Model-based Spreadsheet Engineering

Jácome Cunha

Universidade do Minho

PhD Defense
March 22, 2011

Motivation

- Spreadsheets are widely used;
- Their freedom makes people quickly start work with them;
- This freedom is what makes them notoriously error-prone;
- We will present techniques to help spreadsheet end users;

Motivation - cont.

- We believe models can help spreadsheet end users;
- Most spreadsheets do not have a model/specification;
- It is difficult to an end user to create models;
- Thus, we wish to automatically infer them from spreadsheet data;
- Using these models we will make them more efficient and effective.

An Example

- This spreadsheet represents a movie renting system:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**movieID**	**title**	**year**	**director**	**language**	**renterNr**	**renterNm**	**renterPhone**	**rentStart**	**rentFinished**	**rent**	**totalToPay**
2	mv23	Little Man	2006	Keenen Wayans	English	c33	Paul	3334433	01-04-2010	26-04-2010	0,5	12,50
3	mv1	The OH in Ohio	2005	Billy Kent	English	c33	Paul	3334433	30-03-2010	23-04-2010	0,5	12,00
4	mv21	Edmond	2005	Stuart Gordon	English	c26	Smith	4445467	02-04-2010	04-04-2010	0,5	1,00
5	mv102	You, Me and D.	2001	Anthony Russo	English	c3	Michael	5551212	22-03-2010	03-04-2010	0,3	3,60
6	mv23	Little Man	2006	Keenen Wayans	English	c26	Smith	4445467	02-12-2009	04-04-2010	0,5	61,50
7	mv23	Little Man	2006	Keenen Wayans	English	c14	John	3332425	12-04-2010	16-04-2010	0,5	2,00
8	mv3	Alice	2009	Mark Jones	English	c33	Paul	3334433	12-04-2010	23-04-2010	0,5	5,50
9	mv5	I'm legend	2005	Paul Billy	English	c33	Paul	3334433	05-04-2010	06-04-2010	0,4	0,40
10	mv102	You, Me and D.	2001	Anthony Russo	English	c26	Smith	4445467	22-03-2010	25-03-2010	0,3	0,90

- It stores information about movies, renters and leases.

Functional Dependencies

- We wish to automatically infer models from spreadsheet data;
- We discover relationships among spreadsheet data using *functional dependencies*;
- *Functional dependencies* express that some set of columns A uniquely determines another set of columns B, $A \rightharpoonup B$;
- Using data mining algorithms and making use of spreadsheet idiosyncrasies, we can generate a set functional dependencies characterizing the spreadsheet data.

Relational Model

Using the functional dependencies inferred, we generate a relational model characterizing the spreadsheet:

Language (*language*)
Payment (*rentStart*, *rentFinish*, *rent*, *totalToPay*)
Renter (*renterNr*, *renterNm*, *renterPhone*)
Movie (*movieID*, *title*, *year*, *director*, *rent*)
$<Rent>$ (*#language*, *#rentStart*, *#rentFinish*, *#renterNr*, *#movieID*)

