Supporting the Spreadsheet Idea for Interactive Database Applications

Mario Gleichmann[1], Thomas Hasart[1], Ilvio Bruder[1], and Peter Forbrig[2]

[1] IT Science Center Rügen gGmbH,
Circus 14,
D-18581 Putbus, Germany
Tel.: +49 38301 88290
{gleichmann,hasart, bruder}@it-science-center.de
[2] University of Rostock,
Albert-Einstein-Str. 21
D-18051 Rostock, Germany
Tel.: +49 381 498 7620
peter.forbrig@uni-rostock.de

Abstract. Data base applications allow the analysis of complex and large data. There are many analysis functions showing different relations between the data. End users have often new requirements to see data and relations which can not be shown by the existing analysis software. They need possibilities to create new user interfaces to fit their requirements. Generally, users don't have programming knowledge and cannot wait until the development department has specified the corresponding software. They need a tool which can easily and quick produce corresponding results. The tool must allow navigating via complex data structures of data bases.

This paper discusses a tool that allows end users to specify interactive applications like spreadsheets. The tool supports OLAP applications and is based on the Qt Designer.

Keywords: End user development, OLAP, Business Intelligence, Qt Designer.

1 Introduction

Our project on "Model-Driven Account Management in Data Warehouse Environments" (Monicca) aims at developing data base applications by end users. A tool for key account managers has been developed (key account management, see [9]). With this tool, such a manager can offer clients different views of aggregated data from a data warehouse via specific designed user interfaces. To generate the user views, a model language was developed which can describe necessary OLAP operations for the views, relations, the definition of the outputs and interactions.

This paper starts with a typical scenario of one of our industrial project partner. A part of a sales process is presented. It shows how an interactive application can be used to support business processes. For clarity, only a few exemplary data are presented.

P. Forbrig, F. Paternó, and A. Mark-Pejtersen (Eds.): HCIS 2010, IFIP AICT 332, pp. 122–132, 2010.

The second part of the paper shows how end users can create the GUI without any programming knowledge. Afterwards, a short overview is given that discusses, which existing technologies were used and how these technologies had to be modified to support end user development on OLAP data.

Finally, some data from usability tests of the tool are presented.

2 Sales Talk Scenario

In this scenario, a salesperson of a bakery producer is negotiating with a customer on sales volumes. An essential part of the negotiations are the product prices and things like discounts. In the special case important aspects are the annual sales volumes of different types of bread.

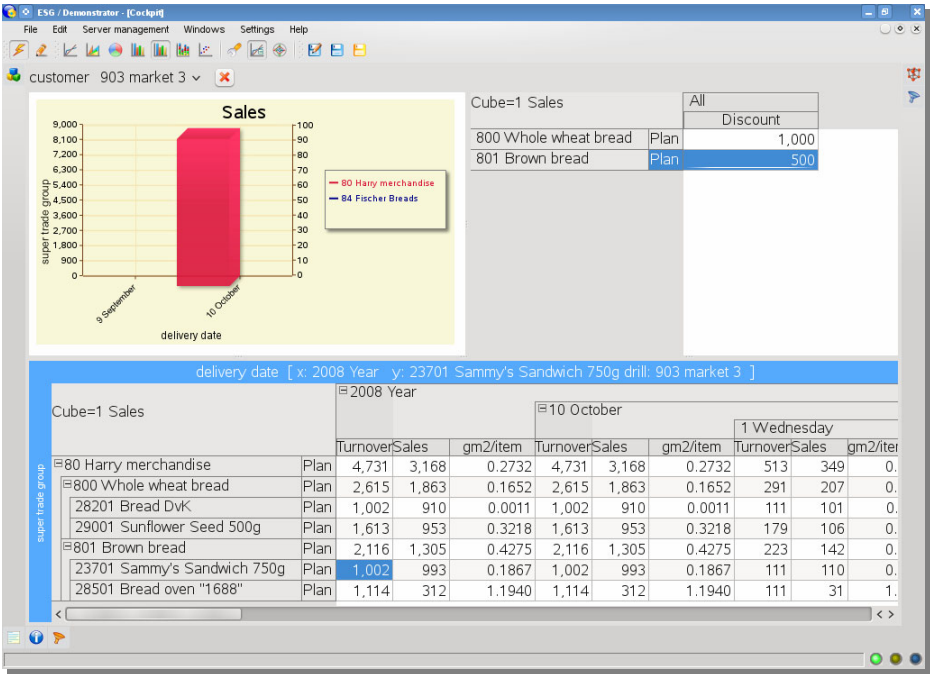
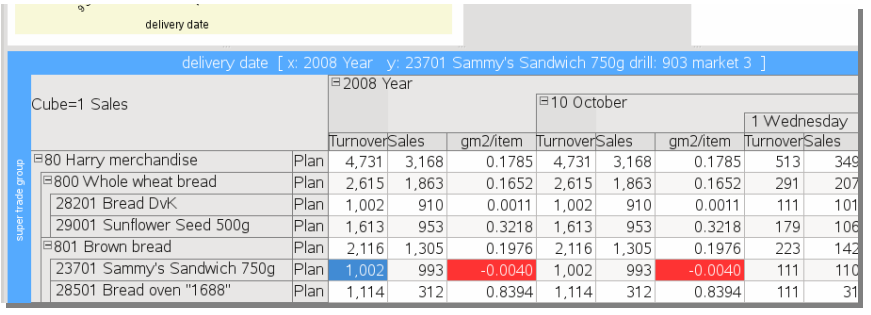


