An Approach to Heterogeneous Data Translation based on XML Conversion

Paolo Papotti
Riccardo Torlone

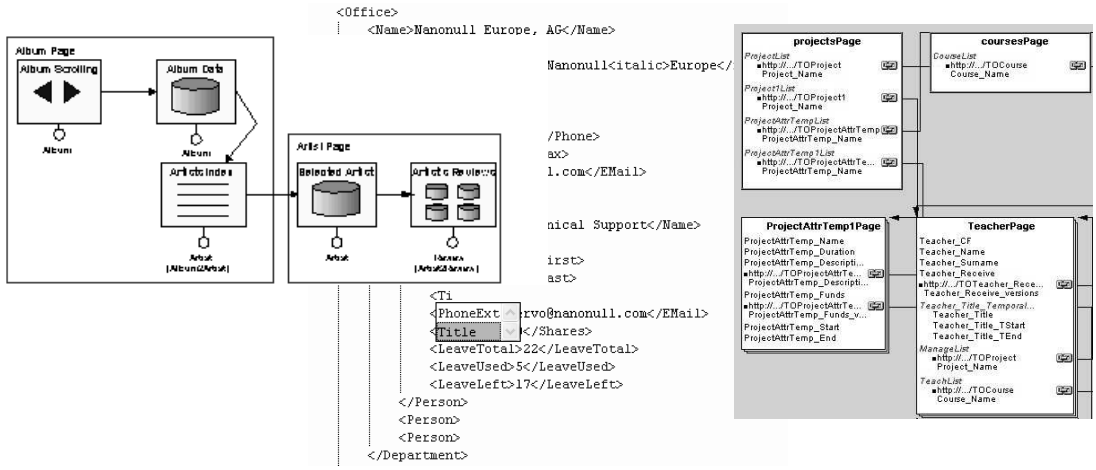
Università Roma Tre

Outline

- Scenario
- Metamodel
- Data translation
- Example
- Future direction

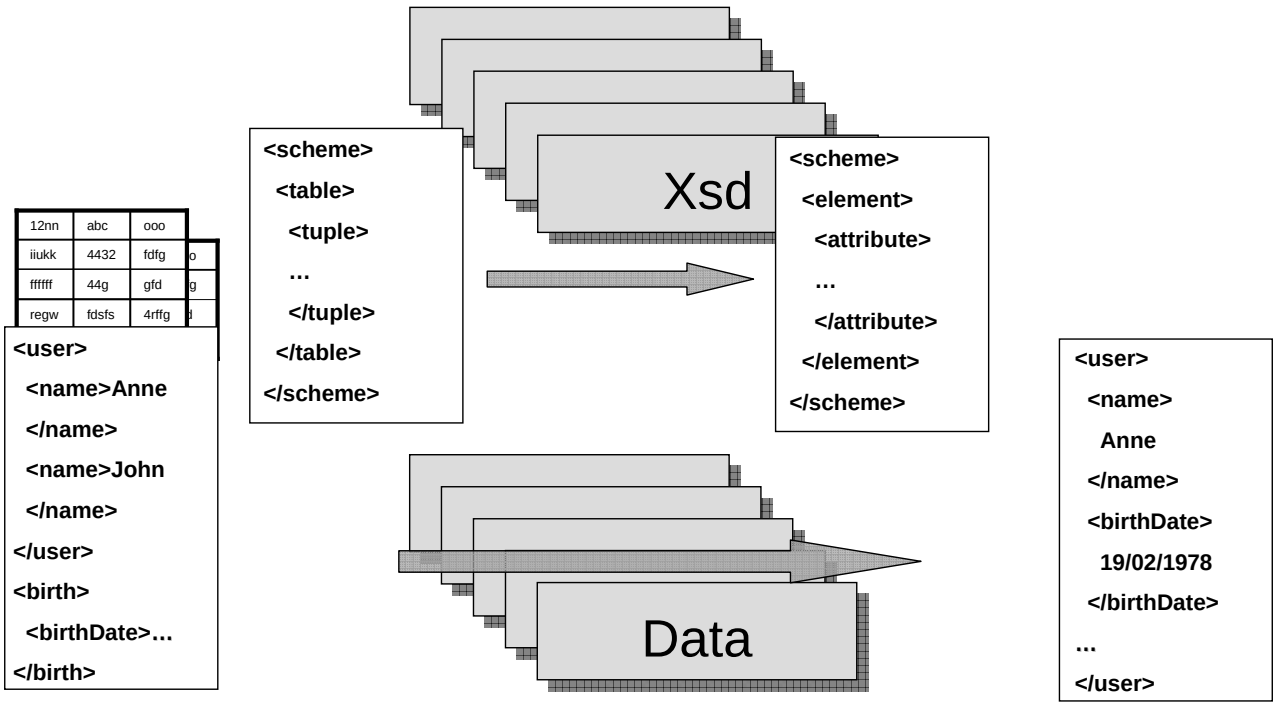
Scenario: the Web Babel

- Many different data models



Many others…

Scheme and instance



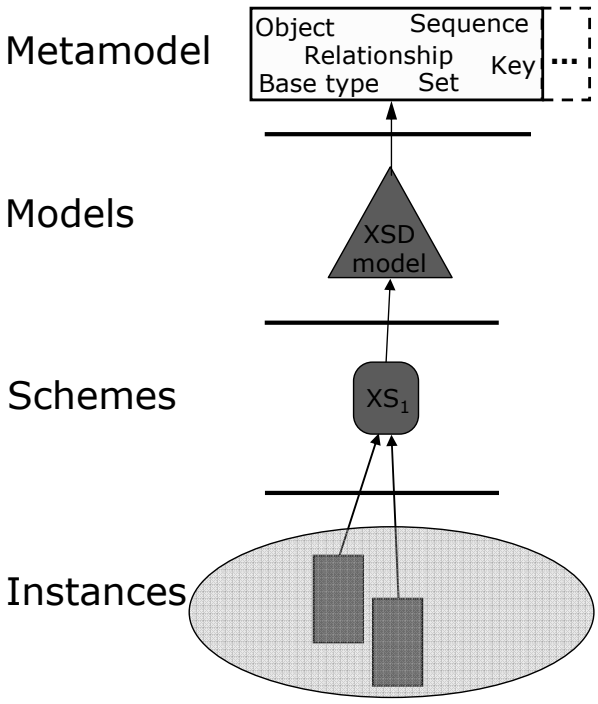
Goals

- Supporting cooperation and data interchange between different organizations with distinct and heterogeneous data sources
- Development of a tool for (semi)automatic translation of schemes and instances from one model to another
  - Models not fixed a priori

Steps

1. the definition of a “meta-formalism” that captures:
   - main primitives adopted by different schema languages for structured data
   - basic constructs used by traditional database models
2. the definition of an effective method for the translation between models, which makes use of the meta-formalism as a level of reference

