From Classical to Quantum Databases with Applied Pullbacks

Nick Rossiter

Seminar – PSSL, 15th February 2003

http://computing.unn.ac.uk/staff/CGNR1/

nick.rossiter@unn.ac.uk

Database Theory

- Usually based on sets (Jeffrey Ullman, Chris Date, Ted Codd)
  - Relational databases
    - Sets of tuples
    - Functions for dependencies
    - First-order safe predicate calculus for manipulation (SQL)
    - Also an equivalent algebra
  - Network databases
    - Graphs for structures
    - Navigational (traversal) languages for manipulation
  - Object-oriented databases
    - Set-based class and object structures
    - Navigational (traversal) language for manipulation (OQL)

Definition: Database Model (as it varies!)

- Database Model: a representation of policies in a structured form according to some perceived view of reality e.g.
  - Relational model – world is tabular
  - Hierarchical model – world is tree-like
  - Security model – world is task-based
  - Object model – world is based on o-o paradigm

Relationships

- Main classifying feature of databases is how they represent relationships:
  - Relational – including a foreign key (primary key of another table) in the set of attributes
  - Network – including the address of an object in another object (pointer-based)
  - Object-oriented – having a function from one class to another (references)

Challenge of Interoperability

- Interoperability:

  the ability to request and receive services between various systems and use their functionality.
- More than data exchange.
- Implies a close integration
- No longer possible for systems to be stand-alone

Motivations

- Diversity of modelling techniques
- Distributed businesses may exercise local autonomy in platforms
- Data warehousing requires heterogeneous systems to be connected
- Data mining enables new dependencies to be derived from heterogeneous collections

Simple Problem in Interoperability

- Homogeneous Models
  - the same information may be held as attribute name, relation name or a value in different databases
  - e.g. fines in library;
    - could be held in a dedicated relation Fine(amount, borrowed_id)
    - or as an attribute Loan(id, isbn, date_out, fine)
    - or as a value Charge(1.25, ‘fine’)

