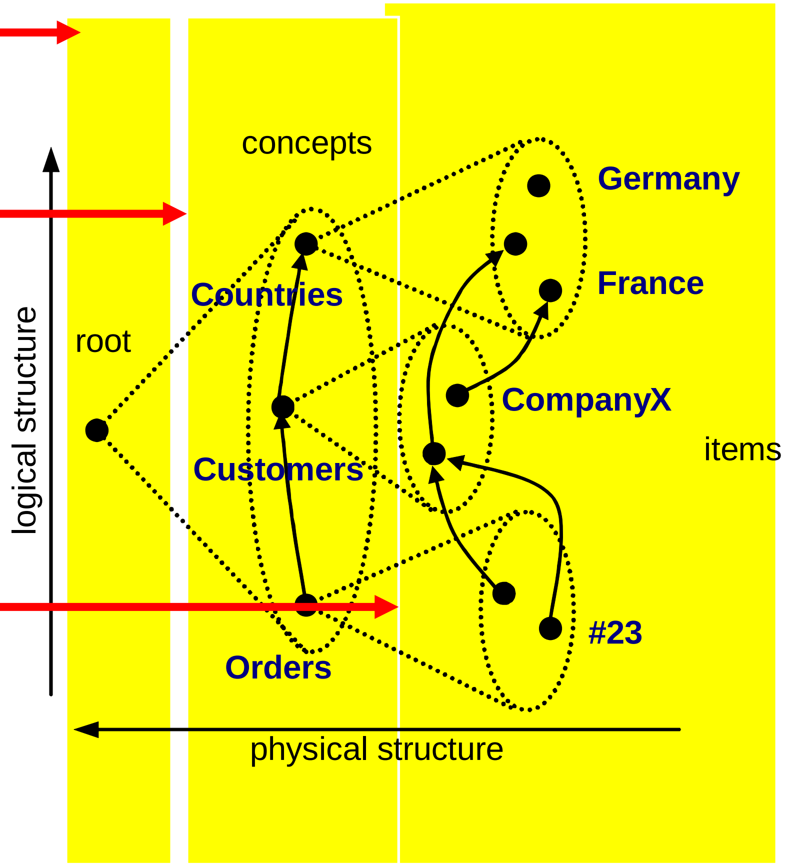


Physical and Logical

Two level model

Fraunhofer Institut Autonome Intelligente Systeme

- **[Root]** One root element is a physical collection of concepts,
- **[Syntax]** Each concept is
  - (i) a combination of other concepts called *superconcepts* (while this concept is a *subconcept*),
  - (ii) a physical collection of *data items* (or concept instances),
- **[Semantics]** Each data item is $i = \{\}$
  - (i) a combination of other data items called *superitems* (while this item is a *subitem*),
  - (ii) empty physical collection,

