

# First Year Topics Data Integration

P.J. McBrien

Imperial College London

# The Problem Domain

$DB_1$

staff			
<u>name</u>	sex	salary	maternity
Peter	M	10000	null
Alex	F	12000	false

↔

$DB_2$

female		
<u>name</u>	salary	maternity
Alex	12000	false

male	
<u>name</u>	salary
Peter	10000

```
SELECT name,salary
FROM staff
WHERE sex='M'
```

↔

```
SELECT name,salary
FROM male
```

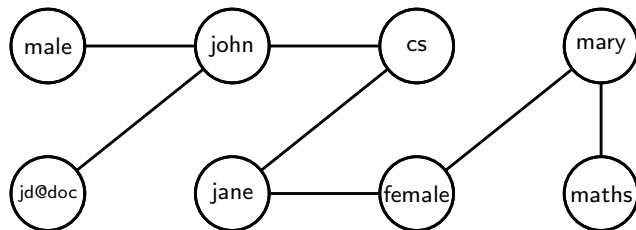
```
INSERT INTO staff
VALUES ('Peter','M',10000,null)
```

↔

```
INSERT INTO male
VALUES ('Peter',10000)
```

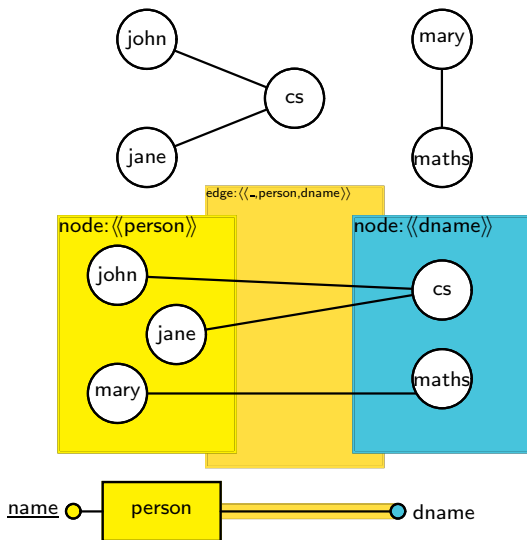
## What is a database schema?

*We often build computer systems to support some activity in the real world* → **Universe of Discourse (UoD)**

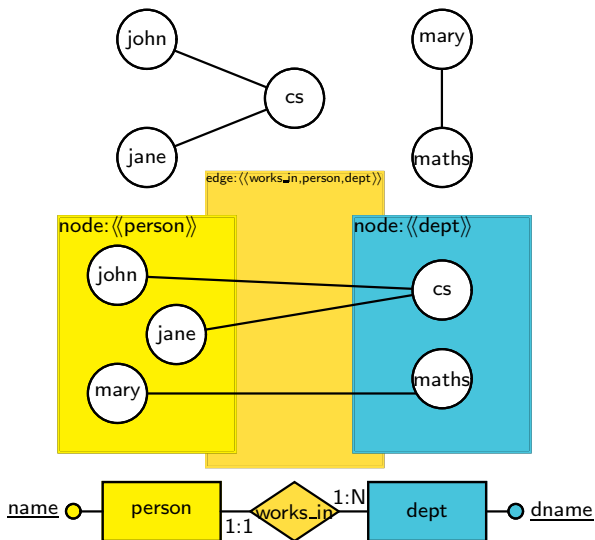


- A **database (DB)** is a computer system to hold data
- A **schema** is a structure for a DB
- Group UoD data into sets, tables, object classes, ...
  - **entity-relationship (ER)** modelling