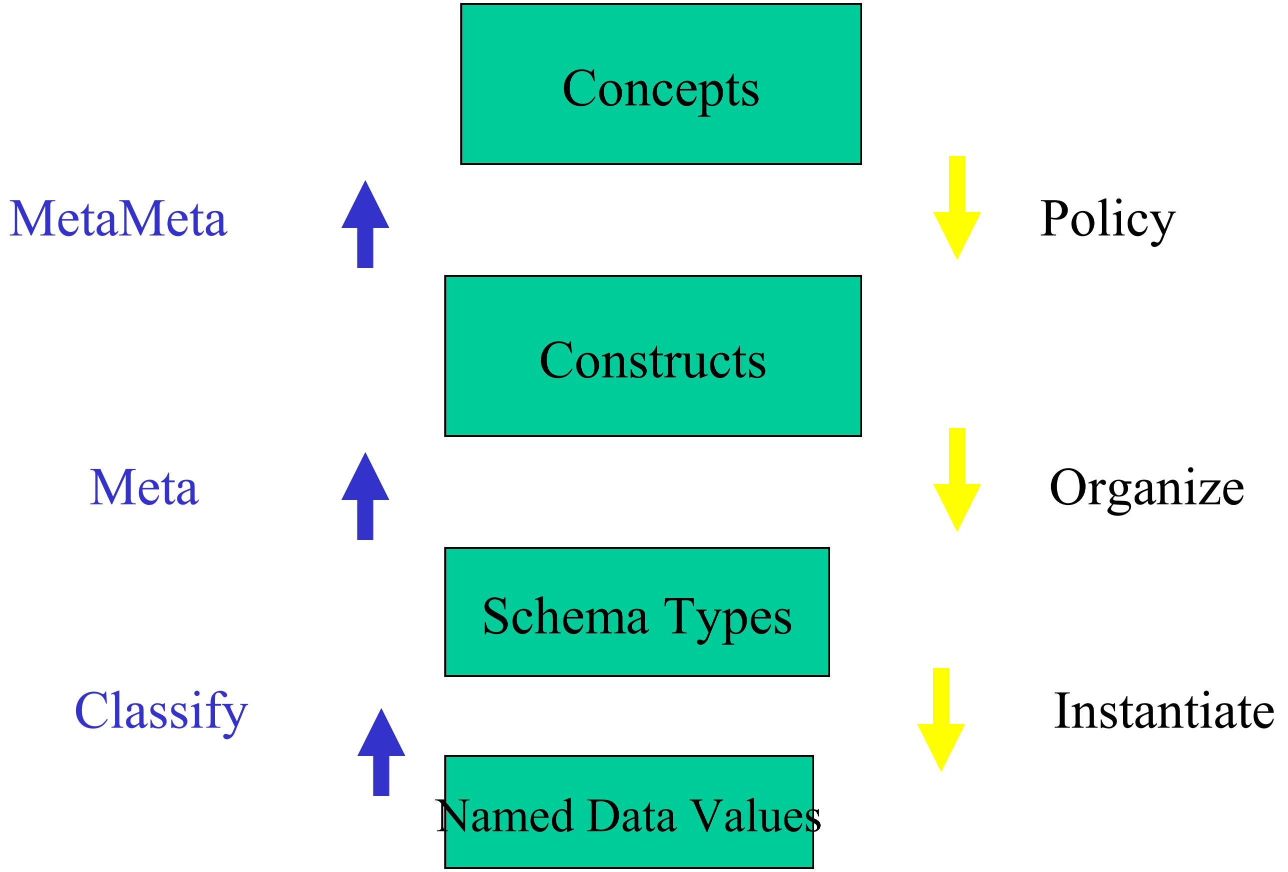
Complex Problems in Interoperability

- Heterogeneous models
- Need to relate model constructions to one another, for example:
  - relate classes in object-oriented to user-defined types in object-relational
- All problems are magnified at this level.

Use of the term Meta Data

- Meta means 'about'
- The basis of schema integration
- Sometimes treated as an object (MOF - Meta Object Facility)
- Better viewed as a relationship:
  - Name (data files)
  - Classify (database classes)
  - Meta (data dictionary)
  - MetaMeta (classify data dictionary)

Mappings are two-way



Downward arrows are intension-extension pairs

Formalising the Architecture

- Requirements:
  - mappings within levels and across levels
  - bidirectional mappings
  - closure at top level
  - open-ended logic
  - relationships (product and coproduct)
- Candidate: category theory as used in mathematics as a workspace for relating different constructions

Choice: category theory

- Requirements:
  - mappings within levels and across levels
    - arrows: function, functor, natural transformation
  - bidirectional mappings
    - adjunctions
  - closure at top level
    - four levels of arrow, closed by natural transformation
  - open-ended logic
    - Heyting intuitionism
  - relationships (product and coproduct)
    - Cartesian-closed categories (like 2NF): pullback and pushout

Work with Databases and Categories

- Michael Johnson, Robert Rosebrugh and RJ Wood, Entity-Relationship-Attribute Designs and Sketches, TAC 10(3) 94-111.
  - sketches for design (class structure)
  - models for states (objects) where model is used in categorical sense
  - lextensive category (finite limits, stable disjoint finite sums) for query language

Sketch/Model

- Developed also in databases by:
  - Zinovy Diskin, Boris Cadish: Algebraic Graph-Based Approach to Management of Multidatabase Systems, NGITS'95 69-79 (1995).
- Sketch originally from Charles Ehresmann.
  - Finite Discrete (FD) sketch D = (E, L, R, S)
    - finite graph E (data structure)
    - set of diagrams L in E (constraints)
    - Finite set R of discrete cones in D (relationships)
    - Finite set S of discrete cocones in D (attributes)
- Model (M) – graph homomorphism
    - maps any E to category V where V is a database state
    - L → commutative diagrams, R → limit cones, S → colimit cocones
    - preserve products
- In FP sketches in Johnson et al:
    - finite sums satisfy the lextensive axiom
    - sums are well-behaved

