QuickMig
Semi-automatic Schema Matching for Data Migration

Christian Drumm, SAP AG
Matthias Schmitt, SAP AG
Hong-Hai Do, SAP AG
Erhard Rahm, University Leipzig

01.10.2007
QuickMig: Overview

Problem Description
- Data migration from unknown source to well-known target system
- Data export schemas of legacy systems not designed for interoperability
- Development of mappings requires significant effort

Solution
- Development of new matching algorithms
- Development of a new methodology
- Combination of new approaches with existing COMA++ functionality
- Prototype implementation including evaluation of new approach

Automatic Schema Matching
- COMA++
  - Used to evaluate schema based matching approaches
  - Enables flexible combination of different matching approaches
  - Schema-based matching approaches not applicable to data migration use case
1. QuickMig
   1.1. Data Migration
   1.2. New Concepts
   1.3. Evaluation

2. Open Research Questions
Data Migration: Scenario

Scenario
- Data migration for „SAP Business ByDesign“
  - SAP's comprehensive new mid market solution
  - Financials, Customer Relationship Management, Human Resources Management, Supply Chain Management, Project Management, …
- Volume business
- Legacy data migration within one week

Challenges
- Customers lack the expertise to perform data migration
- Customers cannot afford expensive data migration projects
- High diversity of possible source systems
  - 3rd party systems
  - Custom solutions

⇒ Migration of any (unknown) source system to well-known targets system
Data Migration: Opportunities – Domain Knowledge

Limited value of source schema information
- Data export schemas of legacy systems are not designed for interoperability
- Technical names, abbreviations, proprietary structures, codes, flat structures, …
- Low quality of meta-data (e.g. element documentation)

Availability of domain knowledge
- Software vendor has detailed knowledge about business capabilities and import interfaces of target system
- Customer has detailed knowledge about business capabilities of source system

➤ Utilize target system knowledge
➤ Leverage business knowledge of source system
Scope of data migration projects

- Rich target system capabilities which can be easily configured to customers' needs
- Data Migration highly dependent on source system capabilities

Accumulated mapping knowledge

- Several mapping tasks in one data migration project
  e.g. mapping of customer data, supplier data, purchase orders, …
- Schemas usually contain similar elements and sub-structures
  e.g. address data, bank data, currency code, …

➤ Flexibly reduce migration scope
➤ Reuse previous mapping results
New Concepts: Schema Reduction

Goal
- Reduce complexity of matching task
- Simplify verification of proposed mapping

Approach
- Electronic questionnaire to exploit source system capabilities and customer specific scope
- Each question targets a specific capability of the target system
  - Irrelevant parts of target schema can be removed
  - Additional simplifications (e.g. cardinality, time dependency, …)
- Mappings between original schema and reduced schema can be derived automatically

Customer
- Name
- Address [1..n]
  - Street
  - City
- BankDetails [1..n]
  - AccountNumber
  - AccountHolder

Customer
- Name
- Address [1..1]
  - Street
  - City

Shall bank details be migrated?
- Yes □ No ☑

Shall multiple addresses by migrated?
- Yes □ No ☑
New Concepts:
Sample Data

**Goal**
- Use instance data for schema matching
- Achieve high precision value (=1.0)

**Approach**
- Pre-deliver specially prepared instance data for target structure
  - For elements with explicit business content only (no technical elements, no codes, …)
  - Highly selective content for every element (no identical values, no substrings, …)
- Enforce identical instance in source system
  - Sample instance is provided in business terms (PDF, Excel, Email, …)
  - Customer can easily create sample instance via standard business user interface
New Concepts: Domain Knowledge

Goal
- Utilize detailed target system knowledge
- Achieve higher recall value

Approach
- Model domain knowledge
  - Alternative information (about data formats and common standards)
    - Different date and time formats
    - Phone Number, Street and House Number
  - Additional information (about target system)
    - Code lists including texts describing the code value
    - Default values for required fields (context dependent)
    - Query attributes for resolution of foreign key associations
- Provide additional instance based matchers exploiting domain knowledge
New Concepts: 
Reuse

Goal
- Reuse previous mapping results within one data migration project
- Achieve higher recall value

Approach
- Provide mapping between similar structures of the target system
- Match corresponding source structures
- Compose required mapping by reusing previously developed correct mapping
**New Concepts: Mapping Categories**

**Goal**
- Determine executable mapping code (beyond simple “Move”)

**Approach**
- Develop a list of semantic mapping categories covering all required transformations
- Extend matchers to enable determination of mapping categories by exploiting sample data and domain knowledge
- Extend “Reuse” matcher to support composition of mapping categories
- Derive mapping code in runtime system
  - Generation of mapping code (“Move”, “Concatenate”, …)
  - Generation of code templates (“Value Mapping”, “Internal ID”, …)
  - Manual implementation (“Complex”)

<table>
<thead>
<tr>
<th>Mapping Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Instance</td>
</tr>
<tr>
<td>Key Mapping</td>
</tr>
<tr>
<td>Internal ID</td>
</tr>
<tr>
<td>Look Up</td>
</tr>
<tr>
<td>Move</td>
</tr>
<tr>
<td>Value Mapping</td>
</tr>
<tr>
<td>Default Value</td>
</tr>
<tr>
<td>Split</td>
</tr>
<tr>
<td>Concatenate</td>
</tr>
<tr>
<td>Complex</td>
</tr>
</tbody>
</table>
Evaluation: Schema Reduction

Evaluation

- Schemas used for evaluation*

<table>
<thead>
<tr>
<th>Scenario</th>
<th># of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Schema</td>
<td>4639</td>
</tr>
<tr>
<td>SAP R/3 4.0</td>
<td>953</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>2150</td>
</tr>
<tr>
<td>SAP B1</td>
<td>480</td>
</tr>
</tbody>
</table>

