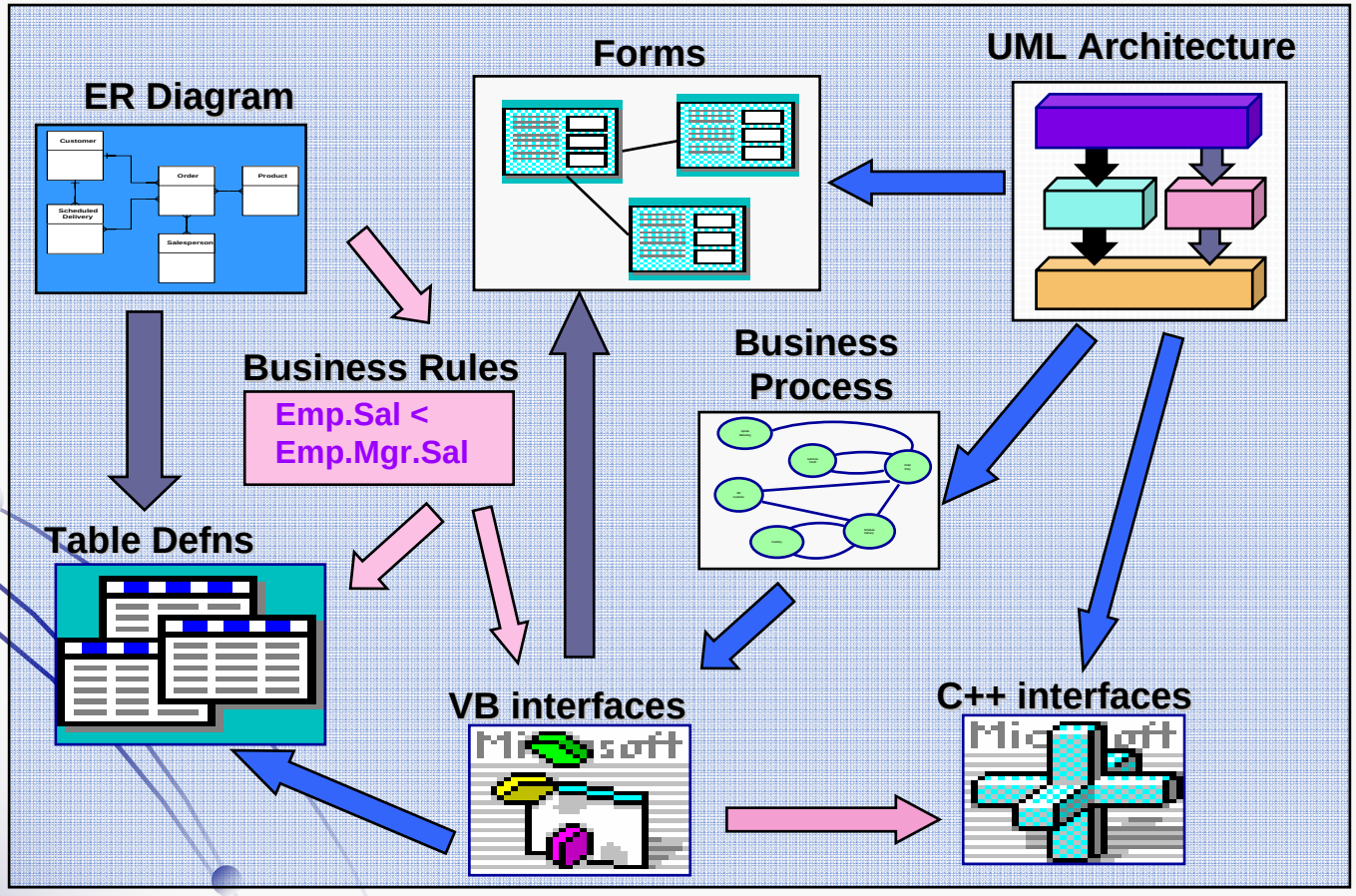
Seminari di sistemi informatici 2005 - 2006

Integrazione e Traduzione di Sorgenti Informative Eterogenee

Riccardo Torlone
torlone@dia.uniroma3.it
Paolo Papotti
papotti@dia.uniroma3.it

The translation problem

- Today, information needs to be shared and exchanged continuously but different organizations collect, store, and process data differently