Pullbacks are used extensively for database relationships

Here of S and M in Context of IMG

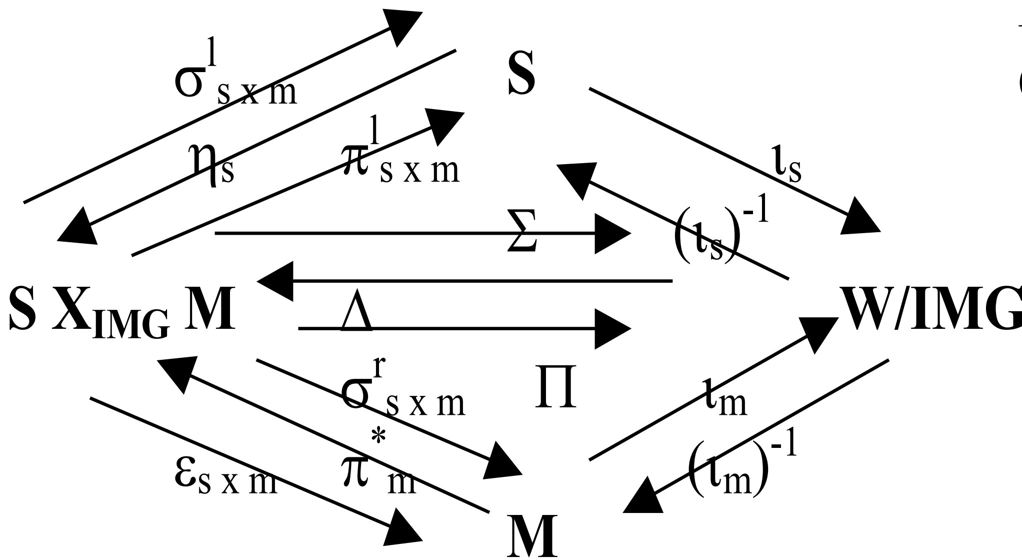


Figure 2: Pullback showing fuller collection of arrows

S = source, M = medium, IMG = image, W = world

Categories

- Each level is represented by a category:
  - Named data values by **DATA** (**DT**)
    - value ⟶ name
  - Schema types by **SCHEMA** (**SM**)
  - Constructions by **CONSTRUCTS** (**CS**)
  - Concepts by **CONCEPTS** (**CC**)

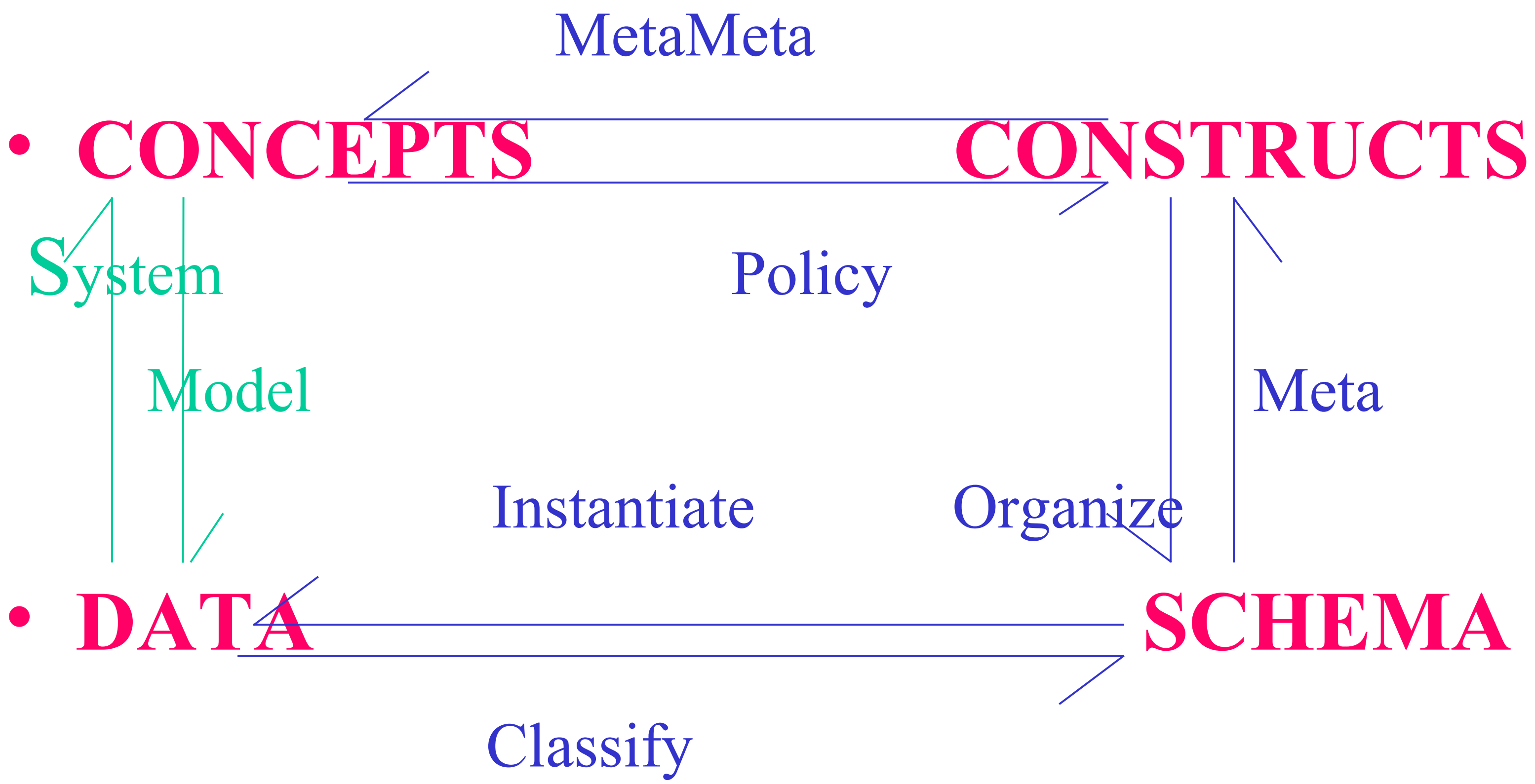
Red font -- categories

Functors

- Relationships between categories at adjacent levels are given by a functor
  - For example:
  - Meta: **SCHEMA** ⟶ **CONSTRUCTS**
  - Meta is a functor

