Gupta, K.M., Zang, M., Gray, A., Aha, D.W., & Kriege, J. (2007). *Enabling the interoperability of large-scale legacy systems* (Technical Note AIC-07-127). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Enabling the Interoperability of Large-Scale Legacy Systems

Kalyan Moy Gupta[1], Mike Zang[2], Adam Gray[2], David W. Aha[3], and Joe Kriege[2]
[1]Knexus Research Corp.; Springfield, VA 22153
[2]CDM Technologies Inc.; San Luis Obispo, CA 93401
[3]Navy Center for Applied Research in Artificial Intelligence;
Naval Research Laboratory (Code 5514); Washington, DC 20375
kalyan.gupta@knexusresearch.com {mzang,adgray,jkriege}@cdmtech.com david.aha@nrl.navy.mil

Abstract

Legacy system data models can interoperate only if their syntactic and semantic differences are resolved. To address this problem, we have developed the Intelligent Mapping Toolkit (IMT), which enables mixed-initiative mapping of meta-data and instances between relational data models. IMT employs a distributed multi-agent architecture so that, unlike many other efforts, it can perform mapping tasks that involve thousands of schema elements. This architecture includes a novel federation of matching agents that leverage case-based reasoning methods. As part of our pre-deployment evaluation for USTRANSCOM and other DoD agencies, we evaluated IMT's mapping performance and scalability. We show that combinations of its matching agents are more effective than any that operate independently, and that they scale to realistic problems (i.e., that involve thousands of schema elements).

Introduction

The interoperability of information systems is an important issue for many organizations. It is a major concern for integrating systems both within and across organizations. For example, the United States Transportation Command (USTRANSCOM) maintains information entities, called *reference data*, which are shared across client organizations at national and international levels. Example reference data entities include airports, vehicles, and citizens. Automated interchange of such reference data across information systems ideally requires that they subscribe to a common, all-encompassing data model – an impractical requisite, given that the local requirements of client applications are typically in constant flux. Instead, mapping meta-data (i.e., schema) and instances across systems is a practical way to manage such changes. The essential operation in schema mapping is *Match*, which takes two schemas as input and produces a mapping between their semantically corresponding elements (Rahm & Bernstein, 2001). For two schemas with n and m elements respectively, the number of possible matches is $n*m$. Therefore, this effort can be prohibitive when mapping schemas with hundreds of thousands of elements. For example, at USTRANSCOM, 25 full-time staff members maintain and distribute over 800 data entities to over 1000 client applications, and four full-time analysts perform mapping. Unfortunately, this approach to mapping is time-intensive and prone to human error. Thus, methods are needed to automate all or part of the mapping task to significantly speed it up and reduce errors.

Several existing research prototypes, including Clio (Miller *et al.*, 2001) and Delta (Clifton, Houseman, & Rosenthal, 1996), provide various levels of intelligent data mapping. Despite their demonstrated utility, these prototypes were not designed to support large-scale operational data mapping. That is, they do not provide adequate support for mixed-initiative mapping as required in an operational setting. Additionally, they do not provide a flexible plug-and-play architecture to accommodate emerging mapping methods for large-scale mapping tasks. Protoplasm (Bernstein *et al.*, 2004), a recent data mapping system, attempts to address this issue. However, to our knowledge, none of these prototypes have been tested or deployed for large-scale operational data mapping efforts (Do, Melnik, & Rahm, 2002), and their operational benefits have not been quantified.

Although many commercial data mapping systems are also available, most only provide graphical user interfaces for manual mapping (e.g., see MapForce (2007)). Very few offer even limited intelligent mapping support. Thus, there is a need for an extensible robust architecture for mixed-initiative relational data mapping.

To meet this need, we created the Intelligent Mapping Toolkit (IMT), which we introduce in this paper. IMT is novel in several ways. It maps large-scale schema (i.e., meta-data) and instance data. It employs a distributed multi-agent architecture that includes a federation of matching agents for case-based similarity assessment and learning. IMT semi-automatically acquires domain-specific lexicons and thesauri to improve its mapping performance. Also, it provides an explanation capability for mixed-initiative mapping.

