Data Integration: Schema Mapping

Jan Chomicki

University at Buffalo and Warsaw University

March 8, 2007

Data integration

Data sources
- data in any format/data model

Wrappers
- typically: relational or XML
- data/query translation, data publishing
- using source query interfaces

Mediators
- restructuring, merging, reconciliation,...
- eager or lazy
Relational data integration

Data integration system

- **target** (integrated) schema, incl. integrity constraints
- one or more **source** schemas, incl. constraints
- **assertions** relating the contents of the target to the contents of the source(s)

Assertions

- **source-to-target (ST) dependencies:**
  \[ \forall t. \phi_S(t) \Rightarrow \phi_T(t). \]
- **local-as-view (LAV):**
  \[ \forall t. R(t) \Rightarrow \phi_T(t). \]
- **global-as-view (GAV):**
  \[ \forall t. \phi_S(t) \Rightarrow R(t). \]

Data integration vs. data exchange

Data integration [Len02]

- source schema given
- target schema and/or assertions to be constructed
- target instance corresponding to the given source instance may or may not be materialized

Data exchange [FKMP05]

- source and target schemas given
- assertions to be constructed
- target instance needs to be materialized
Problems

Schema matching
Establishing correspondences between elements of the source and target schemas.

Schema mapping
Generation of assertions from schema correspondences.

Data reconciliation
- underspecification: selecting the target instance (uniqueness, nulls)
- overspecification: what if target constraints cannot be satisfied?
- ambiguity: object identification (record linkage)

Schematic discrepancies
- variables in assertions involve schema elements
- source-to-target dependencies not first-order

Schema matching
Finding a “best” match
- start with some initial match and try to improve it
- rank the results

Similarity Flooding [MHR02]
- matching schemas represented as labelled directed graphs
- relational, XML, ontologies, ...

Pairwise connectivity graph $PCG(A, B)$
- $N(PCG(A, B)) = \{(x, y) \mid x \in N(A), y \in N(B)\}$
- $E(PCG(A, B)) = \{((x, y), p, (x', y')) \mid (x, p, x') \in E(A), (y, p, y') \in E(B)\}$

Induced propagation graph $IPG(A, B)$
- $N(IPG(A, B)) = \{(x, y) \mid x \in N(A), y \in N(B)\}$
- $E(IPG(A, B)) = \{((x, y), (x', y')) \mid \exists p \ ((x, y), p, (x', y')) \in E(PCG(A, B)) \lor ((x', y'), p, (x, y)) \in E(PCG(A, B))\}$
- edges are labelled by propagation coefficients $w((x, y), (x', y'))$
Similarity Flooding algorithm

**Algorithm**

1. construct an initial mapping (similarity measure) $\sigma^0$, consisting of weighted pairs of nodes in two schemas
2. construct the mapping $\sigma^i$ based on neighborhood information
3. repeat Step 2 for $i + 1$ if necessary
4. filter the result

**Adjustment step**

$$\sigma^{i+1}(x, y) = \sigma^i(x, y) + \sum_{((x', y'), (x, y)) \in E(IPG(A, B))} \sigma^i(x', y') \cdot w((x', y'), (x, y)).$$

The new values are normalized to $[0, 1]$.

**Termination**

- when the changes to the mapping are below a threshold
- after a fixed number of iterations
- guaranteed for strongly connected graphs.

---

**Schema mapping in CLIO [PVM+02]**

**Data model**

- nested relational
- covers both relational and XML
- source and target constraints: keys, foreign keys

**Values**

- atomic: constants, Skolem constants
- finite sets $\{e_1, \ldots, e_n\}$: set IDs and children $e_1, \ldots, e_n$
- records $\langle A_1 = a_1, \ldots, A_k = a_k \rangle$
- set and record values can be nested

**Method**

- identify source and target atomic elements related by the schema itself to obtain primary paths
- identify source and target atomic elements related by foreign key constraints to obtain logical relations
- interpret schema correspondences to obtain source-to-target dependencies
Tableaux and primary paths

Expression
- schema root
- a variable
- a path expression $e.A_1.A_2.\ldots.A_m$

Tableau
A tableau: $(x_1 \in e_1, x_2 \in e_2, \ldots, x_n \in e_n; C)$ where
- $x_1, \ldots, x_n$ variables
- $e_1, \ldots, e_n$ set-valued expressions; $e_i$ can only refer to $x_j$, $j < i$
- $C$ a conjunction of equalities of the form $e = e'$ where $e$ and $e'$ are path expressions involving only $x_1, \ldots, x_n$

Primary path
A tableau in which $e_i$ refers only to $x_{i-1}$ and $C$ is true.