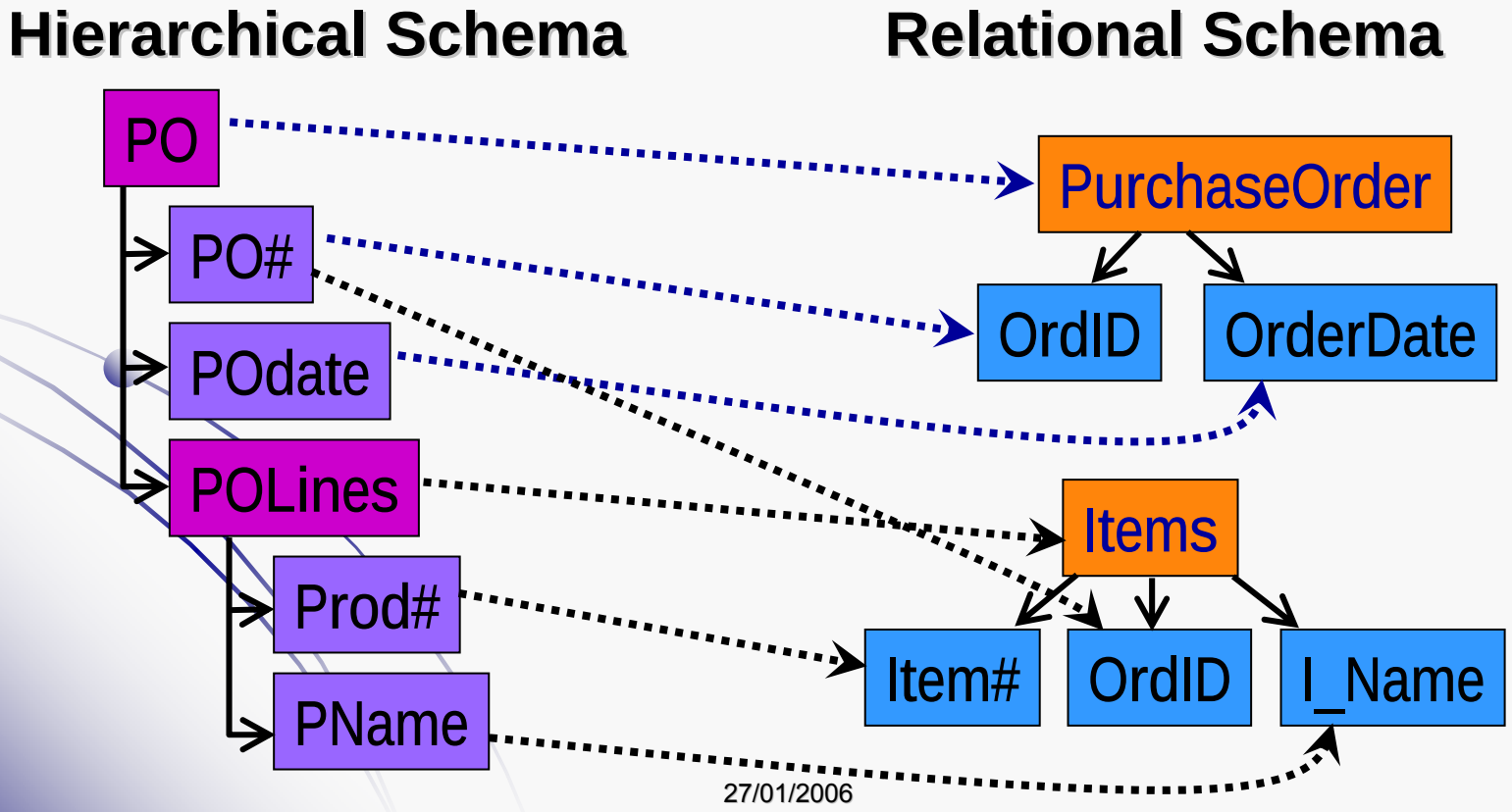


This Problem is Pervasive

- Data translation
- Schema evolution & data migration
- XML message translation for e-commerce
- Integrate custom apps with commercial apps
- Data warehouse loading (clean & transform)
- Design tool support (DB, UML, …)
- Database-driven portal generation
- OO or XML wrapper generation for SQL DB
- …

Schema mappings

- The problem involves meta-data information, in particular, mappings between schemes



Solutions to the Problem

- Solutions strongly resemble each other, but
  - usually are problem-specific
  - usually are language-specific
    SQL, ODMG, UML, XML, RDF, ….
  - usually involve a lot of object-at-a-time programming

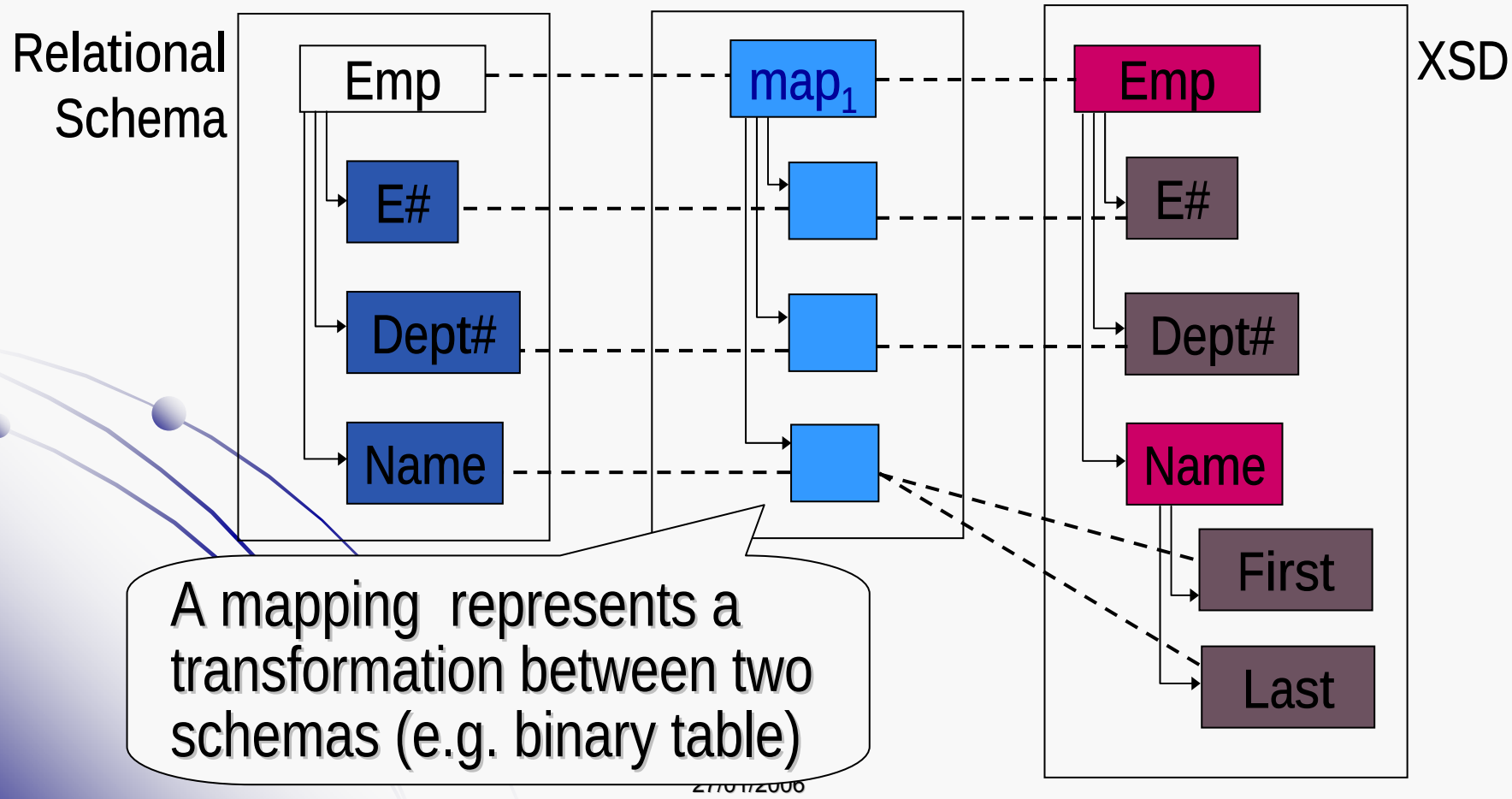
- Goals
  - Generic solutions
  - “Set”-at-a-time programming

Model Management

- A generic approach to this problem
- Model Mgmt operators manipulate *schemas* and *mappings* as bulk objects
  - Their representation is generic
  - Operators:
    - Match, Merge, Diff, Compose, ModelGen, …
- Avoids problem-specific and language-specific solutions

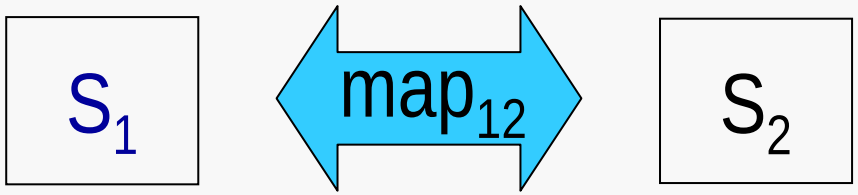
Models and Mappings

- A schema is a rooted directed graph, which represents a complex information structure.



