IMT: A Mixed-Initiative Data Mapping and Search Toolkit

Michael Zang¹, Adam Gray², Joe Kriege¹, Kalyan Moy Gupta³, David W. Aha⁴

¹CDM Technologies, Inc.; San Luis Obispo, CA 93401
²Collaborative Agent Design Research Center (CADRC); California Poly Technical State University; San Luis Obispo, CA 93405
³Knexus Research Corp.; Springfield, VA 22153
⁴Navy Center for Applied Research in Artificial Intelligence; Naval Research Laboratory (Code 5514); Washington, DC 20375

{mzang,adgray,jkriege}@cdmtech.com kalyan.gupta@knexusresearch.com david.aha@aic.nrl.navy.mil

Abstract
Interoperability requires the resolution of syntactic and semantic variations among system data models. To address this problem, we have developed the Intelligent Mapping Toolkit (IMT), which employs a distributed multi-agent architecture to enable mixed-initiative mapping of metadata and instances. This architecture includes a novel federation of service-encapsulated matching agents that leverage case-based reasoning methods. We have recently used the IMT matching service to develop several domain-specific search applications in addition to the IMT mapping application.

The Motivation for Developing IMT
Interoperability among information systems is a primary concern in integrating processes both within and across organizations. As the distribution process owner (DPO) for the U.S. Military, this is particularly true for the United States Transportation Command (USTRANSCOM), which integrates distribution processes (e.g., supply requisition, inventory management, and transportation) across the individual military services, suppliers, shippers, and host nation support systems. To facilitate the requisite levels of interoperability among system-specific information models, USTRANSCOM has developed the Distribution Process Information Exchange Data Model (DPIEDM) and initiated an effort to map existing system-to-system interfaces to this logical data model. DPIEDM’s goal is to provide a much improved semantic and contextual specification to information exchanges, thus improving current and future process integration across the extended enterprise.

The essential operation in data mapping is Match, which takes two schemas (or table extensions) as input and produces a mapping between elements of them that correspond semantically (Rahm and Bernstein 2001). For two schemas with n and m elements respectively, the number of possible matches is \( n^m \), implying a manually prohibitive effort when mapping to schemas containing thousands of elements, such as the DPIEDM. This implication prompted USTRANSCOM to automate aspects of their mapping task to significantly decrease the requisite level of effort (i.e., time and expertise) while reducing errors. No usefully-applicable, commercial products for semantic mapping automation exist. Thus, USTRANSCOM sponsored the development of the IMT operational prototype, which applies Artificial Intelligence (AI) techniques to this compelling problem. The IMT project was a collaborative endeavor involving CDM, CADRC, Knexus, and NRL, and USTRANSCOM’s semantic mapping community.

The IMT Prototype Description
We introduced IMT in our IAAI-08 paper Enabling the Interoperability of Large-Scale Legacy Systems (Gupta, et. al. 2008). Here we summarize it only briefly. IMT proves novel in several ways. It maps large-scale schema (i.e., metadata) 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. It also provides an explanation capability for mixed-initiative mapping. IMT’s primary goal is to suggest mappings to users for final verification and acceptance. Its architecture includes the three layers of components shown in Figure 1 and described below.

The GUI Layer comprises a graphical user interface that allows users to perform actions such as importing, selecting, and visualizing problem elements; acquiring auxiliary resources; invoking matching agents; consulting the agent explanation facility; and exporting mapping solutions for use in other applications.

The Agent Layer provides Matching agents that compute the similarity between problem elements (i.e., tables and fields) by employing similarity assessment procedures typically used in case-based reasoning (CBR). Each agent uses 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, to address 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.
Interoperability requires the resolution of syntactic and semantic variations among system data models. To address this problem, we have developed the Intelligent Mapping Toolkit (IMT), which employs a distributed multi-agent architecture to enable mixed-initiative mapping of metadata and instances. This architecture includes a novel federation of service-encapsulated matching agents that leverage case-based reasoning methods. We have recently used the IMT matching service to develop several domain-specific search applications in addition to the IMT mapping application.
for linguistic matching. Unlike the N-gram Matcher, the Word Matcher users inputs from the 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 **Database Layer** includes JDBC-compliant repositories for persisting the mapping problem and solution representation—supporting mapping among schemas, tables, fields, and instances—and the resources for storing the abbreviations and synonyms—supporting the strength of association among synonyms for use by matching. Additionally, schema and instance data may be imported directly from mapping problem sources.