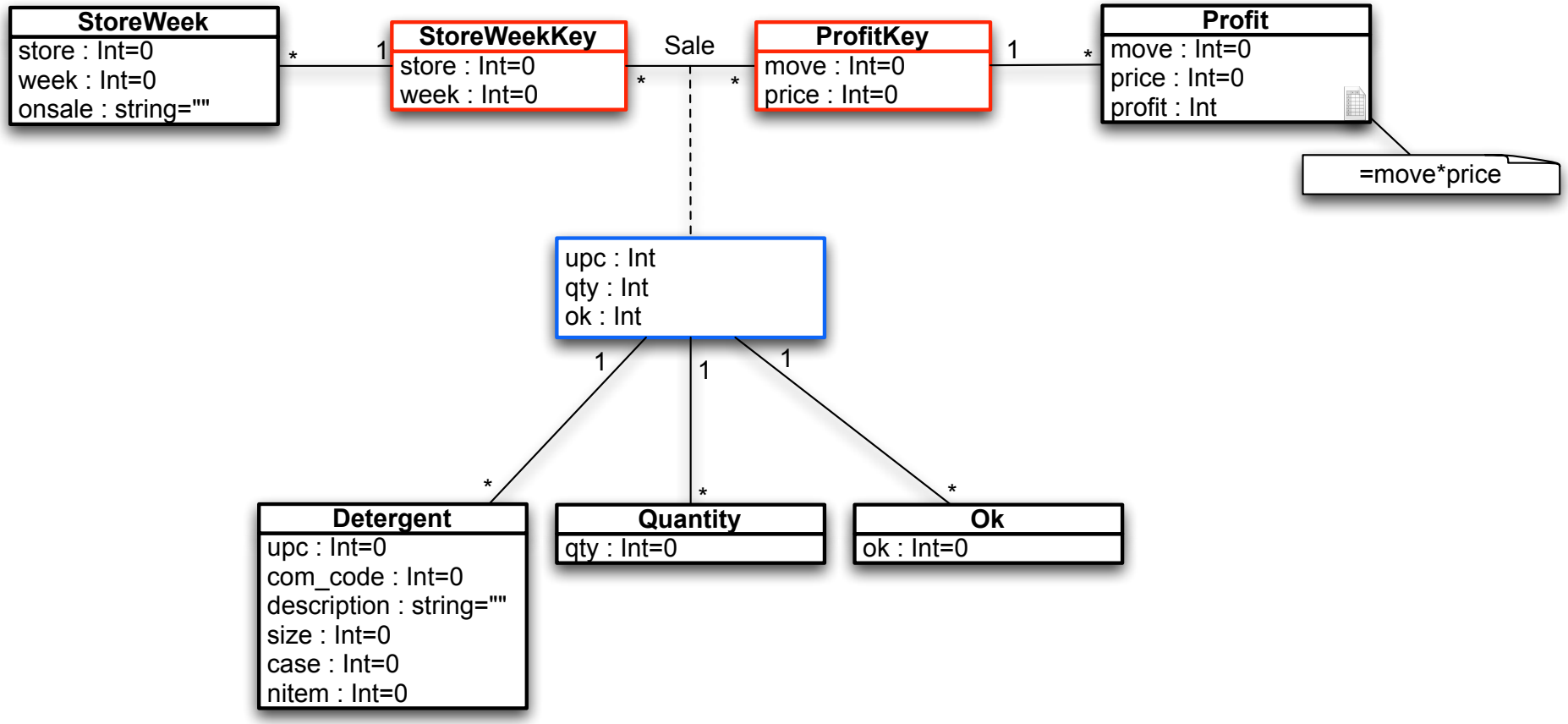
ClassSheet

From the relational model, we can generate a *ClassSheet* diagram fully specifying the spreadsheet:

	A	B	C	D	E	...
1	**Rent**		**MovieKey**			
2			**movieID**	**language**	**renterNr**	
3			movieID=**Movie**.movieID	language=**Language**.language	renterNr=**Renter**.renterNr	
4	**Payment**					
5	**rentStart**	**rentFinish**	**totalToPay**			
6	rentStart=""	rentFinish=""	totalToPay=(rentFinish-rentStart)***Movie**.rent			
⋮						
7						
8	A	B	C	D	E	
9	**Movie**					
10	**movieID**	**title**	**year**	**director**	**rent**	
11	movieID=""	title=""	year=""	director=""	rent=0	
⋮						
12						
13	A	B	C			
14	**Renter**					
15	**renterNr**	**renterNm**	**renterPhone**			
16	renterNr=""	renterNm=""	renterPhone=""			
⋮						
17						
18	A					
19	**Language**					
20	**language**					
21	language=""					
⋮						

UML Class Diagram

Given the similarities between *ClassSheets* and UML class diagrams, we generate the latter from the former:



Edit Assistance

Using the functional dependencies inferred before, we generate a spreadsheet with edit assistance:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**movieID**	**title**	**year**	**director**	**language**	**renterNr**	**renterNm**	**renterPhone**	**rentStart**	**rentFinished**	**rent**	**totalToPay**	
2	mv23	Little Man	2006	Keenen Wayans	English	c33	Paul	3334433	01-04-2010	26-04-2010	0,5	12,50	Delete
3	mv1	The OH in Ohio	2005	Billy Kent	English	c33	Paul	3334433	30-03-2010	23-04-2010	0,5	12,00	Delete
4	mv21	Edmond	2005	Stuart Gordon	English	c26	Smith	4445467	02-04-2010	04-04-2010	0,5	1,00	Delete
5	mv102	You, Me and D.	2001	Anthony Russo	English	c3	Michael	5551212	22-03-2010	03-04-2010	0,3	3,60	Delete
6	mv23	Little Man	2006	Keenen Wayans	English	c26	Smith	4445467	02-12-2009	04-04-2010	0,5	61,50	Delete
7	mv23	Little Man	2006	Keenen Wayans	English	c14	John	3332425	12-04-2010	16-04-2010	0,5	2,00	Delete
8	mv3	Alice	2009	Mark Jones	English	c33	Paul	3334433	12-04-2010	23-04-2010	0,5	5,50	Delete
9	mv5	I'm legend	2005	Paul Billy	English	c33	Paul	3334433	05-04-2010	06-04-2010	0,4	0,40	Delete
10	mv102	You, Me and D.	2001	Anthony Russo	English	c26	Smith	4445467	22-03-2010	25-03-2010	0,3	0,90	Delete
11	mv1	The OH in Ohio	2005	Billy Kent							0,5		Delete
12	mv1												
13	mv21												
14	mv102												
	mv23												