We evaluate IMT on USTRANSCOM's reference data and show the effectiveness of its multi-agent architecture. In particular, we show that IMT outperforms a single agent variant of itself, and that its multi-agent architecture can solve realistic problems.

The rest of this paper is organized as follows. In the next section, we describe the relational data mapping task and related research. Next, we describe IMT's architecture,

matching agents, and resource acquisition agents, followed by a report on its performance evaluation. Finally, we conclude with thoughts on potential future research.

Background and Related Work

Data mapping is a key task for enabling the seamless exchange of data across heterogeneous systems. It establishes semantic concordances (i.e., *mappings*) between elements of two distinct schemata such that a query issued on their data, with suitable transformations, produces identical results (Fletcher & Wyss, 2005).

Mapping is typically performed by matching *schemata elements*, and its methods can be categorized by the following dimensions (Rahm & Bernstein, 2001):

- Object: Matching *schemata* versus matching *instances*;
- Abstraction: *Elemental* (matching each schema element) versus *structural* (matching groups of structurally related elements);
- Mechanism: *Linguistic* (matching elements based on names and textual descriptions) versus *constraint-based* (matching elements using constraints such as keys and relationships) matching;
- Cardinality (e.g., 1:1, 1:n, and n:m); and
- Auxiliary knowledge resources (e.g., lexicons, thesauri).

Most schema matching systems perform 1:1, linguistic, elemental, and structural schema matching; some use auxiliary resources (Rahm & Bernstein, 2001). Some apply information retrieval and machine learning techniques (e.g., SemInt uses neural networks to cluster attributes and find likely mappings (Clifton *et al.*, 1996)).

At the core, all matching methods must contend with syntactic and semantic variations of the schemata vocabulary. Common syntactic variations include abbreviations (e.g., Arpt vs. Airport) and conventions (e.g., AirportCode, vs. Airport_Code). Semantic variations include the use of *synonyms* (e.g., code vs. id), *hypernyms* and *hyponyms* (e.g., vessel vs. ship), *meronyms* (e.g., first and last name vs. name), and *homonyms* (stud [part] vs. stud [horse]). Syntactic variations can be addressed by exploiting methods for assessing string similarity. These vary from finding exact matches to using edit distances. In contrast, semantic variations cannot be effectively addressed using conventional string matching techniques. Instead, *auxiliary knowledge resources* (e.g., thesauri, linguistic ontologies) must be used. The use, development, and maintenance of knowledge resources with suitable coverage and validity pose challenging issues, which we address in IMT. The large variations in schemata vocabulary motivate the adoption of a multi-pronged approach for matching – the approach we take in IMT, where several configurable linguistic and structural matching agents are applied to each pair of schema elements to assess their similarity. In addition, IMT can be easily extended to include new matching agents. In contrast, existing systems rely on a single method or fixed set of linguistic matching methods.

Only a few mapping systems have been systematically evaluated. For example, SemInt (Clifton *et al.*, 1996) was evaluated with only 5 attributes, and Protoplasm only underwent evaluation for its scalability (Bernstein *et al.*, 2004). We instead evaluated IMT's mapping performance to assess the effectiveness and scalability of its multi-agent architecture.

System Description

USTRANSCOM's Master Model is a model of reference data that aims to standardize all relational database tables maintained and distributed by USTRANSCOM. Schemas pertaining to new DoD processes, continuously being developed, need to be mapped to the Master Model as they are introduced or changed. We designed IMT to support schema and Master Model management professionals at USTRANSCOM and other DoD agencies with schema and instance mapping tasks (CDM, 2006). When fully deployed, we expect IMT to significantly reduce the time required to complete the mapping task.

