The IFIP Conference on

October 26 – 28, 2005, Poznan, Poland

ifip

Integration of XML data in P2P E-commerce applications

Tadeusz Pankowski

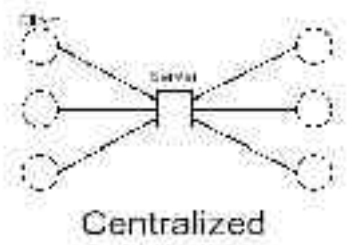
Poznań University of Technology, Institute of Control and Information Engineering

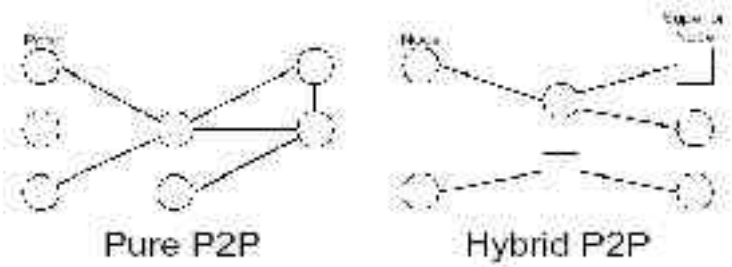
Adam Mickiewicz University, Faculty of Mathematics and Computer Science

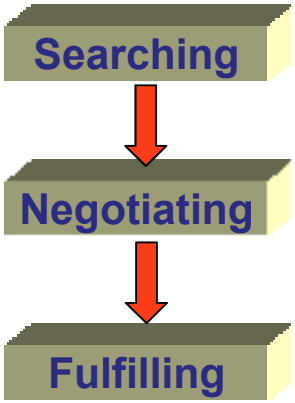
Overview

1. Motivation – XML documents in e-commerce and needs of their integration
2. Approaches to data integration: materializing (*data transformation*) vs virtual data integration (*query reformulation*)
3. XML schema mapping – a language XDMap
4. Automating mapping specification and query reformulation
5. Conclusion

