Building Mashups by Example
Tuchinda, Szekely, and Knoblock

CS690 – Week 9 Slides

Jordan Osecki

Summary

- Mashup: A web application that integrates data from multiple web sources to provide a unique service.
- Involves the problems of data retrieval, source modeling, data cleaning, data integration, and data visualization.
- Existing works relies on a widget paradigm, but it is hard to locate the right widget and customization usually requires programming knowledge.
- The Karma system that they have developed is an approach of a unified interactive framework that requires no widgets, but rather allows users with no programming experience to easily create Mashups by example.

Summary

- Data retrieval: Extracting data from web pages into a structured data source.
- Source modeling: Assigning the attribute name for each data column for relationship purposes.
- Data cleaning: Transforming extracted data into an appropriate format.
- Data integration: Combining two or more data sources together.
- Data visualization: Displaying the final data generated by the user.

Example – Karma, GUI

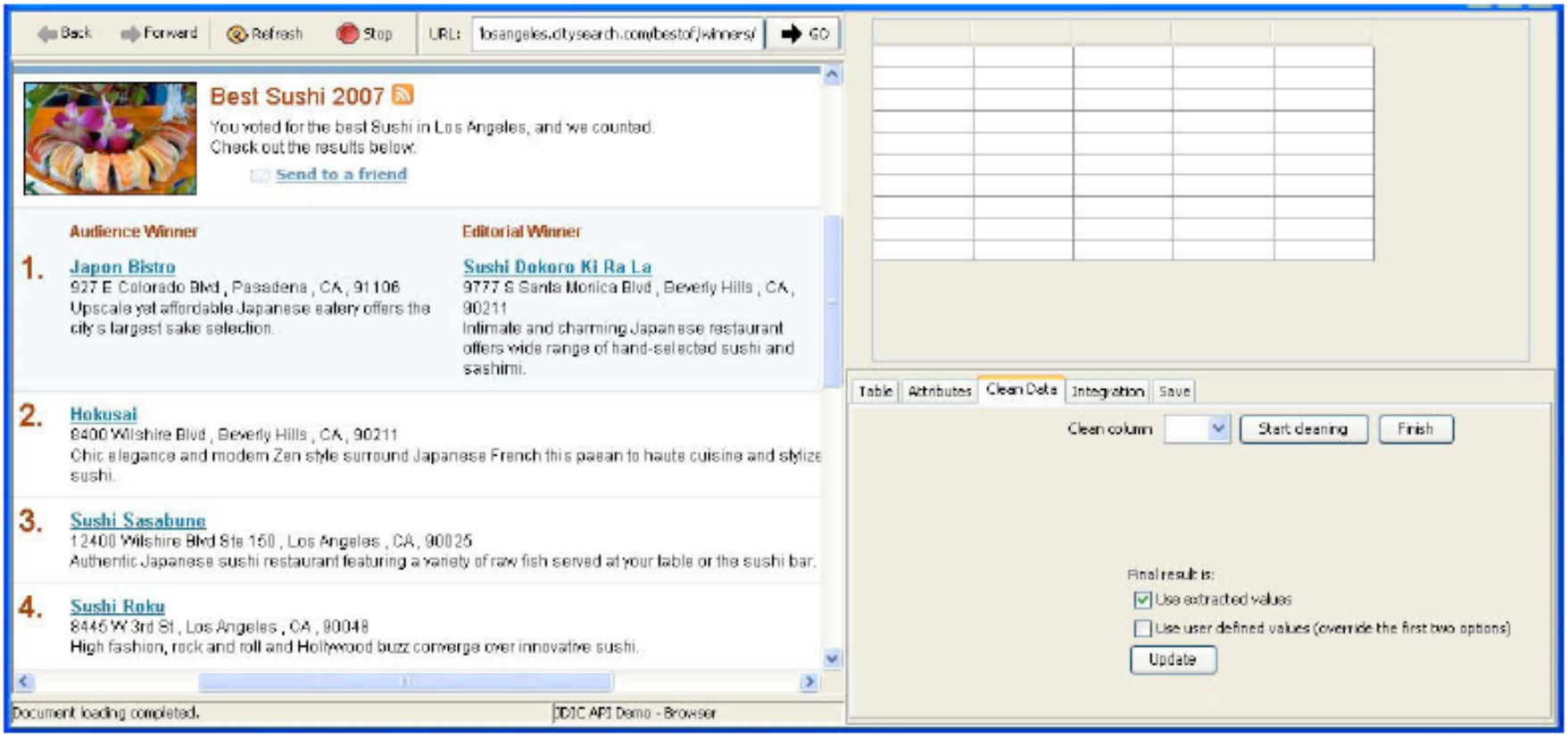
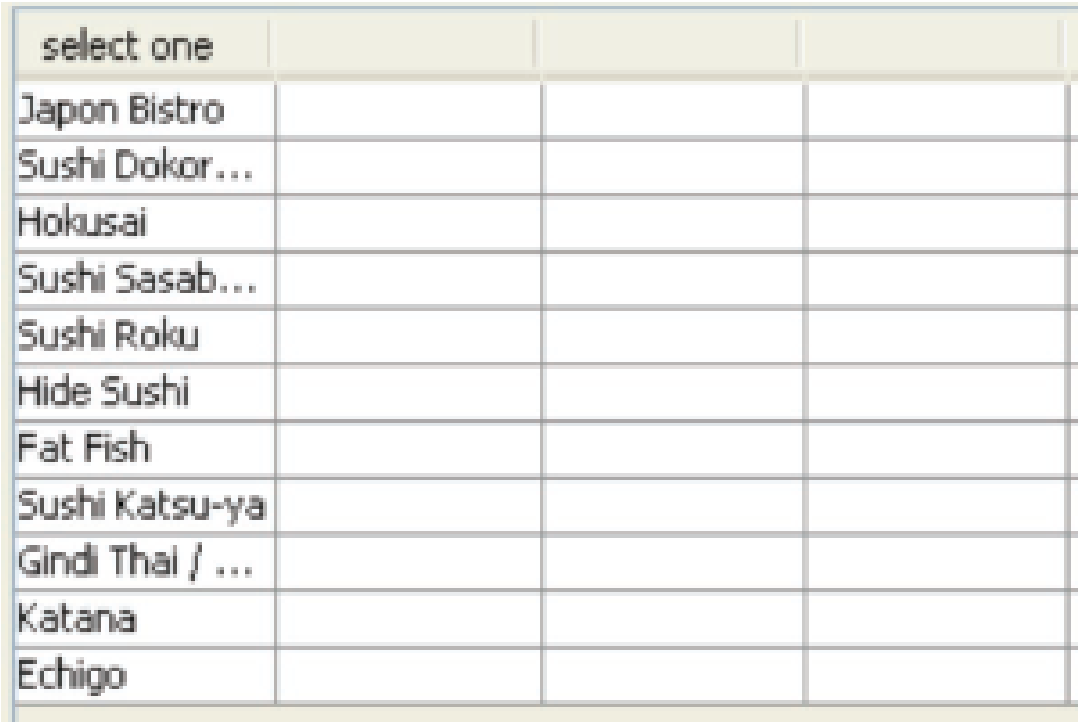


