From Spreadsheets to Relational Databases and Back

Jácime Cunha    João Saraiva    Joost Visser

Universidade do Minho
Portugal

SIG & CWI
The Netherlands

GRACE International Meeting on Bidirectional Transformations
14-18 December, 2009
Japan
Row 3 “says”: john has rented a property owned by tony, with address 5 Novar Dr., price per day 70, between dates 1/7/00 and 8/31/01 and paid 25550 for it
This unstructured model is valid and serves its purpose
But, it contains data redundancy
For example, the client name is repeated several times
Motivation

Updating Problems

As a result, updates can cause data inconsistency

For example, updating the renting value of property *pg4* must be performed in several places
Deleting rows can be problematic, too

For example, deleting the row 5 will remove all the information about property *pg36*
Motivation

A Database Solution

- A normalized (3NF) data representation
- No redundancy
- No update/delete problems
A modular refactored spreadsheet

Which performs error detection
From a Spreadsheet to a Database

HaExcel Framework (PEPM’09)

Infer Functional Dependencies (FDs) from data (fun)
Normalize (3NF) those functional dependencies (synthesize)
Create a relational database schema (3NF)
Use data refinements to perform schema and data migration
We use data refinements:

\[ A \preceq B \]

\( to : A \rightarrow B \) is an injective function;
\( from : B \rightarrow A \) a surjective function;
\( from \cdot to = id_A \) (identity function on \( A \));

Example used hierarchical-relational mapping: \( A^* \preceq \mathbb{N} \rightarrow A \)

In Haskell: 2LT - Two-Level Transformation
Refining a Relational Table to a Spreadsheet Table

\[ A \rightarrow B \leq (A \times B)^*_{\text{list2set}\circ\text{compList}\subseteq\text{list2set}\circ\text{listId}} \]

\[
\text{data } PF \ a \ \text{where}
\]

\[
\ldots
\]

\[
S\text{table2stable} :: PF \ (\[(a, b)\] \rightarrow (a \rightarrow b))
\]

\[
T\text{able2sstable} :: PF \ ((a \rightarrow b) \rightarrow [(a, b)])
\]

table2sstable :: Rule

\[
\text{table2sstable} \ (a \rightarrow b)_{\text{inv}} = \text{return} \ (\text{View rep} \ [a \times b]_{\text{inv'}})
\]

\text{where}

\[
\text{inv'} = \text{trySimplify (inv} \circ S\text{table2stable} \wedge \text{fd) }
\]

\[
\text{rep} = \text{Rep}\{\text{to} = T\text{able2sstable}, \text{from} = S\text{table2stable}\}
\]
From a Spreadsheet to a Database

The Strategy

\[
\text{rdb2ss :: Rule}
\]

\[
rdb2ss = simplifyInv \triangleright
\]

\[
(many (aux \_tables2table)) \triangleright
\]

\[
(many ((aux \_tables2sstables) \triangleright (aux \_tables2sstables'))) \triangleright
\]

\[
(many (aux \_table2sstable))
\]

where \(aux \_r = ((once \_r) \triangleright simplifyInv) \triangleright
\]

\[
((many (once \_r)) \triangleright simplifyInv)
\]
HaExcel is a framework to manipulate and transform spreadsheets

- **Library:** generic and reusable library to transform spreadsheets into RDBs and back
- **Front-ends:** Excel/Gnumeric (XML) formats can be imported and exported
- **Tools:**
A talk about HaExcel framework (PEPM paper)

2LT

HaExcel