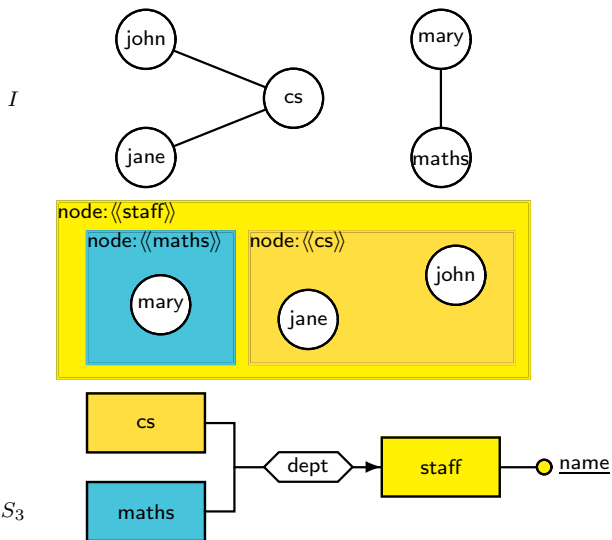
# Alternative Models of the UoD (1)



## Alternative Models of the UoD (2)



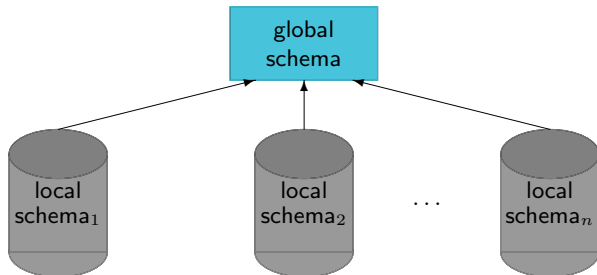
## Alternative Models of the UoD (3)



# Schema Integration

*Different designers may structure data differently*

- **schema integration** integrates DBs to form a common **global schema** (or **mediated schema**)
- required to build **distributed systems** and **data warehouses**

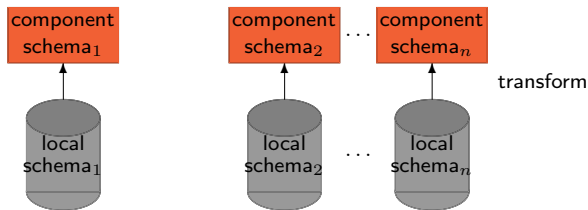


# Logical Model for Heterogeneous Databases



## local schemas transformed into component schemas

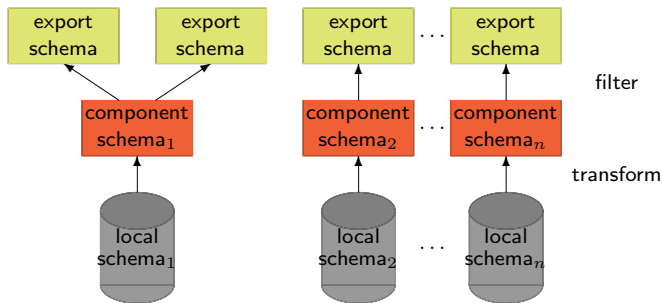
- need a common data model (CDM)
- could use relational, UML, ER, XML, OWL-DL, ...



# Logical Model for Heterogeneous Databases

component schemas **filtered** to export schemas

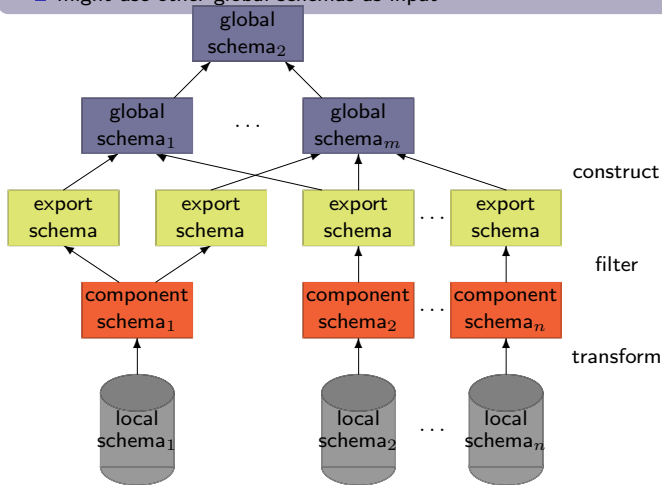
- legal use of data
- security
- operational efficiency