Figure 2. The interface of Karma. The left window is an embedded web browser. The top right window contains a table that a user would interact with. The lower right window shows options that the user can select to get into different modes of operation.

Example – Karma, Data Retrieval



select one			
Japon Bistro			
Sushi Dokor…			
Hokusai			
Sushi Sasab…			
Sushi Roku			
Hide Sushi			
Fat Fish			
Sushi Katsu-ya			
Gindi Thai / …			
Katana			
Echigo			

Figure 3: By dragging "Japon Bistro" into the first row, Karma automatically fills the rest of the column

Example – Karma, Source Modeling

select one	address	select one	select one
Japon Bistro	927 E Color...	Upscale yet ...	28 Reviews
Sushi Dokor...	9777 S Sant...	Intimate an...	3 Reviews
Hokusai	8400 Wilshir...	Chic eleganc...	30 Reviews
Sushi Sasab...	12400 Wilshi...	Authentic Ja...	66 Reviews
Sushi Roku	8445 W 3rd ...	High fashion...	62 Reviews
Hide Sushi	2040 Sawtel...	No fuss, jus...	25 Reviews
Fat Fish	616 N Rober...	Inventive ro...	38 Reviews
Sushi Katsu-ya	11680 Vent...	The MOCA o...	49 Reviews
Gindi Thai / ...	4017 W Riv...	Burbank res...	29 Reviews
Katana	8439 W Sun...	Rustic Japa...	96 Reviews
Echigo	12217 Sant...	Stellar sushi ...	49 Reviews

Figure 4: The user extracts the whole list by dragging only four values into the first row of the table.