Metamodel

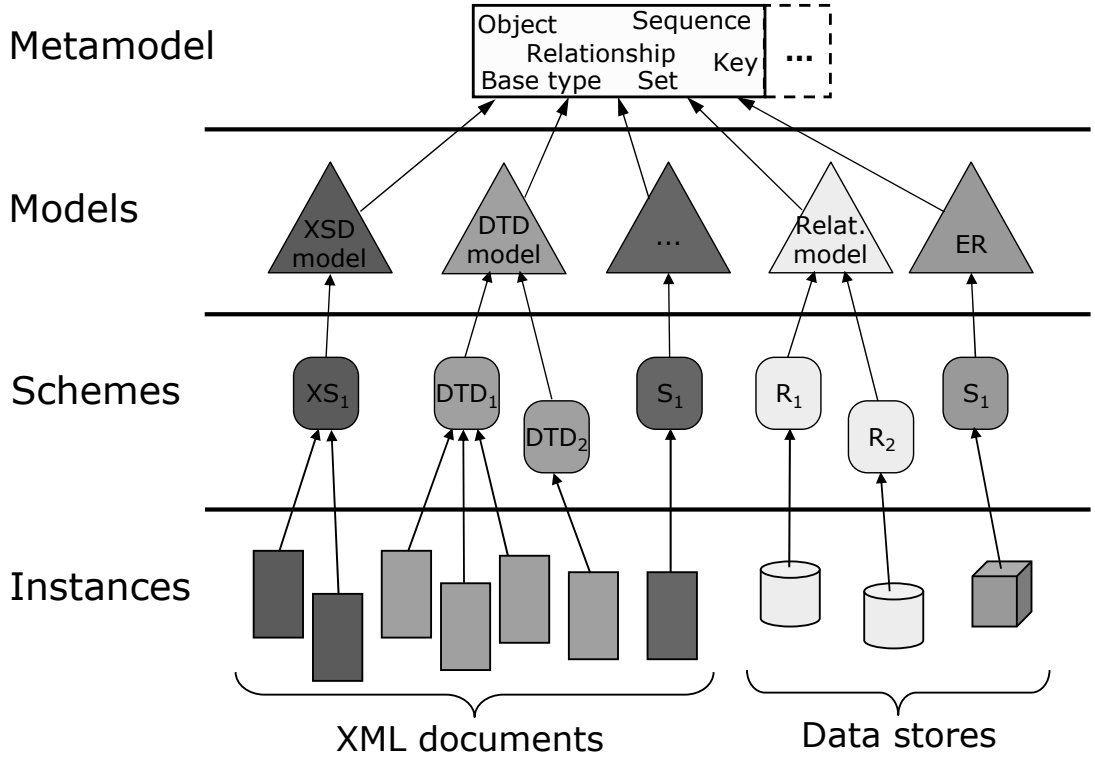


- Metamodel:
  - Set of classes of constructs
- Model:
  - Set of constructs to define schemes
- Scheme:
  - XSD and DTD files
  - Database schemes
- Data:
  - Relational tables
  - XML files
  - Semi structured data

First step: metamodel

- Classification of primitives adopted by the various models
- Definition of a metamodel having a *metaprimitive* for each of these class
- A model is defined by associating its primitives with the metaprimitive in the metamodel (syntax translation)
- Metaprimitives: Base type, User define type, Ordered sequence, Unordered sequence, Choice, Cardinality, Key, Foreign key, ...

The reference scenario



Second step: translation

- Translations operate on individual (or combinations of) primitives
- Complex translation can be obtained as composition of elementary steps
- A *supermodel* is used as “pivot element” in the translations
- Work supported by a library of *procedures*, set of elementary transformations implementing translations between primitives