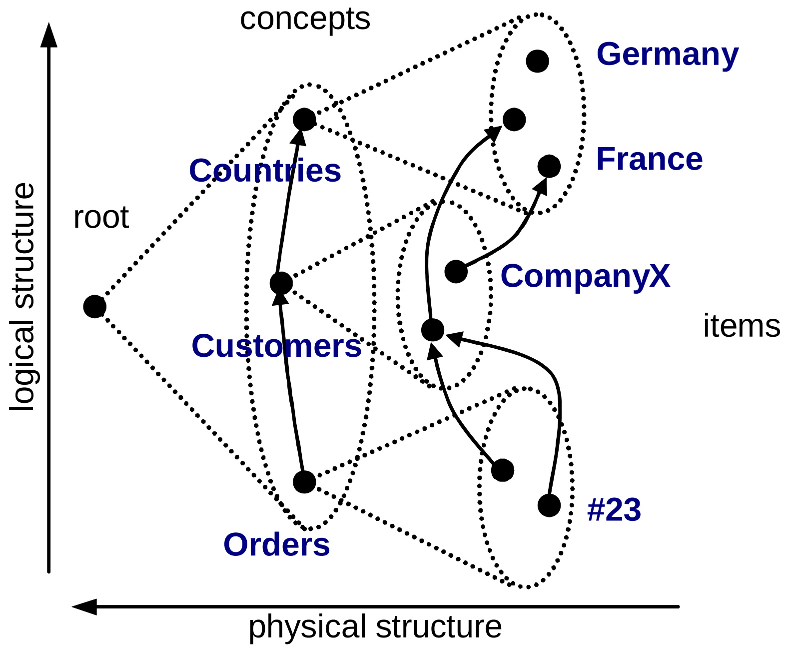


Physical and Logical

Two level model

Fraunhofer Institut Autonome Intelligente Systeme

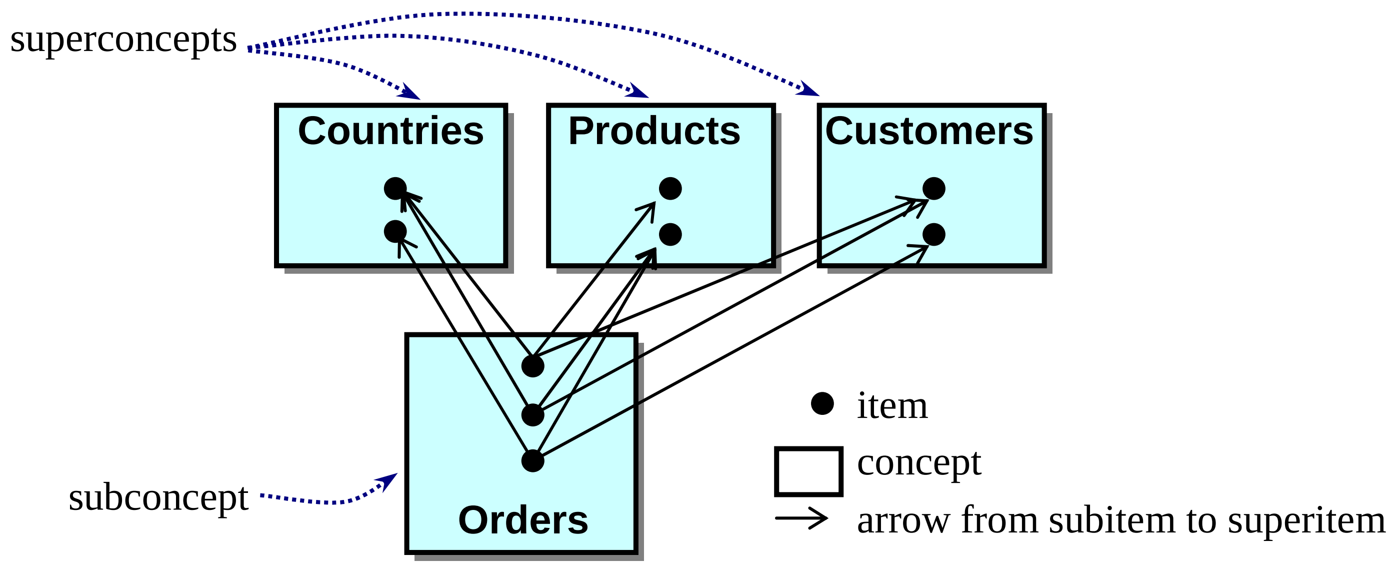
- **[Special elements]**
  - Top and bottom concepts
  - Primitive concepts
  - Null item
- **[Cycles]** Cycles in subconcept-superconcept relation and subitem-superitem relation are not allowed,
- **[Syntactic constraints]** Each data item from a concept may combine only items from its superconcepts.



Model Dimensionality

Multidimensional space

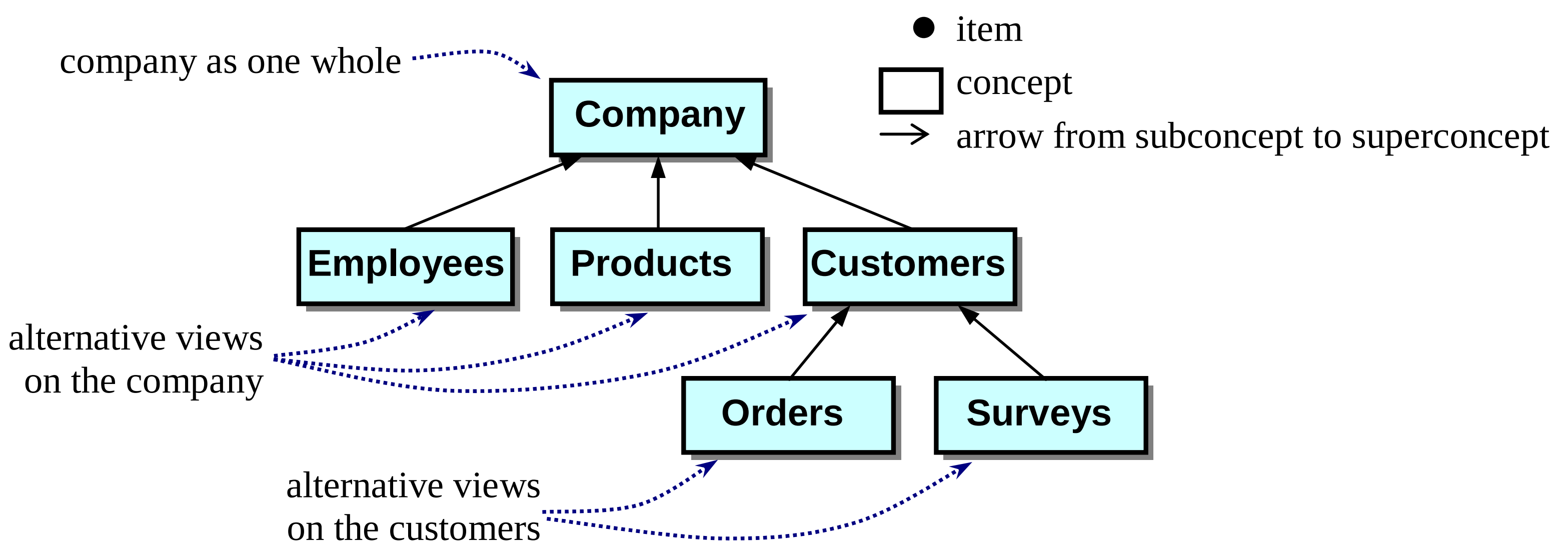
- Superconcept is a domain of a dimension
- A common subconcept is a multidimensional space
- More levels can be added to the multidimensional space