### The New Capabilities and Applications

Since completion of the initial IMT prototype for USTRANSCOM, the underlying similarity assessment framework and agents have been re-factored and cleanly partitioned into a Similarity Assessment Service supporting a number of interfaces (e.g., Java, SOAP, and REST), and a new IMT semantic data mapping toolkit revision. This approach has generalized the original GUI Layer into an Application Layer supporting other problem domains.

In addition to the IMT application, the Similarity Assessment Service now supports domain-specific search tools including: (1) an application to identify desired records in military reference data, and (2) an application to identify vehicles or people of interest. Capabilities currently under development for the IMT mapping application include an agent providing schema match scores from corresponding sample data values, and the generation of data transformation code from the semantic mappings produced by IMT.

### The Demonstration

This demonstration employs a combination of display posters, self-running slide-shows, hands-on software interaction by attendees, and narrated software presentations to show the ability of IMT to specify, import, and refine a metadata or instance data mapping problem. The demonstration further illustrates the practical decision support assistance provided by the IMT towards resolving these problems. Additionally, our demonstration will incorporate one or more intuitive IMT-technology-derived search applications developed as Cal Poly student senior projects under the auspices of the CADRC². These applications will show the weighted combination of multiple Match methods—including N-Gram, Word with Synonym replacement, Measured Quantity, Geographic Location, and Sample Value comparison—to assess similarity between distinct data elements such as the schemas, tables, and fields of two databases to be mapped.

### References


**Demo Summary**

This demonstration employs a combination of display posters, self-running slide-shows, hands-on software interaction by attendees, and narrated software presentations to show the ability of Intelligent Mapping Tool (IMT) to specify, import, and refine a metadata (i.e., schema) or instance data (i.e., record) mapping problem. The demonstration further illustrates the practical decision-support assistance IMT provides towards resolving these problems. Additionally, our demonstration will incorporate intuitive IMT-technology-derived search applications developed as student senior projects under the auspices of the California Polytechnic State University Collaborative Agent Design and Research Center (CADRC). These applications will demonstrate the combination of multiple data matching methods to assess similarity between distinct data elements. Semantic methods include the statistically-weighted comparison of n-grams and words—with synonym replacement. Numeric methods include comparison of measured quantities and geographic locations. Data elements may correspond to schemas, tables, fields and records in the *Mapping* problem, or a query and database for *Search*. 
Demo Storyboard

This document describes the representational elements of two distinct software demonstrations, the IMT Mapping Tools and the IMT Search Tools, proposed for demonstration at IAAI-08.

**IMT Mapping Tools**

**Overview**

Analysts for the MusicMiner software application desire interoperability with the MusicBrain system, a popular application with an overlapping domain. To support the creation of semantic mappings between the two systems’ underlying schemas the analysts have turned to the Intelligent Mapping Toolkit (IMT).

**Step 1**

The analyst imports xml-based schemas for both MusicMiner and MusicBrain into IMT and selects the tables of interest are from each. He also selects the similarity agents to use for mapping suggestion generation providing the following confidence factors for each:

- Feature Similarity Agent (1.0)
- N-gram Similarity Agent (0.5)
- Semantic Similarity Agent (0.5).

The analyst then clicks the “Generate Suggested Mappings” button.
Step 2
The analyst selects the “Define Table/Field Mappings” tab. Suggested mappings for the elements of the MusicMiner and MusicBrain schemas are displayed in the main pane and ordered by similarity score.

Step 3
The analyst expands the combo box located by the “ARTIST” field in the MusicBrain schema. The top 10 most similar fields to “ARTIST” from the MusicMiner schema are displayed. The analyst selects the “ARTIST_IN_BAND” element from the combo box and clicks the “Map…” button.
Step 4
The analyst expands the “TRACK” table under the MusicMiner schema to display its associated fields. The analyst notices that the “TRACK_GID” field’s highest ranked suggested mapping is to the “SONG_GROUP_ID” field from the MusicBrain schema.

![Image of the schema table with highlighted columns for TRACK_GID and SONG_GROUP_ID]

Step 5
The analyst selects the “TRACK_GID” and “SONG_GROUP_ID” row in the schema table. The individual similarity agent results are displayed in the pane below and the analyst notices that both the n-gram and semantic similarity agent results are near 50% despite a strong disparity in names. Upon further inspection, he sees that while the two elements names are entirely dissimilar, their descriptions share 4 words out of 4.

![Image of the similarity agent results]

Step 6
The analyst selects the “Element Details” tab and notices that the elements share near identical descriptions.

