Information Systems 71 (2017) 123–136

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Rule-based spreadsheet data transformation from arbitrary to relational tables

Alexey O. Shigarov*, Andrey A. Mikhailov

Matrosov Institute for System Dynamics and Control Theory of SB RAS, 134 Lermontov st., Irkutsk 664033, Russia

ARTICLE INFO

Article history:
Received 28 June 2017
Revised 11 August 2017
Accepted 12 August 2017
Available online 14 August 2017

Keywords:
Spreadsheet data transformation
Table understanding
Table model
Table analysis
Table interpretation
Rule-based programming

ABSTRACT

The paper discusses issues of rule-based data transformation from arbitrary spreadsheet tables to a canonical (relational) form. We present a novel table object model and rule-based language for table analysis and interpretation. The model is intended to represent a physical (cellular) and logical (semantic) structure of an arbitrary table in the transformation process. The language allows drawing up this process as consecutive steps of table understanding, i. e. recovering implicit semantics. Both are implemented in our tool for spreadsheet data canonicalization. The presented case study demonstrates the use of the tool for developing a task-specific rule-set to convert data from arbitrary tables of the same genre (government statistical websites) to flat file databases. The performance evaluation confirms the applicability of the implemented rule-set in accomplishing the stated objectives of the application.

1. Introduction

Spreadsheets provide a popular way for creating and circulating arbitrary tables (e.g. cross-tabulations, invoices, roadmaps, and data collection forms). They can be considered as a general form for representing tabular data with an explicitly presented layout (cellular structure) and style (graphical formatting). For example, HTML tables presented in web pages can be easily converted to spreadsheet formats. The arbitrary tables can be a valuable data source in business intelligence and data-driven research. However, difficulties that inevitably arise with extraction and integration of the tabular data often hinder the intensive use of them in the mentioned areas.

The number of genuine tables in the Web reaches hundreds of millions [1–3]. Many of them are relational tables that can be considered as flat databases. Nevertheless, there are other popular types of tables [4–7] having layout features designed for human understanding (e.g. merged cells, footnotes, and indentations). These include about 50% of tables presented in 0.4M spreadsheets of ClueWeb09 Crawl[1] [5] and 147M (61%) of 233M web tables extracted from Common Crawl[2] [3]. They lack explicit semantics required for computer programs to interpret their layout and content.

Table understanding is to recover the missing semantics. The papers [8,9] defines the five consecutive stages of the table understanding: *detection* of a table in a document, *recognition* (segmentation) of its cellular structure, *functional* and *structural analysis* for recovering its logical structure, and *interpretation* that aspire to recover its semantics through linking its content with target schema or domain concepts.

We regard the transformation of spreadsheet tabular data (Fig. 1a) into the relational form (Fig. 1b) as a process of table understanding. In the general case, this transformation includes all the enumerated stages:

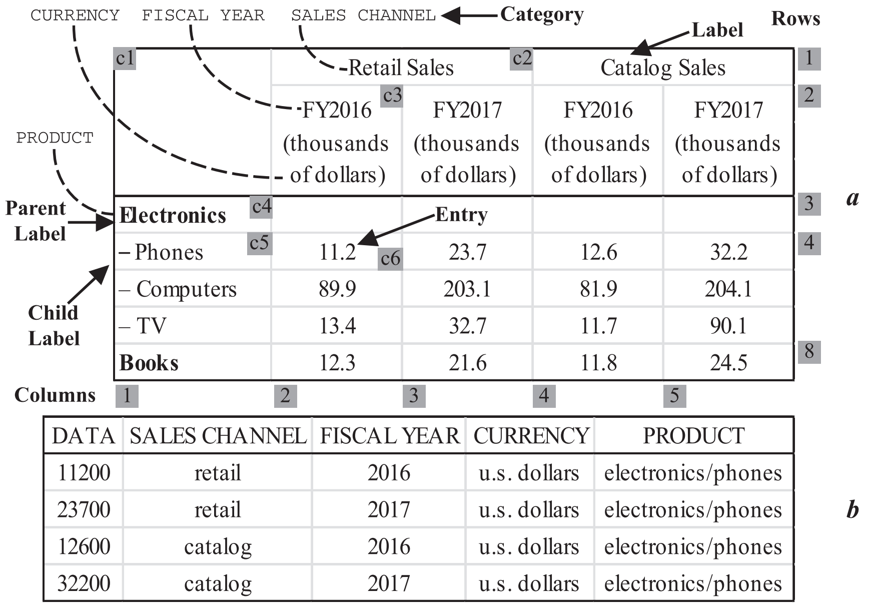
1. *Detection*. A spreadsheet document can contain several arbitrary tables surrounded by text and graphics.
2. *Recognition*. A human-readable structure of an arbitrary table can differ from its machine-readable structure presented in a spreadsheet, e.g. one logical cell can be visually composed of several physical cells through drawing their borders.
3. *Role (functional) analysis*. A spreadsheet cell stores a text, where a human can distinguish one or more data items that play some functional roles in a table (e.g. values or attributes). However, there are no spreadsheet metadata that separate data items from a cell value and determine their functional roles.
4. *Structural analysis*. A spreadsheet also contains no metadata for representing relationships between data items of a table.