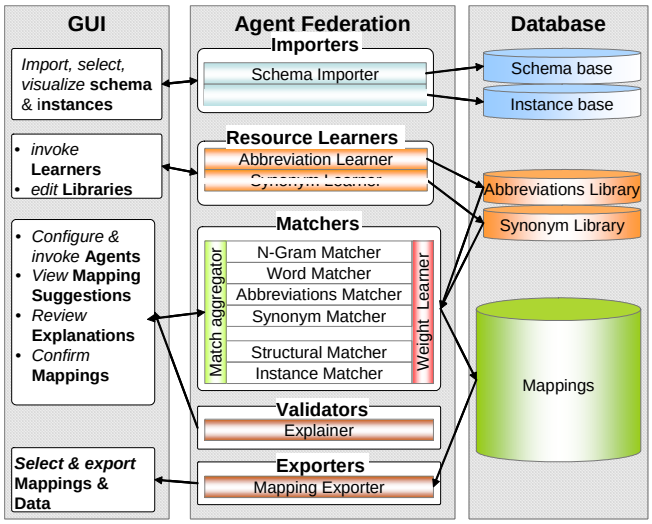

Figure 1. IMT's functional architecture

IMT's primary goal is to suggest mappings to users for final verification and acceptance. Its architecture includes the following three layers of components (see Figure 1).

GUI Layer: This comprises a graphical user interface that allows users to perform the following actions:

- import, select, and visualize relational schemata and instances, the elements of which are to be mapped;
- acquire auxiliary resources (e.g., abbreviation and synonym libraries) by invoking the matching agents;
- create, load, and work on mapping sessions during which users can configure and invoke matching agents, receive mapping suggestions, review mapping explanations, and accept, change, and save mappings (See Figure 2); and
- export the mappings for use in other applications.

An IMT user maps schemas by creating a mapping

session, in which he or she selects a pair of schemas and the subsets to be mapped. The user then configures and invokes the mapping agents, reviews and accepts mappings from the ranked list of mapping recommendations, and saves them with relevant comments.

For example, Figure 2 shows mapping recommendations generated for two schemas from USTRANSCOM containing over 2000 Tables and 13,000 fields. The source schema elements are hierarchically displayed (highlighted in blue in Figure 2). For each element, the corresponding target schema element with the highest similarity score is shown (highlighted in red in Figure 2). For example, a Master Model field "TRANSPORTATION-UNIT SHIPMENT-UNIT-IDENTIFIER" may be suggested as a mapping for the field "Container Transportation Control Number" in the WPS-GTN schema. The WPS-GTN refers to a schema for data exchange between the Worldwide Port System (WPS) and the Global Transportation Network(GTN) System[1].

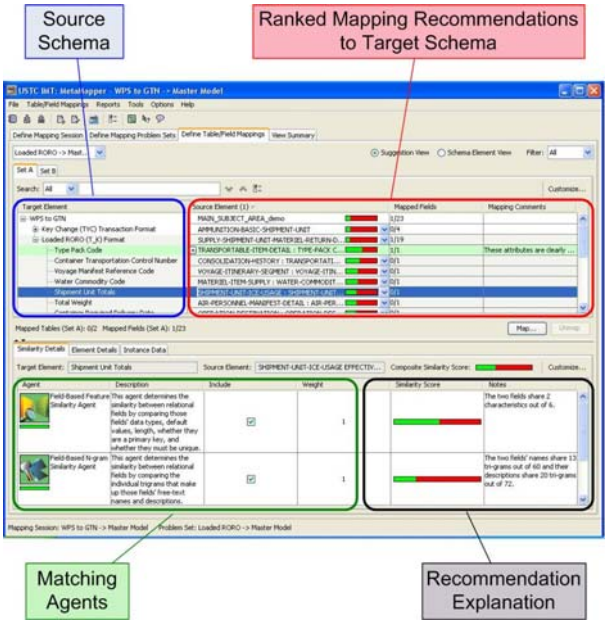

Figure 2. IMT user interface

The lower panel of the interface displays explanations about the computed similarity related to the currently selected recommendation. The users can review such explanations when deciding whether to accept or reject a recommendation. The sub-panel highlighted in green shows the relevant matching agents and the sub-panel highlighted in black shows the corresponding similarity explanations. For example, the field-based N-gram Similarity Agent may calculate a similarity score of 0.75 and present an explanation that reads "the two field names share 24 out of 30 trigrams (segments generated by passing a window, 3 characters long, over a string) and their descriptions share 56 out of 79 trigrams".