Fig. 1. Screenshot of a GUI giving an overview for planning

In Fig. 1., the data of the customer can be seen. In the groups, "Wholemeal" and "Wheat bread" that belong to the upper category of "Merchandise Harry", are the aggregate values of the "Turnover", "Sales" and "gm2 / item" (gross margin 2 per item) for the year 2008 and the October of that year. One can also see the aggregation of items into their group. The sales of the category "Whole wheat bread" are, e.g., the result of the sum of sales by its articles "Bread DvK" and "Sunflower Seed 500g". The same is true for the turnover.



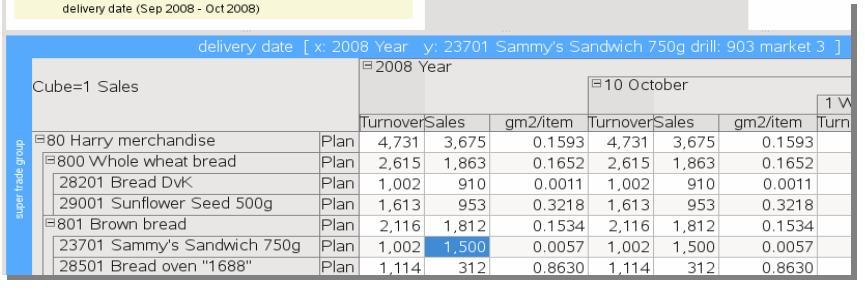
Cube=1 Sales		2008 Year			10 October			1 Wednesday	
		Turnover	Sales	gm2/item	Turnover	Sales	gm2/item	Turnover	Sales
80 Harry merchandise	Plan	4,731	3,168	0.1785	4,731	3,168	0.1785	513	349
800 Whole wheat bread	Plan	2,615	1,863	0.1652	2,615	1,863	0.1652	291	207
28201 Bread DvK	Plan	1,002	910	0.0011	1,002	910	0.0011	111	101
29001 Sunflower Seed 500g	Plan	1,613	953	0.3218	1,613	953	0.3218	179	106
801 Brown bread	Plan	2,116	1,305	0.1976	2,116	1,305	0.1976	223	142
23701 Sammy's Sandwich 750g	Plan	1,002	993	-0.0040	1,002	993	-0.0040	111	110
28501 Bread oven "1688"	Plan	1,114	312	0.8394	1,114	312	0.8394	111	31

Fig. 2. Negative gross margin in red (if not colorful darkest fields are red)

In contrast, the calculation of "gross margin 2 / item" is not as simple as the "Turnover". There are a variety of variable costs, such as discounts, rebates and discounts from the gross proceeds (gross revenues) deducted to get the net sales. The gross margin 2 is the difference of direct sales and marketing costs such as advertising subsidies and the like. In the upper right table, products and current discounts are shown. The top left graphics represents the sales date of the months September and October.