![Image of the element details comparison]

Step 7
The analyst closes the IMT application.
**IMT Search Tools**

**Overview**
A police officer arrives at the scene of a hit and run and interviews an eyewitness who tells him the following information about the suspect’s vehicle:

- it was a green SUV,
- it was relatively new, and
- it had a license plate number ending in ‘4A’.

**Step 1**
The next day at the police station, the officer attempts to compile a list of potential suspects. He starts by opening the Marvel Search web application.

![Marvel Search](image)

**Step 2**
The officer types ‘SUV’ into the *Type* field and ‘4A’ into the *License Plate Number* field, then clicks the *Search* button.

<table>
<thead>
<tr>
<th>State</th>
<th>License Plate Number</th>
<th>Make</th>
<th>Model</th>
<th>Year</th>
<th>Type</th>
<th>VIN</th>
<th>Owner’s Zip Code</th>
<th>Color</th>
<th>Registered Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV</td>
<td>MWA4DA</td>
<td>Chevrolet</td>
<td>Tracker</td>
<td>2003</td>
<td>SUV</td>
<td>X9MFG0DND9UJFFSPZ8J0</td>
<td>69620</td>
<td>Dark Brown</td>
<td>Alex Baxx</td>
<td></td>
</tr>
<tr>
<td>IL</td>
<td>B486PA</td>
<td>Geo</td>
<td>Tracker</td>
<td>1996</td>
<td>SUV</td>
<td>P4R88HGQZVYUPPEZD64</td>
<td>62376</td>
<td>Olive</td>
<td>Mercedes Bele</td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td>R46960A</td>
<td>Jeep</td>
<td>Grand Cherokee</td>
<td>1994</td>
<td>SUV</td>
<td>67AAT9G06Q2BU2H4HIM</td>
<td>94009</td>
<td>Khaki</td>
<td>Lisa Tamis</td>
<td></td>
</tr>
<tr>
<td>NJ</td>
<td>4AKBBBII</td>
<td>Chevrolet</td>
<td>Tracker</td>
<td>1998</td>
<td>SUV</td>
<td>B1H9X97104X6CDDO</td>
<td>8872</td>
<td>Magenta</td>
<td>Anthony Ansley</td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>PBF9KJ</td>
<td>Buick</td>
<td>Rainier</td>
<td>2007</td>
<td>SUV</td>
<td>906G0A49Y3SA461HP8XZ</td>
<td>28044</td>
<td>Light Brown</td>
<td>Curt Mulachi</td>
<td></td>
</tr>
<tr>
<td>IN</td>
<td>0H391W</td>
<td>Acura</td>
<td>MDX</td>
<td>2004</td>
<td>SUV</td>
<td>T0S53E3895Z7RVATX</td>
<td>40348</td>
<td>Dark Orange</td>
<td>Omar Calten</td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td>C8FG0G</td>
<td>Honda</td>
<td>CR-V</td>
<td>2005</td>
<td>SUV</td>
<td>8YP91MR095HA00036OH</td>
<td>49374</td>
<td>Dark Orange</td>
<td>Bayley Amos</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td>2DOX</td>
<td>GMC</td>
<td>Yukon</td>
<td>1995</td>
<td>SUV</td>
<td>PNN01SFTV2J2Z8POS</td>
<td>33330</td>
<td>Violet</td>
<td>Cutter Hulik</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>W8PFJKX</td>
<td>Ford</td>
<td>Explorer Sport</td>
<td>2001</td>
<td>SUV</td>
<td>TDOKAL88HT4213VY8B</td>
<td>19713</td>
<td>Blue</td>
<td>Timo Riedel</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>PH13TR</td>
<td>Hummer</td>
<td>H1</td>
<td>2006</td>
<td>SUV</td>
<td>3K4V20FTTWH00PH4D</td>
<td>72399</td>
<td>Green Yellow</td>
<td>Mandob Beve</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td>SWEZ4I</td>
<td>Nissan</td>
<td>Xterra</td>
<td>2005</td>
<td>SUV</td>
<td>100ARJ505650OKY09E</td>
<td>32004</td>
<td>Blue</td>
<td>Shulamith Ocean</td>
<td></td>
</tr>
<tr>
<td>NE</td>
<td>6HYK4R</td>
<td>Ford</td>
<td>Excursion</td>
<td>2004</td>
<td>SUV</td>
<td>6QWQCD0V2QR24S1G0</td>
<td>6936</td>
<td>Silver</td>
<td>Darci Link</td>
<td></td>
</tr>
<tr>
<td>TX</td>
<td>DZ1U9D</td>
<td>Lexus</td>
<td>GX470</td>
<td>2003</td>
<td>SUV</td>
<td>449P65040AS4H170ZFY</td>
<td>76452</td>
<td>Pink</td>
<td>Dubai Axton</td>
<td></td>
</tr>
<tr>
<td>MN</td>
<td>VYU5QD</td>
<td>Honda</td>
<td>Passport</td>
<td>2000</td>
<td>SUV</td>
<td>6LZW10504CL02121Y</td>
<td>56160</td>
<td>Hot Pink</td>
<td>Lukins Dengus</td>
<td></td>
</tr>
<tr>
<td>IL</td>
<td>SHPS0YC</td>
<td>GMC</td>
<td>Sierra</td>
<td>1991</td>
<td>SUV</td>
<td>R38TES47VH613Y20V7</td>
<td>61644</td>
<td>Violet</td>
<td>Merlin Ables</td>
<td></td>
</tr>
</tbody>
</table>
Step 3
The officer looks at the results returned from his search and sees that there are several SUVs with a license plate ending in ‘4A’. He clicks on the green score bar for the first result and sees that its combined score is 60.1%, resulting from a 100% match on Type and a 20.3% match on License Plate Number.

