A System for Efficient Cleaning and Transformation of Geospatial Data Attributes (Demo Paper)

Yao-Yi Chiang
University of Southern California
Spatial Sciences Institute
Los Angeles, CA 90089, USA
yaoyic@usc.edu

Bo Wu
University of Southern California
Department of Computer Science and Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
bowu@isi.edu

Akshay Anand
Ketan Akade
University of Southern California
Department of Computer Science and Spatial Sciences Institute
Los Angeles, CA 90089, USA
[akshayan, akade]@usc.edu

ABSTRACT

A significant challenge in handling geographic datasets is that the datasets can come from heterogeneous sources with various data qualities and formats. Before these datasets can be used in a Geographic Information System (GIS) for spatial analysis or to create maps, a typical task is to clean the attribute data and transform the data into a unified format. However, conventional GIS products focus on manipulating the spatial component of geographic features and only offer basic tools for editing the attribute data (e.g., one row at a time). This limits the capability for handling large datasets in a GIS since manually editing and transforming attribute data between different formats does not scale for thousands of geographic features. In this demo, we present ArcKarma, which is built on our previous work on data transformation to efficiently clean and transform data attributes in a GIS. ArcKarma generates transformation programs from a few user-provided examples and applies the programs to transform individual attribute columns to the desired formats. We show that ArcKarma produces accurate results and eliminates the need for laborious manual data cleaning and scripting tasks.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous—*Geographic Information Systems*; H.2.8 [**Database Management**]: Database Applications—*Spatial Databases and Geographic Information Systems*

General Terms

Algorithms, Design, Human Factors

SIGSPATIAL '14 Dallas, Texas USA

Keywords

Geographic information system, data cleaning, data transformation

1 INTRODUCTION

A key problem facing geographic information system (GIS) users is that the geographic datasets used in a project very often come from multiple sources and the attribute data of these datasets can be in a variety of data formats (e.g., telephone numbers stored as "213-740-2311" or "(213) 740-2311"). In addition, even individual geographic features within one data source can have different formats for an attribute because of the datasets were collected at different time periods or a consistent attribute format was not enforced during data entry. While a GIS supports a wide range of operations to integrate the spatial component of geographic datasets (e.g., data overlay and spatial join), linking and joining geographic features using attribute data generally relies on manual data preparation steps or requires task dependent programming work to ensure attributes of geographic features are in a unified format. These data preparation steps can be laborious and require scripting or programming skills.

This paper presents a system called *A*rcKarma, which is an interactive data transformation tool integrated with a conventional GIS software (Esri ArcMap). ArcKarma employs a training-by-example strategy to generate transformation programs and automatically transforms attribute data to a unified format. ArcKarma does not require expert knowledge in scripting or programming. In the reminder of this paper, Section 2 presents the ArcKarma user interface and workflow, Section 3 describes the ArcKarma data cleaning and transformation algorithm, Section 4 presents related systems, and Section 5 discusses the future work.

2 ArcKarma – WORKFLOW AND USER INTERFACE

Figure 1 shows the ArcKarma architecture and a typical use case. The "Well Information" layer in this ArcMap project contains an attribute table of oil well information for maintaining the records of leased wells from a petroleum production company. This list is an integration of well data from