Assuming the customer wants a higher discount for "Brown bread", which is changed to 800 €. The system immediately performs all necessary calculations.

As shown in Fig. 2., this would cause a negative gross margin. Consequently, such a reduction would be unacceptable for the seller. To not refuse the higher rebate the manager may look for other compensations and changes the number of sales to 1.500.



Cube=1 Sales		2008 Year			10 October			1 W
		Turnover	Sales	gm2/item	Turnover	Sales	gm2/item	Turn
80 Harry merchandise	Plan	4,731	3,675	0.1593	4,731	3,675	0.1593	
800 Whole wheat bread	Plan	2,615	1,863	0.1652	2,615	1,863	0.1652	
28201 Bread DvK	Plan	1,002	910	0.0011	1,002	910	0.0011	
29001 Sunflower Seed 500g	Plan	1,613	953	0.3218	1,613	953	0.3218	
801 Brown bread	Plan	2,116	1,812	0.1534	2,116	1,812	0.1534	
23701 Sammy's Sandwich 750g	Plan	1,002	1,500	0.0057	1,002	1,500	0.0057	
28501 Bread oven "1688"	Plan	1,114	312	0.8630	1,114	312	0.8630	

Fig. 3. Results of change the annual sales

This yields to a positive gross margin, which is even more than the old one. This is in spite of a higher discount (Fig. 3). This is a typical win-win situation that will contribute to the success of the sales talk.

In the following we will demonstrate how the discussed user interface can be designed using our tool.

3 End User Development

The result of a possible graphical user interface is illustrated in Fig. 1. With our tool it is possible to develop such an interactive system without programming knowledge.

This is as easy as the development of a spread sheet. There are a few steps necessary to perform this. First the tool has to be started. A blank widget will be created initially. Different visualization objects can be dropped on that widget. Fig. 4. shows the situation after an "ENodeSelectorWidget" (left) and an "ESumGrid" (right) were specified. (The visualisation objects are explained in detail in paragraph 3.1.)

On the right hand side the docked window "Cube Selection" can be seen, from which a cube can be selected and with drag and drop attached to the "ESumGrid". Afterwards specific date can be assigned to horizontal and vertical axis (Fig. 5.).

For other visualization objects like "ENodeSelectorWidget" other interactions are necessary. Each object needs to know to which cube it is assigned to and which details of the data have to be displayed.