Motivation – P2P E-commerce





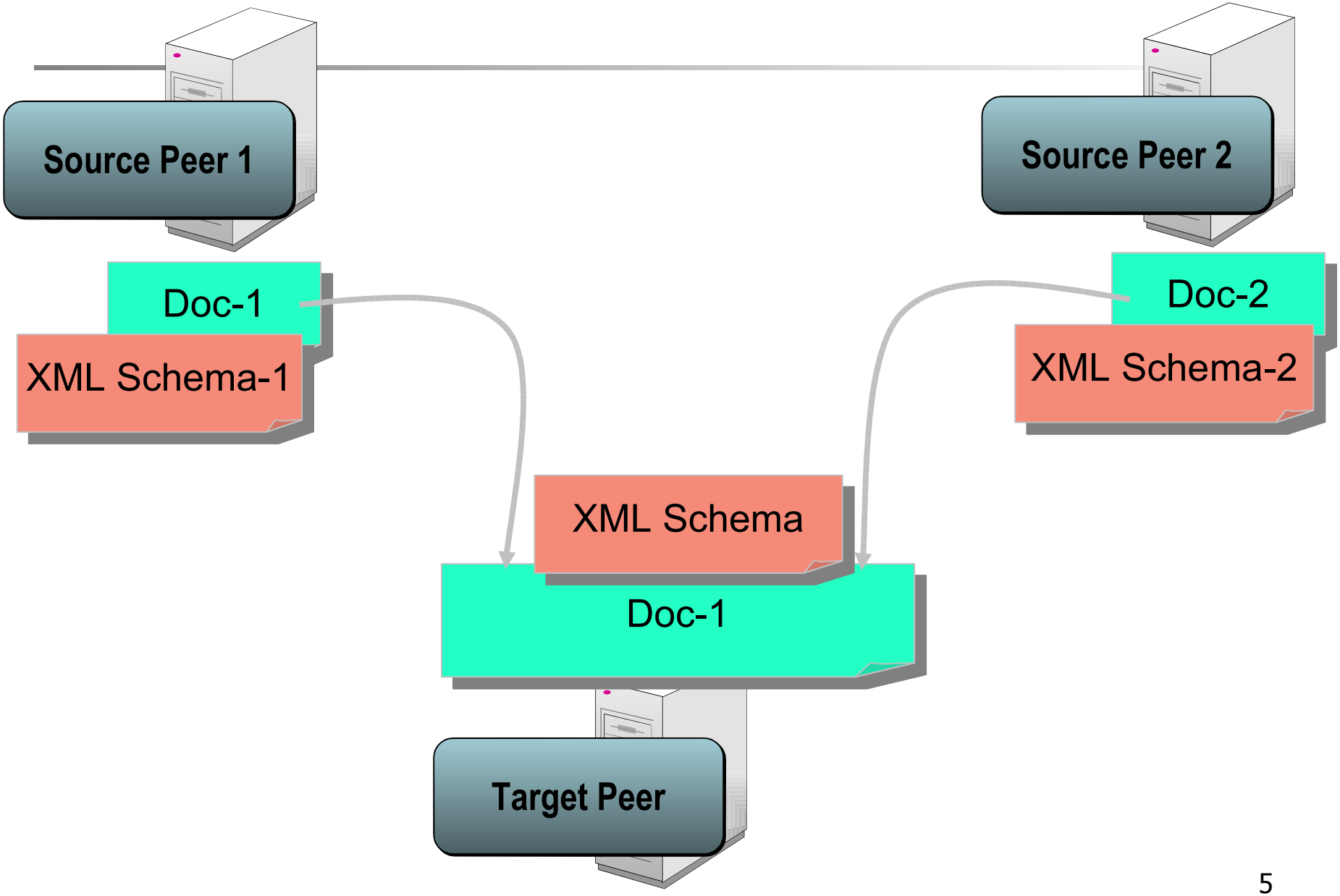


1. E-commerce applications need new solutions in the field of integration, interchange, and transformation of data across the global marketplace.
2. Traditional client-server systems are replaced by P2P systems that offer more flexibility.
3. Advanced data management features are expected to function automatically or semi-automatically.
4. A basic functionality of a system supporting e-commerce applications is the **integration** of external data sources or data services.

Sample scenario

- a newly hired employee requires a laptop for his workplace,
- the employee defines characteristics of the laptop and some additional conditions using an online website which is actually composed of several distinct and interacting Web Services,
- based on the specified order, catalog Web Service contacts Web Services of some hardware vendors and collects a list of offers,
- a price comparisons can be done and some additional negotiations can be performed (e.g. a service contract, prices, payment, etc.).