Model Dimensionality

Hierarchical space

- It is one-dimensional space with many levels of details
- Subconcepts are alternative views on their common superconcept

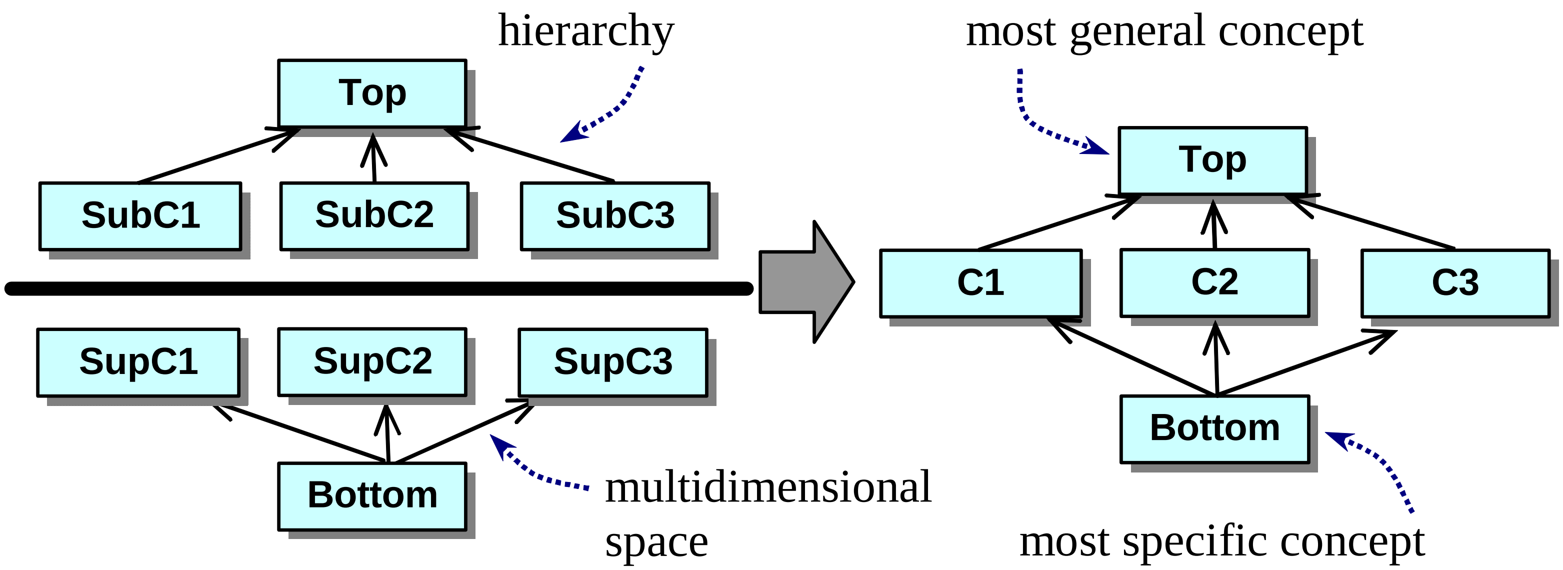


Model Dimensionality

Hierarchical multidimensional space

Fraunhofer Institut Autonome Intelligente Systeme

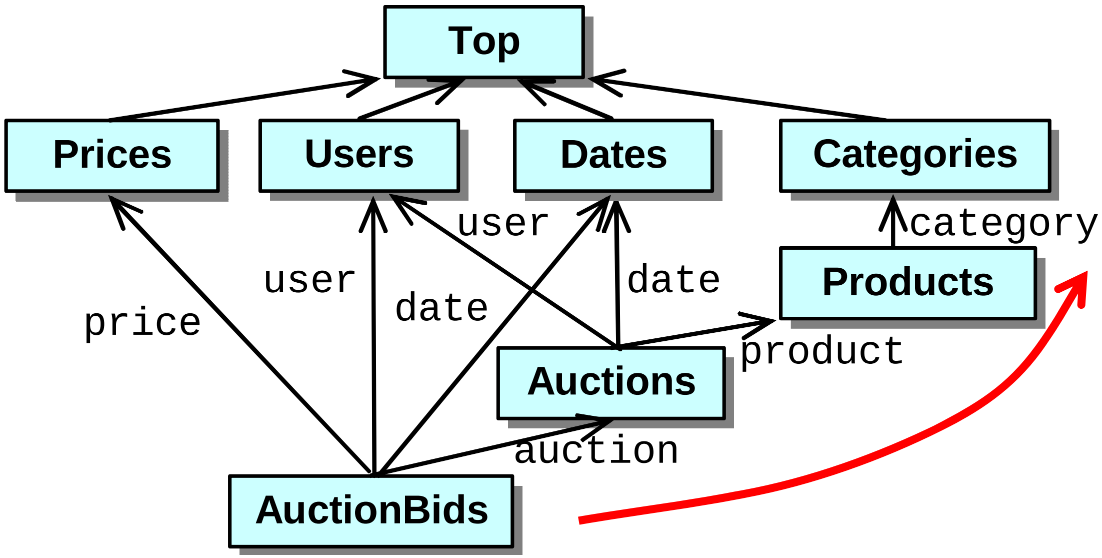
- Both structures are combined in one concept graph
- The concept graph possesses both multidimensional and hierarchical properties