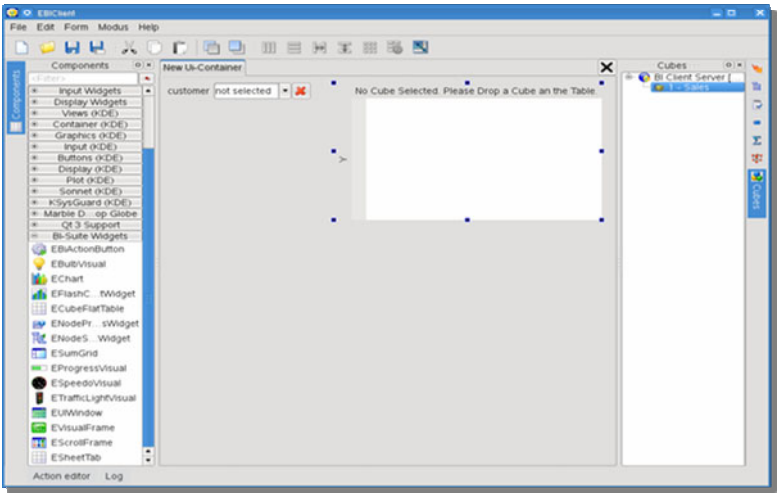


Fig. 4. "ENodeSelectorWidget" and "ESumGrid" dropped into the UI-Container

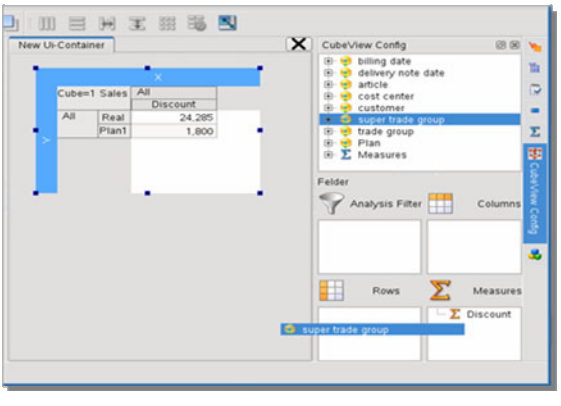


Fig. 5. Assignment of dimensions

In order specify connections between different objects in the user interface, their signals have to be linked. In this way changes in one object result in changes in the other objects.

Fig. 6**.** visualizes the dependency of "ESumGrid" from "ENodeSelectorWidget". The grid will be informed if another selection was made in "ENodeSelectorWidget". It will be updated and will display the corresponding data.

Fig. 6. Visualization of dependencies between object of the user interface

Finally, the individual widgets can be arranged by layouts. In this way they can respond to resizing. Therefore we select simultaneously the chart component and the upper right "ESumGrid" (Fig. 7.) and chose one of the different layout types. By default a horizontal layout is used that results in displaying all components with the same size. Thus the user could later change the sizes we use a horizontal splitter.

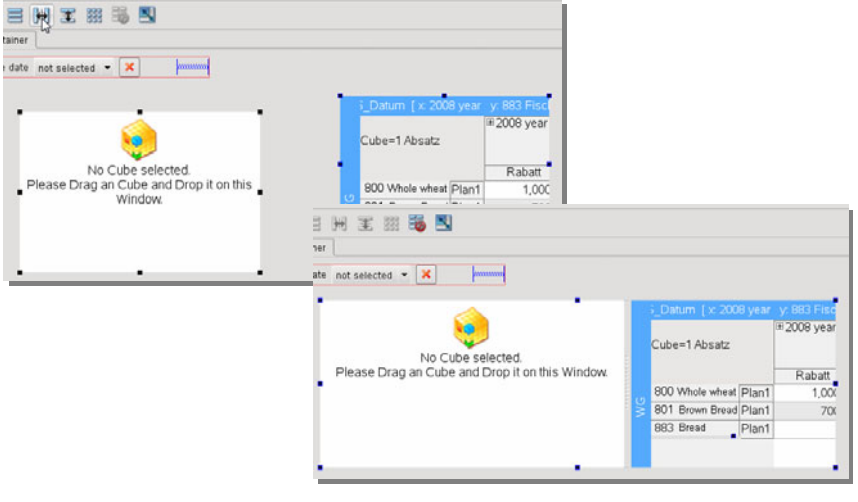


Fig. 7. Layout objects

As result we have a fully featured user interface which reacts on resizing and data changes. It can now be stored and can be used as a template for further developments.

We have demonstrated how our tool can be used to create graphical user interfaces. In this example specific visual objects were used. In the following we want to given an overview of existing objects and how they can be used.

3.1 Visualization Objects

The visualisation of data by different graphical object pays an important role for the usability of software. On the one hand it is very important to attract the attention of users on the other hand users should not pay too much attention to understand the visualisation of an application.

3.1.1 "ESumGrid" – OLAP Table

The "ESumGrid" (e.g. Fig. 3.) represents an OLAP table which can be used to display complex matters. In particular, this table is characterized by numerous and powerful navigation capabilities in multidimensional space.

Based on the example above on the x-axis is the dimension "billing date" and on the y-axis the dimension "super trade group" arranged. On the first start the aggregations of the chosen data, in our case "sales", "turnover" and "gross margin 2 / item", for the highest layers of the dimensions, "Year" and "super trade group" are shown. Now, if the column for the year is opened the months of the year appears and with it the values of data for the top category is shown. Because the column of the year remains visible it can be easily seen, that the addition of the total revenue columns for every month is the sum in the year column. Now, every month can be expanded again and releases the individual values of the days in that month.