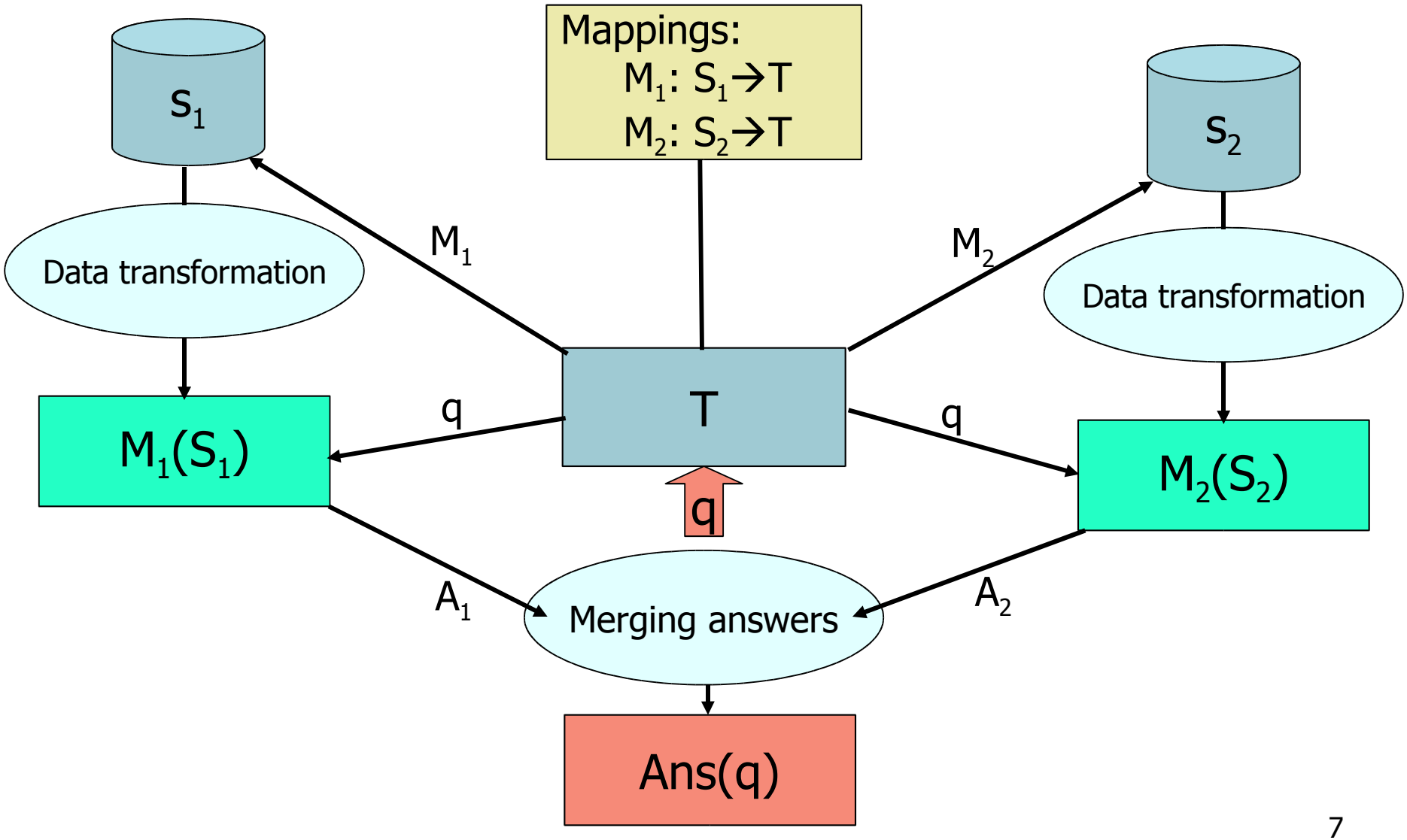
Integrating data from XML documents



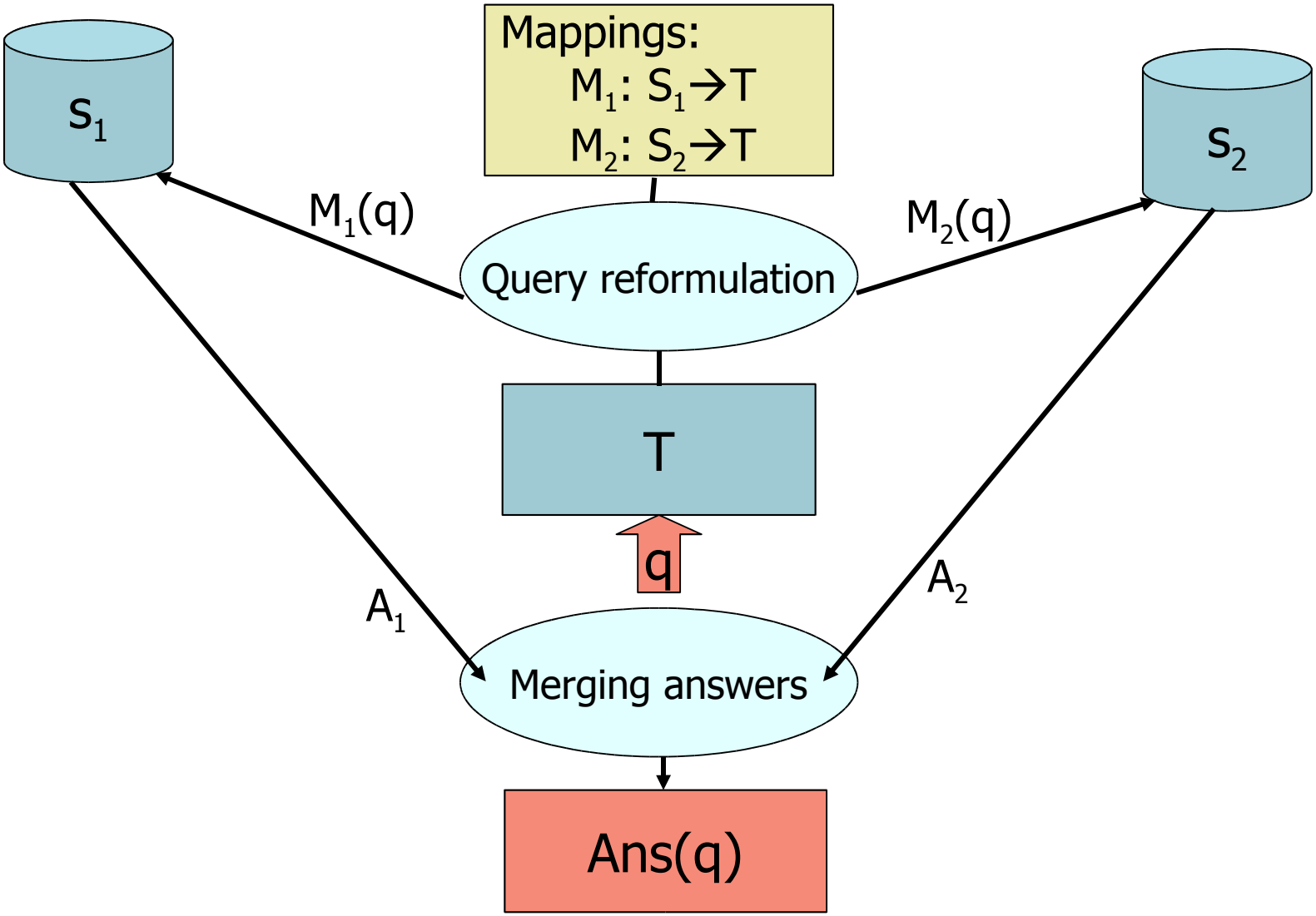
Data integration

1. System architecture
   - Client-server
   - P2P Data Management Systems (PDMS)
2. Strategy of integration
   - Data transformation (materializing integration)
   - Query reformulation (virtual integration)

