Interactively Building Geospatial Mashups

Craig A. Knoblock
University of Southern California

Work in collaboration with Shubham Gupta, Pedro Szekely, and Rattapoom Tuchinda
MASHUPS

a) LA crime map
   - Crime Report from different counties
   - Map

b) zillow.com
   - Real Estate Listing
   - Property Tax

c) Ski bonk
   - Weather
   - Snow Report
   - Snow Resorts

Combined Data gives new insight / provides new services
PROBLEM

• Most Mashups require significant expertise to create
• Demand for creating integrated applications is huge
• Every user has their own unique requirements for an integrated application
• Available sources and needs to integrated data continues to grow
MASHUP BUILDING ISSUES

Data Retrieval

Wrapper

Attribute

Clean

Wrapper

Attribute

Clean

Combining

Integrate

Customize Display

Introduction • Approach • Evaluation • Related Work • Conclusion
EXISTING APPROACHES

**Goal**: Create Mashups without Programming
- Doesn’t translate to not having to understand programming

**Widget Paradigm**
- Widgets (i.e., 43 for Pipes, 300+ for MS) represents an operation on the data
- Locating and learning to customize widget can be time consuming
- Most tools focus on particular issues and ignore others

Can we come up with a framework that addresses all of the issues while still making the Mashup building process easy?

---

**Introduction**  •  **Approach**  •  **Evaluation**  •  **Related Work**  •  **Conclusion**
KEY CONTRIBUTIONS

• A programming by demonstration approach that uses a single table for building a Mashup

• An integrated approach that links data extraction, source modeling, data cleaning, and data integration together

• A query formulation technique that allows users to specify examples to build complicated queries
KEY IDEAS

• Focus on data, not operations
  – Users are more familiar with data

• Leverage existing data
  – Help source modeling, cleaning, and data integration

• Consolidate as opposed to Divide-And-Conquer
  – Solving a problem in one issue can help solve another issue
  – Interacting within a single spreadsheet platform
KARMA USER INTERFACE

Introduction

- Approach
- Evaluation
- Related Work
- Conclusion
INTEGRATION SCENARIO

Evacuation Centers CSV

- EvacCenter_ID, Address, City

Emergency Coordinator MySQL Database

- Extract
- {EvacCenter_ID, Address, City}
- Extract
- {Name, City, Phone No.}
- Clean
- \{EvacCenter_ID, Address, City, Name, Phone No.\}

Injury statistics in Excel Spreadsheet

- Extract
- \{Date, Injuries, Fatalities\}
- Visualize as chart

Google News Website

- Extract
- \{Headlines, Summary, Date, Link\}
- Visualize as bulleted list
RETRIEVING DATA FROM DIVERSE SOURCES

- Karma facilitates retrieval of data from structured data-sources, such as Excel spreadsheets, MySQL databases and CSV files.
- Karma also facilitates the extraction of data from semi-structured data sources such as web pages.

---

**Introduction** • **Approach** • **Evaluation** • **Related Work** • **Conclusion**
EXTRACTION BY EXAMPLE

• The retrieval of data from structured data-sources, such as Excel sheets and CSV files is done through a drag and drop mechanism.
• The user is only required to select a sample data-element and drop it into Karma’s data table.

![Image of Karma's data table with drag and drop functionality]
1. **Japon Bistro**
   970 E Colora Blvd., Pasadena, CA, 91106
   Upscale yet affordable Japanese eatery offers the city's largest sake selection.

2. **Hokusai**
   8400 Wilshire Blvd., Beverly Hills, CA, 90211
   Chic elegance and modern Zen style surround Japanese French this paean to haute cuisine and stylized sushi.

3. **Sushi Sasabune**
   12400 Wilshire Blvd, Ste 150, Los Angeles, CA, 90025
   Sushi is the singular star at this Zen Westside palace that bows only to the royalty of chef and fish.

4. **Sushi Roku**
   8445 W 3rd St, Los Angeles, CA, 90048
   High fashion, rock and roll and Hollywood buzz converge over innovative sushi.