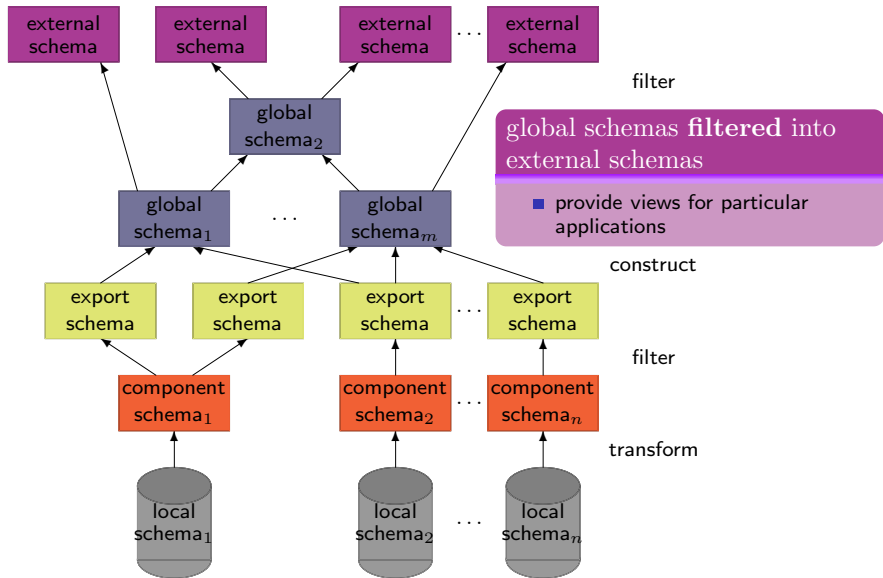
# Logical Model for Heterogeneous Databases

exports schemas **integrated** to form global schemas

- transformation of schemas
- might use other global schemas as input



# Logical Model for Heterogeneous Databases



## 1 schema conforming

- **conflict detection** and **conflict resolution**
- *versions of schemas are produced which represent the same concepts in the same manner*

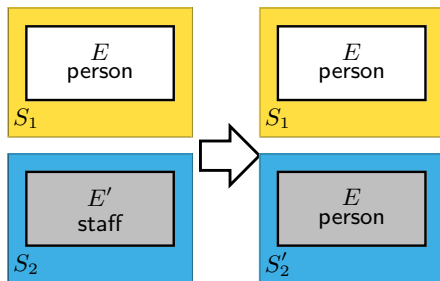
## 2 schema merging

- identify related concepts found in different schemas
- produce a single output merged schema

## 3 schema improvement

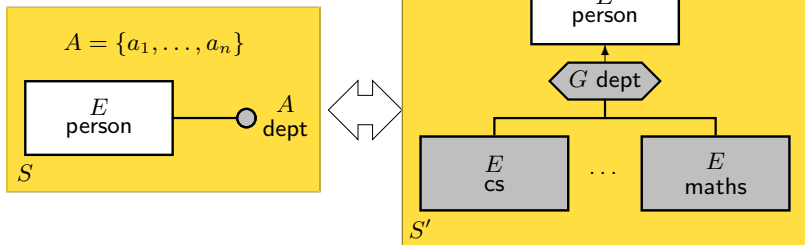
- to improve the quality of the model presented

## Example of Conflict resolution: Synonym removal

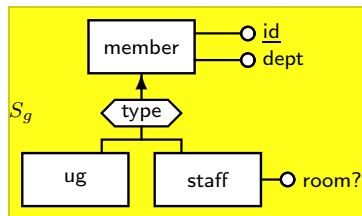
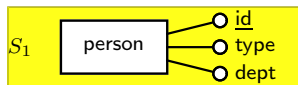


# Example of Structural Equivalences

attribute and generalisation equivalence



# Data Integration: Global As View (GAV)

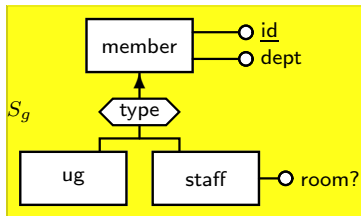
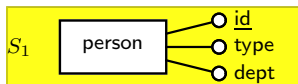