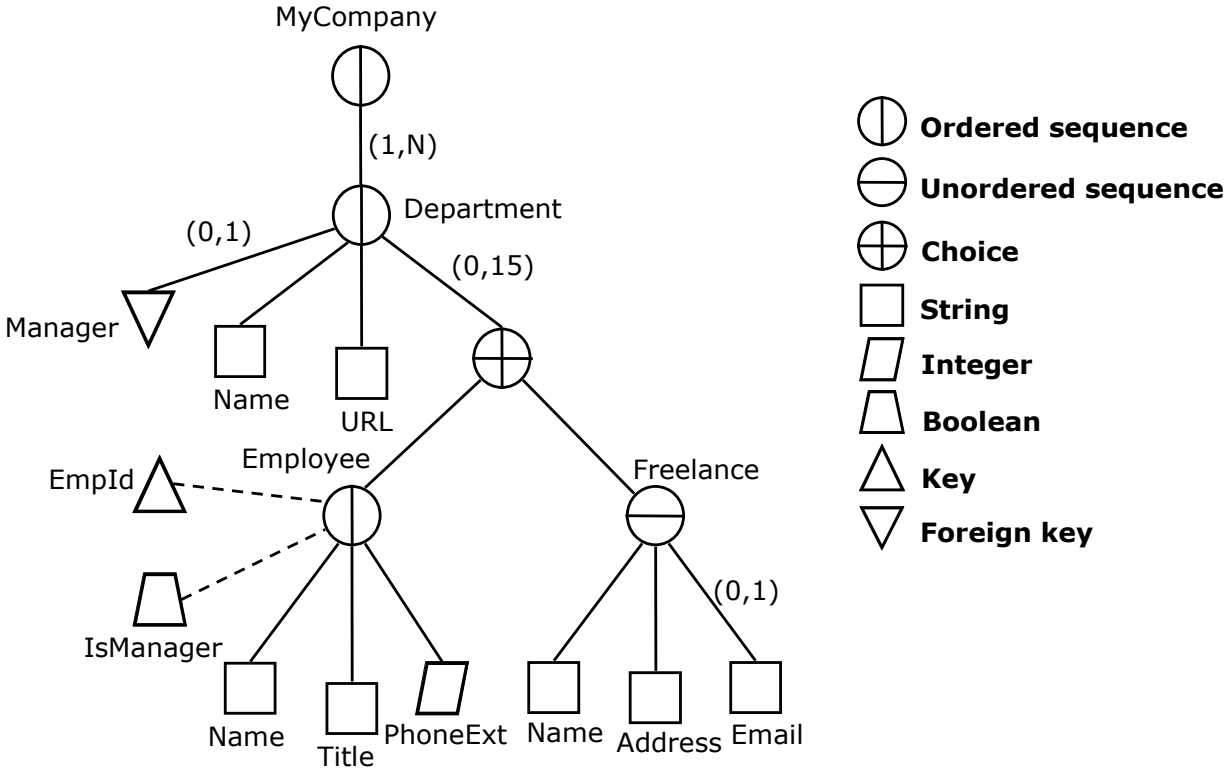
Data translation

- Input: a scheme $\mathbf{S_S}$ of a model $\mathbf{M_S}$, a library of procedures **L**, and the target model $\mathbf{M_T}$
- Output: a set of procedures **t**, a scheme $\mathbf{S_T}$ for $\mathbf{M_T}$

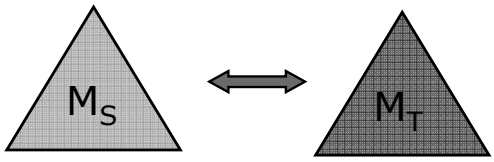
- Used within the supermodel
  1. Syntactical translation of the scheme into the supermodel
  2. Model matching
  3. Identification of metaprimitives to be transformed
  4. Selection of **procedures** from the library
  5. **Procedures** application