---

**Content Selection**

- Japon Bistro
- Sushi Dokoro
- Hokusai
- Sushi Sasabune
- Sushi Roku
- Hide Sushi
- Fat Fish
- Sushi Katsu-ya
- Gindi Thai
- Katana
- Echigo

---

**Structure Diagram**

- **Tbody/tr[1]/td[2]/a**
- **Tbody/tr*/td*/a**

---

**Table of Contents**

- Introduction
- Approach
- Evaluation
- Related Work
- Conclusion
1. **Japon Bistro**
   927 E Colorado Blvd, Pasadena, CA, 91106
   Upscale yet affordable Japanese eatery offers the city’s largest sake selection.

2. **Hokusai**
   8400 Wilshire Blvd, Beverly Hills, CA, 90211
   Chic elegance and modern Zen style surround Japanese French this paean to haute cuisine and stylized sushi.

3. **Sushi Sasabune**
   12400 Wilshire Blvd Ste 150, Los Angeles, CA, 90025
   Sushi is the singular star at this Zen Westside palace that bows only to the royalty of chef and fish.

4. **Sushi Roku**
   8445 W 3rd St, Los Angeles, CA, 90048
   High fashion, rock and roll and Hollywood buzz converge over innovative sushi.

---

**Introduction** • **Approach** • **Evaluation** • **Related Work** • **Conclusion**
EXPLOITING WRAPPER LIBRARIES

Wrapper Library: Karma lists all the available wrappers on the local machine.

Enter values for input parameters in yellow column to execute wrapper!
SOURCE MODELING

- Karma automatically generates the semantic types of each attribute to learn the underlying model of the data source
- Supervised machine learning techniques are used to generate a set of patterns for each semantic type from training data

<table>
<thead>
<tr>
<th>Initial Type</th>
<th>Manually label the data with the correct semantic type to train Karma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1</td>
<td>PR-String</td>
</tr>
<tr>
<td>StreetAddress</td>
<td>PR-String</td>
</tr>
<tr>
<td>2353 Portland St.</td>
<td>Los Angeles</td>
</tr>
<tr>
<td>643 Orchard Ave.</td>
<td>Burbank</td>
</tr>
<tr>
<td>417 Glennmoor Cir.</td>
<td>Milpitas</td>
</tr>
<tr>
<td>6472 Hawthorne Blvd</td>
<td>Pasadena</td>
</tr>
<tr>
<td>325 Abbot Ave.</td>
<td>Santa Clara</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source 1</th>
<th>Source 2</th>
<th>Source 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreetAddress</td>
<td>City</td>
<td></td>
</tr>
<tr>
<td>2353 Portland St.</td>
<td>Los Angeles</td>
<td></td>
</tr>
<tr>
<td>543 Orchard Ave.</td>
<td>Burbank</td>
<td></td>
</tr>
<tr>
<td>417 Glennmoor Cir.</td>
<td>Milpitas</td>
<td></td>
</tr>
<tr>
<td>6472 Hawthorne Blvd</td>
<td>Pasadena</td>
<td></td>
</tr>
<tr>
<td>325 Abbot Ave.</td>
<td>Santa Clara</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source 1</th>
<th>Source 2</th>
<th>Source 3</th>
<th>Source 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR-Address</td>
<td>PR-String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 W Woodruff Ave</td>
<td>Evac Center 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12116 Hallwood Dr</td>
<td>Evac Center 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4005 Marshall St</td>
<td>Evac Center 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>131 W Cypress Ave</td>
<td>Evac Center 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12021 Exline St</td>
<td>Evac Center 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>857 Chapea Rd</td>
<td>Evac Center 9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the new data is imported of same type, Karma automatically labels it correctly
LEARNING SEMANTIC TYPES

✓ Idea: Learn a model of the content of data and use it to recognize new examples

