

# **Relational Wrapper for Navigation-Driven Lazy Mediator**

---

**Michail Petropoulos**

**Database Lab/CSE DEPARTMENT  
San Diego Supercomputer Center**



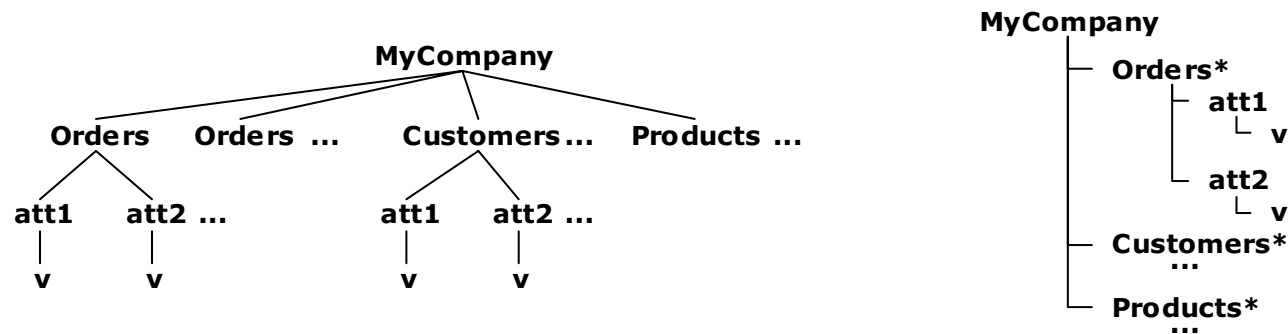
# Outline

---

- Generic X-View (GX-View)
  - Representing a relational database using an XML document
- X-Views
  - How to build a custom view for relational databases
- Architecture of the Relational Wrapper
  - The role of the assistant mediator
- Query Processing
  - Translating XMAS queries to SQL queries
- Future Directions
  - Constructing results for the Lean XML Fragment/LXP Protocol
- MIX Architecture
  - Virtual XML Document/VXD Architecture
- XMAS Algebra
  - Query language for XML data
- Rewriting Rules

# Generic X-View (GX-View)

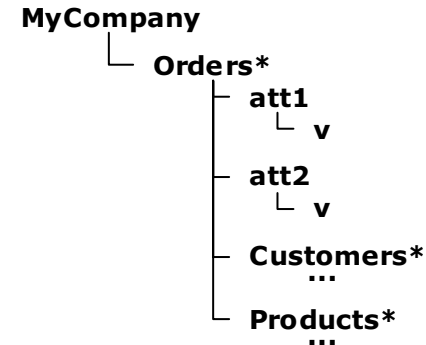
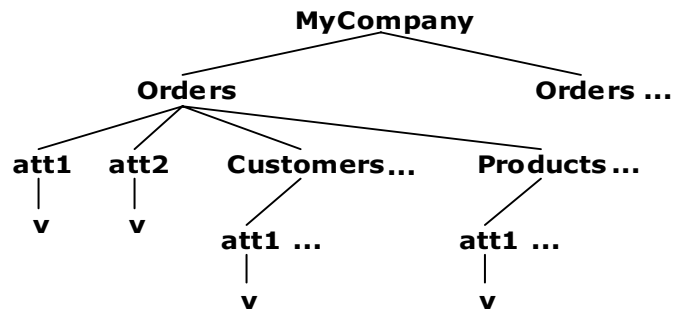
## Relational DB as an XML Document



- The root of the XML document is the name of the RDB
- Tuples are represented as subtrees, that contain attributes and values, and are labeled after the the relation they belong to

# X-Views

## Custom view for Relational DB

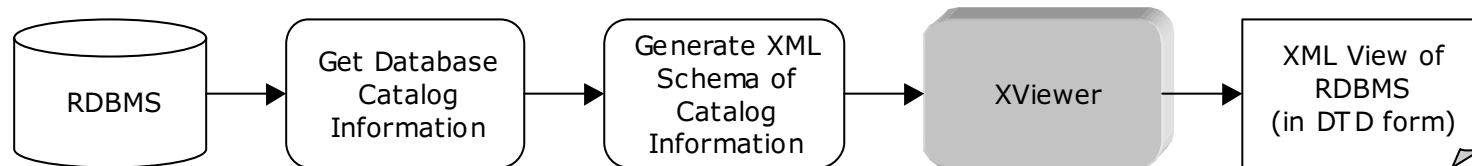


- The mediator engineer would like to choose a custom view of the RDB, other than the generic one, which would make more sense to the mediator's client and would represent better the semantics of the relational schema
- The client may use the DTD of this XML representation to construct a query via BBQ