Constraints

NRI
Nested relational integrity constraint (NRI) is of the form $\forall P_1 \exists P_2 C$ where
- $P_1$ and $P_2$ are primary paths; $P_2$ may be specified relative to a variable in $P_1$
- $C$ is a conjunction of equalities

We assume that the variables in each NRI in the schema are different.

NRIs generalize foreign key constraints and keyrefs (XML Schema).
Chase step

Given an NRI $N_0$

$$IC = \forall (x_1 \in e_1) \ldots (x_n \in e_n) \exists (z_1 \in e_{n+1}) \ldots (z_m \in e_{n+m}) \ C$$

a tableau $T = \langle \ldots , x'_1 \in e'_1 , \ldots , x'_n \in e'_n , \ldots ; C_1 \rangle$, and a substitution

$h(x_1) = x'_1, \ldots , h(x_n) = x'_n, h(z_1) = z'_1, \ldots , h(z_m) = z'_m$ such that

$h(e_1) = e'_1, \ldots , h(e_n) = e'_n$ and $z'_1, \ldots , z'_m$ are fresh variables, a chase step of $T$ with $N_0$ produces the tableau $T'$

$$T' = \langle \ldots , x'_1 \in e'_1 , \ldots , x'_n \in e'_n , \ldots , z'_1 \in h(e_{n+1}) , \ldots , z'_m \in h(e_{n+m}) ; C_1 \land h(C) \rangle$$

The step is applicable if $T'$ is not equivalent to $T$.

Properties of chase

- the order of application of multiple applicable NRI’s does not matter
- **logical relation**:  
  - body: tableau to which no more chase steps are applicable  
  - head: list of atomic attributes occurring in the body at any depth
- chase terminates for acyclic constraints

Schema mapping

**Algorithm**

**Input:**
- source schema $S$ with NRIs $\Sigma_s$
- target schema $S$ with NRIs $\Sigma_t$
- set $V$ of correspondences

**Phase 1:**
- Chase $S$ with $\Sigma_s$ to get logical relations $\{A_1, \ldots , A_n\}$
- Chase $T$ with $\Sigma_t$ to get logical relations $\{B_1, \ldots , B_m\}$

**Phase 2:**

For each pair $(A_i, B_j)$

For each $V' \subseteq V$ covered by $(A_i, B_j)$

Create s-t dependency $(\forall W_i) C_i \Rightarrow (\exists U_j) D_j \land E$

where $body(A_i) = \langle W_i ; C_i \rangle$, $body(B_j) = \langle U_j ; D_j \rangle$

and $E$ are the equalities generated by $V'$

**Output:** the set of all s-t dependencies.
Correspondence

NRI of the form \( v = \forall P_1 \exists P_2 \ e = e' \) where

- \( P_1 \) is a primary path of the source
- \( P_2 \) is a primary path of the target
- \( e = e' \) is an equality relating the last elements of both paths

Coverage

A pair \((A, B)\) of logical relations covers:

- a correspondence \( v = \forall P_1 \exists P_2 \ e = e' \) if there are functions \( h_A \) and \( h_B \) renaming the variables of \( V' \) into those of \( A \) and \( B \) such that \( h(P_1) \) is a subset of \( \text{body}(A) \) and \( h(P_2) \) is a subset of \( \text{body}(B) \) (this generates \( h_A(e) = h_B(e') \))
- a set of correspondences \( V \) if it covers each correspondence in \( V \) (this generates the conjunction of the equalities obtained from covering individual correspondences)

References: