

Rapid Development of Spreadsheet-based Web Mashups

Woralak Kongdenfha¹, Boualem Benatallah¹, Julien Vayssière²,
Régis Saint-Paul³, Fabio Casati⁴

¹CSE, University of New South Wales, Sydney, Australia

²SAP Research, Brisbane, Australia

³CREATE-NET, Trento, Italy

⁴DIT, University of Trento, Italy

22 April 2009

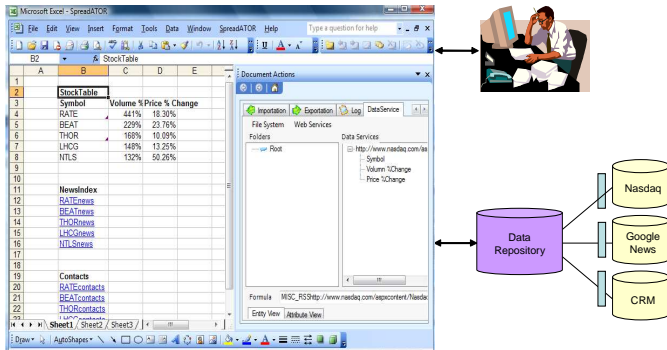
- Integrated Applications
 - Applications and data sources are autonomously developed and deployed
 - Proprietary technologies (communication protocols, data formats, business and presentation logic)
 - Costly development and maintenance of integrated applications especially in large and dynamic environments
- SOA - Main Principles
 - Compliance to standards (protocols and languages)
 - Virtualisation (functionality, data)
 - Composition (reuse)

- SOA: connects systems by process-based composition
- Web 2.0 enables communities and sharing among users
- Mashups: Proliferation of creating websites by composing other websites as seen in ProgrammableWeb.com (novel form of composition)
- Key Enabler of Web 2.0 is the APIs (e.g., SOAP, REST)
- Mashups are typically developed by professional programmers (using AJAX and HTML)
- Needs to bring benefits of composition to end-users

Spreadsheet-based Web data mashups

- Aim: to bring the benefits of composition to end-users
- How: provide a spreadsheet-based framework for Web mashups development
- Why: spreadsheets is one of the most successful end-user programming environment [SSM05, EAKC06, NM90, JBB03]
 - flexible data model,
 - incremental approach for building fairly complex applications with immediate feedback,
 - analysis and manipulation functions spanning various application domains, e.g., financial, statistics.

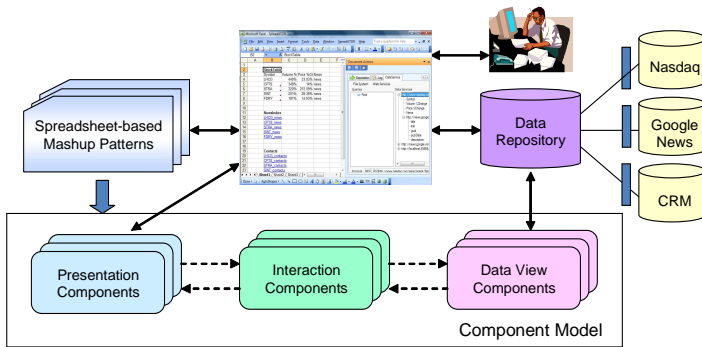
Spreadsheet-based Web Data Mashups



Challenges

- Accessing and representing complex data within spreadsheets
- Synchronization of spreadsheet data and Web data
- Reuse-driven of spreadsheet-based Web mashups
- Easy-manipulation of complex data in spreadsheets

Spreadsheet-based Web Mashups Development Framework



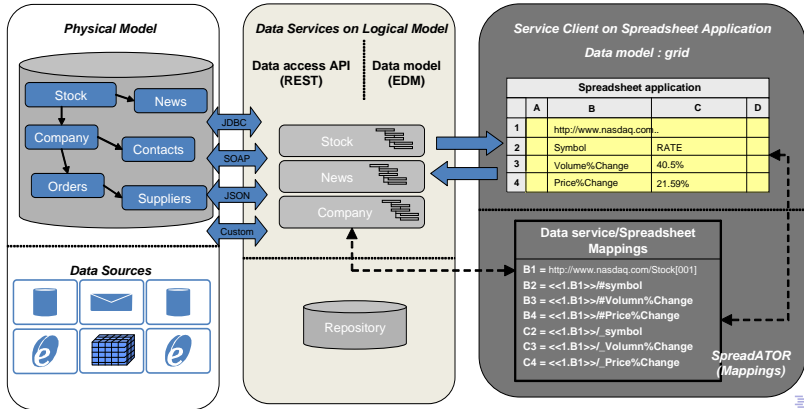
Contributions

- Mapping language between data services/spreadsheet presentation (EDBT'08)
- Component model for spreadsheet-based Web data mashups
- Spreadsheet-based Web mashup patterns
- Drag-and-drop reorganization of structured data in spreadsheets