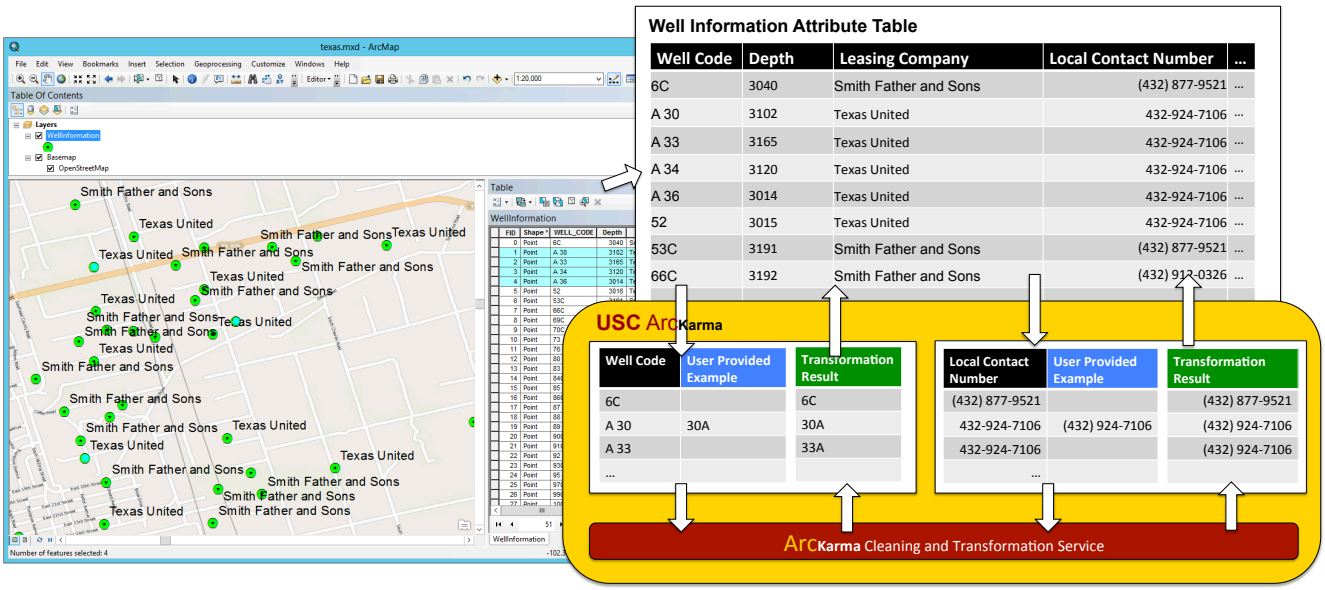

Figure 1: ArcKarma workflow

multiple sources that contain data created at different time periods. The spatial component of the "Well Information" layer contains the well locations (in this case, point data) and the attribute table stores auxiliary information about individual wells. Before the year 2003, the petroleum company used one alphabetic character, a space, and a numeric string to identify the well types (e.g., "A 30") and the current system uses a new naming convention of one or more numeric characters followed by an alphabetic character, e.g., "6C"). As the information of the wells leased from the company Texas United exists before the system change, their well codes follow the old naming convention. In addition, the previous system user interface does not enforce domain integrity during data entry and hence the contact numbers can be in various formats and do not follow the current standard.

To unify the data format for both the "Well Code" and "Local Contact Number" attributes, the user enters example values of the desired format (the blue columns in Figure 1). ArcKarma takes the original data (the black columns in Figure 1) and the example values to invoke the cleaning and transformation service for transforming individual attribute values. Finally, the transformation results (the green columns in Figure 1) are pushed back to the 'Well Information" attribute table in ArcMap.

Figure 2 shows the ArcKarma user interface. Within the ArcMap environment, the user clicks on "Karma Transformation" and a table view shows up for the user to select the desired attribute for transformation (by clicking on the column header) (steps 1 and 2). In this example, the user selects to transform the "WELL_CODE" attribute and provides an example value of "30A" for the original value "A 30" (step 3). After the user finishes providing the transformation example(s), the user clicks on "Transform" to generate the transformation results. The user can then click on "Save" to fetch the results back to the attribute table in ArcMap (steps 4, 5, and 6). In addition to directly saving the transformation results, the user can edit, discard, or add more examples if the results are not satisfactory. During the entire process, the user stays in ArcMap and does not need to manually import/export datasets from/to different software platform.

3 ArcKarma – DATA CLEANING AND TRANSFORMATION SERVICE

The ArcKarma data cleaning and transformation service (DCTS) is an implementation of our previous work in [6] that synthesizes transformation programs using examples. This previous work is based on the programming-by-example approach developed by Gulwani [2], which defines a string transformation language that supports a restricted, but expressive form of regular expressions including conditionals and loops. In this section, we first present a sample transformation program and then describe how DCTS generates the program.

Figure 3 shows a sample transformation program generated by DCTS using an example with the original value as "A 30" and the target value as "30A". For each input raw data, the program first uses a classifier to identify the format of the input string and then selects a transformation branch to apply (i.e., the *switch label* statement). Each transformation branch contains the code that can convert the input string to a target format (e.g., *case* "$format1''$). The transformation code is essentially a concatenation of several substring expressions (e.g., $value.substring$). A substring expression can either be (1) a constant string or (2) an extraction rule describing how to extract specific string parts from raw data. The program in Figure 3 contains two substring expressions: the extraction rule (e.g., $substring(pos_3, pos_4)$) to extract uppercase alphabetic characters and the extraction rule (e.g., $substring(pos_1, pos_2)$) to extract numeric characters.

