

Type-safe Evolution of Spreadsheets

Jácome Cunha Joost Visser Tiago Alves João Saraiva

Universidade do Minho, Portugal, {jacome,jas}@di.uminho.pt
Software Improvement Group, The Netherlands, {j.visser,t.alves}@sig.eu

to be presented at FASE (ETAPS) 2011

17 – 21 January

- Spreadsheets are notoriously error-prone;
- Many errors are introduced when changing data;
- But many others when changing the structure;
- Models capturing the interdependencies between data can help (inferable);
- Co-evolution of models and instances.

An Example

- Budget for travel, hotel and local transportation expenses.

	A	B	C	D	E	F	G	H	I
1	Budget		Year			Year			
2			Year=2010			Year=2011			
3	Category	Name	Qty	Cost	Total	Qty	Cost	Total	Total
4		travel	2	200	400	2	450	900	1300
5		hotel	5	100	500	8	80	640	1140
6		local travel	4	20	80	2	35	70	150
7	Total				980			1610	2590

- At the beginning of each year, it needs to be modified to accommodate the next year;
- Several steps are necessary:
add three new rows, labels, update formulas, etc.
- Very prone to errors.

An Example - Changing the Model

- For expenses before and after tax, additional columns need to be inserted in the block of each year.

Cost	Tax tariff	After tax	Total
200	0,12	224	400
100	0,2	120	500
20	0,2	24	80
			980

- The user needs to change all the year in the spreadsheet.

ClassSheets: Specifying Spreadsheets

- Erwig *et al.* have introduced *ClassSheets* to specify spreadsheets;
- *ClassSheets* allow to express business object structures within a spreadsheet using concepts from the OO paradigm;

	A	B	D	E	F	...	G
1	Budget		Year				
2			year=2010				
3	Category	Name	Qty	Cost	Total		Total
4		name="abc"	qty=0	cost=0	total=qty*cost		total=SUM(total)
:							
5	Total				total=SUM(total)		total=SUM(Year.total)

Modeling our Example

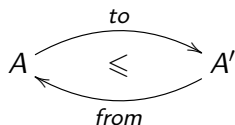
- Class to represent a year;
- Class to represent budget line;
- Class to represent the relationship between them.

	A	B	D	E	F	...	G
1	Budget		Year				
2			year=2010				
3	Category	Name	Qty	Cost	Total		Total
4		name="abc"	qnty=0	cost=0	total=qnty*cost		total=SUM(total)
:							
5	Total				total=SUM(total)		total=SUM(Year.total)

Cost	Tax tariff	After tax	Total
cost=0	tax tariff=0	after tax=cost+ cost*tax tariff	total=qnty*cost
			total=SUM(total)

Spreadsheet Models in HASKELL

- We have created a representation for spreadsheet models based on *ClassSheets*;
- Reused the 2LT framework: when specifying a model transformation, we get for free functions to migrate data back and forward.



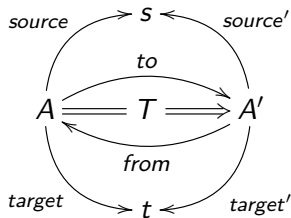
A, A' data type and transformed data type
 to witness function of type $A \rightarrow A'$
(injective and entire relation)
 $from$ witness function of type $A' \rightarrow A$
(surjective, possibly partial)

$from \circ to = id_A$

We can compose refinements

Spreadsheet Models in HASKELL - References

- References pose a particular challenge;
- We implemented them as a pair of selection functions: one selects the cell which is the reference and another select the referenced cell.



source Projection over type *A*
identifying the reference

target Projection over type *A*
identifying the referenced cell

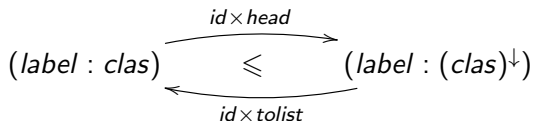
$source' = source \circ from$

$target' = target \circ from$

- Combinators:
 - Pull up all the references: all references must be at the top level;
 - Apply after and friends: applies another rules after something;
- Semantic:
 - Insert column: insert a new column;
 - Make it expandable: allows some part to be added more columns/rows;
 - Split: moves a column and substitutes it by references to the new position;
- Layout:
 - Change orientation: from vertical to horizontal and vice versa;
 - Normalize blocks: some results are not well formatted;
 - Shift: shift vertically, horizontally, up or down.

Make It Expandable

It is possible to make a block expandable:



Its implementation is as follows:

```
expandBlock :: String → Rule  
expandBlock str (label : clas) | compLabel label str = do  
  let rep = Rep { to = id × tolist, from = id × head }  
  return (View rep (label : (clas)↓))
```

- We have created a safe representation of spreadsheet models (and instances);
- We shown rules for coupled evolution of models and instances;
- We want to make this available for spreadsheet users;
- We are working on an extension for OpenOffice;