Example – Karma, Data Cleaning

s	description	number o...	user defi...	final
lora...	Upscale yet ...	28 Reviews	28	
ant...	Intimate and...	3 Reviews		
shir...	Chic eleganc...	30 Reviews		
ilshi...	Authentic Ja...	66 Reviews		
3rd ...	High fashion...	62 Reviews		
vtell...	No fuss, just...	25 Reviews		
ber...	Inventive rol...	38 Reviews		
ntu...	The MOCA o...	49 Reviews		
Rive...	Burbank rest...	29 Reviews		
Sun...	Rustic Japan...	96 Reviews		
anta...	Stellar sushi ...	49 Reviews		

Figure 5: Karma in the cleaning mode. The user can specify the clean result and Karma will try to induce the cleaning transformation.

Methodology – Karma, Data Retrieval

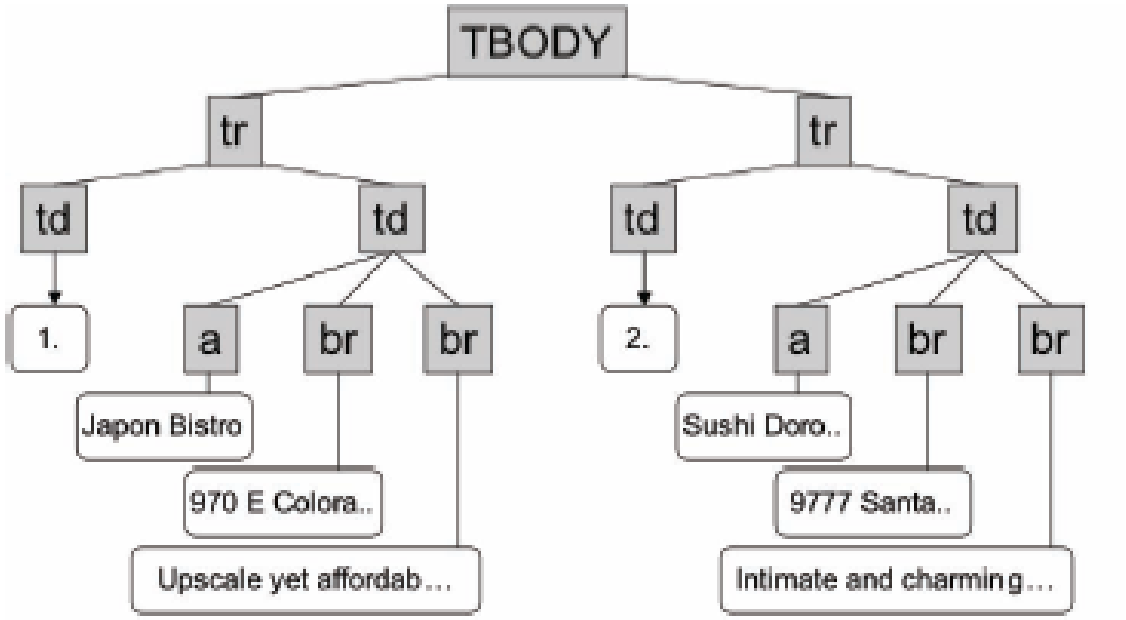


Figure 6. A simplified DOM tree that represents the best restaurant page in the motivating example. The gray nodes represent the HTML tags, while the white nodes represent the data embedded within those tags.