Model Dimensionality

Dimensions

- Dimension is a named position of superconcept
- Superconcept is referred to as the domain
- Dimensions of higher rank consists of many (local) dimensions
- Dimension with the domain in a primitive concept is a *primitive dimension*
- The number of primitive dimensions is the model *primitive dimensionality*

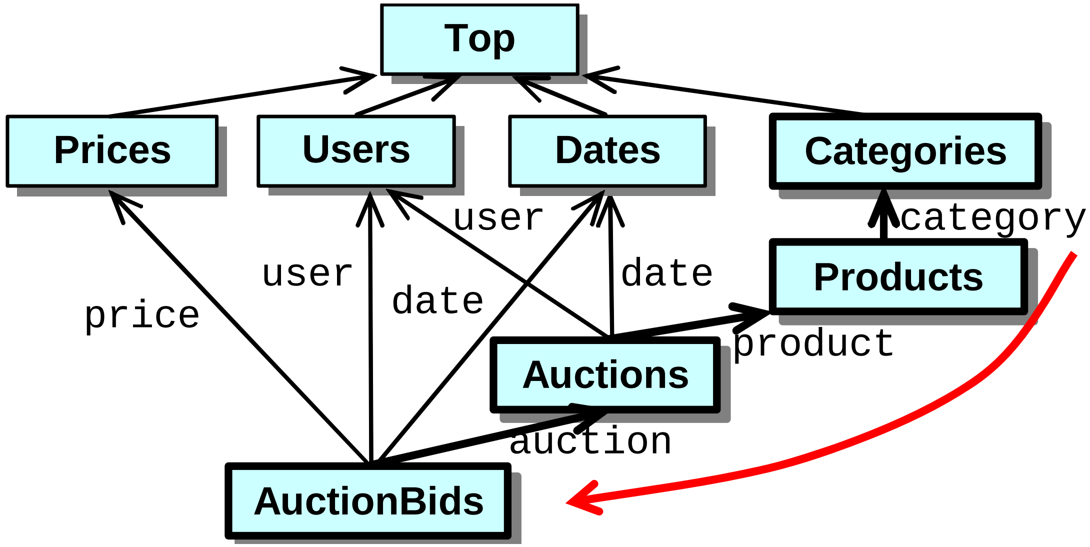


Model Dimensionality

Inverse dimensions

Fraunhofer Institut Autonome Intelligente Systeme

- Inverse dimension has an opposite direction
- Inverse dimension identifies a subconcept
- Inverse dimensions are multi-valued (while dimensions are one-valued)
- The number of primitive dimensions is equal to the number of primitive inverse dimensions
- `{AuctionBids.auction.product.category}`



Projection and De-projection

Two retrieval operations

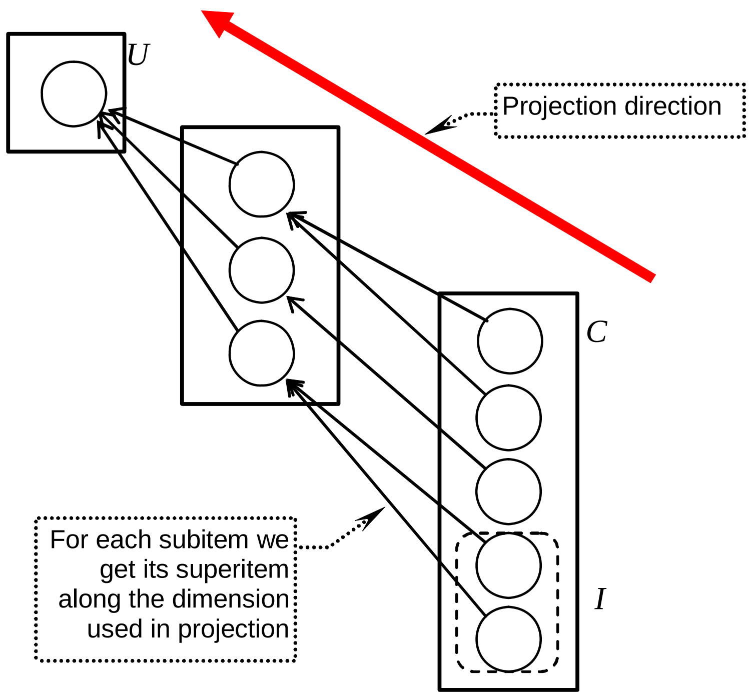
Fraunhofer Institut Autonome Intelligente Systeme