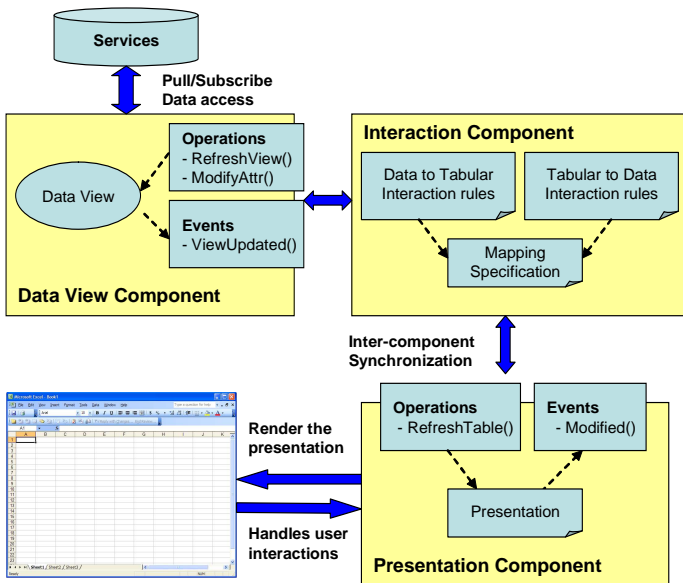
- Background and Challenges
- **SpreadATOR approach**
 - Mapping language between data services/spreadsheet presentation (EDBT'08)
 - **Component model for spreadsheet-based Web data mashups**
 - **Spreadsheet-based Web mashup patterns**
 - **Drag-and-drop reorganization of structured data in spreadsheets**
- Related Work
- Conclusion and Future Work

Mapping language between data services/spreadsheet presentation (EDBT'08)

- interposes an ER-based data model (data services [CN08]) between spreadsheets and heterogeneous data sources; thus enables uniform data access
- proposes a mapping language between data service and spreadsheet data models; thus allows accessing and representing structured data within spreadsheets



Component model for spreadsheet-based Web data mashups



1. Data view components

- allow accessing data from external data sources accessible through data services
- use a variant of the ER model to describe structure of underlying data sources

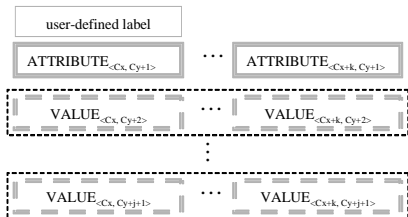
NasdaqStock(Symbol, Volume, Price, News, *GoogleNews)

GoogleNews(title, link, guid, category, pubDate, description)

- adapters are required when accessing data from non-data service sources
- two types of data view components: push component and pull component
- consist of a set of operations and events that allows its interactions with other components of the model
 - operations: `dv:RefreshView()`, `dv:ModifyAttr()`
 - events: `D_ViewUpdated()`

2. Presentation components

- allow displaying data in the tabular grid of spreadsheets
- *presentation specification*
 - describes how the contents of a data view component is mapped to the tabular display of spreadsheet
 - is modeled with the following constructs: ATTRIBUTE, VALUE, RECORD, SET, SHEET



- consists of a set of operations and events
 - operations: `modifyVALUE()`, `insertATTR()`
 - events: `P_VALUEChanged()`, `P_VALUEDeleted()`

3. Interaction components

- the role of interaction components is to synchronize data view and presentation components
- Data-to-Tabular Mappings (DTM)* definition
 - binds contents of a data view component to presentation specification

$$\begin{aligned} \langle C_x, C_y \rangle &= dv:getObjects() && \text{----- (1)} \\ \langle C_x+k, C_y+1 \rangle &= dv:getAttrName(o_1, a_k) ; 0 < k < dv:countAttrs(o_1) && \text{----- (2)} \\ \langle C_x+k, C_y+2+j \rangle &= dv:getAttrValue(o_j, a_k) ; 0 < j < dv:countObjs(\langle C_x, C_y \rangle) && \text{----- (3)} \\ &&& 0 < k < dv:countAttrs(o_j) \end{aligned}$$