Similarly, the y-axis can be expended. The super trade group releases by expanding the trade groups and the individual articles.

Fully expanded each individual sale of a certain article on a particular day can be found. It can also be analysed how it is part of the monthly and annual sum and of the trade and super trade group sum.

This type of analysis falls within the ambit of the "Online Analytical Processing", or short OLAP, and describes an entirely new approach to analyse economic data than the previous conventional approach of static snapshots of specific sizes [15].

3.1.2 "ETrafficLightVisual" - Traffic Lights

For easy monitoring of certain indicators thresholds can be defined. One way to visualize these are traffic lights. The thresholds can be shown on this object with certain colours. Red could be signalled that not enough pieces of an article are sold to justify the production or the purchase. Yellow could be used as a concern. Green visualises that there is no critical situation.

This traffic light object allows a quick overview of some specific data.

3.1.3 "ESpeedoVisual" – Tachometer

For the measures in relation to a particular end or maximum value tachometers are suitable.

Fig. 8. Traffic lights in different styles

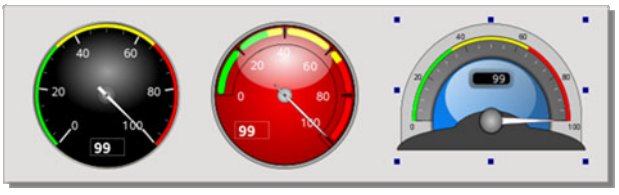


Fig. 9. Tachometers in different styles

3.1.4 "EFlashChartWidget" – Graph

As usual for chart components this widget allows the graphical representation of data. In addition to the shown bar charts (in Fig. 1.) are many other chart types possible (Fig. 10.). With this visualization object can show good temporal developments of certain data.

Fig. 10. Different chart types

3.1.5 "ENodeSelectorWidget" - Node Selection

This combo box allows selecting specific data in one dimension. If this object is used in conjunction with the "ESumGrid" it usually shows the data of a further (not in "ESumGrid" used) dimension. If the "ESumGrid" has the data of the dimensions "billing date" and "super trade group" the selector could have the data of the dimension "customer". With the selection of a certain customer the displayed data in the "ESumGrid" are restricted to that specific customer. There will be an aggregation of turnovers and sales of items that were sold to the selected customers only.

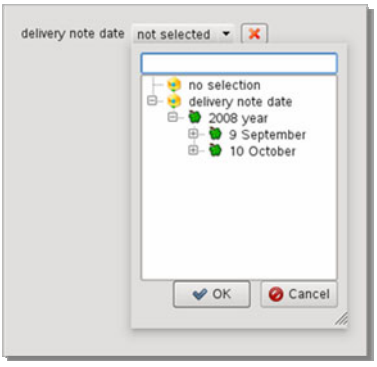


Fig. 11. A widget to select nodes

3.2 "EProgressVisual" - Progress Bar

This UI element is especially suitable for time-dependent quantities or values which achieve a fixed final value and are progressing.

Fig. 12. Progress bar

3.2.1 "EBIActionButton" - Command Button

For this item scripts for mouse actions can be defined. In this way, multiple analyses can be linked together or complete applications can be created.

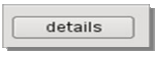


Fig. 13. An example for a button

Other functionalities are calling external programs, contact the properties or invoking actions on other elements of the current presentation.

3.2.2 "ESheet" - Spreadsheet

In these component formulas, pictures and charts can be combined in a flexible manner. The spreadsheet-like approach gives the user more freedom to change the appearance of the application. But for meaningful reports some knowledge is necessary to write the correct formulas.

We have seen how end user development for OLAP works. Special visual objects exist, which can be used for different aspects of analyses. The tool and the resulting application are easily and quick to use.

In the following the technology behind the tool will be shortly described.