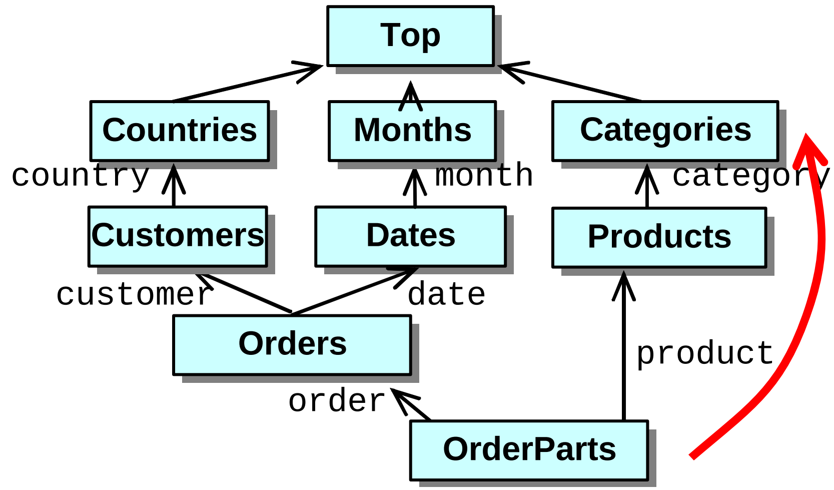
- Two ways to retrieve related items: projection and de-projection
- These two ways are supported by the model structure and correspond to moving up and down in the concept graph
- These two retrieval operations need only dimension names – no complex joins anymore
- These operations are analogous to the corresponding geometrical operations

Projection and De-projection

Projection

- Projection operator returns a set of <u>superitems</u> along some dimension
- Projection operator -> is followed by a dimension:
  `OrderParts->product->category`

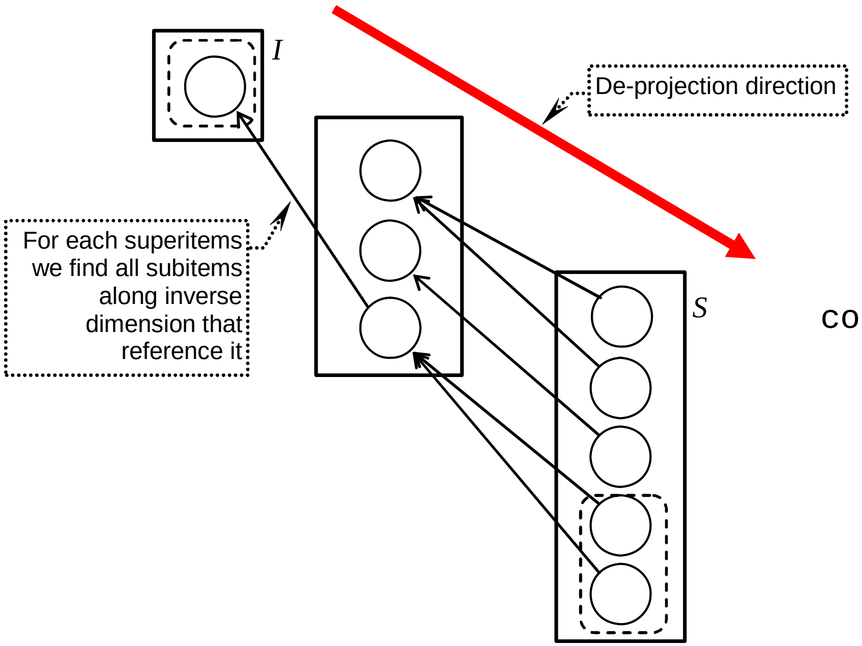


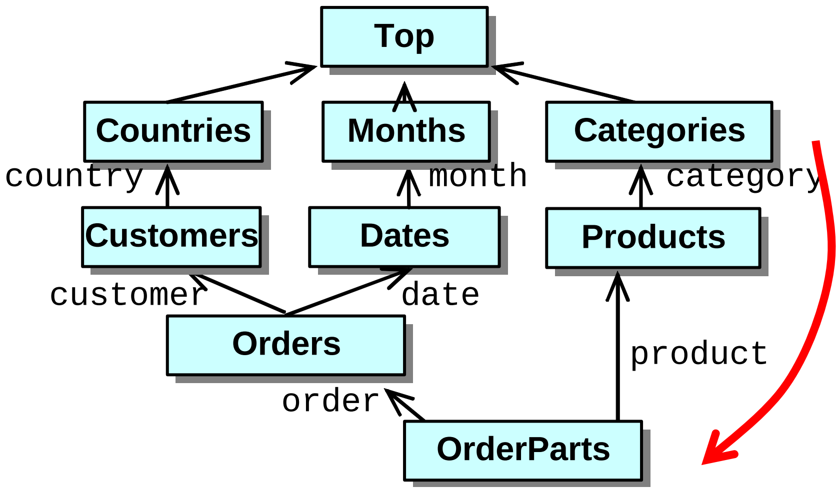


Projection and De-projection

De-projection

- De-projection operator returns a set of <u>subitems</u>
- De-projection operator -> is followed by an inverse dimension: `Category->{product->category}`