- Implementation in XQuery and XSLT

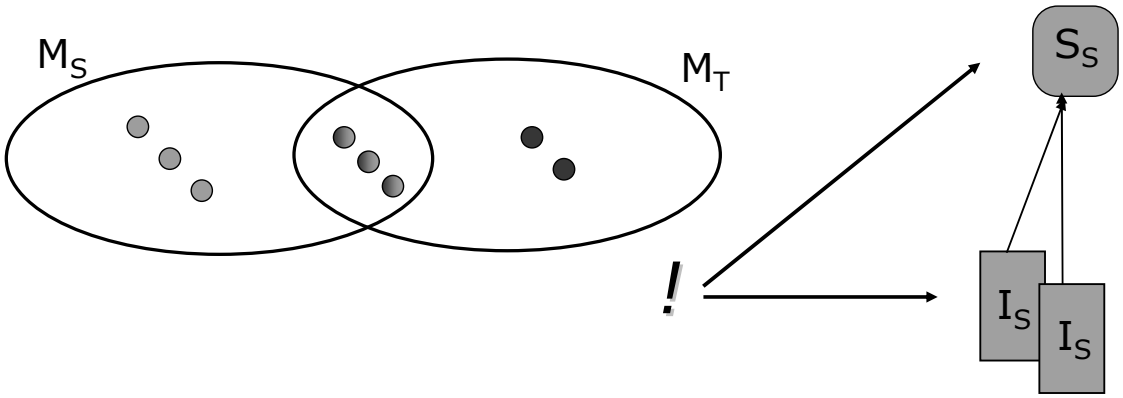
Supermodel



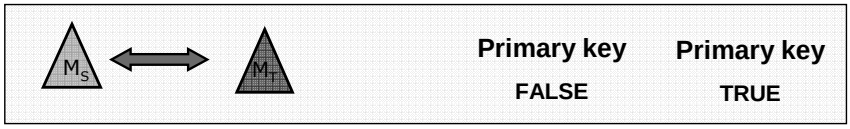
Model matching



Primitive (model)		**Metaprimitive** (metamodel)
All	→	Unordered sequence
Table	→	Relation of lexicals

