Pavel Koupil, Martin Svoboda, Irena Holubová

MM-cat: A Tool for Modeling and Transformation of Multi-Model Data using Category Theory

Motivation

- *Variety* feature of Big Data

- *Level of support* of multiple data models in MMDBMS *varies greatly*
  - No unified approaches exist
  - Necessity of multiple *model-specific model and query constructs*
  - There is *no* solid *formal background*

- We need a representation that would allow us to
  - Capture all the existing data models, preferably in a *unified way*
  - *Query* across multiple *interconnected*, possibly *overlapping* models
  - Perform correct and complete *evolution management*
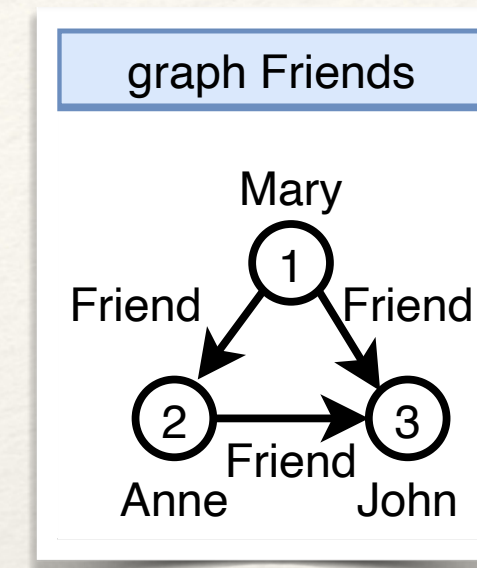 - Enable *data migration*
  - Permit *integration* of new data models

graph Friends

Mary
1
Friend   Friend
2 ———→ 3
   Friend
Anne   John

table Customer

CustomerId	FirstName	Address	Credit
1	Mary	...	3000
2	Anne	...	2000
3	John	...	5000

collection Order

{ OrderId : 220,
  Paid: true,
  Items: [
  { ProductId: T1, Name: toy,
    Price: 200, ItemQuantity: 2},
  { ProductId: B4, Name: book,
    Price: 150, ItemQuantity: 1 }] }

column family Orders

CustomerId	Orders
1	[220, 230, 270, ...]

CustomerId	Orders
2	[10, 217]

| CustomerId	Orders
3	[94, 137, 214, 370]

key / value pairs Cart

1 → Product: T1, Quantity: 2
    Product: B4, Quantity: 1

2 → Product: H1, Quantity: 1

3 → Product: B3, Quantity: 2

MM-cat: An Extensible Framework

❖ Multi-model data management framework *based on category theory* (strong formal background)

❖ Enables to *represent a multi-model schema* using a schema category

   ❖ Can be *automatically extracted* from a conceptual model, e.g., ER model

   ❖ Can *cover* (combinations of) *existing* logical *data models*

   ❖ Can be *visualized using a multigraph*

❖ Enables to map the categorical model to any (combination of) DBMS(s)

   ❖ *Implementation specifics* of particular DBMS(s) are *hidden* to user (demonstrated using MongoDB and PostgreSQL)

❖ Enables to transform the data from the underlying DBMS(s) to an instance category and vice versa, i.e., *data migration*