- Target schema after reduction

<table>
<thead>
<tr>
<th>Scenario</th>
<th># of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Schema</td>
<td>4639</td>
</tr>
<tr>
<td>SAP R/3 4.0</td>
<td>645</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>612</td>
</tr>
<tr>
<td>SAP B1</td>
<td>639</td>
</tr>
</tbody>
</table>

- Resulting target schemas differ largely depending on source system
- Significant reduction of target schema possible

Evaluation: Sample Data

Evaluation

- Purely schema based matching

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>0.40</td>
<td>0.04</td>
<td>0.08</td>
</tr>
<tr>
<td>SAP B1</td>
<td>0.31</td>
<td>0.18</td>
<td>0.23</td>
</tr>
<tr>
<td>Average</td>
<td>0.23</td>
<td>0.07</td>
<td>0.10</td>
</tr>
</tbody>
</table>

- Sample data based matching

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>1</td>
<td>0.27</td>
<td>0.43</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>1</td>
<td>0.38</td>
<td>0.55</td>
</tr>
<tr>
<td>SAP B1</td>
<td>1</td>
<td>0.58</td>
<td>0.70</td>
</tr>
<tr>
<td>Average</td>
<td>1</td>
<td>0.41</td>
<td>0.56</td>
</tr>
</tbody>
</table>

⇒ Schema-based matching approaches not applicable
⇒ Sample data matcher achieves precision of 1
⇒ Improvement of recall required
Evaluation: Domain Knowledge

Evaluation

- Sample data based matching

- Combination of domain knowledge and sample data

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>1</td>
<td>0.27</td>
<td>0.43</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>1</td>
<td>0.38</td>
<td>0.55</td>
</tr>
<tr>
<td>SAP B1</td>
<td>1</td>
<td>0.58</td>
<td>0.70</td>
</tr>
<tr>
<td>Average</td>
<td>1</td>
<td>0.41</td>
<td>0.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>1</td>
<td>0.50</td>
<td>0.66</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>1</td>
<td>0.49</td>
<td>0.65</td>
</tr>
<tr>
<td>SAP B1</td>
<td>1</td>
<td>0.65</td>
<td>0.79</td>
</tr>
<tr>
<td>Average</td>
<td>1</td>
<td>0.55</td>
<td>0.70</td>
</tr>
</tbody>
</table>

- Utilization of domain knowledge significantly improves recall
- Matcher still achieves precision of 1
Evaluation: Reuse

Evaluation

- Reuse matcher

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>0.80</td>
<td>0.50</td>
<td>0.61</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>0.81</td>
<td>0.43</td>
<td>0.56</td>
</tr>
<tr>
<td>SAP B1</td>
<td>1</td>
<td>0.80</td>
<td>0.89</td>
</tr>
<tr>
<td>Average</td>
<td>0.87</td>
<td>0.58</td>
<td>0.69</td>
</tr>
</tbody>
</table>

- Combination of reuse, sample data and domain knowledge

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP R/3 4.0</td>
<td>0.99</td>
<td>0.67</td>
<td>0.80</td>
</tr>
<tr>
<td>SAP ERP</td>
<td>0.99</td>
<td>0.70</td>
<td>0.82</td>
</tr>
<tr>
<td>SAP B1</td>
<td>1</td>
<td>0.80</td>
<td>0.89</td>
</tr>
<tr>
<td>Average</td>
<td>0.99</td>
<td>0.72</td>
<td>0.84</td>
</tr>
</tbody>
</table>

- Reuse matcher achieves mediocre precision and recall values
- Reuse matcher and sample data determine complementary matches
- Combination of all approaches delivers very good results
Evaluation

- 97% of mapping categories identified correctly
- Automatic code generation for more than 40% of the matches possible („Move“ and „Split“)

<table>
<thead>
<tr>
<th>Mapping Categories</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Instance</td>
<td>0.07</td>
</tr>
<tr>
<td>Key Mapping</td>
<td>0.02</td>
</tr>
<tr>
<td>Internal ID</td>
<td>0.02</td>
</tr>
<tr>
<td>Look Up</td>
<td>0.13</td>
</tr>
<tr>
<td>Move</td>
<td>0.36</td>
</tr>
<tr>
<td>Value Mapping</td>
<td>0.14</td>
</tr>
<tr>
<td>Default Value</td>
<td>0.02</td>
</tr>
<tr>
<td>Split</td>
<td>0.07</td>
</tr>
<tr>
<td>Concatenate</td>
<td>0.00</td>
</tr>
<tr>
<td>Complex</td>
<td>0.15</td>
</tr>
</tbody>
</table>

→ Significant reduction of manual implementation effort possible
**Agenda**

1. SAP Research
2. QuickMig
3. Open Research Questions
Open Research Questions

1. **How can mapping code be generated automatically based on mapping categories?**
   - Obvious in case of „Move“
   - What about more complex categories
   - Templates my allready be very helpful

2. **How can (useful) user input be acquired to support the automatic mapping process?**
   - Sample data just a first step
   - Important that user input is acquired non-intrusively
   - Where is the sweet-spot?

3. **How to handle codes and code lists**
   - Sample data not helpful due to large number of similar values
   - Unknown codes and code lists
Thank you!
QuickMig: Resources

COMA++
- http://dbs.uni-leipzig.de/de/Research/coma.html

QuickMig

Contact Details
- Christian Drumm: christian.drumm@sap.com
- Hong-Hai Do: hong-hai.do@sap.com