$S_1 \rightarrow S_g$

```
sg:member(Id,Dept) :-  
    s1:person(Id,-,Dept).  
sg:staff(Id,-) :-  
    s1:person(Id,'staff',-).  
sg:ug(Id) :-  
    s1:person(Id,'ug',-).
```

- global schema defined as views over local schemas (normally using **Datalog**)
- problem with source schema evolution

# Data Integration: Global As View (GAV)



$S_1 \rightarrow S_g$

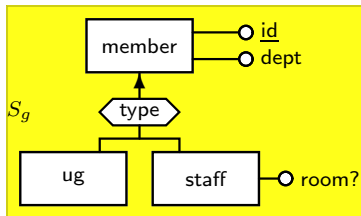
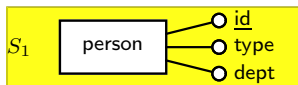
```
sg:member(Id,Dept) :-  
    s1:person(Id,_,Dept).  
sg:staff(Id,_) :-  
    s1:person(Id,'staff',_).  
sg:ug(Id) :-  
    s1:person(Id,'ug',_).
```

$S_2 \rightarrow S_g$

```
sg:member(Id,_) :-  
    s2:staff(Id,_.  
sg:staff(Id,Room) :-  
    s2:staff(Id,Room).
```

- exact definition of global constructs not given

# Data Integration: Local As View (LAV)



$S_g \rightarrow S_1$

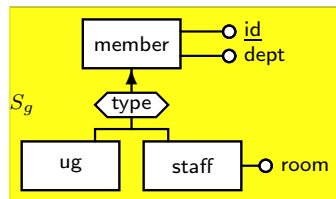
```
s1:person(Id,'ug',Dept) :-  
    sg:member(Id,Dept),  
    sg:ug(Id).  
s1:person(Id,'staff',Dept) :-  
    sg:member(Id,Dept),  
    sg:staff(Id,_).
```

$S_g \rightarrow S_2$

```
s2:staff(Id,Room) :-  
    sg:staff(Id,Room).
```

- source schemas defined as views over global schema

## Both As View (BAV)



$S_1 \rightarrow S_g$  BAV

- ① `addEntity(⟨⟨ug⟩⟩, {X | person_type(X, 'ug')})`
- ② `addEntity(⟨⟨staff⟩⟩, {X | person_type(X, 'staff')})`
- ③ `addGeneralisation(⟨⟨person, type, ug, staff⟩⟩)`
- ④ `deleteAttribute(⟨⟨person, type, notnull⟩⟩, {X, Y | ug(X), Y = 'ug'; staff(X), Y = 'staff'})`
- ⑤ `renameEntity(⟨⟨person⟩⟩, ⟨⟨member⟩⟩)`
- ⑥ `extendAttribute(⟨⟨staff, room, notnull⟩⟩)`

## Deriving GAV from BAV

$S_1 \rightarrow S_g$	BAV	GAV
①	addEntity( $\langle\langle ug \rangle\rangle, \{X \mid \text{person\_type}(X, 'ug')\}$ )	✓
②	addEntity( $\langle\langle staff \rangle\rangle, \{X \mid \text{person\_type}(X, 'staff')\}$ )	✓
③	addGeneralisation( $\langle\langle \text{person, type, ug, staff} \rangle\rangle$ )	✗
④	deleteAttribute( $\langle\langle \text{person, type, notnull} \rangle\rangle, \{X, Y \mid \text{ug}(X), Y = 'ug'; \text{staff}(X), Y = 'staff'\}$ )	✗
⑤	renameEntity( $\langle\langle \text{person} \rangle\rangle, \langle\langle \text{member} \rangle\rangle$ )	✓
⑥	extendAttribute( $\langle\langle \text{staff, room, notnull} \rangle\rangle$ )	✓

## Deriving GAV from BAV

From Step      GAV Rule Derived

- ①  $\text{ug}(X) :- \text{person\_type}(X, 'ug')$ .
- ②  $\text{staff}(X) :- \text{person\_type}(X, 'staff')$ .
- ⑤  $\text{member}(X) :- \text{person}(X)$ .  
 $\text{member\_id}(X, Y) :- \text{person\_id}(X, Y)$ .  
 $\text{member\_dept}(X, Y) :- \text{person\_dept}(X, Y)$ .
- ⑥  $\text{staff\_room}(X, Y) :- \text{staff}(X)$ .