   ❖ Instance category serves as a mediator for a unified representation of data instances
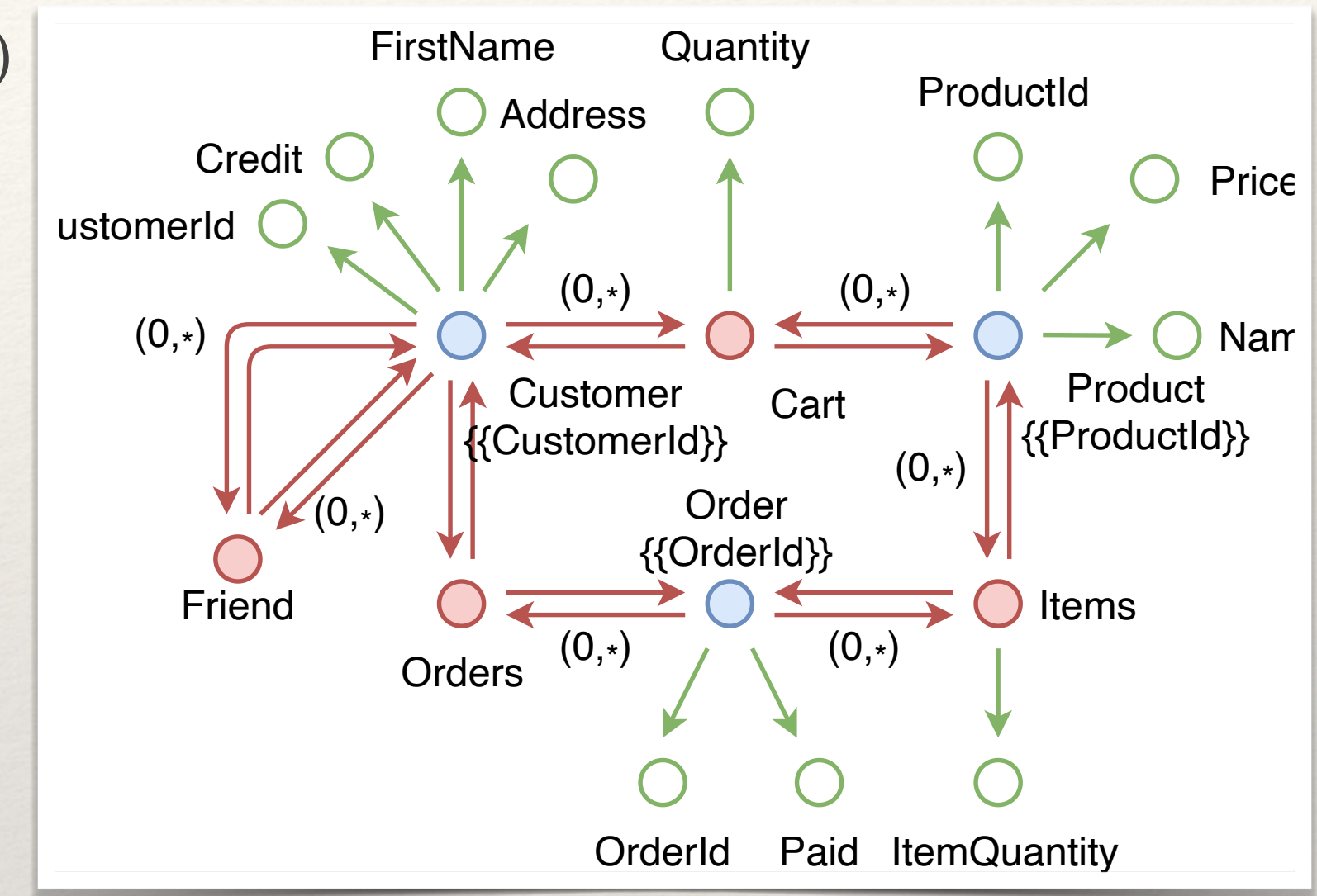
MM-cat: Modular Architecture

❖ Modular architecture

 ❖ *Allowing simple extension* towards additional data models (DMBSs)

❖ *Advanced data management tasks* built on top of the categorical representation