Methodology – Karma, Source Modeling

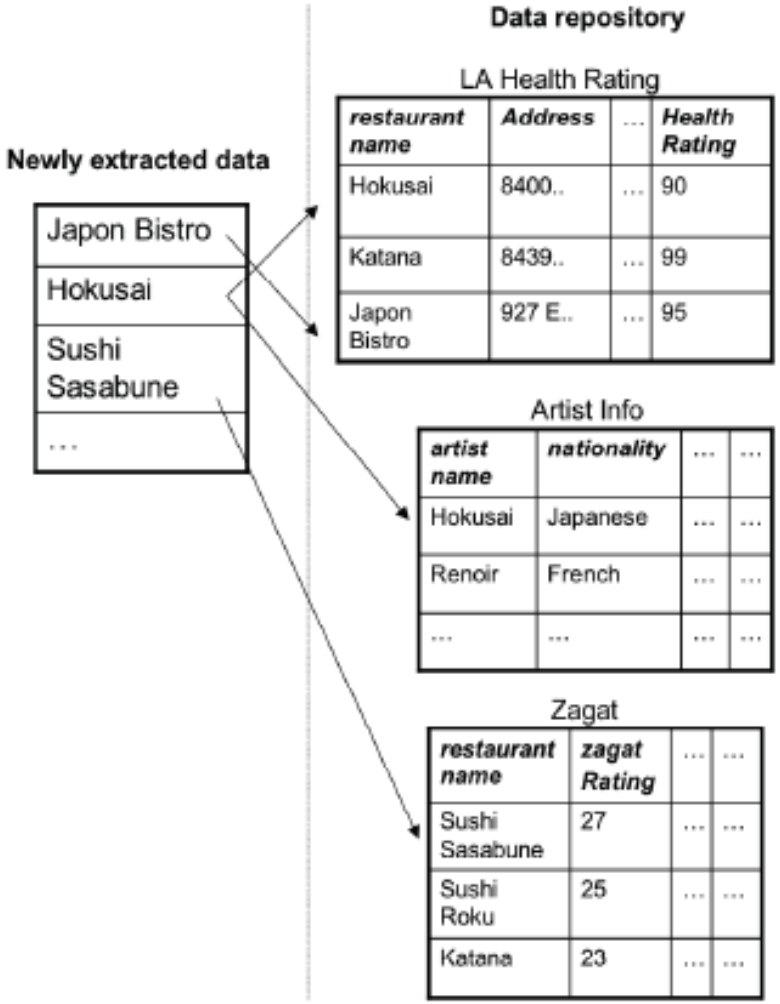


Figure 7. A view of the overlapping between newly extracted data and existing data in the repository.

Methodology – Karma, Data Cleaning

In Karma, we use a cleaning by example approach that lets users specify how the cleaned data should look like. Karma then will try to induce the cleaning transformation rule. We adapt our cleaning by example approach from Potter's Wheel [13]. Given a string of data, we first break the string into different tokens based on the following data types: <word>, <number>, <blankspace>, and <symbol>. For example, "jones, norah" would correspond to {<*word1*>, <*symbol*>, <*blankspace*>, <*word2*>}. Once the user specifies the cleaned result, for example "Norah Jones", the user-defined data will also be broken into different tokens {<*word1*>, <*blankspace*>, <*word2*>}. Karma then tries to determine the transformation as follows:

First locate tokens with the same value between the *O* (original) and *D* (user-defined) set, and determine if the ordering has been swapped or not. If yes, add the swap instruction for that token into the set *T*, which stores all transformation sub-rules.

For each token in *O* that cannot be matched to *D*, apply a set of pre-defined transformations *S* and see if the result of the transformation can be matched to any value in D.

If no, then discard that token from O. If yes, add the pre-defined transformation and the swap instruction, if any, to *T*.

Methodology – Karma, Data Integration

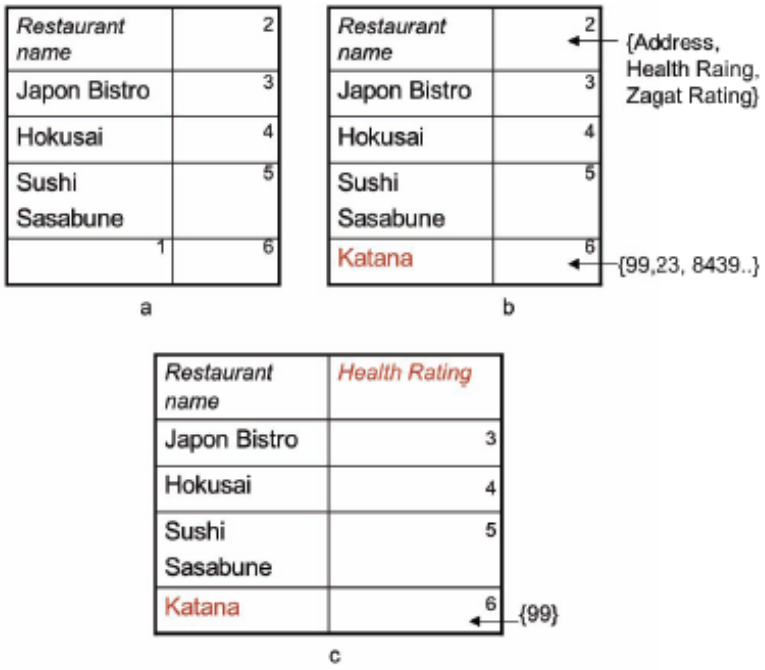


Figure 8 shows how the user can integrate new data with existing data through examples. When the user selects more examples, the table becomes more constrained. The value 1-6 designated empty cells.

Figure 9. Selecting Katana in cell 1 limits the choices in other cells, such as cell 6 and cell 2, through the horizontal constraint.

On the other hand, cell 2 is only limited to three attributes (shaded in the attribute rows in Figure 9) since these attributes come from sources that have "Restaurant name"

Rough Architecture Diagram

CS690 – Week 9 Slides

Jordan Osecki

Core Component Diagram