# X-Views (cont.)

---

## X-Viewer

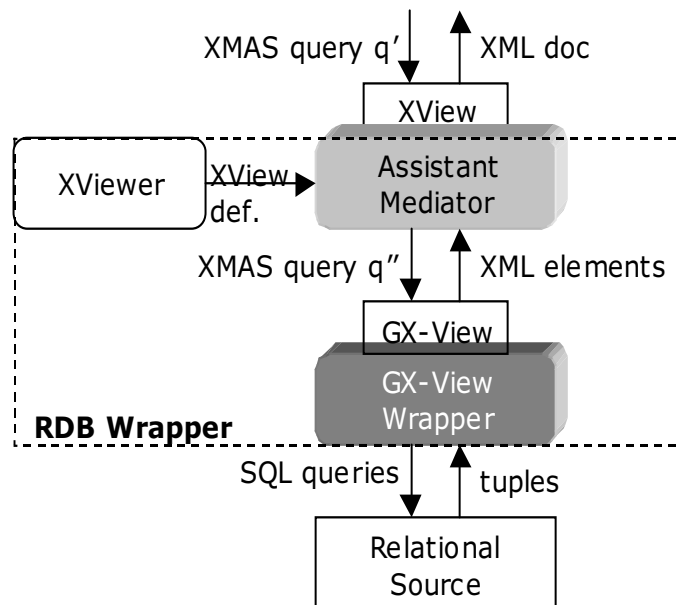


The X-Viewer is the tool that generates possible XViews (DTDs) to the mediator engineer when the following three methods are applied to the relational schema:

- *MaxIndegree*: The relation which has the highest incoming edges (mostly referenced) is chosen as the starting relation.
- *ZeroIndegree*: The relation which has no incoming edges (many N:M relationships) is chosen as the starting relation
- *User Defined*: The user enters the name of the relation to start with as a parameter

# Architecture for RDB Wrapper

## The Role of the Assistant Mediator

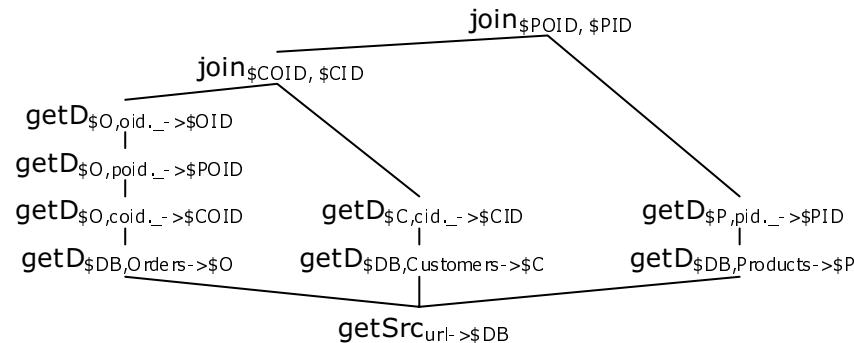


- Modular architecture leverages mediator technology
- GX-View wrapper supports queries against the GXView
- Assistant mediator translates a query  $q'$  against an XView to a query  $q''$  against the GXView
- The XView definition is imported to the assistant mediator

# Query Processing by GX-View Wrapper

## Translating XMAS queries to SQL

- GX-View wrapper accepts only the body part of a XMAS query
- Example: The body of the view definition in XMAS algebraic form for the above XView is the following:



The list of bindings that this body produces are generated from the following SQL query:

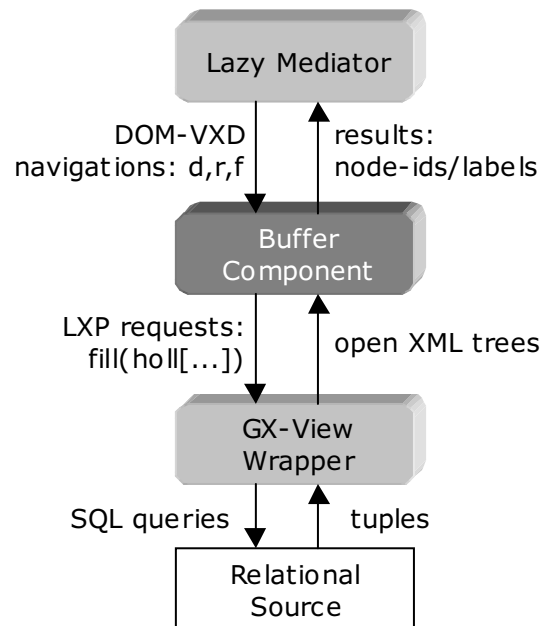
```
SELECT *
FROM Orders AS $O, Customers AS $C, Products AS $P
WHERE coid = cid AND poid = pid
```





# Future Directions

## Constructing Results for the Lean XML Fragment/LXP Protocol



- The lazy mediator uses a buffer component to handle the different granularities of the sources
- The LXP protocol is used to extract (groups of) XML elements from the sources as needed from the DOM commands
- The fill request causes the GX-View wrapper to produce a list of bindings from a predefined number of tuples, and send them to the buffer in XML format