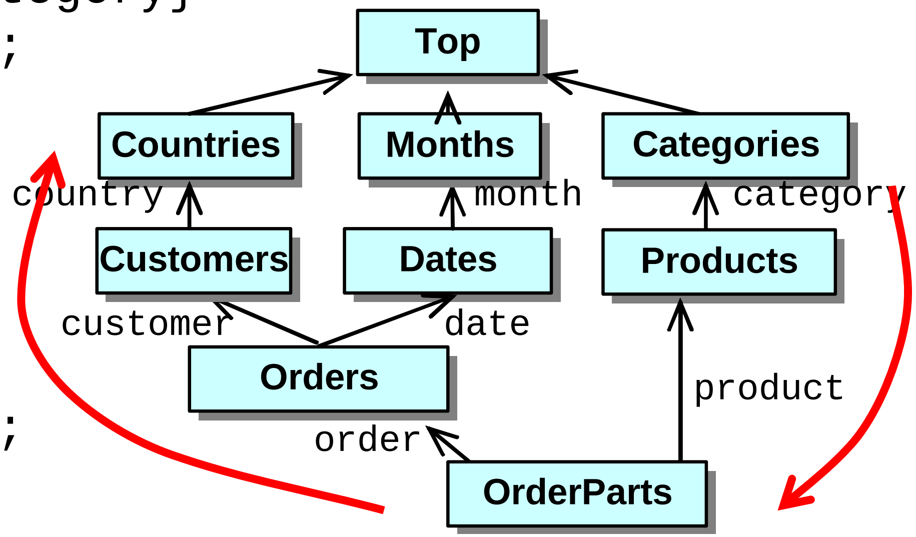


Projection and De-projection

Access path

Fraunhofer Institut Autonome Intelligente Systeme

- Access path is a sequence of projections and de-projection where each next operator is applied to the result of the previous operator
- ```
  Category.getOrders = this->
    {OrderParts->product->category}->
    order;
  ```
- ```
  Category.getOrders = this->
    {OrderParts->product->category}->
    order->customer->country;
  ```
- Zigzag paths are possible
- Aggregation can be applied to sets of items
- ```
  Category.meanPrice = avg(
    this->getOrders->price );
  ```