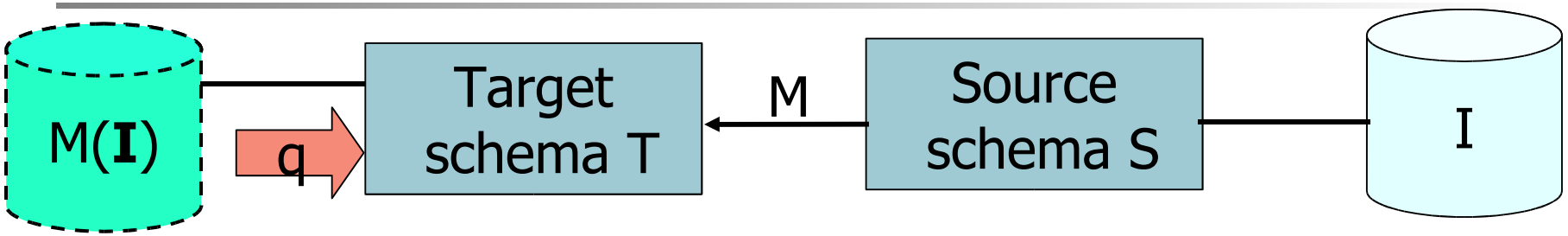
Data integration by data transformation

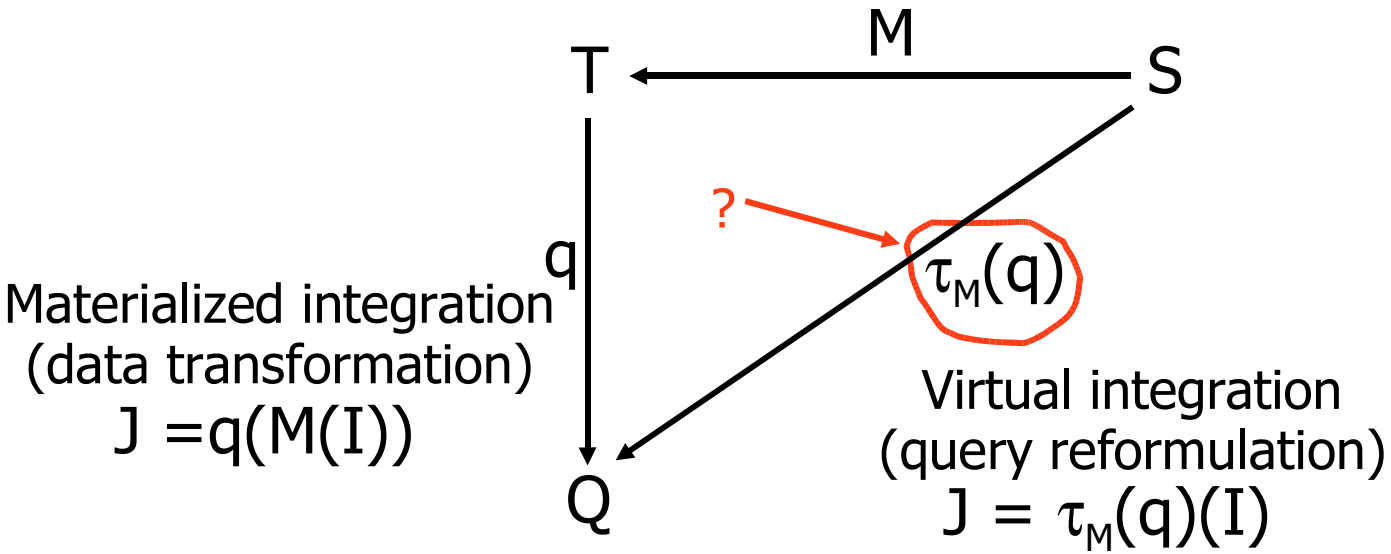


Data integration by query reformulation



Materialized vs Virtual Data integration





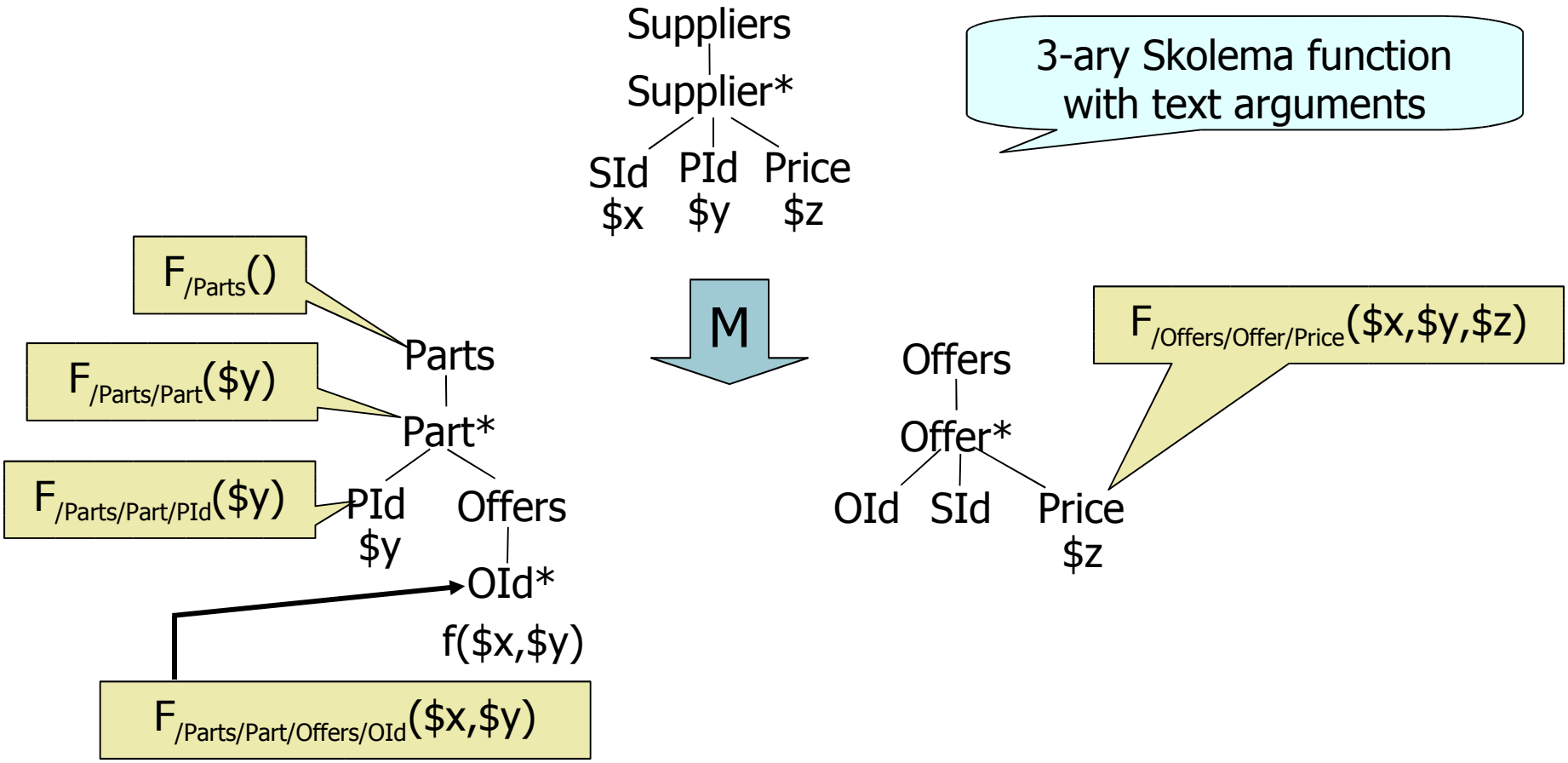
$M \circ q = \tau_M(q)$

Mappings

1. In both cases mappings between XML schemas must be established
2. What do we need to define mappings?

- a language to express relationships between schemas (XDMap)
- a method to create specification – as automatically as possible
- mappings can be generated automatically from key constraints (within XML Schema)