## Deriving LAV from BAV

$S_g \rightarrow S_1$	BAV	LAV
6	contractAttribute( $\langle\langle$ staff,room,nonnull $\rangle\rangle$ )	X
5	renameEntity( $\langle\langle$ member $\rangle\rangle$ , $\langle\langle$ person $\rangle\rangle$ )	✓
4	addAttribute( $\langle\langle$ person,type,nonnull $\rangle\rangle$ , $\{X, Y \mid \text{ug}(X), Y = \text{'ug'}; \text{staff}(X), Y = \text{'staff'}\}$ )	✓
3	deleteGeneralisation( $\langle\langle$ person,type,ug,staff $\rangle\rangle$ )	X
2	deleteEntity( $\langle\langle$ staff $\rangle\rangle$ , $\{X \mid \text{person\_type}(X, \text{'staff'})\}$ )	X
1	deleteEntity( $\langle\langle$ ug $\rangle\rangle$ , $\{X \mid \text{person\_type}(X, \text{'ug'})\}$ )	X

## LAV rules extracted from BAV rules

### From Step LAV Rule Derived

- 5 person(X) :- member(X).  
person\_id(X,Y) :- member\_id(X,Y).  
person\_dept(X,Y) :- member\_dept(X,Y).
- 4 person\_type(X,Y) :- ug(X),Y='ug'.  
person\_type(X,Y) :- staff(X),Y='staff'.

## Mappings

LAV

GAV

BAV

GLAV

## Tools

AutoMed [BKL<sup>+</sup>04, SRM08]

Clio [PHV02, HHM<sup>+</sup>09]

DB-Main [HEH<sup>+</sup>94, HH03, Hai05, CH10]

## Approaches

Dataspaces [FHM05, HBF<sup>+</sup>09, BPE<sup>+</sup>10]

Model Mangement [Ber03, BM07, SM08]

Ontologies [Noy04, UGM07]

Peer-to-Peer [CDLR04, HIST03, MP03b]

- **schema** is the structure of a DB
- UoD may be represented in many alternative schema
- independent DB designs means that one application must access alternative schemas
- **schema integration** combines independent DB designs into one **global schema**
- Mappings express
  - Transformation of schema from one modelling language to another
  - Transformations of the structure of schema
- **data integration**=schema integration+**distributed query processing**

The ER modelling language was first described by Chen [Che76]. However, it should be noted that there are many possible variants of the ER modelling language. A good textbook covering the subject of ER modelling is [EN94].

The Federated Database architecture model was proposed by [SL90]. This describes the five level federated database architecture, and the process by which a global schema (called in that paper a federal schema) is built from the local schemas of various databases. Note that federated databases are often called heterogeneous databases. An alternative view of how to conduct data integration is found in the mediator approach proposed by [Wie92].

GAV and LAV have been studied by many researchers, a good overview of both of the approaches can be found in [Len02]. The term BAV was first proposed in [MP03a], but the technique was used in [MP98, MP99b, MP99a, MP01, MP02].

A survey of various techniques used in schema integration, and a selection on some of the transformations that can be applied to database schemas can be found in [BLN86]. A mathematical approach to the description of schema transformations using the BAV type approach is found in [PM98].

A general framework for the construction heterogeneous databases (also known as federated databases) is found in [SL90].

Further descriptions of the transformation language and query transformation process of BAV may be found in [MP99a]. How to support various modelling languages in BAV, and to translate between different modelling languages using BAV, is described in [MP99b]. The latest development and implementation work of BAV may be found at <http://www.doc.ic.ac.uk/automed>

Examples of tools implementing the GAV approach are TSIMMIS [CGMH<sup>+</sup>94], InterViso [THEB95], Garlic [RS97], and DB-MAIN [HH03, CH10], examples of the LAV approach are IM [LRO96], Agora [MFK01] and Piazza [HIST03]. For GLAV there is Clio [HHM<sup>+</sup>09] and BAV there is AutoMed [BKL<sup>+</sup>04].