Agent Layer: This layer includes five sets of configurable agents that support user actions:

- *Import agents*: These import relational schemata and instances from a variety of source files (e.g., Microsoft Excel, comma-delimited, XML) and from databases via JDBC or ODBC connections.
- *Resource Learners*: Auxiliary knowledge resources (i.e.., abbreviations and synonyms) are acquired semi-automatically. The textual elements of verified mappings, either imported from an external file or from the current session, are used to generate abbreviation and synonym suggestions. The Abbreviation Learner detects and extracts <abbreviation, expansion> pairs using a heuristic that assumes an abbreviation's letters preserve their relative ordering in the expansion, while the Synonym Learner recommends two words as synonymous based on their conditional probability of association. This probability is also used by the mapping agents (discussed in the next bulleted section) to define the strength of their synonymy relation. The user can select from a library of synonyms and abbreviations and configure them for use by the matching agents.
- *Matching agents*: They compute the similarity between elements (i.e., tables and fields) of a pair of schemata. The IMT agents' matching techniques employ similarity assessment procedures typically used in case-based reasoning (CBR), a problem-solving methodology that reuses solutions from similar cases to solve a new problem (Aamodt & Plaza, 1994). Similarity assessment constitutes a critical step in case retrieval.

  IMT represents schema elements using a feature vector. In particular, it performs a linguistic analysis of element names and descriptions to create a bag-of-words representation (Gupta, Aha, & Moore, 2006). The process of matching elements compares two feature vectors and yields a similarity value ranging from 0 to 1, where 1 implies that two schema elements are identical and 0 indicates they are distinct.

  The IMT agents' similarity function computes a ratio of the weighted combinations of matching features (i.e., their intersection) and the union of all features in the two vectors (Gupta & Montazemi, 1997). Feature weights are automatically set by the feature-weighting agents, which we describe later in this section.

  IMT includes four *linguistic matching* agents, each utilizing a different feature representation, to address a variety of syntactic and semantic variations. For example, the N-gram Matcher converts element names and descriptions into n-grams, each of which becomes a feature. This addresses the morphological variations in the text pertaining to verbs and nouns (e.g., description vs. describe). Likewise, the Word Matcher tokenizes multi-word descriptions into words that will be used as features for linguistic matching. Unlike the N-gram Matcher, the Word Matcher uses inputs from the

[1] The WPS system tracks all DoD shipments across all ports in the World and the GTN system provides in-transit visibility of shipments within the Defense Transportation System (DTS).

Synonym Matcher to process semantic variations. The Synonym Matcher computes the similarity of two features by using the Abbreviations and Synonyms Libraries. The Word Matcher then incorporates these results into the overall similarity assessment.

The Weight Learner supports IMT's linguistic matching agents. It implements a modification of the TF-IDF method commonly used in information retrieval systems. We use this method because, in the schema mapping task, only one instance per class is available, which prevents using feature-weighting algorithms (e.g., information gain) that need multiple instances per class.

In addition to linguistic matching agents, IMT includes an implemented Structural Matcher and an Instance Matcher, which we will implement and include in a future version of IMT. The Structural Matcher uses elemental attributes (e.g., keys, key types, data types, and other constraints such as field lengths) to assess structural constraint similarity. The Instance Matcher will examine the data content of two fields to determine their similarity (e.g., it will use the identity function for string matches, and both max-min ranges and averages for numeric features).

The Match Aggregator combines and weights the results of the linguistic and structural agents into an overall similarity score. IMT allows users to control the contribution of each agent. By default, all agents are equally weighted. In our future work, we will add a weight-learning component to the Match Aggregator.

- *Validation Agents*: Currently, IMT implements a limited automated validation capability: an explanation capability for each matching agent. Users can review these explanations to confirm or refute mapping suggestions. We included this capability because our research on explanation in CBR demonstrated its ability to improve decision-making performance (Montazemi & Gupta, 1997).
- *Export Agents*: These export the mappings in a variety of formats such as XML for use by other systems.

Database Layer: This includes the following repositories:

- *Schema Base*: This contains relational schemas and their elements (i.e., tables and fields).
- *Instance Base*: This contains data records for a given schema. Data records from different sources can be associated with a single schema. They can also be partitioned into subsets to support schema mapping or to map a record from one data source into a record from another.
- *Mapping Base*: IMT supports mapping among schemas, tables, fields, or instances. Mappings are stored in the Mapping Base, along with any additional information (e.g., user comments, mapping decision history) that can be used to improve mapping performance as well as abbreviation and synonym learning.
- *Resource Base*: This stores the abbreviations and synonyms. It also includes the strength of association among synonyms for use by matching.