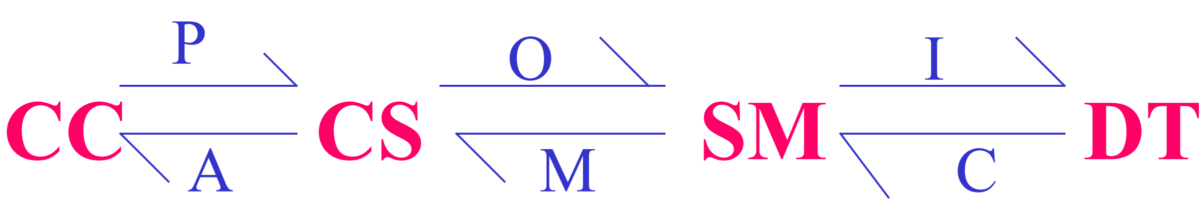
Blue font -- functors

Levels in Functorial Terms



Green font - composed functors: System = MetaMeta o Meta o Classify

Composition of Adjoint Functors

- Classify -- C
- Meta -- M
- MetaMeta -- A
- Policy -- P
- Organise -- O
- Instantiate -- I
- 

Composed adjunction

Adjunctions

- The adjointness between two functors is given by a 4-tuple e.g. for
- **CC** $\underset{A}{\overset{P}{\rightleftarrows}}$ **CS**
- <P, A, η, ∈>
  - ← η unit of adjunction measures change from initial cc to cc obtained by following P and A ($1_{CC} \longrightarrow$ AP(cc))
  - ∈ counit of adjunction measures PA(cs) $\longrightarrow 1_{cs}$
  - Unit and counit give measure of creativity of arrows and preservation of style in mapping by functors.
  - If complete preservation of style (∈ =1) and no creativity (η=0) -- isomorphism.

Composed Adjunction for Four Levels

Represents complex mappings across the levels of the system

$$< IOP, AMC, AM\overline{\overline{\eta_{cc}}}OP \bullet A\overline{\eta_{cc}}P \bullet \eta,$$

$$\overline{\overline{\varepsilon_{dt}}} \bullet I\overline{\varepsilon_{dt}}C \bullet IO\varepsilon_{dt}MC >$$

Unit of adjunction is a composition of :

$\eta_{cc} : 1_{cc} \to AP(cc)$ with $A\overline{\eta_{cc}}P : AP(cc) \to AMOP(cc)$

with $AM\overline{\overline{\eta_{cc}}}OP : AMOP(cc) \to AMCIOP(cc)$

Counit of adjunction is a composition of :

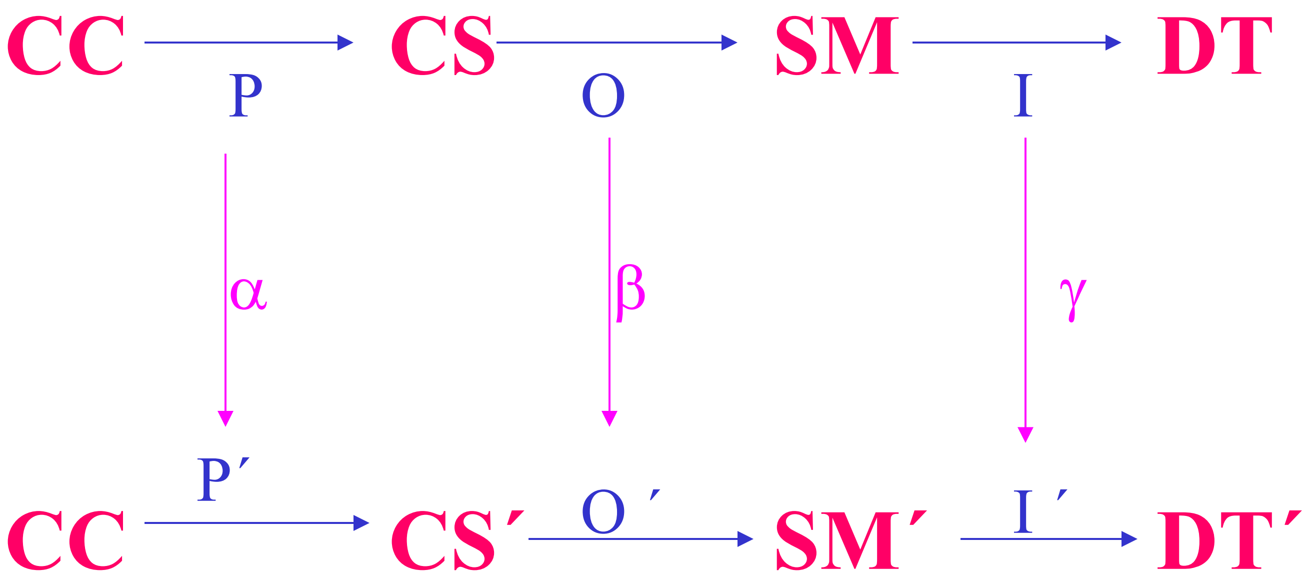
$IO\varepsilon_{dt}MC : IOPAMC(dt) \to IOMC(dt)$ with