- [Ber03] P.A. Bernstein.  
Applying model management to classical meta data problems.  
In *Proc. CIDR'03*, 2003.
- [BKL<sup>+</sup>04] M. Boyd, S. Kittivoravitkul, C. Lanzanitis, P.J. McBrien, and N. Rizopoulos.  
AutoMed: A BAV data integration system for heterogeneous data sources.  
In *Proc. CAiSE'04*, volume 3084 of *LNCS*, pages 82–97. Springer, 2004.
- [BLN86] C. Batini, M. Lenzerini, and S. Navathe.  
A comparative analysis of methodologies for database schema integration.  
*ACM Computing Surveys*, 18(4):323–364, 1986.
- [BM07] P.A. Bernstein and S. Melnik.  
Model management 2.0: Manipulating Richer Mappings.  
In *Proc. SIGMOD*, 2007.

- [BPE<sup>+</sup>10] K. Belhajjame, N.W. Paton, S.M. Embury, A.A.A. Fernandes, and C. Hedeler.  
Feedback-based annotation, selection and refinement of schema mappings for dataspace.  
In *Proc. EDBT*, pages 573–584, 2010.
- [CDLR04] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati.  
Logical foundations of peer-to-peer data integration.  
In *Proc. PODS*, pages 241–251, 2004.
- [CGMH<sup>+</sup>94] S.S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J.D. Ullman, and J. Widom.  
The TSIMMIS project: Integration of heterogeneous information sources.  
In *Proc. 10th Meeting of the Information Processing Society of Japan*, pages 7–18, October 1994.
- [CH10] A. Cleve and A. Brogneaux J. Hainaut.  
A conceptual approach to database applications evolution.  
In *Proc. ER*, pages 132–145, 2010.
- [Che76] P.P. Chen.  
The Entity-Relationship model — toward a unified view of data.  
*ACM Trans. Database Systems*, 1(1):9–36, 1976.

- [EN94] R. Elmasri and S. Navathe.  
*Fundamentals of Database Systems*.  
 The Benjamin/Cummings Publishing Company, Inc., 2nd edition, 1994.
- [FHM05] M. Franklin, A. Halevy, and D. Maier.  
 From databases to dataspace: a new abstraction for information management.  
*SIGMOD Record*, 34:27–33, 2005.
- [Hai05] J-L. Hainaut.  
 Transformation-based database engineering.  
 In P. van Bommel, editor, *Transformation of Knowledge, Information and Data: Theory and Applications*, chapter 1, pages 1–28. IGI Global, 2005.
- [HBF<sup>+</sup>09] C. Hedeler, K. Belhajjame, A.A.A. Fernandes, S.M. Embury, and N.W. Paton.  
 Dimensions of dataspace.  
 In *Proc. BNCOD*, pages 55–66, 2009.
- [HEH<sup>+</sup>94] J-L. Hainaut, V. Englebert, J. Henrard, J-M. Hick, and D. Roland.  
 Database evolution: the DB-MAIN approach.  
 In *Proc. ER'94*, LNCS, pages 112–131. Springer, 1994.
- [HH03] J-M. Hick and J-L. Hainaut.

Strategy for database application evolution: The DB-MAIN approach.  
 In *Proc. ER'03*, volume 2813 of *LNC3*, pages 291–306. Springer, 2003.