- Interaction rules* establish publish/subscribe relationships between data view and presentation components

Presentation-data interaction rule

```
<interaction publisher="StockTable" event="P_VALUEChanged"
subscriber="StockDataView" operation="dv:modifyValue()" >
```

Data-presentation interaction rule

```
<interaction publisher="StockDataView" event="D_ViewUpdated()"
subscriber="StockTable" operation="ui:Refresh()" >
```

- Background and Challenges
- SpreadATOR approach
 - Mapping language between data services/spreadsheet presentation (EDBT'08)
 - Component model for Web data mashups
 - **Spreadsheet-based Web mashup patterns**
 - Drag-and-drop reorganization of structured data in spreadsheets
- Related Work
- Conclusion and Future Work

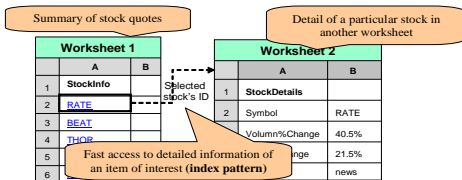
- Benefits of the proposed component model
 - enables the synchronization of spreadsheet data and Web data
 - enables different tabular presentations
 - enables bulk loading
- However, it is difficult for non-professional users to build such a component model
- We propose the notion of spreadsheet-based Web mashup patterns
 - each pattern captures default behavior for creating Web data mashups (default data access method, presentation, and interaction features)
 - shifts the development efforts from scratch to reuse

Spreadsheet-based Web mashup patterns (characterization)

- CAISE'08: analysis of tabular presentations frequently found in spreadsheet applications
- WWW'09: captures common spreadsheet data presentations as presentation components

	A	B	C	D
1				
2	St			
3	Symbol	Volumn%Change	Price%Change	
4	RATE	40.5%	21.59%	
5	BEAT	37.0%	32.17%	
6	THOR	27.0%	18.78%	
7	LHCG	21.4%	41.35%	
8	NTLS	19.3%	34.68%	

Side-by-side comparison of stock quotes
(Table pattern)



	A		
1	StockDetails		
2	Symbol		
3	Change in Volumn	40.5%	
4	Change in Price	21.5%	
5	News		
6			
7	StockList		StockList
8	Symbol	RATE	Symbol
9	Volumn%Change	40.5%	Volumn%Change
10	Price%Change	21.5%	Price%Change
11			
12	Symbol	BEAT	Symbol
13	Volumn%Change	37.0%	Volumn%Change
14	Price%Change	32.17%	Price%Change
15

Stock information with user-defined labels (Content pattern)

Stock details shown in a list (Repeater pattern)

Worksheet 1

	A	B
1	NewsIndex	
2	RATEnews	
3	BEATnews	
4	THORnews	
5	LHCGnews	
6	NTLSnews	

Selected stock's ID

Worksheet 2

	A	B	C	D
1	NewsTable			
2	title	link	pubDate	description
3	BBB warns...	http://new.	Mon,29..	<table bc...
4	Filmmaking...	http://new.	Wed,15..	<table bc...
5	American...	http://new.	Thu,16..	<table bc...
6	Sculpture...	http://new.	Fri,17..	<table bc...
7	RATE...	http://new.	Wed,01..	<table bc...

A list of news related to a particular stock in another worksheet (Relationship Index pattern)

A list of stocks and their related contact information organized in a nested structure (Hierarchical pattern)

					F	G
1	Contacts					
2		Prospect	ID	Name	ACC_Owner	ACC_Mngr
3	RATE	3088299	56388	Plasma ...	Elizabeth Alex...	Elizabeth Alex...
4				Cell Culture...	Brian Mcnamee	Brian Mcnamee
5						
6	BEAT	1111111	444555	SK	Alex	Alex
7		2222222	555444	SK	N/A	N/A
8	...					