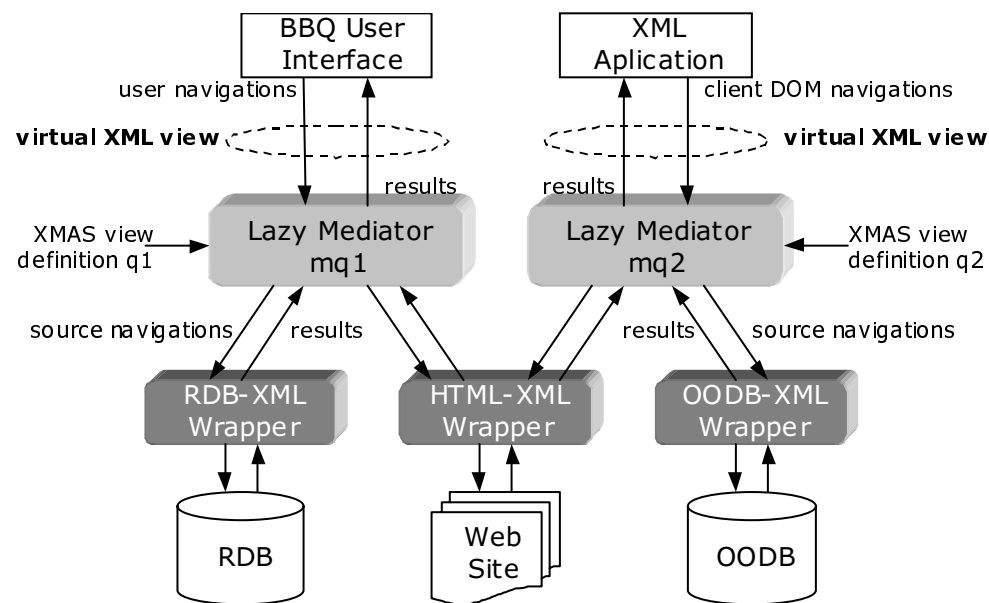
# References

---

- "*Navigation-Driven Evaluation of Virtual Mediated Views*", B. Ludäscher, Y. Papakonstantinou, P. Velikhov
- "*XML-Based Information Mediation with MIX*", C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov
- "*XViews: XML views of relational schemas*", C. Baru
- "*The Lorel Query Language for Semistructured Data*", S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. Wiener
- "*Optimizing Queries across Diverse Data Sources*", L. Haas, D. Kossmann, E. Wimmers, J. Yang
- "*A Query Translation Scheme for Rapid Implementation of Wrappers*", Y. Papakonstantinou, A. Gupta, H. Garcia-Molina, J. Ullman
- "*Object Fusion in Mediator Systems*", Y. Papakonstantinou, S. Abiteboul, H. Garcia-Molina

# MIX Architecture

## Virtual XML Document/VXD Architecture



- The mediator composes the client's query with the view definition and produces an algebraic plan
- Queries are executed against the wrappers

- Client gets back a virtual answer document
- Client uses navigation commands to browse the document
- Client's navigation commands are translated to source commands in order to retrieve the requested data

# XMAS Algebra

---

## Query Language for XML data

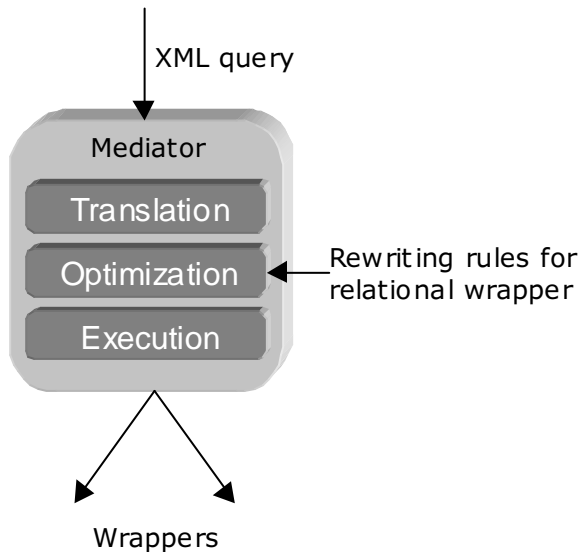
An XML QL operator takes as input a list of bindings(trees) for various element variables and produces a new one

- getDescendants <sub>$e, re \rightarrow ch$</sub> : Extracts the children of variable  $e$  using the general path expression  $re$  and binds them to variable  $ch$
- groupBy <sub>$\{u_1, \dots, u_k\}, u \rightarrow l$</sub> : Groups bindings of variable  $u$  by bindings of  $u_1, \dots, u_k$  and  $l$  is the label of the resulting list
- concatenate <sub>$x, y \rightarrow z$</sub> : Concatenates the bindings for variables  $x$  and  $y$  and binds them to variable  $z$
- createElement <sub>$label, ch \rightarrow e$</sub> : For each binding of  $ch$  it outputs a new element labeled  $label$  and binds it to variable  $e$
- orderBy <sub>$x_1, \dots, x_k$</sub> : Orders by bindings of  $x_1, \dots, x_k$
- Relational select, project, union, join and anti semi-join

# Query Processing by Mediator

## Rewriting Rules

Mediator's optimizer must be provided with a set of rewriting rules in order to be able to produce a plan that conforms the query pattern supported by the relational wrapper. These rules have the following form:



- Normalization of general path expressions in operators
- Joins involving one source must be expressed as selections
- Operators supported by the wrapper must pass below operators that are not (e.g. *join* under *groupBy*)