- [HHM<sup>+</sup>09] L.M. Haas, M.A. Hernandez, R.J. Miller, L. Popa, and Y. Velegrakis.  
 Clio: Schema Mapping Creation and Data Exchange.  
 In A.T. Borgida, V. Chaudhri, P. Giorgini, and E.S. Yu., editors,  
*Conceptual Modeling: Foundations and Applications, Essays in Honor  
 of John Mylopoulos*, 2009.
- [HIST03] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov.  
 Schema mediation in peer data management systems.  
 In *Proc. ICDE'03*. IEEE, 2003.
- [Len02] M. Lenzerini.  
 Data integration: A theoretical perspective.  
 In *Proc. PODS'02*, pages 233–246. ACM, 2002.
- [LRO96] A. Levy, A. Rajamaran, and J. Ordille.  
 Querying heterogeneous information sources using source description.  
 In *Proc 22nd VLDB*, pages 252–262, 1996.
- [MFK01] I. Manolescu, D. Florescu, and D. Kossmann.  
 Answering XML queries on heterogeneous data sources.  
 In *Proc. of 27th International Conference on Very Large Data Bases*,  
 pages 241–250, 2001.

- [MP98] P.J. McBrien and A. Poulovassilis.  
Automatic migration and wrapping of database applications — a schema transformation approach.  
Technical Report TR98-10, King's College London, 1998.
- [MP99a] P.J. McBrien and A. Poulovassilis.  
Automatic migration and wrapping of database applications — a schema transformation approach.  
In *Proc. ER'99*, volume 1728 of *LNCS*, pages 96–113. Springer, 1999.
- [MP99b] P.J. McBrien and A. Poulovassilis.  
A uniform approach to inter-model transformations.  
In *Proc. CAiSE'99*, volume 1626 of *LNCS*, pages 333–348. Springer, 1999.
- [MP01] P.J. McBrien and A. Poulovassilis.  
A semantic approach to integrating XML and structured data sources.  
In *Proc. CAiSE'01*, volume 2068 of *LNCS*, pages 330–345. Springer, 2001.
- [MP02] P.J. McBrien and A. Poulovassilis.  
Schema evolution in heterogeneous database architectures, a schema transformation approach.  
In *Proc. CAiSE'02*, volume 2348 of *LNCS*, pages 484–499. Springer, 2002.

- [MP03a] P.J. McBrien and A. Poulovassilis.  
Data integration by bi-directional schema transformation rules.  
In *Proc. ICDE'03*, pages 227–238. IEEE, 2003.
- [MP03b] P.J. McBrien and A. Poulovassilis.  
Defining peer-to-peer data integration using both as view rules.  
In *Proc. DBISP2P, at VLDB'03*, pages 91–107, 2003.
- [Noy04] N.F. Noy.  
Semantic integration: A survey of ontology-based approaches.  
*SIGMOD Record*, 33(4):65–70, 2004.
- [PHV02] L. Popa, M.A. Hernandez, and Y. Velegrakis *et al.*  
Mapping XML and relational schemas with Clio.  
In *Proc. ICDE'02*, pages 498–499, 2002.
- [PM98] A. Poulovassilis and P.J. McBrien.  
A general formal framework for schema transformation.  
*Data and Knowledge Engineering*, 28(1):47–71, 1998.
- [RS97] M.T. Roth and P. Schwarz.  
Don't scrap it, wrap it! A wrapper architecture for data sources.  
In *Proc. 23rd VLDB Conference*, pages 266–275, Athens, Greece, 1997.
- [SL90] A. Sheth and J. Larson.

Federated database systems.

*ACM Computing Surveys*, 22(3):183–236, 1990.

- [SM08] A.C. Smith and P.J. McBrien.  
A generic data level implementation of modelgen.  
In *Proc. BNCOD*, pages 63–74, 2008.
- [SRM08] A.C. Smith, N. Rizopoulos, and P.J. McBrien.  
AutoMed model management.  
In *Proc. 27th ER*, volume 5231 of *LNCS*, pages 542–543, 2008.
- [THEB95] M. Templeton, H.Henley, E.Maros, and D.J. Van Buer.  
InterViso: Dealing with the complexity of federated database access.  
*The VLDB Journal*, 4(2):287–317, April 1995.
- [UGM07] O. Udrea, L. Getoor, and R.J. Miller.  
Leveraging data and structure in ontology integration.  
In *Proc. SIGMOD*, 2007.
- [Wie92] G. Wiederhold.  
Mediators in the architecture of future information systems.  
*IEEE Computer*, 25(3):38–49, March 1992.