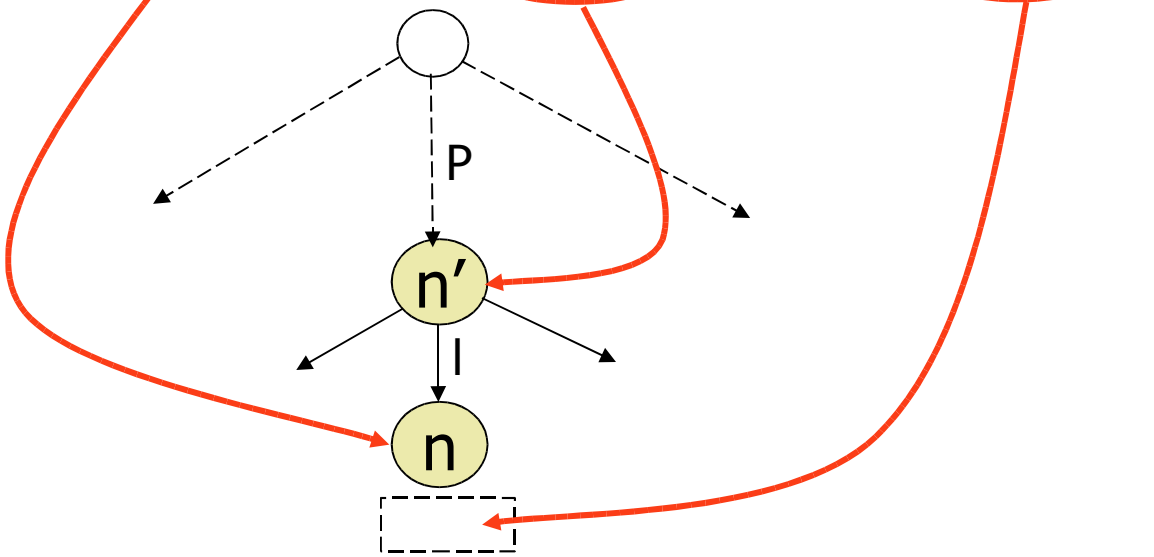
The idea of mappings



The idea of mapping specification

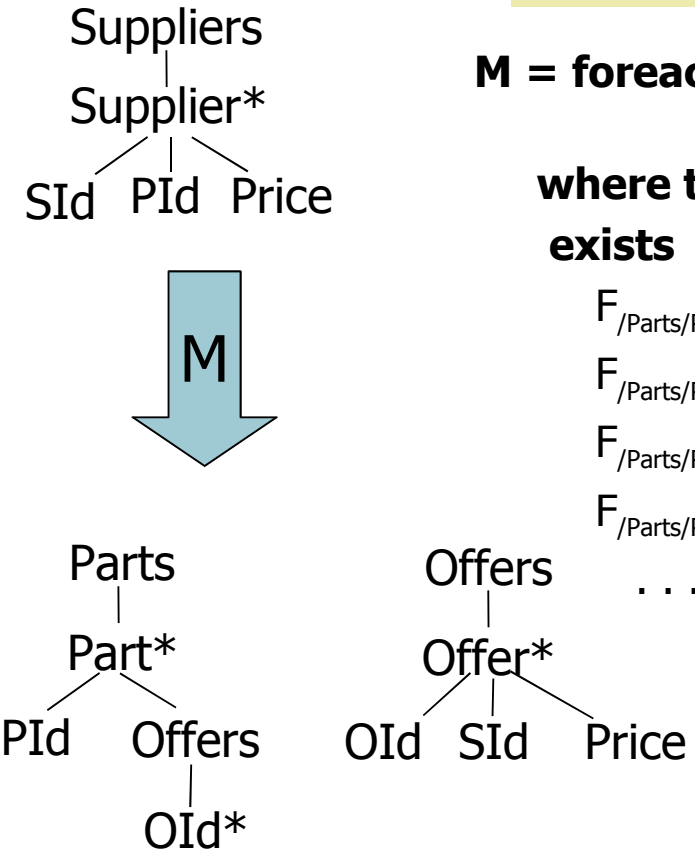
- $\forall \underline{x}\ (\Phi_S(\underline{x}) \Rightarrow \exists \underline{y}\ \Psi_T(\underline{x},\underline{y}))$ – *data generating dependency.*
- $\forall \underline{x}\ (\Phi_S(\underline{x}) \Rightarrow \Psi_T(\underline{x}))$ – *<u>full</u> data generating dependency.*

$M :=$ **foreach** $g, \ldots, g$
where Φ
exists $F_{P/l}(\overline{\$x})$ **in** $F_P(\overline{\$x'})\,[l$ **with** $f(\overline{\$x''})\ ]$



XDMap – a language for mapping specification

$$\forall \underline{x}\ (\Phi_S(\underline{x}) \Rightarrow \Psi_T(\underline{x}))$$



M = foreach $s in /Suppliers/Supplier, $x in $s/SId,
$y in $s/PId, $z in $s/Price

where true

exists

$F_{/Parts/Part}($y) **in** $F_{/Parts}()$/Part

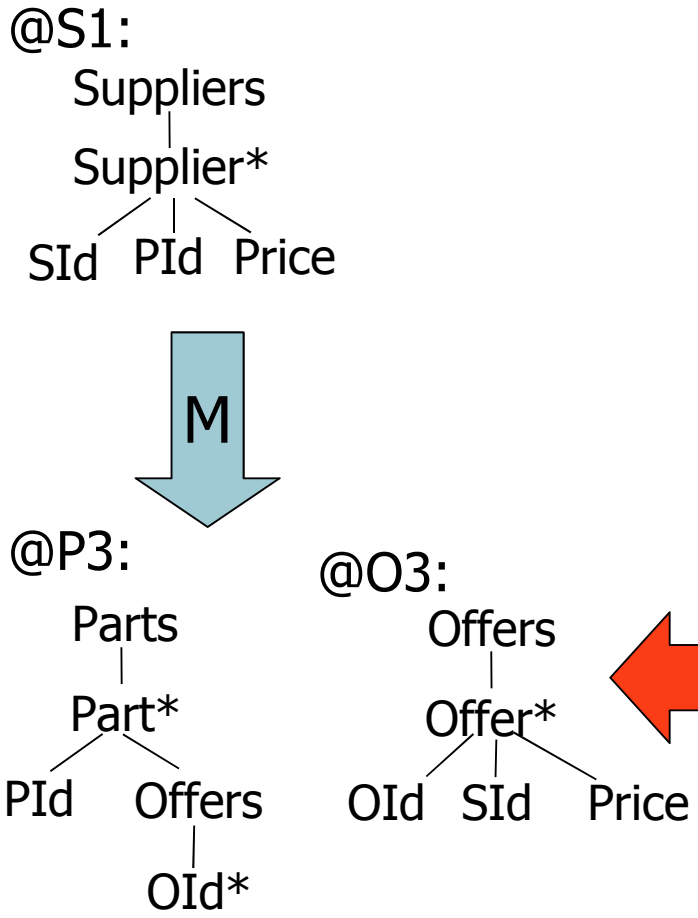
$F_{/Parts/Part/PId}($y) **in** $F_{/Parts/Part}($y)$/PId **with** $y

$F_{/Parts/Part/Offers}($y) **in** $F_{/Parts/Part}($y)$/Offers

$F_{/Parts/Part/Offers/OId}($x, $y) **in** $F_{/Parts/Part/Offers}($y)$/OId **with** f($x, $y)

. . .