<table>
<thead>
<tr>
<th>Person</th>
<th>Address</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>E Lewis</td>
<td>3518 Hilltop Rd</td>
<td>(419) 531 - 0504</td>
</tr>
<tr>
<td>Andrew Lewis</td>
<td>3543 Larchmont Pkwy</td>
<td>(518) 474 - 0593</td>
</tr>
<tr>
<td>C. S. Lewis</td>
<td>555 Willow Run Dr</td>
<td>(612) 578 - 0590</td>
</tr>
<tr>
<td>Carmen Jones</td>
<td>355 Morgan Ave N</td>
<td>(612) 522 - 0533</td>
</tr>
<tr>
<td>John Jones</td>
<td>3574 Brookside Rd</td>
<td>(555) 531 - 9566</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>State_prov</th>
<th>Postal_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>64325-3000</td>
</tr>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>64356</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>8422</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>8435</td>
</tr>
<tr>
<td>Omaha</td>
<td>NE</td>
<td>52456-6444</td>
</tr>
</tbody>
</table>

### Background knowledge

- **Idea:** Learn a model of the content of data and use it to recognize new examples.

### Patterns

<table>
<thead>
<tr>
<th>:FullName:</th>
<th>:StreetAddress:</th>
<th>:Telephone:</th>
</tr>
</thead>
<tbody>
<tr>
<td>E Lewis</td>
<td>3518 Hilltop Rd</td>
<td>(419) 531 - 0504</td>
</tr>
<tr>
<td>Andrew Lewis</td>
<td>3543 Larchmont Pkwy</td>
<td>(518) 474 - 0593</td>
</tr>
<tr>
<td>C. S. Lewis</td>
<td>555 Willow Run Dr</td>
<td>(612) 578 - 0590</td>
</tr>
<tr>
<td>Carmen Jones</td>
<td>355 Morgan Ave N</td>
<td>(612) 522 - 0533</td>
</tr>
<tr>
<td>John Jones</td>
<td>3574 Brookside Rd</td>
<td>(555) 531 - 9566</td>
</tr>
</tbody>
</table>

- **Location:**
- **State:**
- **Zipcode:**
DATA CLEANING

• Karma performs the data cleaning by learning and applying the transformation rules that are learned from examples.

<table>
<thead>
<tr>
<th>Initial data source</th>
<th>User provides example</th>
<th>Data source after cleaning</th>
<th>Karma learns a transformation rule and applies to remaining data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source1</td>
<td>Source2</td>
<td>PR-String</td>
<td>City</td>
</tr>
<tr>
<td>PR-String</td>
<td>City</td>
<td>Coordinator_Name</td>
<td></td>
</tr>
<tr>
<td>City of Agoura Hills</td>
<td>Carol Tubellis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City of Alhambra</td>
<td>Bat. Chief Ray Mosack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City of Arcadia</td>
<td>Chief Dave Odel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City of Artesia</td>
<td>Madalena Galindo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City of Duarte</td>
<td>Brian Villalobos</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City of El Monte</td>
<td>Ralph Nuneez</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source1</td>
<td>Source2</td>
<td>Cleaning Table</td>
<td>Source1</td>
</tr>
<tr>
<td>PR-String</td>
<td>City</td>
<td>Data Type</td>
<td>User Defined Values</td>
</tr>
<tr>
<td>City of Agoura Hills</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
<tr>
<td>City of Alhambra</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
<tr>
<td>City of Arcadia</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
<tr>
<td>City of Artesia</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
<tr>
<td>City of Duarte</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
<tr>
<td>City of El Monte</td>
<td>City</td>
<td>User Defined Values</td>
<td></td>
</tr>
</tbody>
</table>

Introduction • Approach • Evaluation • Related Work • Conclusion
# DATA CLEANING: PREDEFINED TRANSFORMATIONS