 ❖ Querying

 ❖ Evolution Management

 ❖ Data Migration

 ❖ Schema Inference
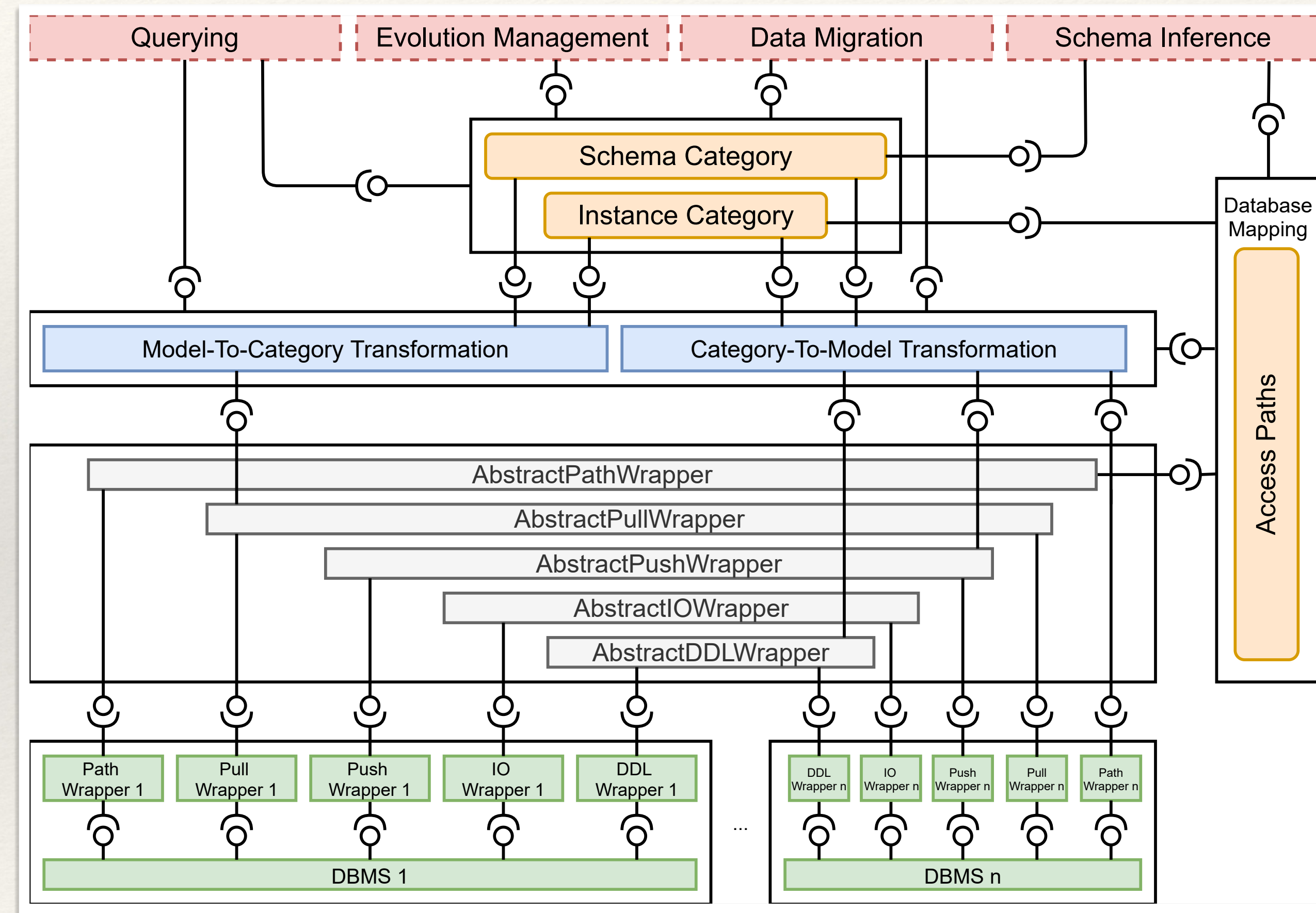
Running Example

1. An *ER schema* of the target problem domain is created

2. This ER schema is automatically transformed to the *schema category*

3. *Mapping* of logical data models (DBMS components)

   ❖ Scripts with the resulting CREATE statements are generated

4. PostgreSQL to MongoDB *data migration*

   ❖ Data is loaded from PostgreSQL to the instance category

   ❖ Collections in MongoDB are created, if necessary

   ❖ Contents of the instance category is materialized in MongoDB

Summary

❖ Current version of MM-cat provides

   ❖ Modular and extensible framework

   ❖ A tool for *user-friendly modeling* of multi-model data

   ❖ Captures specifics of popular data models

   ❖ *Migration* of data between multiple DBMSs

❖ Extensions currently being added and future work:

   ❖ Implementation of a *unifying conceptual query language*

   ❖ Intra and inter model modifications of the schema, i.e., *evolution management*

   ❖ Automatic *schema inference* over schema-less (and schema-mixed) multi-model data and its *mapping to schema category*

Thank you for your attention…