In DCTS, an extraction rule has two expressions to identify the start and end positions of a substring (e.g., the value.indexOf function). The positions can be expressed using (1) an absolute position or (2) restricted regular expressions that identify the context of the given position, which

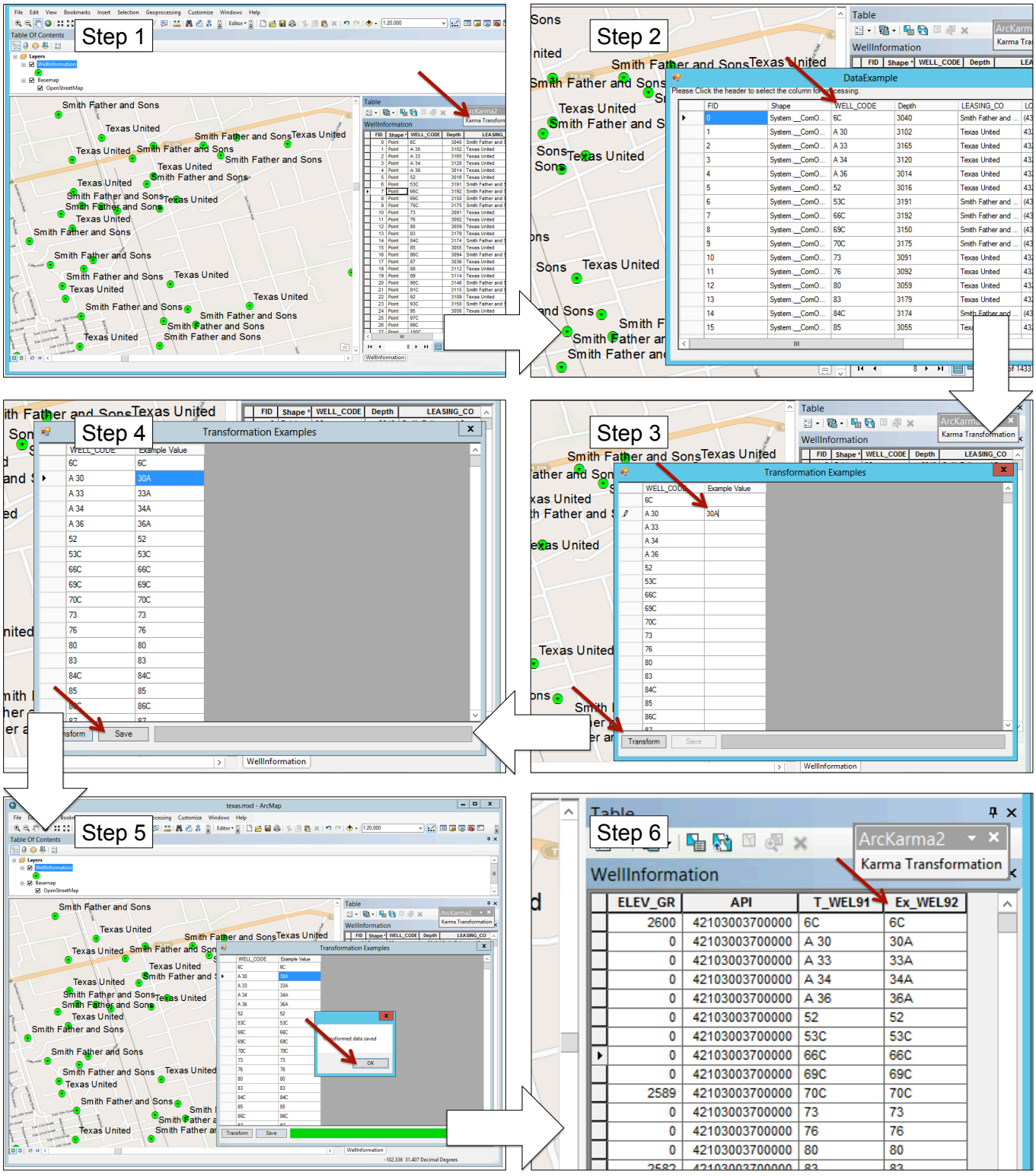

Figure 2: ArcKarma user interface

```
Transform(value)
  label = classify(value)
  switch label:
    case "format1":
      pos1 = value.indexOf('BNK', 'NUM', 1)
      pos2 = 4
      pos3 = 0
      pos4 = value.indexOf('UWRD', 'BNK', 1)
      output = value.substring(pos1,pos2)+value.substring(pos3, pos4)
    ...
    case "formatN":
      ...
  return output
```