## Predefined Rules

<table>
<thead>
<tr>
<th>description</th>
<th>number of reviews</th>
<th>suggest</th>
<th>user defined</th>
<th>final</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upscale yet...</td>
<td>31 Reviews</td>
<td></td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>Intimate an...</td>
<td>3 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chic eleganc...</td>
<td>30 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentic Ja...</td>
<td>66 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High fashion...</td>
<td>62 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No fuss, jus...</td>
<td>25 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inventive ro...</td>
<td>38 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The MOCA o...</td>
<td>49 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Burbank res...</td>
<td>29 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rustic Japa...</td>
<td>96 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stellar sushi...</td>
<td>49 Reviews</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

31 Reviews $\rightarrow$ 31

**Subset Rule:**

$$(s_1 s_2 ... s_k) \rightarrow (d_1 d_2 ... d_t) \land
(k \leq t) \land
s_i \in \{d_1, d_2, ..., d_t\} \land
\forall d_i \neq d_j$$

- Predefined Rules
DATA INTEGRATION

- Karma discovers the related sources by detecting and ranking associations based on the common attribute names and matching semantic types.
- Karma suggests potential joins between the current data sources in the form of column completions.
USER SELECTS FROM COLUMN COMPLETIONS

Karma suggests the possible column completions in a drop down list.

MySQL Database loaded as another source in Karma.

Karma executes the join query once the user selects an option.
DATA VISUALIZATION

• Visualization by demonstration approach
  – The user demonstrates to Karma the kind of visualization desired for the data specified through examples using a drag and drop mechanism
DATA VISUALIZATION

Karma currently supports four types of visualization formats:

1. **Chart Format**: Useful for visualizing numerical statistics, time based events etc.

   ![Chart Format Example]

2. **Paragraph Format**: Useful for visualizing descriptive text data such as Wikipedia definitions.

   ![Paragraph Format Example]
DATA VISUALIZATION

3. List Format: Useful for visualizing information in a bulleted list such as list of summarized news articles

4. Table Format: Useful for visualizing information that is best presented in a row-and-column format such as numerical values etc
RESULTS CAN BE PUBLISHED IN MULTIPLE FORMATS

- Karma lets you export your final mashup in variety of formats:
  - HTML Page
  - Database table
  - KML Layer
  - XML File
  - CSV Text File
AUTOMATICALLY FINDS GEOSPATIAL REFERENCES

- Final mashup output in HTML web page format:
  - Karma identifies geospatial information in the current data with the help of geographic semantic types such as PR-Address, PR-Latitude etc
  - The Google geocoding service is used to find the coordinates for a given address
  - Karma uses the coordinates information to place the markers in the final mashup

<table>
<thead>
<tr>
<th>Source1</th>
<th>Source2</th>
<th>Source3</th>
<th>Source4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR-String</td>
<td>PR-Address</td>
<td>PR-City</td>
<td>PR-String</td>
</tr>
<tr>
<td>Evac Center 1</td>
<td>311 W Woodruff Ave</td>
<td>Arcadia</td>
<td>Chief Dave Odell</td>
</tr>
<tr>
<td>Evac Center 3</td>
<td>12116 Rainbow Dr</td>
<td>El Monte</td>
<td>Ralph Nunez</td>
</tr>
<tr>
<td>Evac Center 4</td>
<td>8805 Marshall St</td>
<td>Rosemead</td>
<td>Donna Wagner</td>
</tr>
<tr>
<td>Evac Center 7</td>
<td>131 W Cypress Ave</td>
<td>Montebello</td>
<td>Dave Durbin</td>
</tr>
<tr>
<td>Evac Center 6</td>
<td>12021 Esmere Dr</td>
<td>El Monte</td>
<td>Ralph Nunez</td>
</tr>
<tr>
<td>Evac Center 9</td>
<td>867 Chapea Rd</td>
<td>Pasadena</td>
<td>Lisa Carverian</td>
</tr>
</tbody>
</table>

Options to publish mashup as HTML web page

Potential geographic information

Introduction • Approach • Evaluation • Related Work • Conclusion
CONSTRUCTS A MAP WITH USER-DEFINED LAYOUT

• Final mashup as a HTML web page:

![Map with user-defined layout](image)
RESULTS CAN BE EXPORTED AS KML

- Final mashup output as a KML layer

Options to publish mashup as KML layer
KML LAYERS CAN BE OPENED IN GOOGLE EARTH

The generated KML layer can be viewed in a GIS software such as Google Earth.
RESULTS CAN BE STORED IN A DB

- The final mashup data can also be saved into a database table by providing the details about the database location, username, and password, etc., in Karma.
EVALUATION

• Baseline: A combination of Dapper/Pipes
• Claims:
  1. Users with no programming experiences can build all four Mashup types.
  2. Karma takes less time to complete each subtask and scales better as the tasks get harder
  3. Overall, the user takes less time to build the same Mashup in Karma compared to Dapper/Pipes
• Users:
  – Programmers (20)
  – Non-programmers (3)
Familiarization
- Programmers (2 assignments on DP)
- Review Package
- 30 minutes tutorial

Practice
- 2-3 tasks using Karma

Test (3 tasks)
- Programmers: Alternating between Karma vs. DP for each task
- Non Programmers: use only Karma
- Screen are recorded using video capture software

<table>
<thead>
<tr>
<th>Task2</th>
<th>Dapper/Pipes</th>
<th></th>
<th></th>
<th></th>
<th>Total</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>E</td>
<td>M</td>
<td>C</td>
<td>I</td>
<td></td>
<td>E</td>
<td>M</td>
<td>C</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>No.1</td>
<td>4:38</td>
<td>0:22</td>
<td>2:45</td>
<td>1:15</td>
<td>9:00</td>
<td>1:26</td>
<td>0:43</td>
<td>0:43</td>
<td>0:00</td>
<td>2:52</td>
</tr>
<tr>
<td>No.2</td>
<td>1:35</td>
<td>0:12</td>
<td>3:30</td>
<td>0:12</td>
<td>5:29</td>
<td>0:50</td>
<td>0:57</td>
<td>0:57</td>
<td>0:00</td>
<td>2:44</td>
</tr>
<tr>
<td>No.3</td>
<td>*5:00</td>
<td>0:25</td>
<td>*5:00</td>
<td>*5:00</td>
<td>15:25</td>
<td>2:52</td>
<td>1:00</td>
<td>3:00</td>
<td>0:00</td>
<td>5:52</td>
</tr>
<tr>
<td>No.4</td>
<td>4:49</td>
<td>0:17</td>
<td>3:29</td>
<td>0:38</td>
<td>9:14</td>
<td>1:26</td>
<td>0:48</td>
<td>1:03</td>
<td>0:00</td>
<td>3:18</td>
</tr>
<tr>
<td>No.5</td>
<td>*5:00</td>
<td>0:29</td>
<td>1:44</td>
<td>1:16</td>
<td>8:29</td>
<td>1:43</td>
<td>0:45</td>
<td>1:20</td>
<td>0:00</td>
<td>3:48</td>
</tr>
<tr>
<td>No.6</td>
<td>*5:00</td>
<td>0:20</td>
<td>*5:00</td>
<td>*5:00</td>
<td>15:20</td>
<td>2:07</td>
<td>0:30</td>
<td>0:50</td>
<td>0:00</td>
<td>3:27</td>
</tr>
</tbody>
</table>

5 minute cut off time
EVALUATION: TASKS

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Mashup Type</th>
<th>Data Extraction</th>
<th>Source Modeling</th>
<th>Data Cleaning</th>
<th>Data Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 (1 source)</td>
<td>Moderate</td>
<td>Simple</td>
<td>Difficult</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>2,3 (union+form)</td>
<td>Difficult</td>
<td>Simple</td>
<td>Simple</td>
<td>Union (simple)</td>
</tr>
<tr>
<td>3</td>
<td>4 (join 2 sources)</td>
<td>Simple</td>
<td>Simple</td>
<td>N/A</td>
<td>Join (difficult)</td>
</tr>
</tbody>
</table>