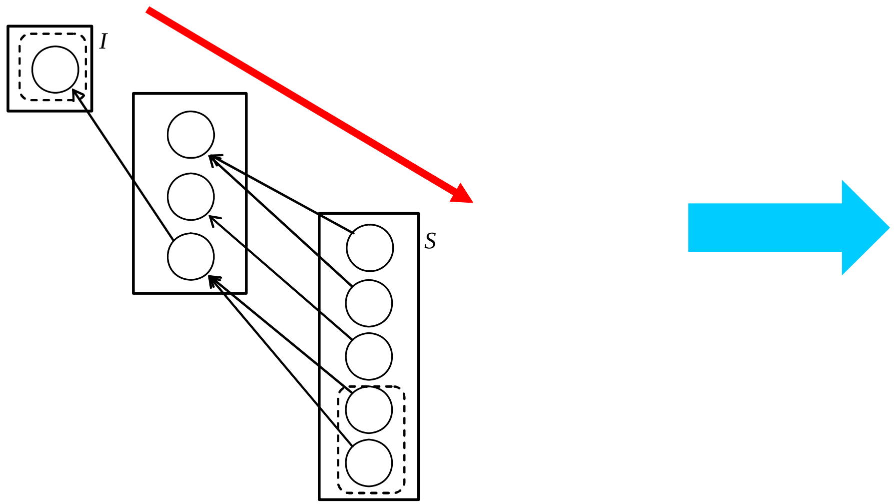

  ```

Multidimensional Analysis

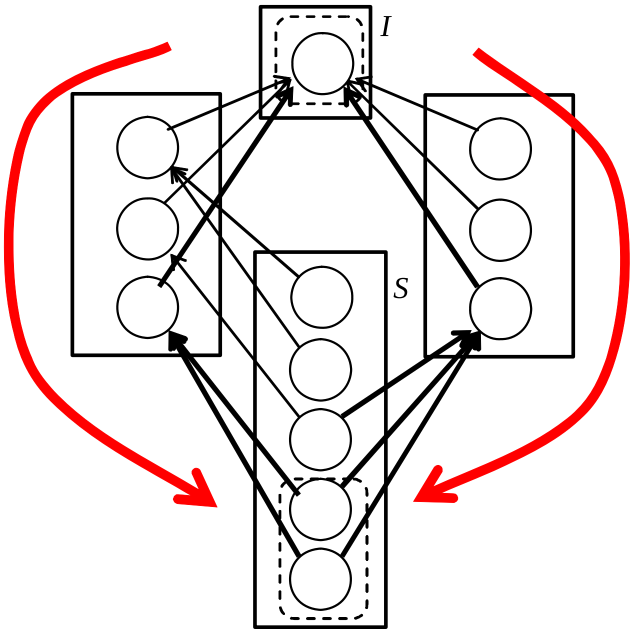
Multidimensional de-projection

- More than one bounding dimension
- Multidimensional de-projection returns a set of subitems referencing source items along <u>all</u> bounding dimensions:

One-dimensional de-projection



Multi-dimensional de-projection

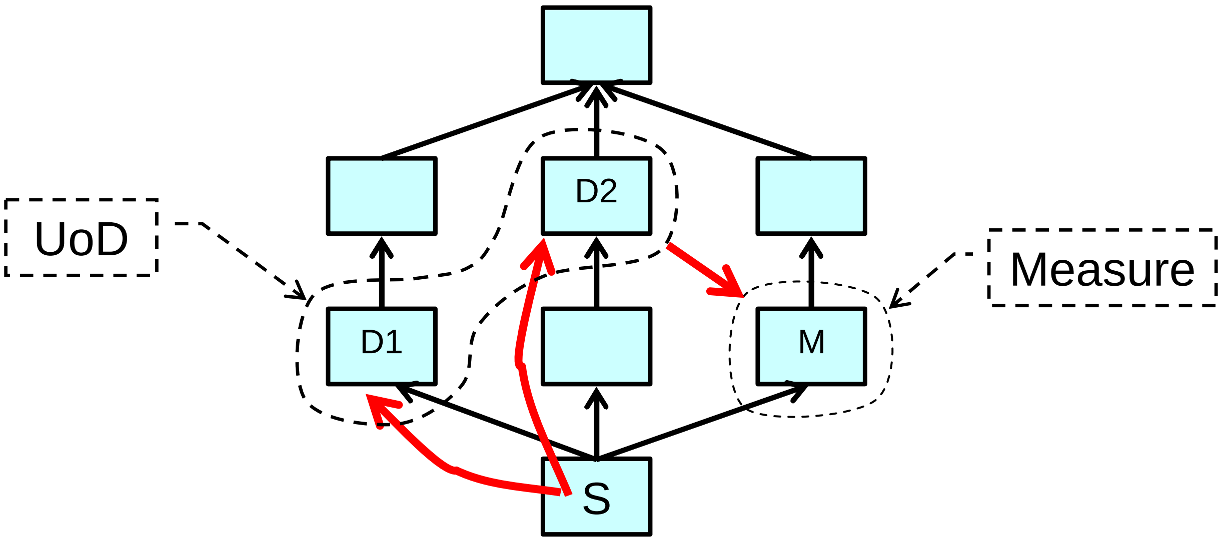


Multidimensional Analysis

Steps of analysis

Fraunhofer Institut Autonome Intelligente Systeme

1. Choose dimension paths along which we want to view our data S
2. Choose the levels along these dimensions
3. Universe of discourse is the Cartesian product of the chosen levels
4. Each point from UoD is de-projected onto the target subconcept S
5. De-projection is aggregated using some property (measure)



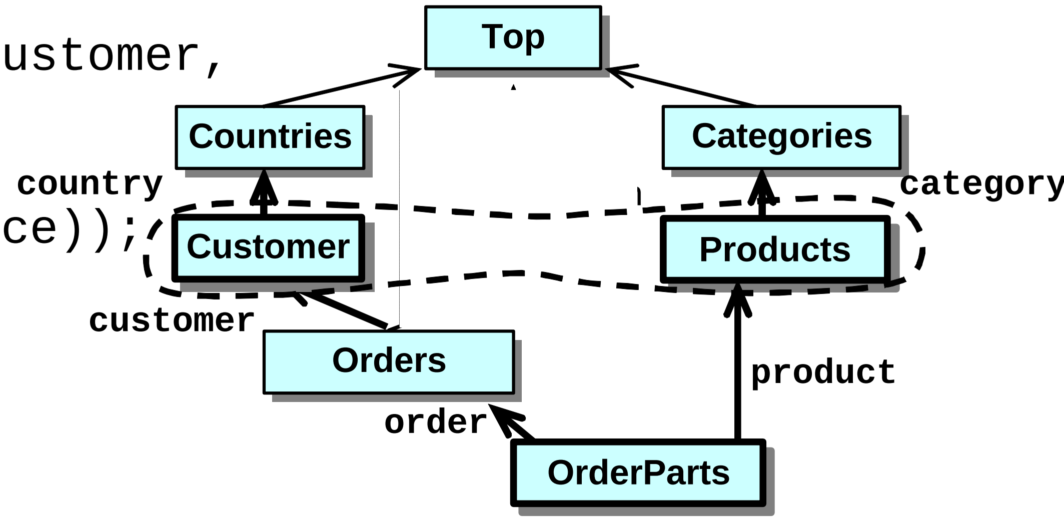
Multidimensional Analysis
Example

- Choose the target concept `OrderParts` and two dimensions leading to concepts `Countries` and `Categories`
- De-project each pair of customer and product to `OrderParts`:

```
<c,p>->{OrderParts->order->customer,
        OrderParts->product}
```

- Aggregate and return average price:

```
FORALL(c Customers, p Products) {
  tmp= <c,p>->{
    OrderParts->order->customer,
    OrderParts->product
    }
  RETURN(c,p,avg(tmp.price));
}
```

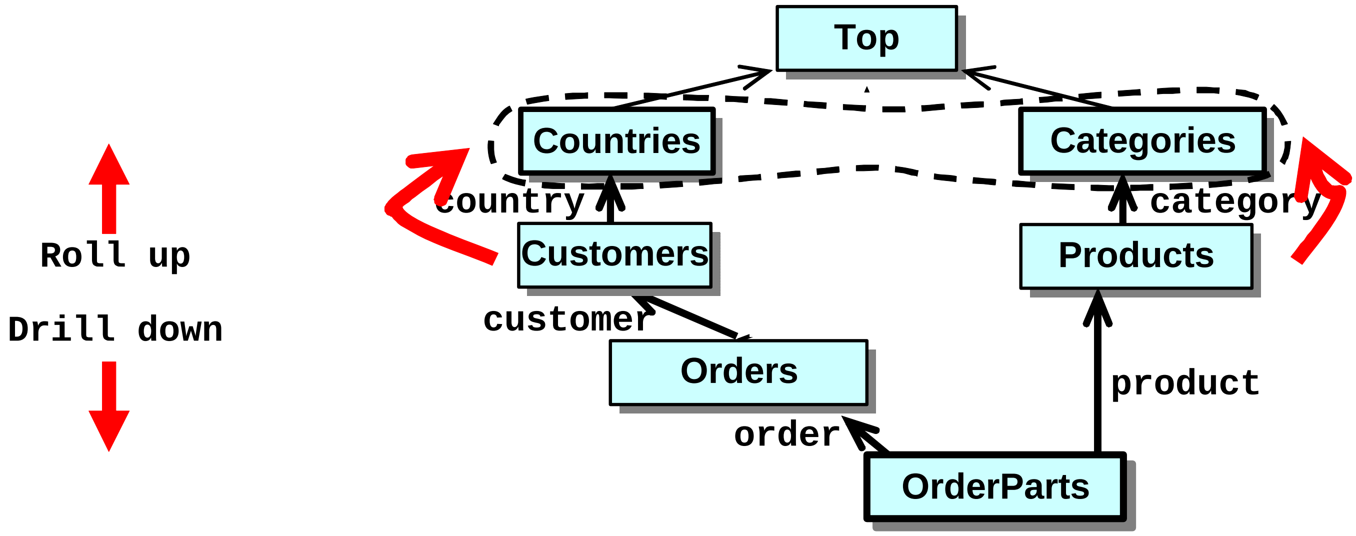


Multidimensional Analysis

Change the level of details

Fraunhofer Institut Autonome Intelligente Systeme

- Choose other domains along dimension paths and apply the same query:
- ```
  FORALL(c Countries, p Categories) {
    tmp= <c,p>->{
      OrderParts->order->customer->country,
      OrderParts->product->category
      }
    RETURN(c,p,avg(tmp));
  }
  ```

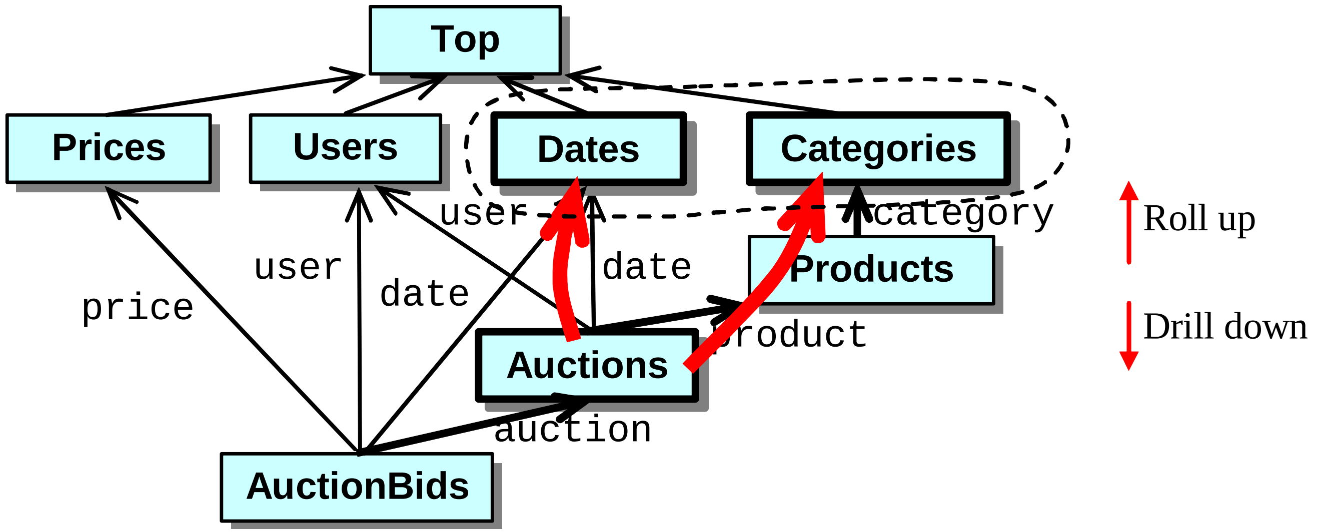

  ```

Multidimensional Analysis

Example 2

Fraunhofer Institut Autonome Intelligente Systeme

- ```
  FORALL{ d Dates, c Categories } {
    tmp = this->{
        Auctions.date,
        Auctions.product.category
        }
    RETURN(d,c, avg(tmp->meanPrice) )
  }
  ```


  ```

Conclusions

Fraunhofer Institut Autonome Intelligente Systeme

- Features:
  - Canonical semantics
  - Logical navigation via access paths, dimensions and inverse dimensions
  - Multidimensional aggregation and analysis
  - Constraint propagation and inference (not described in this presentation)
- Advantages:
  - Grouping and aggregation is integral part of the model
  - Combination in one model hierarchical and multidimensional properties
  - Formal syntax and semantics
  - Simple query language -- no joins anymore