Spreadsheet-based Web mashup patterns (Patterns usage and tool support)

Microsoft Excel - SpreadATOR

File Edit View Insert Format Tools Data Window SpreadATOR Help

Type a question for help

B2 MISC RSShttp://www.nasdaq.com/aspxcontent/NasdaqRSS.aspx?data=uvup

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Document Actions

1. provide URL of data

2. browse schema of data service

3. specify query conditions

4. select a presentation component to display data

5. specify sorting condition on the query result

6. specify the number of content should be displayed at a time

7. specify how the data should be refreshed

FFRSSPreview

RecordPerPage 1 Page Number 1 Return All Pages Refresh

Attribute	Value	Operator	FilterValue	Include
Symbol	RATE	Equal		<input checked="" type="checkbox"/>
Volume %Change	441%	Equal		<input checked="" type="checkbox"/>
Price %Change	18.30%	Equal		<input checked="" type="checkbox"/>
News	news	Equal		<input type="checkbox"/>

Presentation Format

- Content Widget
- Table Widget
- Content Widget
- Repeater Widget
- Index Widget
- Hierarchical Widget

Direction Ascending

Refresh Option Manually

Select All Unselect All Show All Attribute Prev Next Finish

Spreadsheet-based Web mashup patterns (Table Mashup Pattern)

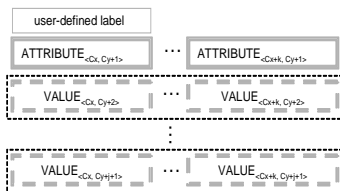
Data view component

NasdaqStock(Symbol, Volume, Price, News)

Interaction component

$\langle C_x, C_y \rangle = dv:getObjects()$
 $\langle C_{x+k}, C_{y+1} \rangle = dv:getAttrName(o_i, a_k) ; 1 \leq k \leq dv:countAttrs(o_i)$
 $\langle C_{x+k}, C_{y+1+j} \rangle = dv:getAttrValue(o_j, a_k) ; 1 \leq j \leq dv:countObjs(\langle C_x, C_y \rangle)$
 $1 \leq k \leq dv:countAttrs(o_i)$

Presentation component



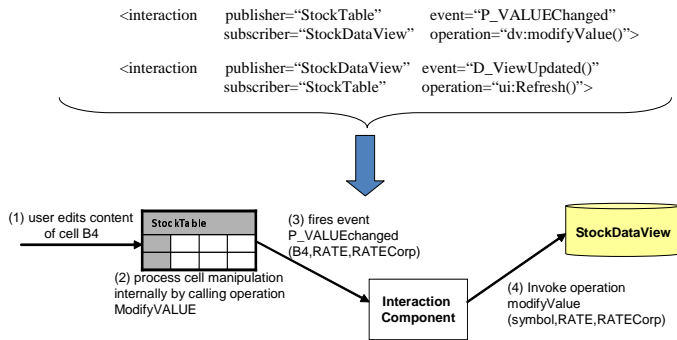
Formulas generated from Table Mashup Pattern

	A	B	C	D
1				
2		http://www.nasdaq.com...		
3		$\langle\langle 1.B2 \rangle\rangle.[0]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[0]_{\#Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[0]_{\#Price\%Change}$
4		$\langle\langle 1.B2 \rangle\rangle.[0]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[0]_{_Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[0]_{_Price\%Change}$
5		$\langle\langle 1.B2 \rangle\rangle.[1]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[1]_{_Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[1]_{_Price\%Change}$
6		$\langle\langle 1.B2 \rangle\rangle.[2]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[2]_{_Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[2]_{_Price\%Change}$
7		$\langle\langle 1.B2 \rangle\rangle.[3]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[3]_{_Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[3]_{_Price\%Change}$
8		$\langle\langle 1.B2 \rangle\rangle.[4]_{_symbol}$	$\langle\langle 1.B2 \rangle\rangle.[4]_{_Volumn\%Change}$	$\langle\langle 1.B2 \rangle\rangle.[4]_{_Price\%Change}$

Contents displayed on the spreadsheet

	A	B	C	D
1				
2		StockTable		
3		Symbol	Volumn%Change	Price%Change
4		RATE	40.5%	21.59%
5		BEAT	37.0%	32.17%
6		THOR	27.0%	18.78%
7		LHCG	21.4%	41.35%
8		NTLS	19.3%	34.68%

Spreadsheet-based Web mashup patterns (Runtime Behavior)



- At runtime, when user edits contents of a VALUE cell, the interaction component performs the following tasks:
 - Capturing event from the presentation component
 - Invoking operations of subscribing components

Spreadsheet-based Web mashup patterns (Tool support)