- Claim 1: Users with no programming experiences can build all four Mashup types
- Claim 2: When the Mashup subtask is difficult, Karma takes less time to complete that subtask
- Claim 3: Overall, the user takes less time to build the same Mashup in Karma compared to Dapper/Pipes
Claim 1: Users with no programming experiences can build all four Mashup types
EVALUATION: NON-PROGRAMMERS

The Result from Non-Programmer Subjects

- Task 1
- Task 2
- Task 3

Time (minute)

- Subject1
- Subject2
- Subject3

Task
Claim 2: Karma takes less time to complete each subtask
As the extraction task gets more difficult, Dapper/Pipes takes
- longer
- more subjects failing to complete the task (11% for moderate and 25% for difficult)
EVALUATION: SOURCE MODELING

- Karma performed worse in task 1 and tasks 2
  - only 30 sec difference
  - subjects take times selecting attributes
  - the saving will be realized in the data integration step.
- Karma performed better in task 3 because it can automatically identify the attribute

Dapper/Pipes  Karma
EVALUATION: DATA CLEANING

- Karma performed better in both tasks
- When the cleaning task gets harder, more subjects are failing in Dapper/Pipes (35% for simple and 83% in hard)
Because of the table structure, subjects can specify union indirectly by dropping data into the right cell.

The time spent in source modeling step allows Karma to suggest the linking source.

Dapper/Pipes: 30% fail in the union case and 95% fail in the join case.
Claim 3: Overall, the user takes less time to build the same Mashup in Karma compared to Dapper/Pipes
EVALUATION: OVERALL

Task 1: Overall

Task 2: Overall

Task 3: Overall

Dapper/Pipes  Karma

Introduction  • Approach  • Evaluation  • Related Work  • Conclusion
EVALUATION: AVERAGE

Time comparison average over three tasks

Problem Type

Extraction  Source Modeling  Cleaning  Integration  Overall

Dapper/Pipes  Karma

2.22x  0.67x  4.16x  6.49x  3.32x
## RELATED WORK: MASHUP BUILDING TOOLS

<table>
<thead>
<tr>
<th>System</th>
<th>Data Retrieval</th>
<th>Source Modeling</th>
<th>Data Cleaning</th>
<th>Data Integration</th>
<th>Mashup Type Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT’s Simile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MIT’s Pot Luck</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,3,4</td>
</tr>
<tr>
<td>Dapper</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,4</td>
</tr>
<tr>
<td>Yahoo’s Pipes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,3</td>
</tr>
<tr>
<td>MS’s Popfly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,4</td>
</tr>
<tr>
<td>CMU’s Marmite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,4</td>
</tr>
<tr>
<td>Intel’s Mashmaker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>Google MyMap</td>
<td>DOM</td>
<td>Database</td>
<td>PBD</td>
<td>PBD</td>
<td>1,2</td>
</tr>
<tr>
<td>Agent Wizard</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,3,4</td>
</tr>
<tr>
<td>Cards</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,4</td>
</tr>
<tr>
<td>Karma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,2,3,4</td>
</tr>
</tbody>
</table>

- Early work. Focus on DOM, too basic
- RDF / Manually specify data int
- Mainly focus on extraction / linear
- Widgets
- Fancier UI/ more widgets
- Fewer Widgets / Confusion on workflow
- Require an expert
- Create points on Map
- Q/A approach / linear / scalability
- Tuple = card. Drawing links for relations

1: Extraction, 2: Union, 3: Form-based Interaction, 4: Join
RELATED WORK: DATA EXTRACTION

• Automatic extraction: table and lists only
  – RoadRunner (exploit HTML structure) [Crescenzi et al., 2001]
  – Adel (grammar induction to detect rows) [Lerman+ 2001]
  – VisualWeb (OCR technique to detect tables) [Gatterbauer+ 2007]

• Semi-Automatic: require more label examples
  – WIEN (inductive – less expressive than stalker) [Kushmerick 1997]
  – Stalker (Cotesting) [Muslea+ 1999]
  – SoftMealy (finite state transducer) [Hsu 1998]
  – WHISK (rigid format, exact delimiter) [Soderland 1998]