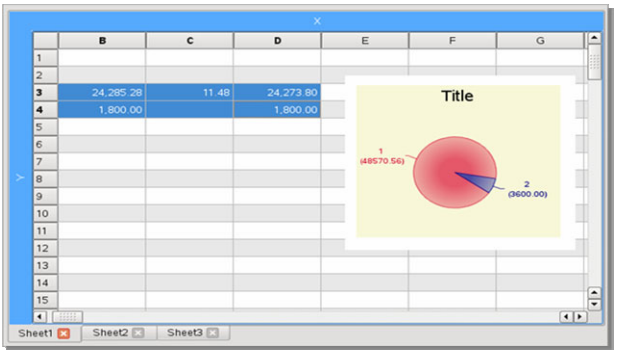


Fig. 14. A sheet with a chart

4 Technologies

To provide end user development we integrated an existing editor into our editor. We use the "Qt Designer", a part of the C++ framework "Qt" from "Nokia".

Usually this tool is used from C++ developer to arrange components to complex GUIs and save the settings in XML files.

We have extended it with dock widgets for the selection of data cubes, dimensions, and layers.

Additionally numerous plug-ins were developed which can be used as visualization objects.

As result a designer for end user development of OLAP applications exists. Like in a spreadsheet language [6] a user can specify the data that have to be displayed and he can use different visualization objects. The user can select a cube, the dimensions, layers, nodes and data. He can set relations between objects of the user interface. In some cases formulas can be specified to compute certain data.

5 Evaluation

To evaluate our tool we performed a usability test with 8 candidates. None of them were programmers. All of them work in a company as consultant. 4 of the candidates were already familiar with the idea of OLAP cubes and corresponding software. They already used the tool for adaptations in projects. The other 4 had to be introduced to the idea first.

All of them had to develop the user interface we discussed in this paper. First the scenario was explained and afterwards the candidates were asked to develop the presented user interface.

The ages ranged between 20 and 50. We observed that younger people were able to work with the tool better than older ones.

It was also recognized that it helped a lot if the candidates already knew the idea of OLAP cubes. More or less all candidates were able to specify the static user interface and to assign data to the tables.

50% of the candidates that already used our tool in projects had problems with the concepts of signals. They were not able to specify the dynamic behaviour of the user interface by specifying dependencies without help. None of the novices were able to perform this task.

In this respect training is necessary to be able to use our existing tool. As a result we have to think about a better user interface for this specific problem. Maybe it is not necessary to ask the user such detailed technical aspects like: "Which signal has to be propagated?" There might be better opportunities. We want to face this challenge in the future.

Our experiences with customers show some further interesting aspects. Even if we speak from end-user programming it is often not the end user who performs this work. The end user wants to have consultants that specify adaptations.

With our tool we are very close to the spreadsheet metaphor [3] for OLAP applications. But even for spreadsheets managers would like to perform the development task to be done by somebody else. For our tool the managers ask the consultants of our company to develop new user interfaces and to adapt existing user interfaces.

The good news is, even if we do not have reached end user programming, we reached programming by non-programmers. New user interfaces can be designed by consultants who are not involved in the development process of the tool and have no knowledge in programming. At the moment they are trained in specifying and we hope that by further improving our tool this training can be reduced. Maybe somewhere in the future no training will be necessary anymore.

6 Summary and Outlook

We have presented a tool for OLAP applications that allows non-programmers to develop own applications. In this paper, a scenario was discussed in which ad hoc reports on mass data in complex relationships are necessary. A GUI was described which makes these computations possible. It was shown how end-user can assemble themselves such a GUI using our tool that is based on the "Qt Designer" provided by "Nokia".

Additionally, some UI elements, developed by our group, were presented. These elements allow an adequate visualisation of specialized data.

In the future we work on a new interface based on the MDX standard replacing the current proprietary data connection interface for our visualization objects. This leads to a more flexible usage of the GUI on different OLAP servers.

Reasonable and reusable arrangements of GUI components developed by end users will be generalized and stored as patterns. These templates provide other users a faster GUI development for new OLAP applications. The variety of patterns will be stored in a database as a pattern library.