Sample target query (subset of XQuery)

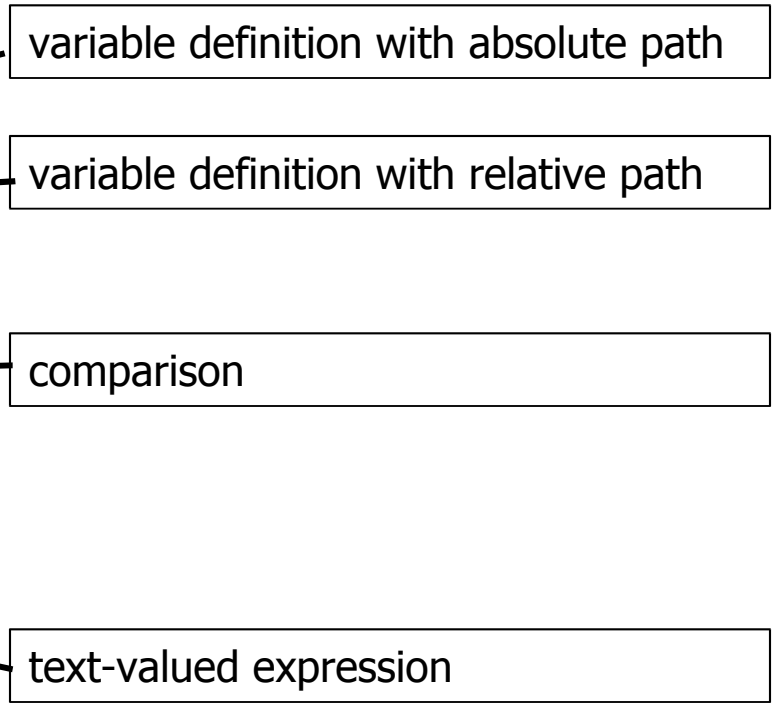


T = (@P3, @O3), **S** = (@S1)

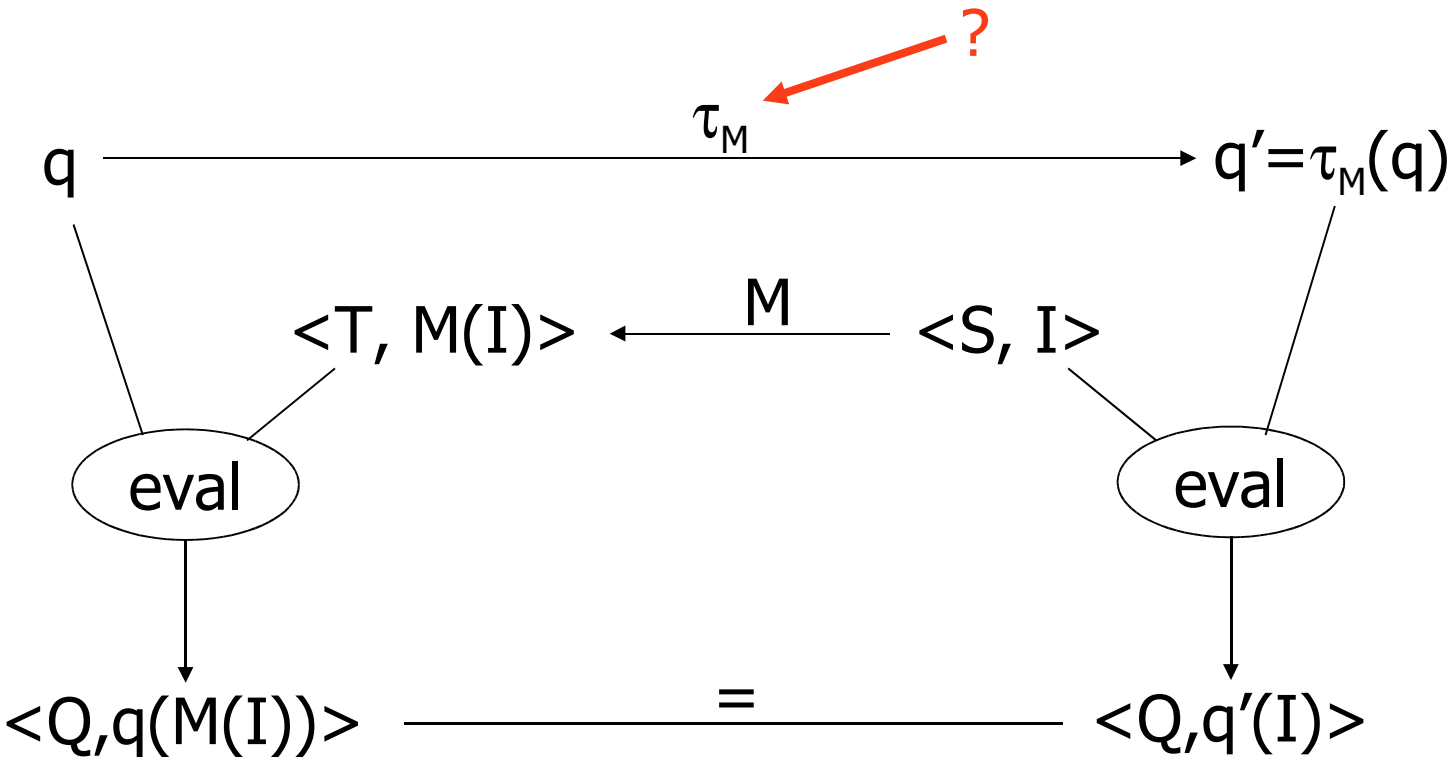
```
q =
<Result>
for   $p in @P3/Parts/Part
      $q in $p/Offers
      $o in @O3/Offers/Offer
where
      $q/OId = $o/OId
return
  <Part>
    <PartId> $p/PId </PartId>
    <Supplier> $o/SId </Supplier>
    <Price> $o/Price </Price>
  </Part>
</Result>
```