$I\overline{\varepsilon_{dt}}C : IOMC(dt) \to IC(dt)$ with $\overline{\overline{\varepsilon_{dt}}} : IC(dt) \to 1_{dt}$

Benefits of Approach

- Can represent relationships between levels, either:
  - abstractly with one relationship from top to bottom levels
  - in much more detail with all combinations of adjoints expressed.

Comparing one System with Another



α,β, γ are natural transformations (comparing functors)

Godement Calculus

- Rules showing:
  - composition of functors and natural transformations is associative
  - natural transformations can be composed with each other
- For example:
- $(I'O')\,\alpha = I'(O'\,\alpha)$; $\gamma(OP) = (\gamma\,O)P$

$\forall\ \gamma\,\beta = (\gamma\,O) \circ (I'\,\beta)$; $\beta\,\alpha = \beta P \circ (O'\,\alpha)$

Four Levels are Sufficient

- In category theory:
  - objects are identity arrows
  - categories are arrows from object to object
  - functors are arrows from category to category
  - natural transformations are arrows from functor to functor
- An arrow between natural transformations is a composition of natural transformations, not a new level

Analogous Levels for Interoperability

Level	Category	Architecture
1. data values	Objects (identity arrows)	id_{dt}
2. named values	Category	**DT**
3. classified values	Functor	C: **DT** $\longrightarrow$ **SM**
4. contrasted representation	Natural transformation	$\alpha^* \circ \beta^*$ (α^* is dual of α)

Discussion

- Category theory shows that:
  - four levels are ideal for interoperability
  - more than four yields no benefits
  - less than four gives only local interoperability
- Categorical approach provides:
  - an architecture for universal interoperability
  - a calculus (Godement) for composing mappings at any level
  - adjunctions for evaluating two-way mappings

Quantum Databases

- Recent area of interest
- Following Grover's work on searching algorithms
- Following initial work by Peter Sellinger, we are developing database query language for the quantum area
- Based on category theory (entanglements as limits, superpositioning as colimts)

References 1

- Our work (available from NR's home page)
  - Heather, M A, & Rossiter, B N, The Anticipatory and Systemic Adjointness of E-Science Computation on the Grid, Computing Anticipatory Systems, Proceedings CASYS`01, Liège, Dubois, D M, (ed.), AIP Conference Proceedings **627** 565-574 (2002).
  - Rossiter, B N, Heather, M A, & Nelson, D A, A Universal Technique for Relating Heterogeneous Data Models, 3rd International Conference on Enterprise Information Systems (ICEIS), Setúbal, **I** 96-103 (2001).
  - Heather, M A, & Rossiter, B N, Constructing Standards for Cross-Platform Operation, Software Quality Journal, 7(2) 131-140 (1998).

References 2

- Category Theory and Computing Science:
  - Barr, M, & Wells, C, Category Theory for Computing Science, Prentice-Hall (1990).
  - Mac Lane, S, Categories for the Working Mathematician, Springer, 2^{nd} ed (1998).

- Category Theory and Information Systems: some other workers
  - Zinovy Diskin (USA, formerly Latvia)
  - Boris Cadish (Latvia)
  - Robert Rosebrugh (Canada)
  - Michael Johnson (Australia)
  - Christopher Dampney (Australia)
  - Michael Heather (Northumbria)
  - David Nelson (Sunderland)
  - Arthur ter Hofstede (Australia, formerly Holland)
- Many other workers on category theory and program semantics