Classifying Meta Data Problems

- Scheme mapping

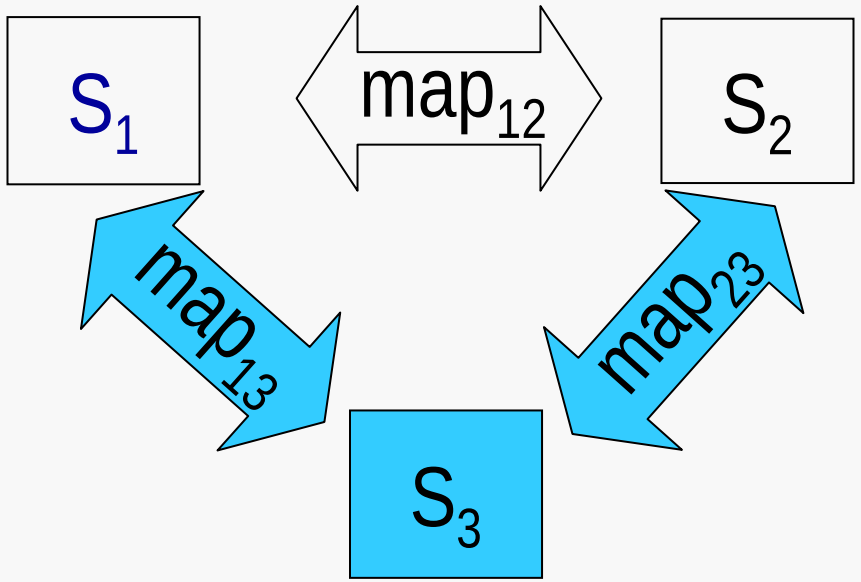


  - Data translation
  - XML message translation for e-commerce
  - Integrate custom apps with commercial apps
  - Data warehouse loading (clean & transform)

- Solution is the **match** “operator”

Categorizing M D Problems (2)

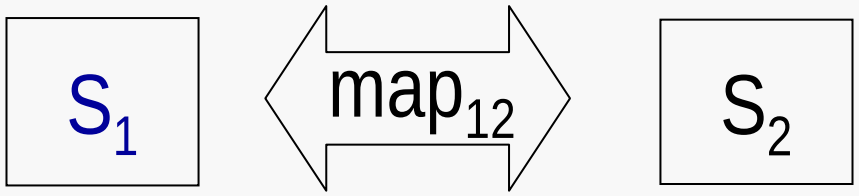
- Scheme integration



  - View integration
  - Data integration
- Solution is the Merge operator

Categorizing M D Problems (3)

- Scheme and mapping generation

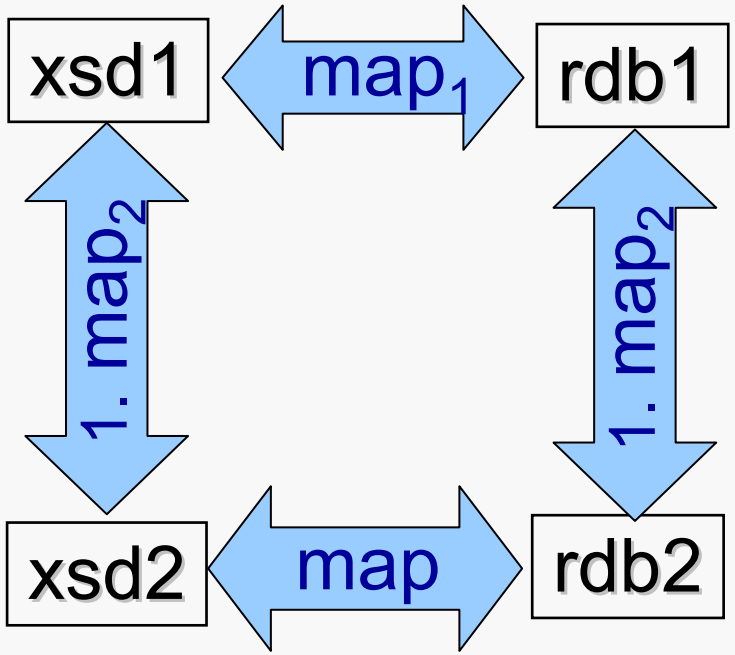


  - Design tools (ER → SQL)
  - Wrapper generation (SQL → OO or XML)

- Solution is the **ModelGen** operator
  - $<S_2, map_{12}>$ = ModelGen(S_1, $model_2$)

E.g. Change Propagation

- Given
  - **map_1 between xsd1 and SQL schema rdb1**
  - **xsd2, a modified version of xsd1**
- Produce
  - **rdb2 to store instances of xsd2**
  - **a mapping between xsd2 and rdb2**



Model Mgmt Algebra

- map = Match (S1, S2)
- <S3, map13, map23> = Merge (S1, S2, map)
- map3 = Compose(map1, map2)
- <S2, map12> = Diff(S1, map)
- <S2, map12> = ModelGen(S1, model2)
- S2 = Copy(S1)
- Apply, Insert, Delete, . . .