* Corresponding author.
E-mail addresses: shigarov@icc.ru, shigarov@gmail.com (A.O. Shigarov), mikhailov@icc.ru (A.A. Mikhailov).

[1] http://lemurproject.org/clueweb09.

[2] http://commoncrawl.org.

http://dx.doi.org/10.1016/j.is.2017.08.004
0306-4379/© 2017 Elsevier Ltd. All rights reserved.



	Retail Sales		Catalog Sales	
	FY2016 (thousands of dollars)	FY2017 (thousands of dollars)	FY2016 (thousands of dollars)	FY2017 (thousands of dollars)
Electronics				
– Phones	11.2	23.7	12.6	32.2
– Computers	89.9	203.1	81.9	204.1
– TV	13.4	32.7	11.7	90.1
Books	12.3	21.6	11.8	24.5

DATA	SALES CHANNEL	FISCAL YEAR	CURRENCY	PRODUCT
11200	retail	2016	u.s. dollars	electronics/phones
23700	retail	2017	u.s. dollars	electronics/phones
12600	catalog	2016	u.s. dollars	electronics/phones
32200	catalog	2017	u.s. dollars	electronics/phones

Fig. 1. A fragment of a source arbitrary table (a); a fragment of a target table in the canonical form generated from the source table (b).

5. *Interpretation*. A data item can be an instance of a concept (category), but its spreadsheet does not explicitly associate it with a domain ontology or a global taxonomy.

The paper covers the rule-based analysis and interpretation of arbitrary tables presented in spreadsheets. Our contribution consists of the following results:

1. We present a novel table object model designed for representing a physical (cellular) and logical (semantic) structure of an arbitrary table in the transformation process (Section 2). Our model associates roles with data items instead of cells or cell regions (e.g. head, stub, or body). Moreover, it provides data provenance for recovered semantics.
2. We propose CRL (Cells Rule Language), a domain-specific language for expressing table analysis and interpretation rules (Section 3). A set of the rules can be implemented for a specific task characterized by requirements for source and target data.
3. We develop TabbyXL, a tool for rule-based transformation of arbitrary tables presented in spreadsheets into the canonical (relational) form (Section 4). The tool implements our table object model and rule language.
4. We evaluate an experimental application that is intended to convert data from tables of the same genre (government statistical websites) to flat file databases (Section 5). It exemplifies the use of our language for developing a task-specific rule-set. The performance evaluation confirms the applicability of the implemented rule-set in accomplishing the stated objectives of this application.

2. Table object model

The table object model is designed for representing both a physical structure and logical data items of an arbitrary table in the process of its analysis and interpretation (Fig. 2). Our model adopts the terminology of Wang's table model [10]. It includes two interrelated layers: *physical* (Section 2.1) represented by the collection of cells and *logical* (Section 2.2) that consists of three collections of entries (values), labels (keys), and categories (concepts). We deliberately resort to the two-way references between the layers to provide convenient access to their objects in table analysis and interpretation rules.

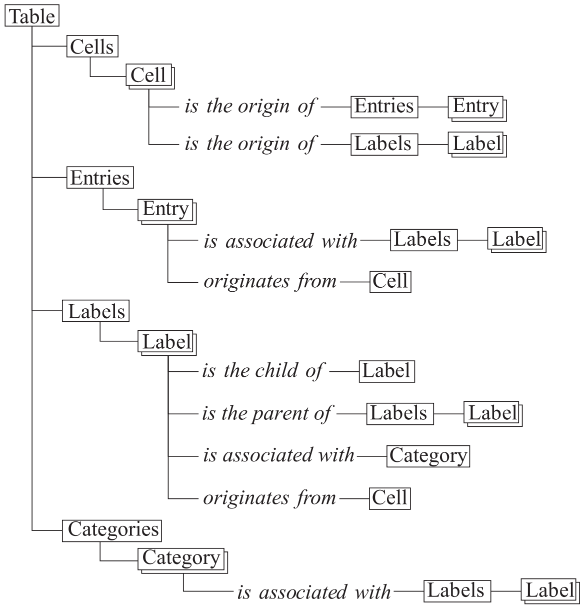


Fig. 2. Two-layered table object model.

2.1. Physical layer

Cell object models common features of a cell that can be presented in tagged documents of well-known formats, such as Excel, Word, or HTML. We define Cell object as a set of the following attributes:

- *Location*: cl — left column, rt — top row, cr — right column, and rb — bottom row. A cell located on several consecutive rows and columns covers a few grid tiles, which always compose a rectangle. Moreover, two cells cannot overlap each other.
- *Style*: font — font features (name, color, height, etc.), bgColor and fgColor — background and foreground colors, rotation — text rotation, horzAlignment