4-types of query components

```
q =
<Result>
for   $p : @P3/Parts/Part
      $q : $p/Offers
      $o : @O3/Offers/Offer
where
      $q/OId = $o/OId
return
  <Part>
    <PartId> $p/PId </PartId>
    <Supplier> $o/SId </Supplier>
    <Price> $o/Price </Price>
  </Part>
</Result>
```



Query reformulation



Rewriting rule (R1-1)

```
q =
<Result>
for     $p in @P3/Parts/Part
. . .
```

F_t : $p in @P3/Parts/Part
M^e : $F_{@P3/Parts/Part}$($y)
M^f : $y in @S1/Suppliers/Supplier/PId

ω : [$p → $p']
F_s : $p' in @S1/Suppliers/Supplier/PId

```
q =
<Result>
for     $p' in @S1/Suppliers/Supplier/PId
. . .
```

Rewriting rule (R2-1)

```
q = <Result>
        for   $p in @P3/Parts/Part
              $q in $p/Offers
              . . .
```

F_t : $q in $p/Offers

M^e : $F_{@P3/Parts/Part/Offers}$($s/PId) **in** $F_{@P3/Parts/Part}$($s/PId)/Offers

M^f : $s in @S1/Suppliers/Supplier

ω : [$p → $p']

ω : [$q → $q']

~~F_s : $q' in @S1/Suppliers/Supplier~~

F_w : $p'=$q'/PId

```
q = <Result>
        for $p' in @S1/Suppliers/Supplier/PId
            $q' in @S1/Suppliers/Supplier
            . . .
        where $p'=$q'/PId and ...
```

Rewriting rule (R1-2)

```
q = <Result>
        for   $p in @P3/Parts/Part
              $q in $p/Offers
              $o in @O3/Offers/Offer
        . . .
```

F_t : $o in @O3/Offers/Offer

M^e : $F_{@O3/Offers/Offer}$($s/SId,$s/PId)

~~M^f : $s in @S1/Suppliers/Supplier~~

ω : [$o → $o']

~~F_s : $o' in @S1/Suppliers/Supplier~~

```
q = <Result>
        for   $p' in @S1/Suppliers/Supplier/PId
              $q' in @S1/Suppliers/Supplier
              $o' in @S1/Suppliers/Supplier
        . . .
        where $p'=$q'/PId and ...
```

Process of query reformulation

We will say that a source query q_s is derived from a target query q_t by means of a reformulation τ_M, denoted by

$$(q_t, \varnothing) \underset{\tau_M}{\overset{*}{\Rightarrow}} (q_s, \Omega^*),$$

iff there is a sequence

$$(q^0, \Omega^0), \ldots, (q^N, \Omega^N),$$

such that:

$$(q^0, \Omega^0) = (q_t, \varnothing),$$
$$(q^i, \Omega^i) \underset{\tau_M}{\Rightarrow} (q^{i+1}, \Omega^{i+1})$$
$$(q^N, \Omega^N) \equiv (q_s, \Omega^*),$$

$$, 0 \le i \le N\text{-}1.$$

Source query (reformulated q)

```
q' =
<Result>
for  $p' : @S1/Suppliers/Supplier/PId
     $q' : @S1/Suppliers/Supplier
     $o' : @S1/Suppliers/Supplier
where
     $p' = $q'/PId and $q'/OId = $o'/OId and $q'/SId = $o'/SId
return
  <Part>
    <PartId> $p' </PartId>
    <Supplier> $o'/SId </Supplier>
    <Price> $o'/Price </Price>
  </Part>
</Result>
```

Conclusion

1. E-commerce activities performed at electronic marketplace and using independently created e-commerce applications require data integration.
2. Data integration can be achieved by data transformation or query reformulation. In both cases mappings between schemas are necessary.
3. We propose a language XDMap for specifying mappings. Mappings are used for query reformulation.
4. Rules for query reformulation are proposed.

Thank You