Evaluation

We evaluated IMT's ability to support the mapping task. In particular, our goal was to evaluate its mapping performance and assess the effectiveness of its multi-agent architecture in comparison with its single-agent variants. Complexities inherent in the schema mapping task imply that multiple concurrent matching techniques are likely to perform better than a single matching technique. However, thus far, this has not been formally investigated. Consequently, it is one of the primary focus of our evaluation. Next, we present our hypotheses, data, tools, easures, test procedure, and results.

Hypothesis. IMT performs better in the multi-agent mode than in the single-agent mode.

Data. USTRANSCOM provided us two schemas to evaluate the mapping task: (1) the WPS-GTN schema and (2) the Master Model schema (see Table 1). This task focuses on mapping WPS-GTN to the Master Model, which has 12,383 fields. This pair of schemas has 10,302,656 1:1 possible field mappings. USTRANSCOM provided 597 of the 832 mappings from WPS-GTN to the Master Model, which we used as the Gold Standard for our investigation. There were no mappings for the remaining 235 of these 832 WPS-GTN fields. None of the mappings involved identical field or table names across the two schemas.

Table 1. Schemas for mapping performance tests

Characteristic	WPS-GTN	Master Model
Tables	47	2039
Fields	832	12,383
Fields per Table (avg.)	18	6

Tools. (1) IMT was used with all its matching agents: Word Matcher (WM), Gram Matcher (NM), and the Structural Matcher (STM). The SYM matcher is only used in conjunction with WM. (2) CDM's multi-agent test platform for simulating mapping tasks.

Measure. We measured the *Rank of the Correct Mapping* (RCM) in the list of ranked suggestions displayed by IMT. A rank of 1 means that the IMT agent performed perfectly. An RCM of 5 implies that a user will likely look through 5 mappings before finding the correct one. Lower RCM values imply better performance.

Procedure. We performed 4 simulation runs of IMT using CDM's test platform to generate and record mapping suggestions for the 597 mappable fields in WPS-GTN. For each of these we measured their RCM. The first three runs involved the individual matching agents (i.e., WM, STM, and NM). In the fourth run, we combined the agents using the Match Aggregator for a multi-agent (MA) mode. To generate the best mapping suggestions, we manually searched for the best weight combination to be used by the Match Aggregator. We only report the best result here and use paired t-statistics for our analysis.

Results. IMT, when used in the MA mode, outperforms all

of the individual matching agents (see Table 2). The average RCM for MA was 12.90. This is significantly better than WM (RCM=23.93, [p=0.000]), STM (RCM=71.30, [p=0.000]) and NM (RCM=34.34, [p=0.000]). Therefore, we accept our hypothesis. The best performing weight combination for MA was 3 (WM), 1 (SM), and 1 (NM). Therefore, the word-matching agent proved to be the most effective contributor of the three.

For 59.05% of the mapping tasks (i.e., WPS-GTN mappable fields), the best performing MA provides the correct mapping within the first five suggestions. Given that USTRANSCOM currently employs no tool with comparable capabilities, their use of IMT could yield substantial savings in effort.

Each simulation run, comprising 597 mapping tasks, took approximately 1 hour and 45 minutes on average. This implies that each mapping task took approximately 11 seconds, which users are likely to consider as an acceptable performance level for mapping against a schema with 12,383 fields. Thus, we conclude that IMT's multi-agent architecture is well suited for realistic mapping tasks.

Table 2. Task performance results

Agent	RCM	Proportion of correct mappings in		
		Top 1	Top 5	Top 10
WM	23.93	25.72%	51.44%	65.99%
STM	71.30	6.43%	27.58%	34.86%
NM	34.34	26.90%	48.22%	54.99%
MA	**12.90**	29.44%	**59.05%**	69.20%

Discussion

Our evaluation shows that IMT can be effective for industrial mapping tasks. For example, nearly 60% of the time, the correct mapping is contained within IMT's top five suggestions, resulting in a significant acceleration of the mapping task. At USTRANSCOM, this implies a potential savings of one full-time staff member.