The screenshot displays the Microsoft Excel - SpreadATOR application. The main window shows a spreadsheet with the following data:

	A	B	C	D	E
1					
2		Stocktable			
3		Symbol	Volume %	Price %	Change
4		RATE	441%	18.30%	
5		BEAT	229%	23.76%	
6		THOR	168%	10.09%	
7		LHCG	148%	13.25%	
8		NTLS	132%	50.26%	
9					
10					
11		NewsIndex			
12		RATEnews			
13		BEATnews			
14		THORnews			
15		LHCGnews			
16		NTLSnews			
17					
18					
19		Contacts			
20		RATEcontacts			
21		BEATcontacts			
22		THORcontacts			
23		LHCGcontacts			
24		NTLScontacts			
25					

The right-hand side of the application features a **Document Actions** panel with tabs for **Formula**, **Importation**, **Exportation**, **Log**, and **DataService**. The **Service Browser** is active, showing a tree view of **Web Services** under **File System**. The tree includes folders like **Sales-oppr-Nov08**, **NasdaqStock**, **GoogleNews**, and **CRMContacts**. A **Query Folders** label points to the **GoogleNews** folder. The **Service Browser** also displays a list of **Data Services** with fields such as **Symbol**, **Volume %Change**, **Price %Change**, **title**, **link**, **guid**, **category**, **pub Date**, and **description**. A **Formula Editor** at the bottom shows the formula `MISC_RSShttp://www.nasdaq.com/aspxcontent/NasdaqRSS.asp`. Below the formula editor is an **Object Browser** showing a tree view with **Stocktable** and a sub-entry for **04 Feb 2009 16:00:00 - T**.

- Background and Challenges
- Integration of spreadsheets with SOA
- SpreadATOR approach
 - Mapping language between data services/spreadsheet presentation (EDBT'08)
 - Component model for Web data mashups
 - Spreadsheet-based Web mashup patterns
 - **Drag-and-drop reorganization of structured data in spreadsheets**
- Example Application
- Related Work
- Conclusion and Future Work

Drag-and-drop reorganization of structured data

- Spreadsheet paradigm supports data manipulation
- When introducing new abstractions in spreadsheets, SpreadATOR also considers providing data manipulation operators for such abstractions

The screenshot displays the Microsoft Excel - SpreadATOR interface. The main window shows a spreadsheet with the following data:

1	A13	B	C	D	E	F
1	RATEContacts	ContactName	ContactTitle	Address	City	Phone
2	CHRSB	Christina E Order adm	Berguvsvä	Lulea		0921-1289890
3	BLAUS	Hanna Moi Sales man	Forsterstr	Mannheim	0621-8856544	
4	BLONP	Blondesdd Marketing	24. place F	Strasbourg	88 60 15 74 23	
5	BONAP	Bolido Cor Owner	C/Araquil, Madrid		(91) 555 4322	
6	LAURL	Laurence L Owner	12, rue de	Marseille	91 24 45 43 35	
7						
8						
9						
10						
11						
12						
13	MISC_ADOhttp	localhost:49566/CRM.svc/Contacts(companyID = "RATE")	ContactName	ContactTitle	Address	City
14	CHRSB	Christina E Order adm	Berguvsvä	Lulea		0921-1289890
15	BLAUS	Hanna Moi Sales man	Forsterstr	Mannheim	0621-8856544	
16	BLONP	Blondesdd Marketing	24. place F	Strasbourg	88 60 15 74 23	
17	BONAP	Bolido Cor Owner	C/Araquil, Madrid		(91) 555 4322	
18	LAURL	Laurence L Owner	12, rue de	Marseille	91 24 45 43 35	
19						
20						
21						
22						

The Document Actions panel on the right shows the following structure:

- Importation
- Exportation
- Log
- DataService
- File System
 - Root
 - Sales-oppr-Nov08
- Web Services
 - Data Services
 - http://www.nasdaq.com/as
 - Symbol
 - Volume %Change
 - Price %Change
 - http://news.google.com/7o
 - title
 - link
 - guid
 - category
 - pubDate
 - description
 - http://localhost:49566/CRM

The Formula bar at the bottom shows: `http://localhost:49566/CRM.svc/Contacts(companyID =`

Existing tools in Spreadsheets (Excel)