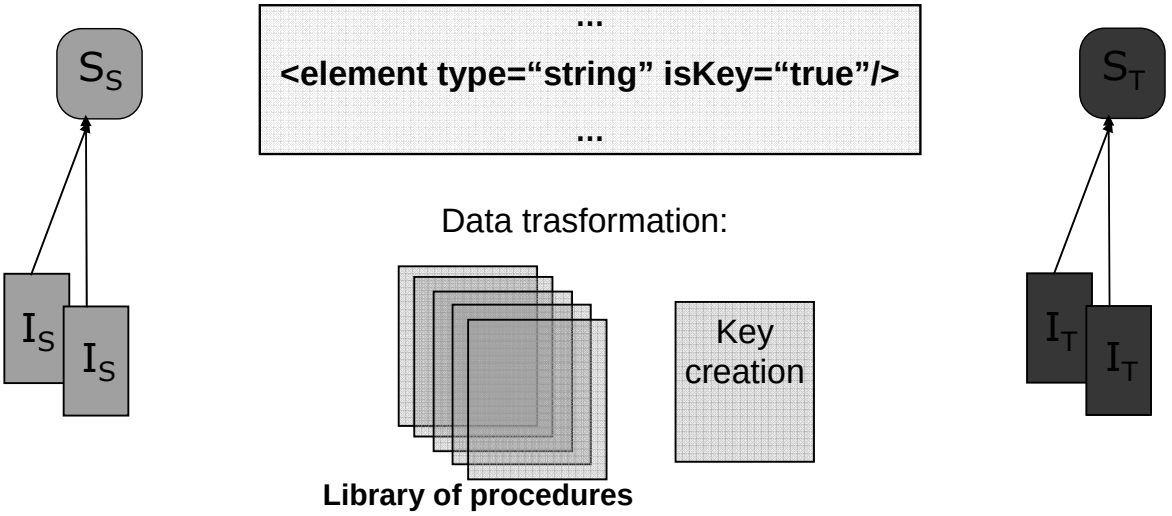


Procedures selection



Scheme modification:

```
...
<element type="string" isKey="true"/>
...
```



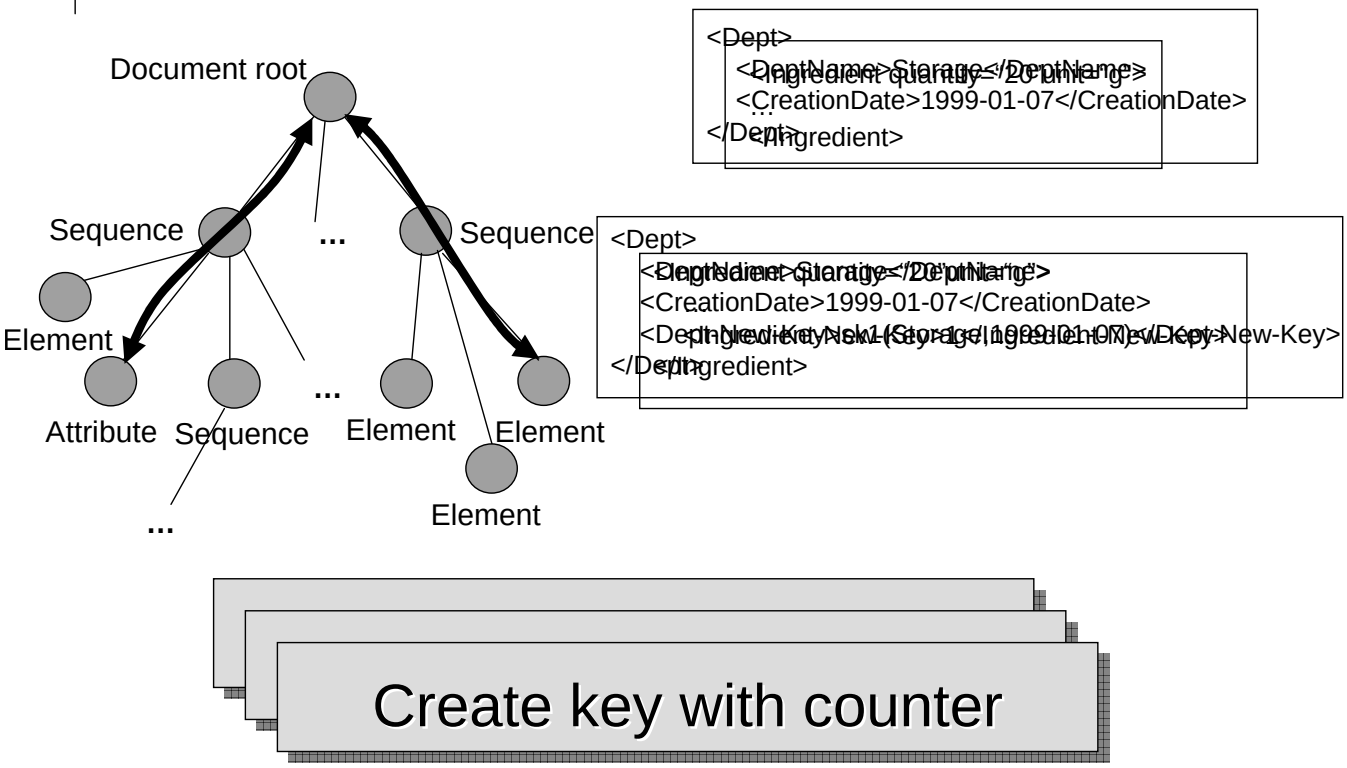
Problems and some solution

- Different translation: user choice
  - Elimination of hierarchies
- Loss of metaprimitives: residual
  - Namespaces
- Semantic loss: residual
  - n-ary cardinality
- Indirect translation:
  - $A \not\rightarrow C$, but $A \rightarrow B$ and $B \rightarrow C$