<table>
<thead>
<tr>
<th>State</th>
<th>License Plate Number</th>
<th>Make</th>
<th>Model</th>
<th>Year</th>
<th>Type</th>
<th>VIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV</td>
<td>MMAD4A</td>
<td>Chevrolet</td>
<td>Tracker</td>
<td>2003</td>
<td>SUV</td>
<td>X9MF7GND1LF</td>
</tr>
<tr>
<td>Type</td>
<td>Geo</td>
<td>Tracker</td>
<td>1996</td>
<td>SUV</td>
<td>P4R5HOV2WY1</td>
<td></td>
</tr>
<tr>
<td>License Plate Number</td>
<td>Jeep</td>
<td>Grand Cherokee</td>
<td>1994</td>
<td>SUV</td>
<td>674ATIGN6CZU</td>
<td></td>
</tr>
<tr>
<td>NJ</td>
<td>4ACD88E1</td>
<td>Chevrolet</td>
<td>Tracker</td>
<td>1998</td>
<td>SUV</td>
<td>BOHDKV91093C</td>
</tr>
<tr>
<td>NC</td>
<td>PBR2NZJ</td>
<td>Buick</td>
<td>Rainier</td>
<td>2007</td>
<td>SUV</td>
<td>GC5GASQVSK</td>
</tr>
<tr>
<td>IN</td>
<td>0H391W</td>
<td>Acura</td>
<td>MDX</td>
<td>2004</td>
<td>SUV</td>
<td>TN0S5E3RO5Z</td>
</tr>
</tbody>
</table>

Step 4
The officer decides that he needs to add additional search criteria. Interpreting what the eyewitness told him, he types in ‘2000’ for the Year, chooses a green color in the Color field, and clicks the Search button for a second time.
Step 5
Looking at his new search results, the officer sees that there are several Olive colored SUVs with a year close to 2000. The first result has a License Plate Number ending in ‘NA’, which could have been misread by the eyewitness as ‘4A’. Since he’s not sure on how good the information was from the eyewitness, the officer decides to change the confidence values of his search terms by clicking on each column name and moving the Confidence slider bar. He changes the confidence on the License Plate Number column to 75%, the confidence on Year to 25%, the confidence on Type to 90%, and the confidence on Color to 50%.

<table>
<thead>
<tr>
<th>Year (25%)</th>
<th>Type (90%)</th>
<th>VIN</th>
<th>Owner’s Zip Code</th>
<th>Color (50%)</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>SUV</td>
<td>GUAT71GD4AFXAFS4P</td>
<td>72342</td>
<td>Olive</td>
<td>Setiawan</td>
</tr>
<tr>
<td>1996</td>
<td>SUV</td>
<td>P4R5HOV2WYPPEZX54</td>
<td>62376</td>
<td>Olive</td>
<td>Mercade</td>
</tr>
</tbody>
</table>

Step 6
The officer clicks Search for a third time and observes the new results. He sees that the top match is an Olive 1996 Geo Tracker with a License Plate of 0AE6P4A. Marvel gave it a composite score of 69.4% due to a 100% match on Type, 84.5% match on Color, 78.9% match on Year, and 19.3% match on License Plate Number.

Step 7
The officer takes down the vehicle VIN number and proceeds to track down its registered owner.
Hardware and Software Requirements

The demonstration will require 3-4 laptop computers which will be provided by the authors/demonstrators of this proposal. The demonstration will require the IAAI-08 to provide No hardware or software in support of this demonstration.