- XML mapping tool, SQL importation, Web data importation, Analysis Services
 - No support for complex types as they are converted to Excel's supported types. Hence it is not possible to access the original document
 - Lose of the nested structure of data as the tool denormalizes the hierarchical structure into flat-table
- Adherence to various tools dealing with external data importation in spreadsheets gives us hope that SpreadATOR will be adopted

Spreadsheet Data Presentations

- Lakshmanan's Tabular Data Model [LSGK98, GLS96]
 - provides a powerful data model to present broad classes of tables (relation/spreadsheets)
 - the specification of presentations needs to be done manually
- Gencel [ACKE, EAKC06, EE]
 - a specific table presentation (since the purpose is to ensure spreadsheet correctness)
 - supports for the specification of presentation through templates
- SpreadATOR approach is in between the two by defining a small set of frequently used spreadsheet data presentations and provides support for their specifications

Inferring spreadsheet data presentation templates

- infer templates from existing spreadsheets [AE06] (enables data exportation)
- SpreadATOR's data presentation templates can be adopted by such approach

Data-driven Web application development

- ASP.Net, PHP, JSF
 - provide widgets for data presentation
 - require web development skills
- WebML [CFB00]
 - provide presentation patterns for Web applications development
- SpreadATOR follows spreadsheet paradigm and provide interactive importation and presentation with immediate results

- SpreadATOR brings the benefits of SOA (Data Services) to end-users by providing a spreadsheet-based framework for Web data mashups development
- The integration of data services and spreadsheets raise challenges in the difference in their data models
- By enabling spreadsheet cells to contain *object reference* and *queries*, SpreadATOR allows the user to access structured data within spreadsheets
- SpreadATOR proposes a new component model that supports the synchronization between spreadsheet data and Web data
- SpreadATOR introduces a set of common spreadsheet-based Web mashup patterns that simplifies mashups development by shifting the development efforts from scratch to that of reuse

- Presentation components
 - We observed that Web data are frequently displayed using visualization components such as timeline, maps, etc.
 - We are investigating how to integrate common visualization components into SpreadATOR, while preserving spreadsheet paradigm
- Composition patterns
 - SpreadATOR currently focuses on the integration at the data layer
 - To enable service composition within spreadsheets, more complex patterns such as workflow patterns need to be considered



Robin Abraham, Irene Cooperstein, Steve Kollmansberger, and Martin Erwig.

Automatic generation and maintenance of correct spreadsheets.

Proc. ICSE'05.



Robin Abraham and Martin Erwig.

Inferring templates from spreadsheets.

In ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 182–191, New York, NY, USA, 2006. ACM.



Stefano Ceri, Piero Fraternali, and Aldo Bongio.

Web modeling language (webml): a modeling language for designing web sites.

In Proc. WWW'00, pages 137–157, 2000.



Pablo Castro and Anil Nori.

Astoria: A programming model for data on the web.

Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pages 1556–1559, April 2008.



Martin Erwig, Robin Abraham, Steve Kollmansberger, and Irene Cooperstein.

Gencel: a program generator for correct spreadsheets.

J. Funct. Program., 16(3):293–325, 2006.



Gregor Engels and Martin Erwig.

Classsheets: automatic generation of spreadsheet applications from object-oriented specifications.

In Proc. ASE '05.



Marc Gyssens, Laks V. S. Lakshmanan, and Iyer N. Subramanian.

Tables as a paradigm for querying and restructuring (extended abstract).

In *PODS '96: Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 93–103, New York, NY, USA, 1996. ACM.



Simon Peyton Jones, Alan Blackwell, and Margaret Burnett.

A user-centered approach to functions in excel.

In *ICFP '03: Proceedings of the eighth ACM SIGPLAN international conference on Functional programming*, pages 165–176, New York, NY, USA, 2003. ACM Press.



Laks V. S. Lakshmanan, Subbu N. Subramanian, Nita Goyal, and Ravi Krishnamurthy.

On query spreadsheets.

In *ICDE '98: Proceedings of the Fourteenth International Conference on Data Engineering*, pages 134–141, Washington, DC, USA, 1998. IEEE Computer Society.



Bonnie A. Nardi and James R. Miller.

The spreadsheet interface: A basis for end user programming.

In *INTERACT '90: Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*, pages 977–983. North-Holland, 1990.



C. Scaffidi, M. Shaw, and B. Myers.

Estimating the numbers of end users programmers.

In *Proc. VLHCC '05*, pages 207–214, 2005.