    - risk of infinite loops

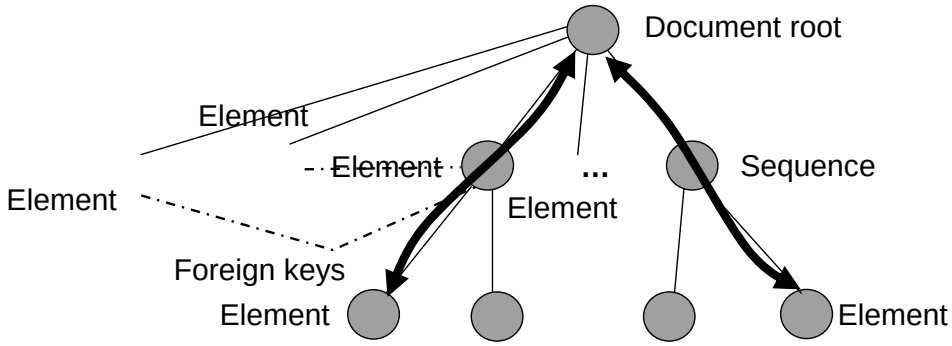
Library of Procedures

- Nesting/unnesting of complex and atomic elements
- Key/foreign key creation
- Management of ordered/unordered sequence
- Management of cardinality (restriction, extension)
- Addition/removal of namespace
- Management of generalization hierarchies
- …

Key creation



Unnesting (scheme)



creation of new references.

