An Extensible Framework for Data Cleaning

Helena Galhardas*
INRIA Rocquencourt
Helena.Galhardas@inria.fr

Daniela Florescu
INRIA Rocquencourt
Daniela.Florescu@inria.fr

Dennis Shasha†
Courant Institute, NYU
shasha@cs.nyu.edu

Eric Simon
INRIA Rocquencourt
Eric.Simon@inria.fr

Abstract

We propose an extensible data cleaning tool, named AJAX, that supports the specification and efficient execution of complex data cleaning programs.

1. Overview

Data quality concerns arise when one wants to correct anomalies in a single data source (e.g., duplicate elimination in a file), or when one wants to integrate data coming from multiple sources into a single new data source (e.g., data warehouse construction). The main quality problem that arises is that the same real object is modeled by different data records. This is called the *Object Identity Problem* and may result from several factors. First, data may contain *errors*, usually due to mistyping, such as "John Smith" and "Joh Smith". Second, when data comes from different origins, different naming conventions may have been used. For instance, the same customer may be referred to in different tables by slightly different but correct names, say "John Smith", "Smith John" or "J. Smith".

Correcting the Object Identity problem is ensured by a set of software solutions called data cleaning tools. We propose a new tool, called AJAX, whose main goal is to facilitate the specification and execution of data cleaning programs either for a single source or for integrating multiple data sources.

2. Features

The main novelty of AJAX is that it allows data cleaning to be described in a declarative way. AJAX proposes a ***framework*** wherein the logic of a data cleaning program is modeled as a directed graph of data transformations that start from some input source data. Four types of data transformations are distinguished. The *mapping* transformation standardizes data formats when possible or simply produces records with a more suitable format. *Matching* finds pairs of records that most probably refer to the same real object. Records are compared using the values of one or several fields via a matching criteria that can be arbitrarily complex. A similarity value representing the result of the matching criteria is attached to each pair of compared records, called a matching pair. *Clustering* groups together matching pairs with a high similarity value by applying a given grouping criteria (e.g. by transitive closure). Finally, *merging* collapses each individual cluster into a tuple of the resulting data source. AJAX provides a ***"declarative" language*** for specifying data cleaning programs, which consists of SQL statements enriched with a set of specific primitives to express these transformations. Declarativeness has several advantages:

1. It allows for ***optimizations***. The execution of the language primitives is supported by both a regular SQL query engine and an external home-grown execution engine. This dual architecture enables to exploit the computing capabilities offered by relational database systems, and particular optimization techniques which are tailored to the specificities of the above transformations.
2. It invites ***extensibility*** through object-relational extensions. For example, the predefined transformations can invoke externally defined domain specific functions (e.g., a string comparison function in the case of the matching macro-operator) that have been previously added to an open library of functions.

AJAX also offers a ***user graphical interface***. It allows the human interaction during the execution of a data cleaning program either to handle exceptional cases arising during the execution or to inspect the intermediate results. Finally, AJAX provides a ***data lineage mechanism*** that permits to backtrack a data cleaning program for debugging purposes.

References

[1] http://caravel.inria.fr/~galharda/cleaning.html.

* Founded by "Instituto Superior Técnico" - Technical University of Lisbon and by a JNICT fellowship of Program PRAXIS XXI (Portugal)

† This work was done while the author was visiting INRIA.