Chameleon

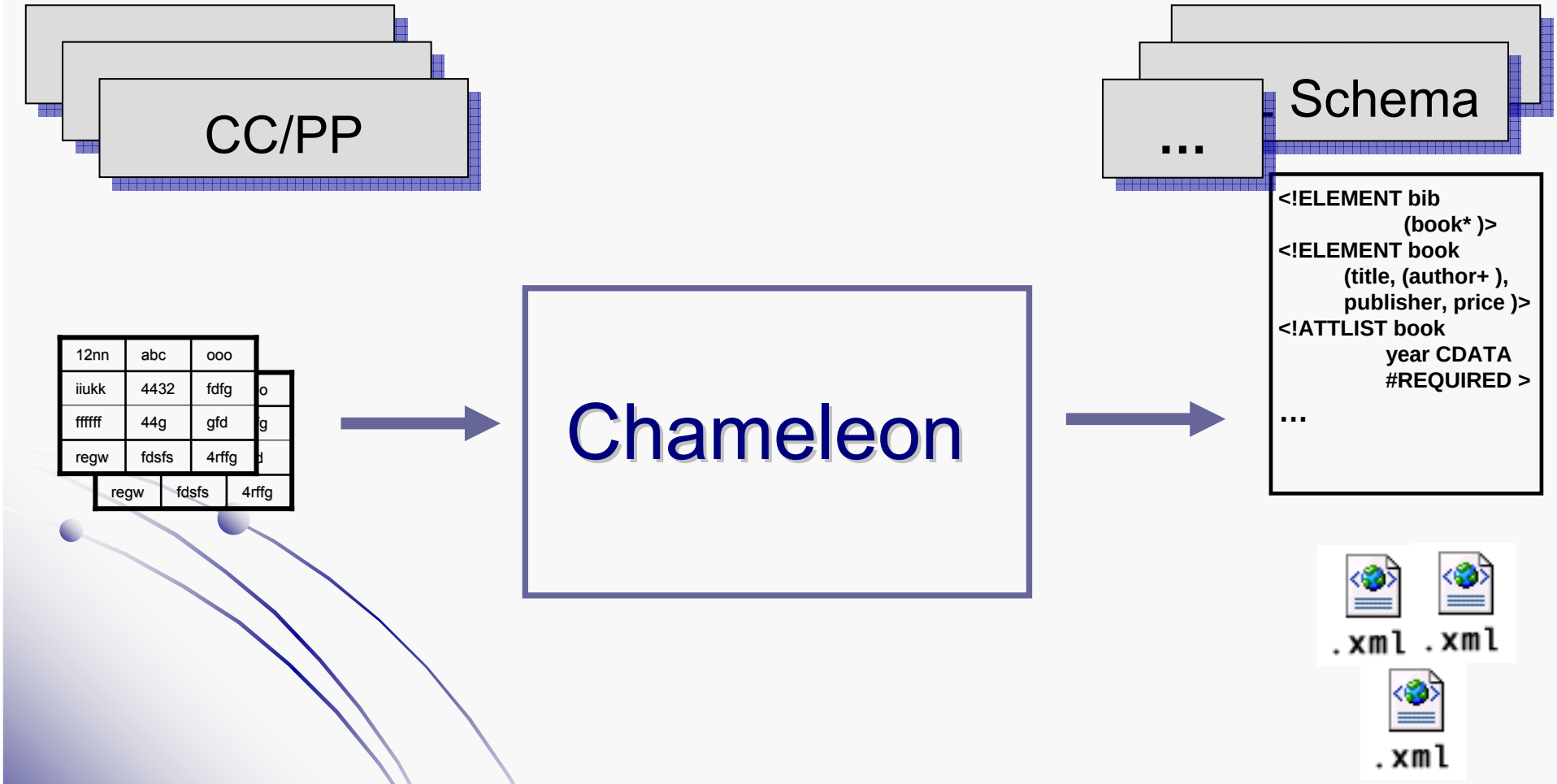
An Extensible and Customizable Tool for Web Data Translation

27/01/2006

Goals

- Supporting cooperation and data interchange between different organizations with distinct and heterogeneous data sources
- Development of a tool for the automatic translation of schemes and instances between models
  - Models are not fixed a priori