Although IMT significantly reduced mapping errors, it did not eliminate them. This validates our emphasis on a mixed-initiative approach and the utility of the user's domain and task expertise.

Independent of the evaluations reported here, we conducted an informal evaluation of IMT at USTRANSCOM. USTRANSCOM users found IMT's explanation capability valuable. For example, based on IMT's recommendations and explanations, they found additional mappings in their schemas that they had previously overlooked. We also tested its plug-and-play multi-agent architecture by developing and testing an agent that performs exact matching, which required only a few hours demonstrating that relatively simple to create and integrate new matching agents into IMT's architecture.

Conclusion

Semantic mapping across heterogeneous data can enable interoperability across organizational systems. Its automation has been the focus of much recent research (Rahm & Bernstein, 2001; Kalfoglou & Schorlemmer, 2005). However, these recent methodologies have not been applied in industry nor evaluated in an industrial context. We introduced and described IMT, a practical integrated tool for semi-automatic schema mapping. It includes many novel features, such as case-based matching agents embedded in a distributed multi-agent architecture with an explanation capability. We demonstrated that IMT's multi-agent version performs better than its single-agent variant and that it performs well for realistic mapping tasks.

We left several issues to be addressed in the future. For example, we will improve our algorithms for schema structure matching and consider instance information for schema matching. We will also exploit existing semantic resources such as WordNet (Felbaum, 1998) and investigate methods for automatically identifying the optimal weight settings to aggregate matching results, rather than rely on manual search. We will investigate the applicability of IMT to mapping XML schemas.

Acknowledgements

We thank the Naval Research Laboratory and CDM Technologies Inc. for supporting this research.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, **7**(1), 39-59.

Bernstein, P.A., Melnik, S., Petropoulous, M., & Quix, C. (2004). Industrial-strength schema matching. *SIGMOD Record*, **33**(4), 38-43.

CDM (2006). *Intelligent mapping toolkit (IMT): Design and development report*. Unpublished Technical Report. San Luis Obispo, CA: CDM Technologies Inc.

Clifton, C., Housman, E., & Rosenthal, A. (1996). Experience with a combined approach to attribute matching across heterogeneous databases. *Proceedings of the Seventh Conference on Database Semantics* (pp. 428-454). Leysin, Switzerland: Chapman & Hall.

Do, H., Melnik, S., & Rahm, E. (2002). Comparison of schema matching evaluations. *Revised papers from NODe Web and Database Related Workshops on Web, Web-services, and Database Systems* (pp. 221-237). London, UK: Springer.

Felbaum, C. (Ed.) (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Fletcher, G.H., & Wyss, C.M. (2005). Relational data mapping in MIQIS. *Proceedings of 2005 ACM SIGMOD conference on management of data* (pp. 912-914). New York, NY: ACM Press.

Gupta K.M., Aha D.W., & Moore P.G. (2006). Rough set feature selection algorithms for textual case-based classification. *Proceedings of the Eighth European Conference on Case-Based Reasoning* (pp. 166-181). Ölündeniz, Turkey: Springer.

Gupta, K.M., & Montazemi, A.R., (1997). Empirical evaluation of retrieval in case-based reasoning systems using modified cosine matching function. *IEEE Transactions on Systems, Man, and Cybernetics*, **27**(5), 601-612.

Kalfoglou, Y. & Schorlemmer, W.M. (2005). Ontology mapping: The state of the art. In Y. Kalfoglou, M. Schorlemmer, A.P. Sheth, S. Staab, & M. Uschold (Eds.), *Semantic interoperability and integration*. Schloss Dagstuhl, Germany: IBFI.

MapForce (2007). MapForce®: A visual flat file data mapping tool. Retrieved from http://www.altova.com/download/mapforce/data_mapping_professional.html on 17 January, 2007.

Montazemi, A.R, & Gupta, K.M. (1997). On the effectiveness of cognitive feedback from an interface agent. *OMEGA International Journal of Management Science*, **25**(6), 643-658.

Miller, R.J., Hernandez, M.A., Haas, L.M., Yan, L., Ho, C.T., Fagin, R., & Popa, L. (2001). The Clio project: Managing heterogeneity. *SIGMOD Record*, **30**(1), 78-83.

Rahm, E., & Bernstein, P.A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, **10**, 334-350.