Refactoring Spreadsheets

From the relational model, we generate a spreadsheet that simulates it, but in a spreadsheet environment:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Country**		**Renter**			**Owner**			**Property**				
2	**country**		**renterNr**	**renterNam**		**ownerNr**	**ownerNam**		**propNr**	**propAddress**	**rent**	**ownerNr**	
3	UK		cr56	Aline		co40	Tina		pg4	6 Lawrence	50	Tina	
4			cr76	John		co93	Tony		pg16	5 Nuvar Dr.	70	Tony	
5						co12	Anne		pg36	2 Manor Rd	60	Anne	
6													

(a) First part of the refactored properties spreadsheet.

M	N	O	P	Q	R	S	T
	Renting						
	renterNr	**propertyNr**	**country**	**rentStart**	**rentFinish**	**nrDays**	**total**
	cr76	pg4	UK	01-07-2000	31-08-2001	426	21300
	cr76	pg16	UK	01-09-2001	01-09-2002	365	25550
	cr56	pg4	UK	01-09-1999	10-06-2000	283	14150
	cr56	pg36	UK	10-10-2000	01-09-2001	326	19560
	cr56	pg16	UK	01-11-2002	10-10-2003	343	24010

(b) Second part of the refactored properties spreadsheet.

Migrating Spreadsheets to Databases and Back

- We have calculated the formal relationship between spreadsheets and relational databases;
- This relationship was expressed using data refinement theory;
- Using the 2LT framework (an implementation of data refinement theory), we can transform spreadsheets into databases and vice versa;
- This theory provides functions to safely migrate the data back and forth.

Co-evolution of Spreadsheet Models and Instances

- We have encoded *ClassSheets* in the 2LT framework;
- In particular, we have encoded reference as type-safe projection functions;
- Thus, rules to spreadsheet evolution can be defined;
- We defined a set of common spreadsheet evolution steps:
  - Add/remove column;
  - Make a block expandable;
  - Split;
- This steps can be safely applied to spreadsheet models and data will be migrated automatically.

Empirical Validation of Model-based Spreadsheets

- We believe that model can help end user be more efficient and effective;
- We organized and run an empirical study with 38 participants;
- These participants are students from the university, but non were studying engineering or computer science;
- They were asked to do several tasks in different model-based spreadsheets;
- We concluded that, in some case, our spreadsheets helped them being more effective and efficient.

HaExcel

- We have developed a framework to integrate our techniques in a single platform;
- It is composed by Haskell libraries, batch and online tools and *OpenOffice.org* extensions;
- It can import and export spreadsheets in different formats;
- It can be used to infer functional dependencies and the different models we presented before;
- The model-based spreadsheets can also be generated with HaExcel.

Contributions

- We presented techniques to infer and reason about functional dependencies in the context of spreadsheets;
- Using the idiosyncrasies of spreadsheets, we presented techniques to automatic inference of relational schemas, *ClassSheets* and UML class diagrams;
- Using functional dependencies we can infer edit assistance for spreadsheets including, for example, the auto-completion of some columns;
- We calculated the formal relationship between spreadsheet models and relational schemas. Rules for the migration between these two fields were designed;

Contributions - cont.

- Based on a relational schema we are able to produce a new spreadsheet that is more organized than the original one and thus better for handling data;
- We improved 2LT to support spreadsheet models. We also develop a series of common evolution steps including, for example, insertion of a column in each instance of a model;
- A study with end users validating the results of our work is presented;
- All the techniques here presented are available under an open source framework, HaExcel, that can be reused in other projects.

Contributions - cont.

- Jácome Cunha, João Saraiva, Joost Visser. From spreadsheets to relational databases and back. PEPM '2009. 179–188.
- Jácome Cunha, João Saraiva, Joost Visser. Discovery-based edit assistance for spreadsheets. VL/HCC '2009. 233–237.
- Jácome Cunha, Martin Erwig, João Saraiva. Automatically inferring ClassSheet models from spreadsheets. VL/HCC '2010. 233–241.
- Jácome Cunha, Joost Visser, Tiago Alves, João Saraiva. Type-safe evolution of spreadsheets. FASE '2011. to appear.
- Laura Beckwith, Jácome Cunha, João Paulo Fernandes, João Saraiva. End-users Productivity in Model-based Spreadsheets: An Empirical Study. IS-EUD ' 2011. to appear.