• DOM: rely on well-formed HTML and less labeling
  – Simile [Huynh+ 2005]
  – Dapper
  – Interactive Wrapper Generation (ML + prediction on DOM)[Irmak+ 2006]
  – PLOW (add natural language) [Allen+ 2007]
  – Cards [Dontcheva+ 2007]
  – Karma [Tuchinda+ 2008]
RELATED WORK:
SOURCE MODELING

• 1:1 mapping, N:M mapping
  – Schema-level match
    • TranScm [Milo+ 98]
    • DIKE [Palopoli+ 99]
    • Artemis [Castano+ 01]
    • Delta [Clifton+ 97]
  – +Instance-based matcher
    • SemInt [Li 00]
    • LSD [Doan 01]
    • ILA [Etzioni 95]
    • iMapp [Dhamanka 04]
    • Clio (interactive) [Ling 01]
    • Inducing Source Description [Carman 07]
• Karma leverages existing techniques to narrow candidate matches
  – String Similarities [Cohen+ 2003]
RELATED WORK:
DATA CLEANING

• Commercial Tools: Focus on writing transformation
  – ACR/Data, Migration Architect [Chaudhuri+ 1997]

• Discrepancy Detection: Use as a stepping stone for record linkage and cleaning system
  – Levenshtein distance [Needleman+ 70]
  – Vector based [Baeza-Yates+ 99]
  – EM [Ristad+ 98]
  – SVM [Bilenko+ 03]

• Record linkage & cleaning systems: Focus on ranking [Winkler 06]
  – Fuzzy Match [Chaudhuri+ 03]
  – Apollo [Michalowski+ 05]
  – Phoebus [Michelson+ 07]
  – Potter’s wheel [Raman+ 01]

• Karma
  – Gains reference sources through source modeling process
  – Provides predefined transformations
RELATED WORK: DATA INTEGRATION

- **Universal Relation**: Make it easier to formulate the query but users still need to formulate the query [Ullman 1980, 1988]
- **Query by example**: Need to know which data sources to use and the query may not return results
  - QBE [Zloof 1975]
- **Retrieval by formulation**: Need to understand domain model to formulate partial description
  - Helgon [Fischer 1989], RABBIT [Williams 1982]
- **Graphical Query Language**: Users still need to navigate through sources (graphs)
- **Question-Answering Techniques**: Understanding about database operations required
  - Agent Wizard [Tuchinda+ 2004]
- **Interactive Schema/data integration**: Understanding about source schema required
  - Clio [Ling 01]
- Karma is based on **Programming by Demonstration** [Cyper 2001; Lau2001]
CONCLUSION

• Mashups are a fast growing area
  – Need an efficient way to for casual web users to build them

• Contributions
  – A PBD approach that uses a single table for building a Mashup
  – An integrated approach that solves the various Mashup building issues
  – A query formulation technique that allows users to specify examples to build complicated queries

• Evaluated the validity of the Karma approach
  – Subjects were able to complete Mashup building tasks in Karma
  – The overall improvement is at least a factor of 3.5
FUTURE WORK

• Learn and generalize over the task
  – Store the integration plan so that it can be reexecuted on current data
• Support the integration of geospatial data types (i.e., vector layers, raster layers)
• Improve the techniques for automatic source modeling
• Learn new transformations from examples for data cleaning

• **Interactive data integration through smart copy & paste.** Zachary G. Ives, Craig A. Knoblock, Steven Minton, Marie Jacob, Partha Pratim Talukdar, Rattapoom Tuchinda, Jose Luis Ambite, Maria Muslea, and Cenk Gazen, *Fourth Biennial Conference on Innovative Data Systems Research (CIDR)*, 2009.

• **Building mashups by example.** Rattapoom Tuchinda, Pedro Szekely, and Craig A. Knoblock. *Proceedings of the 2008 International Conference on Intelligent User Interfaces*, 2008