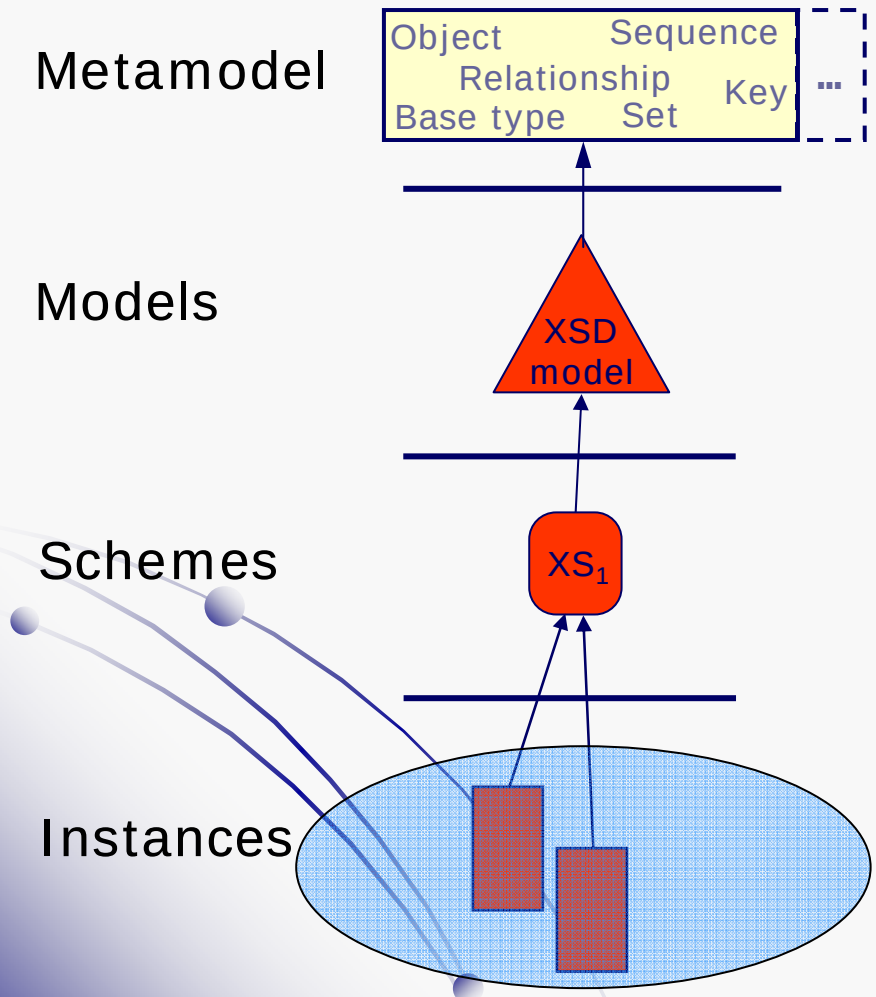
Scheme and instance



Approccio

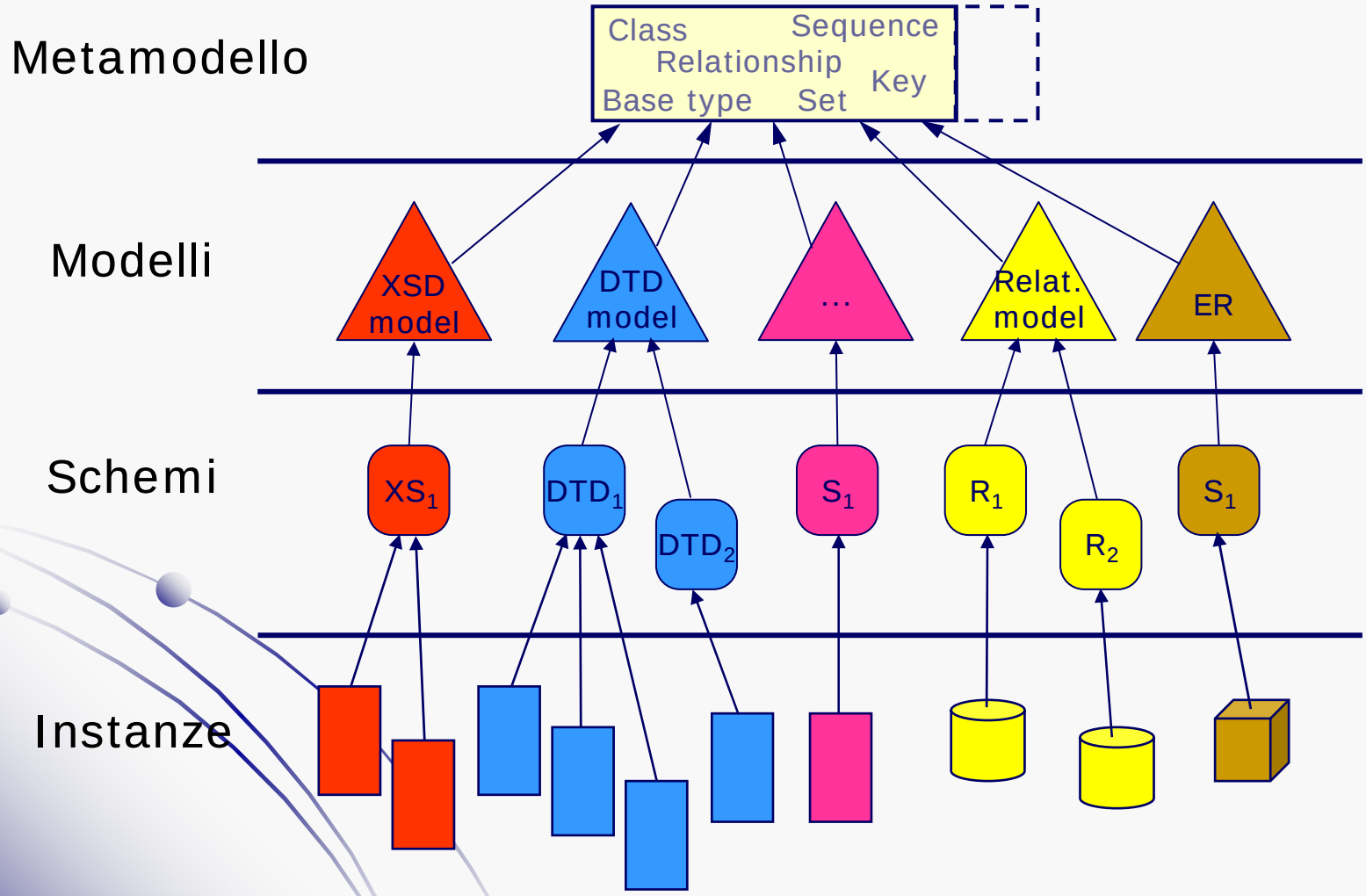
- **Gestione dei modelli**
  - **Chameleon** è basato su un *metamodello* composto da un insieme di *metaprimitive*
  - Una metaprimitiva corrisponde a una classe di costrutti base per i dati: elemento, attributo, relazione, relationship, tipo base, sequenza, … (Hull&King, 1987)
  - Un modello viene definito specificando le metaprimitive che utilizza per rappresentare i dati e le loro caratteristiche (quando sono ammesse, con che limiti, con che sintassi, ...)

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

Scenario riferimento



Definition of a metamodel

- Classification of primitives adopted by the various models into classes (*metaprimitive)*
- A model is defined by associating its primitives with the metaprimitive in the metamodel (syntax translation)
- The supermodel is the “most general model”
- Metaprimitives: Abstract Object, Concrete Object, Base type, User define type, Ordered sequence, Unordered sequence, Choice, Cardinality, Key, Foreign key, ...
- XML-based:
  - models and schemes represented in XML

Motivations

- Two positive aspects:

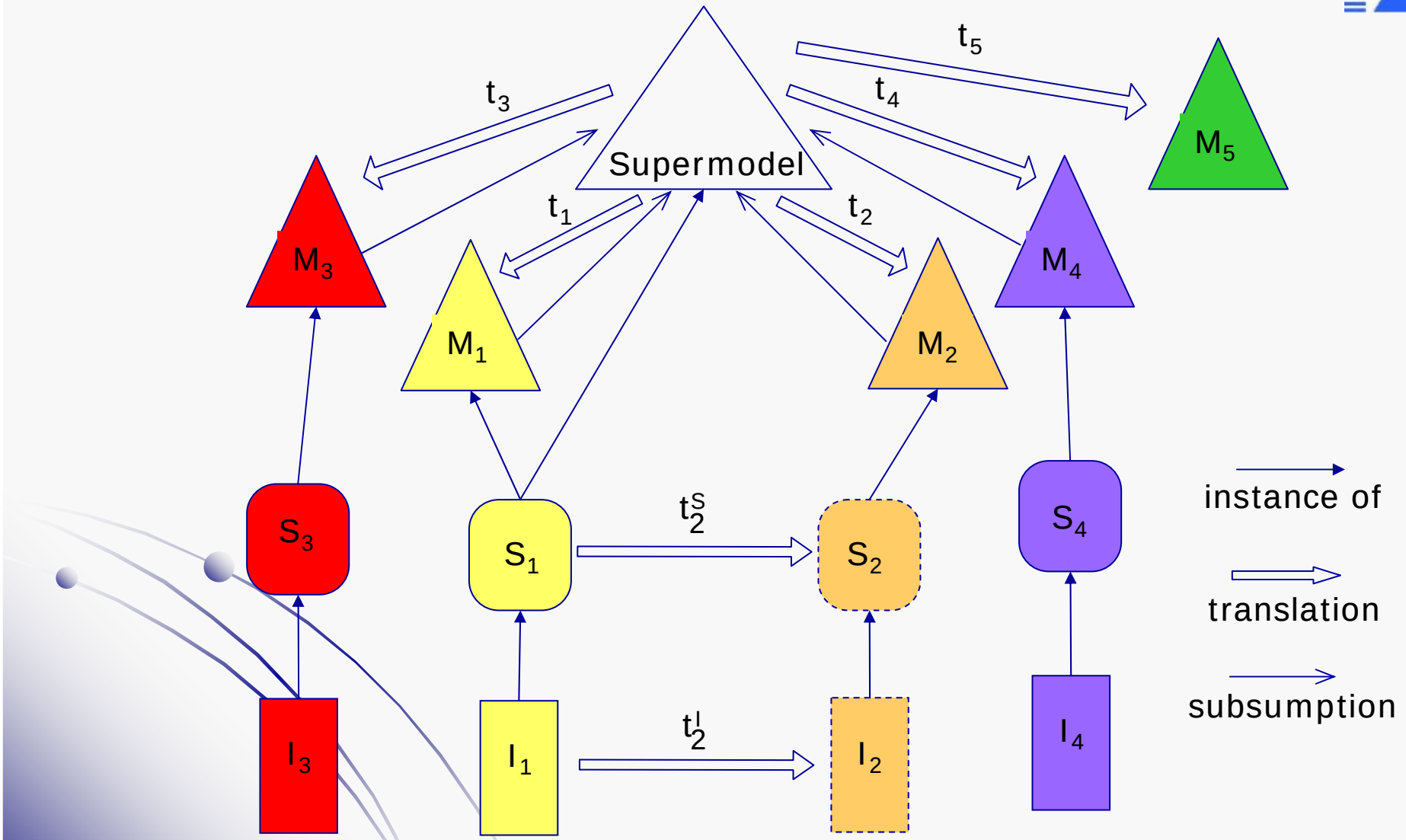
1. Representation of schemes and models with common constructs
   - Add easily new models and constructs
2. Reuse of translations between constructs
   - Translate between models with shared procedures