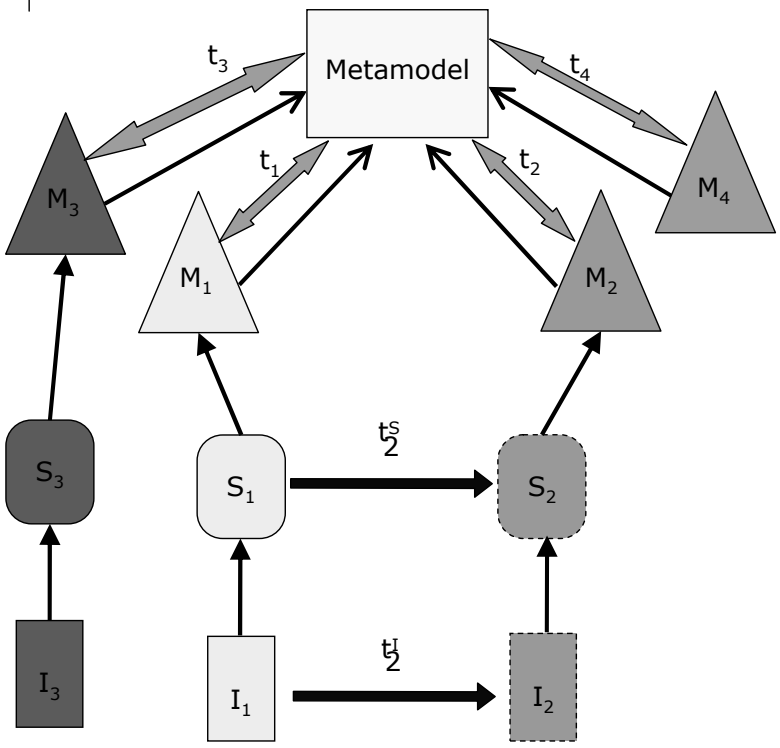
Unnesting (data)

```
<Dept>
   <DeptName>Storage</DeptName>
   <CreationDate>1999-01-07</CreationDate>
   <Emps>
      <Emp>
         <EmpID>37</EmpID>
         <EmpName>Paul</EmpName>
      </Emp>
      <Emp>
         <EmpID>48</EmpID>
         <EmpName>Andrew</EmpName>
      </Emp>
   </Emps>
</Dept>
```

```
<Dept>
   <DeptName>Storage</DeptName>
   <CreationDate>1999-01-07</CreationDate>
   <Dept-New-Key>sk1(Storage,1999-01-07)</Dept-New-Key>
</Dept>


<Emp>
   <Dept-New-Key>sk1(Storage,1999-01-07)</Dept-New-Key>
   <Emp-New-Key>sk2(37,Paul)</Emp-New-Key>
   <EmpID>37</EmpID>
   <EmpName>Paul</EmpName>
</Emp>
<Emp>
   <Dept-New-Key>sk1(Storage,1999-01-07)</Dept-New-Key>
   <Emp-New-Key>sk2(48, Andrew)</Emp-New-Key>
   <EmpID>48</EmpID>
   <EmpName>Andrew</EmpName>
</Emp>
```


New model



- Models not fixed at priori
- New model:
  - Metaprimitive selection
  - Syntax translation (XSLT)
  - New procedures (if necessary)

Instance of

Sintactical Translation

Subsumes

Translation

Future directions

- Theoretical investigation
  - Regular grammar tree
  - Graph transformation
- Integration of tools
  - Graphical mapping for syntactical translation (XSLT)
  - Procedure aided design (XQuery)
  - Model creation wizard from scheme (or instance)
- Optimization of translation
  - Removal of redundancies and useless procedures

The end

Thank you
for your attention!