A first evaluation of our tool with different user groups was discussed and will support our next development steps. The results were promising. With further development of our tool the evaluation of the next prototype will be extended.

References

1. Ballinger, D., Bidle, R., Noble, K.: Spreadsheet Visualisation to Improve End-user Understanding. In: Australiean Symposium on Information Visualisation, Adelaide, Australia (2003)

2. Burnett, M., Cook, C., Pendes, O., Rothermel, G., Summet, J., Wallace, C.: End-User Software Engineering with Assertions in the Spreadsheet Paradigm. In: Proc. International Conference on Software Engineering, Portland, Oregon, USA, pp. 93–103 (2003)
3. Chitnis, S., Yennamani, M., Gupta, G.: ExSched: Solving Constraint Satisfaction Problems with the Spreadsheet Paradigm. In: CoRR. abs/cs/0701109, p. 1 (2007)
4. Dmitriev, S.: Language oriented programming: The next programming paradigm. JetBrains 'onBoard' Electronic Monthly Magazine (2004), http://www.onboard.jetbrains.com/is1/articles/04/10/lop/
5. Erwig, M., Abraham, R., Cooperstein, I., Kollmansberger, S.: Automatic Generation and Maintenance of Correct Spreadsheets. In: Proc. ICSE 2005, St. Louis, Missouri, USA, May 15-21, pp. 136–145 (2005)
6. Fisher II, M., Jin, D., Rothermel, G., Burnett, M.: Test Reuse in the Spreadsheet Paradigm. In: IEEE International Symposium on Software Reliability Engineering, p. 1 (2002)
7. Hodgins, J., Bruckman, A., Hemp, P., Ondrejka, C., Vinge, V.: The Potential of End-User Programmable Worlds: Present and Future. In: Panel SIGGRAPH 2007: ACM SIGGRAPH 2007 Panels (2007)
8. Ruthruff, J.R., Burnett, M.: Six challenges in supporting end-user debugging. ACM SIGSOFT Software Engineering Notes 30(4), 1–5 (2005)
9. McDonald, M., Rogers, B.: Key Account Management – Learning from supplier and customer perspectives. Butterworth Heinemann, Oxford (1998)
10. Meyer, R.M., Masterson, T.: Towards a better visual programming language: critiquing prograph's control structures. The Journal of Computing in Small Colleges 15(5), 181–193 (2000)
11. Millman, A.F., Wilson, K.J.: From Key Account Selling to Key Account Management. Journal of Marketing Practice: Applied Marketing Science 1(1), 9–21 (1995)
12. Mørch, A.I., Stevens, G., Won, M., Klann, M., Dittrich, Y., Wulf, V.: Component-Based Technologies for End-User Development. Communications of the ACM 47(9), 59–62 (2004)
13. Myers, B.A., Burnett, M.: End Users Creating Effective Software. In: CHI 2004 Special Interest Group, Vienna, Austria (2004)
14. Myers, B., Burnett, M.M., Wiedenbeck, S., Ko, A.J.: End User Software Engineering: CHI 2007 Special Interest Group Meeting. In: CHI 2007, San Jose, California, USA (2007)
15. Pendse, N.: What is OLAP? The OLAP Report (1998), http://www.olapreport.com/fasmi.htm (visited: 13.03.2008)
16. Scaffidi, C., Shaw, M., Myers, B.: An Approach for Categorizing End User Programmers to Guide Software Engineering Research. In: First Workshop on End User Software Engineering (WEUSE I), Saint Louis, Missouri, May 21 (2005)
17. Scaffidi, C.: A Data Model to Support End User Software Engineering. In: 29th International Conference on Software Engineering, ICSE 2007 Companion (2007)
18. Sidow, H.D.: Key Account Management. Landsberg am Lech: mi-Fachverlag (2007)
19. Nokia (2010), http://qt.nokia.com/ (visited: 18.01.2010)
20. Winter, S.: Generierung von dynamischen Web-Anwendungen aus visuellen Spezifikationen. In: Schloss Birlinghoven - Sankt Augustin: Fachwissenschaftlicher Informatikkongress – Informatiktage 2005 (2005)