Using the Graph-Model for Schema and Data Mapping

Fritz Laux
Prof. emeritus
Reutlingen University
Dept. of Informatics
Reutlingen, Germany
fritz.laux@reutlingen-university.de

Aim of the Talk

Motivation to use the Graph Model for visualizing schema mapping and data transformation

Contents

- Present the Graph Model and with relevant properties for our purpose
  - Formally compact, yet sufficient for the target aim
- Apply the model to typical situations (patterns)
  - Show benefits and pitfalls

Research challenges (open questions)

- Automate the matching
- Finding the “best” possible mapping
Motivation

Increasing number of data sources need to be integrated to…

- gain added value (knowledge, insights, predictive analysis)
- coordinate complex processes (e.g. traffic control, fight epidemic, industry 4.0)

Challenges

Data Integration problems

- Variety of systems/technologies
  - Incompatible platforms, systems, access technologies

- Logical and semantic reasons
  - Different data models, data structure/representation, synonyms, homonyms

- Social and administrative hindrance
  - Data owners fiefdom, data privacy, performance reasons
Idea for Solution

- Even if we can create transformations to solve these problems, but ongoing maintenance will be difficult since the source systems keep evolving.

- There is a strong need to have visual help to understand the impact and interplay of any changes.

- The idea: Use the Graph Model (GM) to visualize, formalize, and support the data integration.

  Why?
  1. GM is flexible and easy to understand
  2. GM and Category theory allows to check the validity of the integration mappings
  3. GM visualizes interdependencies

Data Integration Scenario (Running Example)

- Hospital patient stats (tabular)
  - For privacy reasons the hospital agrees to provide only the following aggregated Patient statistics
  - region string
  - numPatients int
  - admissDate date
  - Diagnosis text
  - Treatment text

- Mediated Schema (relational)
  - ICD10_classifier
  - ICD10 char(6)
  - description text
  - Patient statistics
  - regionCode char(8)
  - #patients int
  - admissDate date
  - ICD10 Code
  - Treatment text

- Admin office (hierarchical)
  - Population in hierarchically organized administrative areas
  - code | state | #inhabitants
  - code | district | #persons
  - code | province | #persons
  - code | county | #persons
  - code | quarter | #persons
  - urban areas
  - rural areas

Aim  Motivation  Challenges  Idea  Example  Graph Model  Mappings  Quality Criteria  Framework  Solution  Conclusion

5 /22 © F. Laux

6 /22 © F. Laux
Data Integration Scenario (Running Example)

<table>
<thead>
<tr>
<th>Hospital patient stats (tabular)</th>
<th>Mediated Schema (relational)</th>
</tr>
</thead>
<tbody>
<tr>
<td>region string</td>
<td>ICD10 classifier</td>
</tr>
<tr>
<td>numPatients int</td>
<td>ICD10_code</td>
</tr>
<tr>
<td>admissionDate date</td>
<td>diagnosis text</td>
</tr>
<tr>
<td>Treatment text</td>
<td>Treatment text</td>
</tr>
</tbody>
</table>

Possible problems
1. Identify matching elements / metadata (e.g. region to regionCode)
2. Resolve conflicts (e.g. when merging data)
3. Preserve semantics (e.g. extract ICD10 code from diagnosis)
4. Transform models and preserve its structure (e.g. mapping hierarchy to relation)
5. Ensure consistency when multiple mapping paths exist (e.g. from hospital-region to population-regionCode.)

Property Graph Model

Model elements
- Nodes (Vertices) = objects
- Lines (Edges) either directed or undirected = related objects
- Properties (of vertices and/or edges) = detail information about objects or relations

Definition: Graph
- A Graph G := (V,E) is a set of Vertices V and a set of Edges E.
- An Edge e ∈ E is defined by the pair of vertices (u,v), with u,v ∈ V, that connect u and v.

Definition: Property Graph
- A Property Graph PG = (V, E, P) is a Graph where any x ∈ V ⋃ E can have a subset P_x ⊂ P of properties (e.g. key-value pairs) attached to x.

Definition: Hypergraph
- A Hypergraph is a Graph G where the edges e can connect more than two vertices.
Examples of Graph Mappings (1/2)

**Match problem (1, 3)**
- Given schema S and G. A 1:1- or renaming mapping is called a Match. The mapping preserves the semantics.

**Merge problem (2)**
- Given mapping \( M_{12} : S_1 \rightarrow S_2 \), \( G = S_1 \cup S_2 \)

Examples of Graph Mappings (2/2)

**Model Generation problem (4)**
- Given S and G of different meta-models, define mappings to transform S into G
- Goal: preserve semantics as far as possible

Example: S hierarchy, G relational
Important Graph Mapping Types

- **Isomorphism (Edge preserving bijection)** problem (3, 4)
  
  Let $S=(V_1,E_1), G=(V_2,E_2)$
  
  $f: (V_1) \rightarrow (V_2)$ is bijection and
  
  $\forall (v_1,v_2) \in E_1 \iff (f(v_1),f(v_2)) \in E_2$

- **Homomorphism (Edge preserving map)** problem (3, 4)
  
  $f: (V_1) \rightarrow (V_2)$ is mapping and
  
  $\forall (v_1,v_2) \in E_1 \implies (f(v_1),f(v_2)) \in E_2$

Mapping Composition

- **Function Chain**
  
  Let $S_i=(V_i,E_i) (i=1,2,...,n)$ be graphs and
  
  $f_i: (V_i) \rightarrow (V_{i+1}) (1 \leq i < n)$ be mappings.
  