Approach to data translation

- Library of *basic procedures:* set of transformations implementing translations between individual (or combinations of) metaprimitives
- Complex translation can be obtained as composition of elementary steps
- XML Based: XSLT and XQuery

- **Goal: Automatic** generation of a **sequence** of procedures to translate complex schemes and instances

The translation technique



Esempio traduzione

Schema sorgente (XML Schema)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="Order" type="OrderType"/>
 <xsd:complexType name="OrderType">
  <xsd:sequence>
   <xsd:element name="destination" type="USAddress"/>
   <xsd:element name="items"  type="Items"/>
  </xsd:sequence>
  <xsd:attribute name="orderDate" type="xsd:date"/>
 </xsd:complexType>
 <xsd:complexType name="USAddress">
  <xsd:all>
   <xsd:element name="street" type="xsd:string"/>
   <xsd:element name="city"   type="xsd:string"/>
   <xsd:element name="zip"    type="xsd:decimal"/>
  </xsd:all>
  <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
 </xsd:complexType>
 <xsd:complexType name="Items">
  <xsd:sequence>
   <xsd:element name="item" minOccurs="0" maxOccurs="10">
    <xsd:complexType>
     <xsd:sequence>
      <xsd:element name="productName" type="xsd:string" />
      <xsd:element name="quantity" type="xsd:integer" />
      <xsd:element name="USPrice"  type="xsd:decimal"/>
     </xsd:sequence>
    </xsd:complexType>
   </xsd:element>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

Sorgente nel supermodello

```
<META source="xsd">
 <element name="Order" type="OrderType">
  <sequence cardinality="1:1">
   <element name="destination" type="USAddress" cardinality="1:1">
    <unorderedSequence cardinality="1:1" >
     <element name="street" type="string" cardinality="1:1" />
     <element name="city" type="string" cardinality="1:1" />
     <element name="zip" type="decimal" cardinality="1:1" />
    </unorderedSequence>
    <attribute name="country" type="string" cardinality="0:1">
     <fixed>US</fixed>
    </attribute>
   </element>
   <element name="items" type="Items" cardinality="1:1">
    <sequence cardinality="1:1">
     <element name="item" cardinality="0:10">
      <sequence cardinality="1:1">
       <element name="productName" type="string" cardinality="1:1" />
       <element name="quantity" type="integer" cardinality="1:1" />
       <element name="USPrice" type="decimal" cardinality="1:1" />
      </sequence>
     </element>
    </sequence>
   </element>
  </sequence>
  <attribute name="orderDate" type="date" cardinality="0:1" />
 </element>
</META>
```

Trasformazione delle metaprimitive

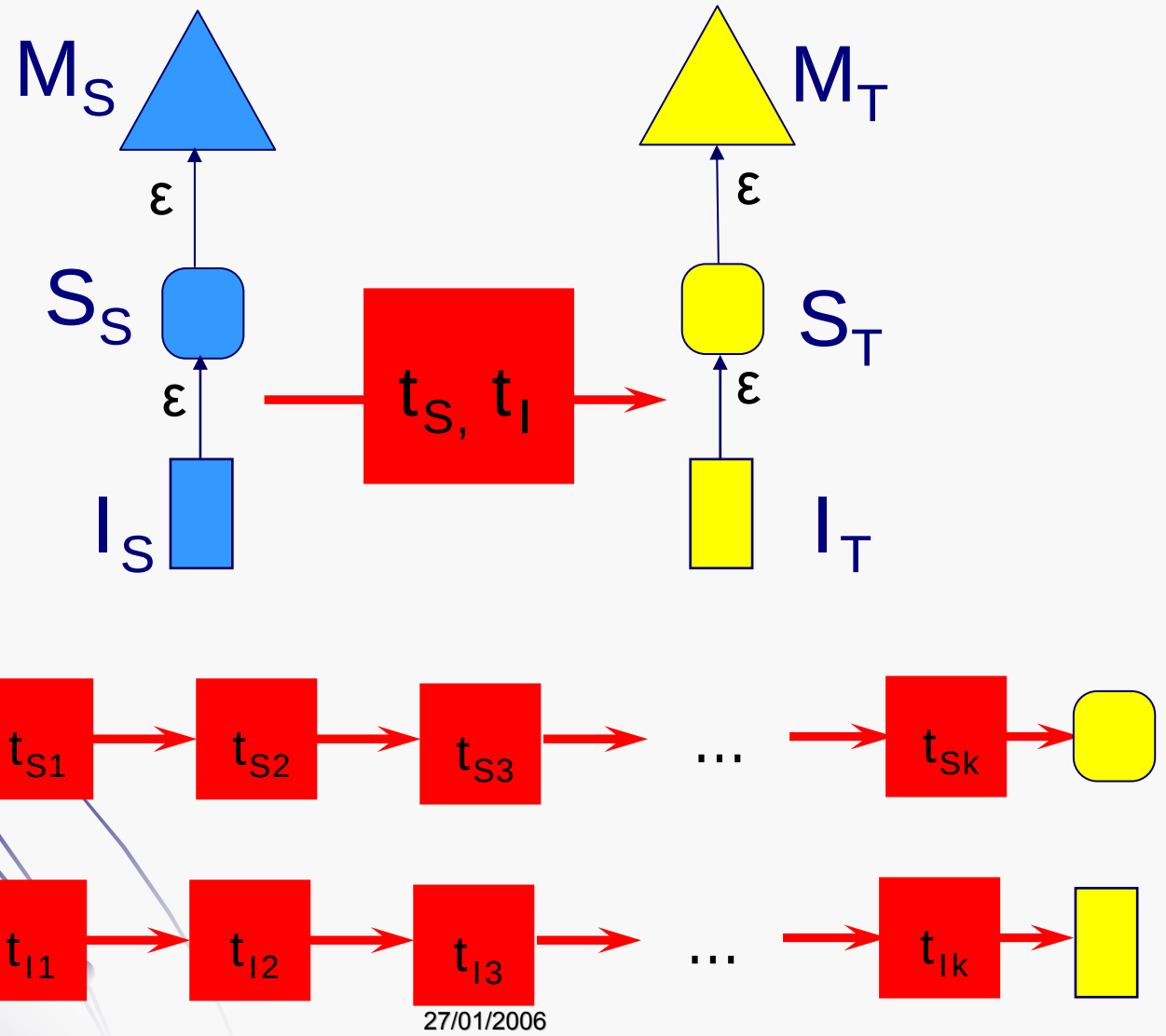
```
<META source="xsd" target="dtd">
 <element name="Order" root="true">
  <sequence cardinality="1:1">
   <element name="destination" cardinality="1:1">
    <sequence cardinality="0:N">
     <element name="street" type="string" cardinality="1:1" />
     <element name="city" type="string" cardinality="1:1" />
     <element name="zip" type="string" cardinality="1:1" />
    </sequence>
    <attribute name="country" type="string" cardinality="0:1" >
     <fixed>US</fixed>
    </attribute>
   </element>
   <element name="items" cardinality="1:1">
    <sequence cardinality="1:1">
     <element name="item" cardinality="0:N">
      <sequence cardinality="1:1">
       <element name="productName" type="string"  cardinality="1:1" />
       <element name="quantity" type="string" cardinality="1:1" />
       <element name="USPrice" type="string" cardinality="1:1" />
      </sequence>
     </element>
    </sequence>
   </element>
  </sequence>
  <attribute name="orderDate" type="string" cardinality="0:1" />
 </element>
</META>
```

Destinazione nel supermodello

Schema destinazione (DTD)

```
<!DOCTYPE Order[
  <!ELEMENT Order (destination,items)>
  <!ELEMENT destination (street,city,zip)>
  <!ELEMENT street (#PCDATA)>
  <!ELEMENT city (#PCDATA)>
  <!ELEMENT zip (#PCDATA)>
  <!ELEMENT items (item*)>
  <!ELEMENT item (productName,quantity,USPrice)>
  <!ELEMENT productName (#PCDATA)>
  <!ELEMENT quantity (#PCDATA)>
  <!ELEMENT USPrice (#PCDATA)>
  <!ATTLIST Order orderDate CDATA #IMPLIED>
  <!ATTLIST destination country CDATA #FIXED "US">
  ]>
```

Traduzione dei dati



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/unions
- Management of built in/extended types
- …

Model 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 scheme $\mathbf{S_T}$ for $\mathbf{M_T}$, a set of procedures **t**, a residual **r**
  - For each instance **I of S_S**, t(I) is an instance of **S_T**

- Algorithm
  1. Serialization (if needed)
  2. **Translation** of the scheme into the supermodel
  3. Model matching: identification of metaprimitives to be transformed
  4. Selection of **procedures** from the library
  5. Application of **procedures**

Esempio

- Portare dati su dipartimenti e impiegati da un insieme di documenti XML a un database relazionale
- Conosciamo lo schema di partenza (XMLSchema) e le istanze (documenti XML)

Schema sorgente

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name = "Dept" >
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="DeptName" type="xsd:string"/>
        <xsd:element name="CreationDate" type="xsd:date"/>
        <xsd:element name = "Emps" >
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "Emp"maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="EmpID" type="xsd:integer"/>
                    <xsd:element name="EmpName" type="xsd:string"/>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Istanza sorgente

```
<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>
```

Supermodello 1

```
<META source="XSD">
  <element name="Dept">
    <sequence occurs="1:1">
      <element name="DeptName" type="string" occurs="1:1"/>
      <element name="creationDate" type="date" occurs="1:1"/>
      <element name="Emps" occurs="1:1">
        <sequence occurs="1:1">
          <element name="Emp" occurs="1:N">
            <sequence occurs="1:1">
              <element name="EID" type="integer" occurs="1:1"/>
              <element name="EName" type="string" occurs="1:1"/>
            </sequence>
          </element
        </sequence>
      </element>
    </sequence>
  </element>
</META>
```

Supermodello 2

```
<META source="Relational">
  <element name="Depts" occurs="0:N">
     <attribute name="DeptName" occurs="1:1" type="string"/>
     <attribute name="CreationDate" occurs="1:1" type="string"/>
     <attribute name="Dept-New-Key" type="key" occurs="1:1"/>
  </element>
  <element name="Emps" occurs="0:N">
     <attribute name="Depts-Emps-Key" type="string">
        <keyref name="Depts-Emps-Key-Est" refer="Dept-New-Key"/>
     </attribute>
     <attribute name="Emps-New-Key" type="key" occurs="1:1"/>
  </element>
  <element name="Emp" occurs="0:N">
     <attribute name="Emps-Emp-Key" type="string">
        <keyref name="Emps-Emp-Key-Est" refer="Emps-New-Key"/>
     </attribute>
     <attribute name="EmpID" occurs="1:1" type="string"/>
     <attribute name="EmpName" occurs="1:1" type="string"/>
  </element>
</META>
```

Schema target

```
<database>
 <table name="Dept">
  <tuple>
   <field name="DeptName" occurs="1:1" type="string"/>
   <field name="CreationDate" occurs="1:1" type="string"/>
   <field name="Dept-New-Key" type="key" occurs="1:1"/>
  </tuple>
 </table>
 <table name="Emps">
  <tuple>
   <field name="Depts-Emps-Key" type="string">
      <keyref name="Depts-Emps-Key-Est" refer="Dept-New-Key"/>
   </field>
   <field name="Emps-New-Key" type="key" occurs="1:1"
  </tuple>
 </table>
 <table name="Emp">
  <tuple>
   <field name="Emps-Emp-Key" type="string">
      <keyref name="Emps-Emp-Key-Est" refer="Emps-New-Key"/>
   </field>
   <field name="Emp-New-Key" type="key" occurs="1:1" />
   <field name="EmpID" occurs="1:1" type="string" />
   <field name="EmpName" occurs="1:1" type="string" />
  </tuple>
 </table>
</database>
```

Istanza target

```
<Dept>
 <tuple>
  <DeptName>Storage</DeptName>
  <CreationDate>1999-01-07</CreationDate>
  <Dept-New-Key>sk1(Storage,1999-01-07)</Dept-New-Key>
 </tuple>
</Dept>
<Emps>
 <tuple>
  <Depts-Emps-Key>sk1(Storage,1999-01-07)</Depts-Emps-Key>
  <Emps-New-Key>1<Emps-New-Key>
 </tuple>
</Emps>
<Emp>
 <tuple>
  <Emps-Emp-Key>1</Emps-Emp-Key>
  <Emp-New-Key>sk2(37,Paul)</Emp-New-Key>
  <EmpID>37</EmpID>
  <EmpName>Paul</EmpName>
 </tuple>
 <tuple>
  <Emps-Emp-Key>1</Emps-Emp-Key>
  <Emp-New-Key>sk2(48, Andrew)</Emp-New-Key>
  <EmpID>48</EmpID>
  <EmpName>Andrew</EmpName>
 </tuple>
</Emp>
```

Istanza finale

- Realizzazione dell'istanza di destinazione secondo il modello relazionale:

Istanza Sorgente (XML)

```
<Biblioteca>
 <NomeBiblio>Feltrinelli</NomeBiblio>
 <CatalogoLibri>
<Genere>Avventura</Genere>
  <Libro>
   <Titolo>Il signore degli Anelli</Titolo>
   <Autore>Tolkien</Autore>
   <Editore>Mondadori</Editore>
   <Prezzo>20.00</Prezzo>
  </Libro>
  <Libro>
   <Titolo>I Promessi Sposi</Titolo>
   <Autore>Manzoni</Autore>
   <Editore>Einaudi</Editore>
   <Prezzo>28.00</Prezzo>
  </Libro>
 </CatalogoLibri>
</Biblioteca>
```

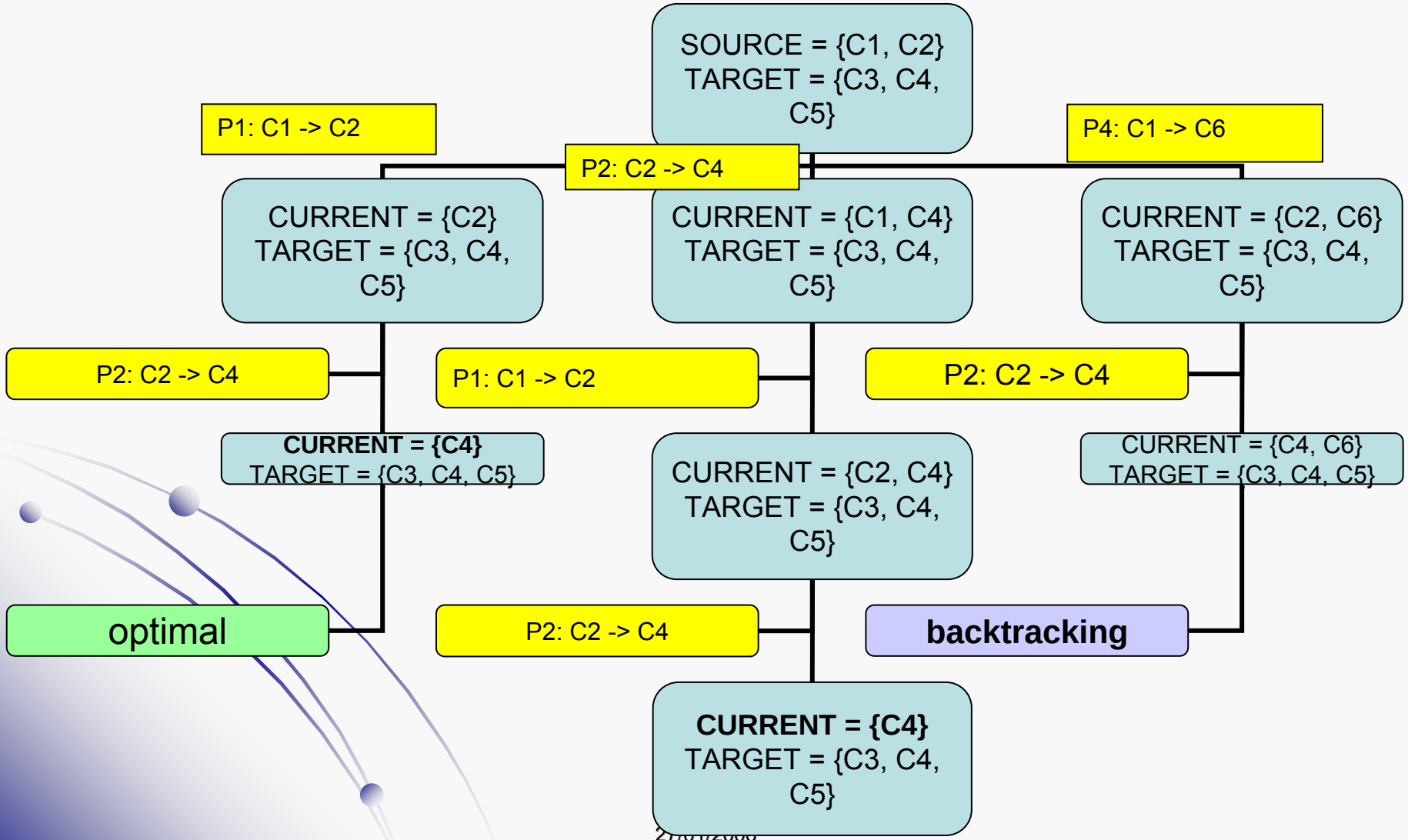
Istanza Destinazione (Relational Model)

```
<Biblioteca>
 <tuple>
  <NomeBiblio>Feltrinelli</NomeBiblio>
  <ChiaveBiblio>Fn(Feltrinelli)</ChiaveBiblio>
 </tuple>
</Biblioteca>

<CatalogoLibri>
 <tuple>
  <Genere>Avventura</Genere>
  <ChiaveRifBiblio>Fn(Feltrinelli)</ChiaveRifBiblio>
  <ChiaveCatalogoLibri>Fn(Avventura)</ChiaveCatalogoLibro>
 </tuple>
</CatalogoLibri>

<Libro>
 <tuple>
  <ChiaveRifCatalogoLibri>Fn(Avventura)</ChiaveRifCatalogoLibri>
  <ChiaveLibro>Fn(Il Signore degli Anelli, Tolkien, Mondadori, 20.00)</ChiaveLibro>
  <Titolo>Il Signore degli Anelli</Titolo>
  <Autore>Tolkien</Autore>
  <Editore>Mondadori</Editore>
  <Prezzo>20.00</Prezzo>
 </tuple>
 <tuple>
  <ChiaveRifCatalogoLibri>Fn(Avventura)</ChiaveRifCatalogoLibri>
  <ChiaveLibro>Fn(I Promessi Sposi, Manzoni, Einaudi, 28.00)</ChiaveLibro>
  <Titolo>I Promessi Sposi</Titolo>
  <Autore>Manzoni</Autore>
  <Editore>Einaudi</Editore>
  <Prezzo>28.00</Prezzo>
 </tuple>
</Libro>
```

Model matching



Progetti

- Gruppi massimo da due persone
  - Studio problemi all'interno del progetto (su tutti proprietà di trasformazioni e gestione mapping)
    - Lettura articoli e verifica
  - Studio di altri strumenti
    - Lettura articolo, esperimenti, relazione/demo
- Progetti da concordare caso per caso a seconda degli interessi