Transformation Branch

Figure 3: Program generated by the Data transformation Module

can be represented as ("leftcxt", "rightcxt", "occ"). "leftcxt" describes the left context of the position, "rightcxt" describes the right context, and "occ" is the occurrence of the position. For example, the start position of "A" in "A 30" can be specified as an absolute position 0 (pos_3) or (START, UWRD, 1). Here, "START" represents the beginning of the raw value. "UWRD" represents an uppercase letter. Other token types are that "END" is for the end of the raw value; "LWRD" means a continuing sequence of lower letters; "NUM" refers to a continuing sequence of digits; the "BNK" means a blank space. Therefore, "(START,UWRD, 1)" means the first occurrence of a position ("occ" = 1), which is at the beginning of the raw data ("leftcxt" = "START") and has a uppercase token at its right ("rightcxt" = 'UWRD"). With a list of position expressions pos_1, pos_2, pos_3 *and* pos_4, the transformation program can specify the start and end positions of the specific substrings. The program then extracts these substrings and concatenates them to form the output.

To learn a transformation program, DCTS first separates the target values into string segments. For example, the raw data "30A" is separated into two segments "30" and "A" as shown in Figure 4. DCTS then tries to independently generate these segments by either extracting a substring from the input or inserting a new string. For instance, the "30" in the target string can be extracted from the original input. With these alignments, DCTS then identifies the locations used to extract the substrings and provides a variety of ways specifying these locations such as using absolute position or using its context.

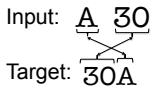

Figure 4: Learning the transformation programs

If there are multiple examples, DCTS uses an efficient algorithm to construct a version space [4] for each example and to merge them into a version space consistent with the maximum number of examples. A version space is essentially a hypothesis space that contains the programs that are consistent with the examples. However, if the examples cannot be covered by a single version space, DCTS partitions the examples and generates a version space for each partition individually. Each partition corresponds to a different input format. A conditional expression can be learned to distinguish multiple different formats. DCTS also supports loop expressions by detecting whether continuous segments of one program can be merged. To efficiently generate the transformation program from the version space, DCTS defines a partial order over the version space. This partial order helps to generate the simpler program first, which is more likely to be correct according to Occam's principle.

4 RELATED WORK

OpenRefine[1] and Potter's wheel [5] allow the user to specify string edit operations. OpenRefine is a data cleaning tool that supports regular expression style of string transformation and data layout transformation. Potter's Wheel has predefined transformation operations and users gradually build transformation programs by adding or undoing transformation operations using an interactive user interface. Many programming-by-demonstration approaches can learn edit operations by asking the user to demonstrate the editing process [1]. Lau [4] presents a system that learns from a user's edit operations to generate a sequence of text editing programs. Data Wrangler [3] is an interactive tool that uses the transformation operations defined in Potter's wheel for creating data transformation programs. In addition to supporting string level transformation, Data Wrangler also supports data layout transformation including column split, column merge, fold, and unfold. In contrast, ArcKarm only requires the user to enter a few examples of the desired data value without asking the user to demonstrate the format conversion steps.

5 DISCUSSION AND FUTURE WORK

We plan to integrate two additional capabilities from [6] to enhance ArcKarma usability: (1) recommending rows to provide examples and (2) highlight the corresponding substrings between the raw and transformed values, which helps the user to understand the applied transformation.

6 ACKNOWLEDGMENTS

This research is based upon work supported in part by the Chevron Corporation and the Center for Interactive Smart Oilfield Technologies at University of Southern California.

References

[1] A. Cypher, D. C. Halbert, D. Kurlander, H. Lieberman, D. Maulsby, B. A. Myers, and A. Turransky, editors. *Watch what I do: programming by demonstration*. MIT Press, 1993.

[2] S. Gulwani. Automating string processing in spreadsheets using input-output examples. In *POPL*, pages 317–330, 2011.

[3] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: interactive visual specification of data transformation scripts. In *CHI*, pages 3363–3372, 2011.

[4] T. Lau, S. A. Wolfman, P. Domingos, and D. S. Weld. Programming by demonstration using version space algebra. *Mach. Learn.*, pages 111–156, 2003.

[5] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB*, pages 381–390, 2001.

[6] B. Wu, P. Szekely, and C. A. Knoblock. Minimizing user effort in transforming data by example. In *IUI*, pages 317–322, 2014.

[1] http://openrefine.org