  The composition (or *chain*) of functions
  
  $f_k \circ f_{k-1} \circ \ldots \circ f_1$ ($k < n$) is defined as $f_k(\ldots f_2(f_1(v_1))\ldots) (\forall v_i \in V_i)$

  Example:
  
  Each quarter is mapped to the city it is located, the city in turn is
  mapped to its district and finally the districts is mapped to the
  state.
Commutative Mappings

A function chain is called **commutative** if and only if
\[ f_2 \circ f_1 = f_1 \circ f_2 , \text{i.e. } f_2(f_1(x)) = f_1(f_2(x)) \quad \forall x \in \text{dom}(f_1) \]

Example: \( g \circ \text{id} = \text{id} \circ g \) (and more general \( \hat{g} \circ \text{iso}_1 = \text{iso}_2 \circ g \))

For a consistent mapping from **patient** to **rCode** it is irrelevant if the projection \( g \) to **region** is done first or the isomorphic mapping \( \text{iso}_1 \) to **patientCode**.

Desirable Mappings

Projection \( \pi \), Homomorphism \( \text{hom} \), and Isomorphism \( \text{iso} \) are good candidates for commutative mappings.
(e.g. \( \pi \circ \text{iso} = \text{iso} \circ \pi \))

Bipartite Graph and Graph Matching

Let \( G = (V,E) \) with \( V = V_1 \cup V_2 \) and \( V_1 \cap V_2 = \emptyset \). If there are no edges within \( V_1 \) and \( V_2 \) then \( G \) is bipartite.

Example 1:

Graph Matching **quality criteria for problem (1)**

Let \( G \) be a bipartite Graph. A matching is a subset of edges where no two edges share an endpoint (node)

- **Maximum matching** = maximum number of vertices are matched
- **Perfect matching** = all vertices are matched

Example 2:
**Theorem of Hall (Marriage Theorem)**

Let $G = (V_1 \cup V_2, E)$ be a bipartite Graph. In $G$ exist a perfect matching if $\forall U_1 \subseteq V_1$: $d(U_1) \geq |U_1|$. 

$d(U_1) := |\{v \in V_2 | u \in U_1 \land (u,v) \in E\}|$

**Example 3 (perfect match)**

- All subsets $U_1$ of $V_1$ have $d(U_1) \geq |U_1|$.
- $(a_1,a_4), (a_2,a_5), (a_3,a_6)$ is the only possible perfect matching.

**Example 4 (no perfect match)**

- Subset $U_1 = \{a_2, a_3\}$ has $d(U_1) = |\{a_5\}| = 1$, but $|U_1| = 2$.
- Perfect matching is not possible.

---

**Integration Framework**

1. **Take source models and target model.**
2. **Make all data elements explicit (nodes) that must be matched or mapped.**
3. **Define a bipartite Graph with all elements from step 2 (sources = $V_1$, target = $V_2$).**
4. **Identify semantic matches between sources and target nodes by making edges.**
5. **Define mappings by giving transformation rules or formulae as properties.**
Checking rules for integration completeness

- Use theorem of Hall to check for integration completeness resp. coverage
- Add relations in source and target models to check formal consistency.
  - If a target node can be reached by more than one path, make sure that the mappings are commutative.
  - When the mapping is an aggregation, then the mapping should be a homomorphism.
  - If the mapping is an isomorphism, the mapping is lossless.

Putting everything together (Running Example 1st part)

- Identify matching schema metadata
- Merge diagnosis & description to lookup ICD10 classifier
- If node can be mapped by multiple path, check for commutative mappings
Lessons learned

**Use the GM on the data/object type level**
- Use different colors for node/edge types
- Only use GM for instances if special details need to be visualized (e.g. aggregation of instances of the same object type)

**In real world scenarios the GM tends to be confusing**
- Model partial data structures separately
- In extreme cases use only 1 source element and model all edges from and to this element only. This visualizes all influencing factors and dependencies.

**Some GM theorems allow (formal and automated) quality checks of the data integration**
- Theorem of Hall: coverage/completeness check
- Commutative mappings: consistency checks
- Hypergraph links need detailed description for mappings
References


Cover comprehensively the different concepts of data integration using conjunctive queries as formal representation.


Propose a REST and ReadCheck for uniform relational access and query.


IncMap can detect and leverage semantic-rich patterns in the relational data sources and use them for data integration.

Discussion