Demonstration of Software Components for End-User Development

Mario Gleichmann[1], Thomas Hasart[2], Ilvio Bruder[3], Andreas Heuer[4], and Peter Forbrig[5]

[1] IT Science Center Rügen gGmbH, Germany, gleichmann@it-science-center.de
[2] IT Science Center Rügen gGmbH, Germany, hasart@it-science-center.de
[3] IT Science Center Rügen gGmbH, Germany, bruder@it-science-center.de
[4] IT Science Center Rügen gGmbH, Germany, heuer@it-science-center.de
[5] University of Rostock, Germany, peter.forbrig@uni-rostock.de

Abstract. This paper demonstrates how "End-User-Development" can be implemented with the Qt4 designer of Trolltech. It provides an example showing how users modify user interfaces by adding functionality that originally was not available.

Keywords: end user development, design, user interface

1. Introduction

This paper demonstrates the design process of user defined user interfaces for accessing OLAP data (Pendse, 1998) at runtime. As an example, a simple dashboard for a controller is designed.

2. Project Monicca

Our Monicca project on "Model-Driven Account Management in Data Warehouse Environments" aims at adapting OLAP applications of the user interface to the functional layer. Thereby, a tool is developed for Key Accounts [6]. With this tool one can offer clients different data from the Data Warehouse or other external resources. It gives key account managers a special interface to the data of key customers. Additionally, this allows applications to offer broad, suitable and adjusted analysis functionalities. The general problem of adaptation of OLAP applications will be solved by using techniques based on metadata. To generate the user views, a model language will be developed which can describe necessary OLAP operations for the views, relations, the definition of the outputs and the following interactions. This model-based approach is the basis for the "end-user development" that aims at adapting and extending applications.

3. Demonstration

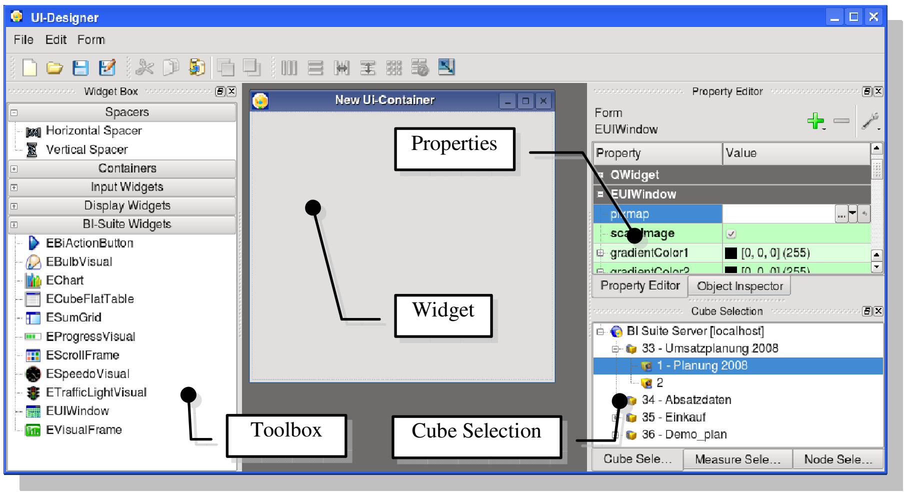


Fig. 1. Designer

Figure 1 shows the embedded Qt-Designer [15] with an empty new window (Ui-Container). On the right hand side the current available UI components are listed. These components can be dragged and dropped on the empty window. From the dock window “Widget Box/BI-Suite Widgets”, which holds the custom designer plug-ins, “ESumGrid” was selected and dragged on the new widget (Figure 2).

Now, a data cube must be specified representing the data of this table. Therefore, a cube from the list in the “Cube Selection” tab window is dropped on the table (Figure 3).

Following the same procedure, the node for the horizontal and the vertical axis (Figure 4) can be assigned.

Measures and plan scenarios can be designed in the same way like the axis definition by drag and drop of the needed items.

To demonstrate the layout mechanism, some more items must be created - in this example two speedometers and one button (Figure 6). As first step the two speedometers are laid out horizontally (Figure 7). The button is being placed at the lower left corner. Therefore, a vertical spacer is created and a layout with the button is horizontally performed. Afterwards all created elements are laid out vertically. Figure 8 shows the final layout of the sample dashboard.

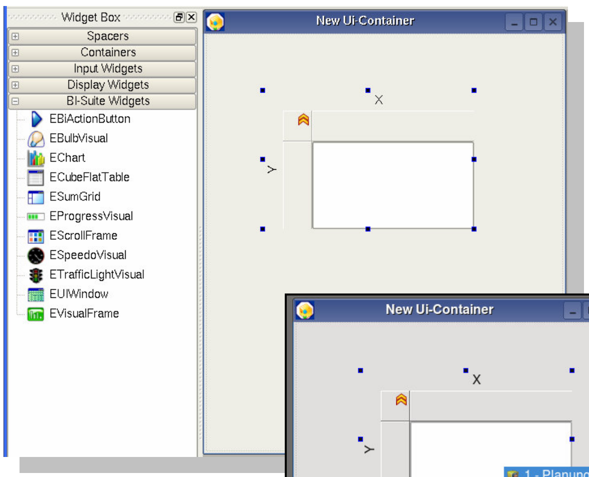


Fig. 5. Window with a slice and dice table

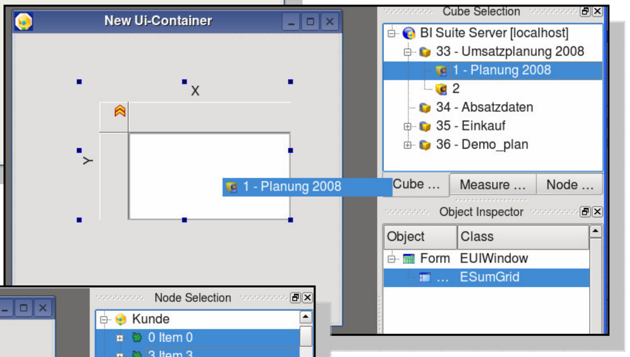


Fig. 5. Drop cube on table

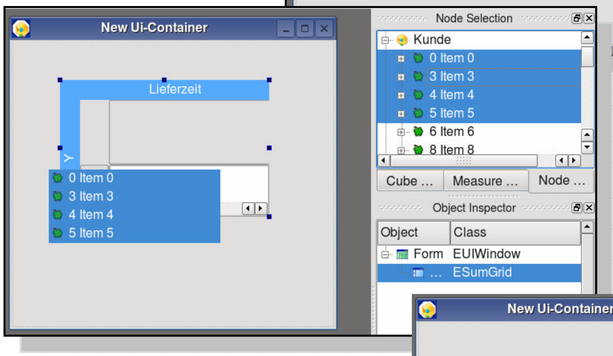


Fig. 55. Definition of x- and y-axis

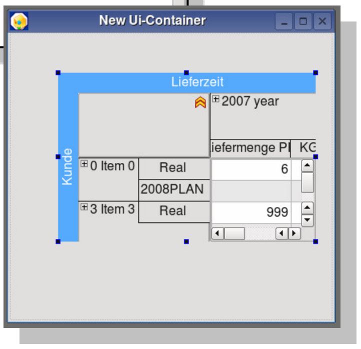


Fig. 5. Finished axis definition

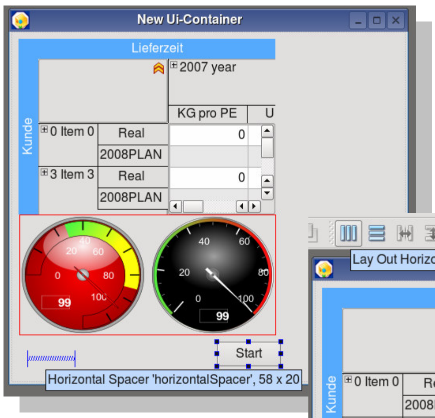


Fig. 7. Layout mechanism

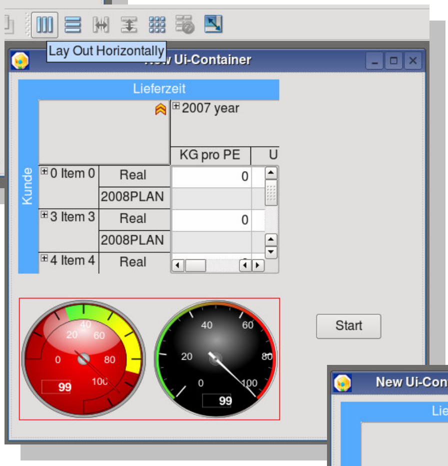


Fig. 7. Layout horizontally

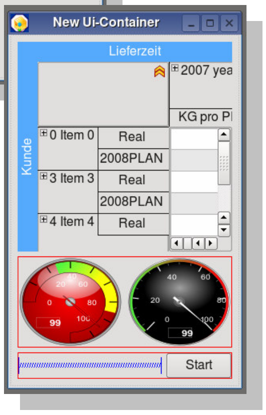
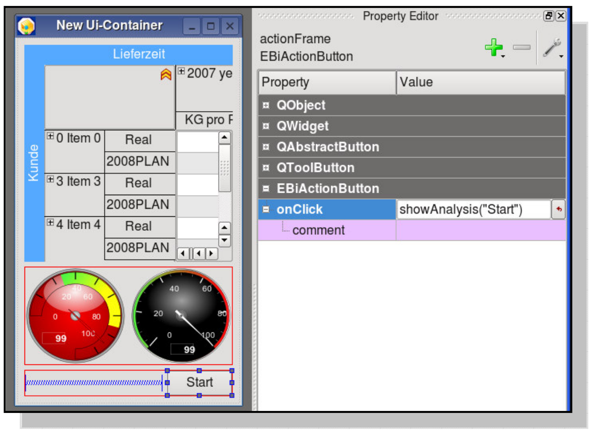


Fig. 8. Final layout

Figure 9 shows the property editor of the Qt-Designer. Most of the properties are provided by the QWidget class. These properties are needed to define the look of the selected component. They allow to change the font or the background image. In the example, a new property "onClick" for the BIActionButton is implemented. Within this property, a script can be defined which is executed if the button is clicked by the user. Currently, functions for switching to another container and for executing external commands are implemented.



actionFrame
EBiActionButton

Property	Value
QObject	
QWidget	
QAbstractButton	
QToolButton	
EBiActionButton	
onClick	showAnalysis("Start")
comment	

Fig. 9. Component properties

4. Summary and Future Work

With the developed plug-ins, we allow users to design interactive applications according to their own needs. A first big step in the direction of "End-User-Development" has been realized. Our approach provides opportunities to represent OLAP cubes as spreadsheet metaphors for end users. Limitation and problems in forms creation are no longer only solvable by software developers. Initial experiments with key account managers (without any programming knowledge) showed very positive results.

As a challenge still remains defining dependencies between visualization components. This would allow users to specify the consequences of interacting with one object by state changes in other objects. A selection box could restrict the data area of all elements or a chart could show the currently selected row in a table.

Indeed, experiments demonstrated that user would like to have these possibilities in existing applications.

It is also planned to consider visualization proposals for spreadsheets discussed in [1]. Constrains analogous to [2] would be more meaningful expansions to a wider security of the programs.

In the current stage of development, very limited opportunities for process control are available. At the moment there are no conditional jumps to other analysis elements. It is also not possible to unlock actions depending on properties of the elements.

For enterprise applications it is not enough to have good individual (local) user interface elements. Complex enterprise models are needed to get the software usable. It must e.g. be possible to specify various user roles in these models.

References

1. Ballinger, D., Bidle, R. & Noble, K. (2003). Spreadsheet Visualisation to Improve End-user Understanding. Australiean Symposium on Information Visualisation, Adelaide, Australia.
2. Burnett, M., Cook, C., Pendes, O. Rothermel, G., Summet, J. & Wallace, C. (2003). End-User Software Engineering with Assertions in the Spreadsheet Paradigm, Proc. International Conference on Software Engineering, Portland, Oregon, USA, p. 93-103.
3. Erwig, M., Abraham, R., Cooperstein, I.. & Kollmansberger, S. (). Automatic Generation and Maintenance of Correct Spreadsheets. In Proc. ICSE'05, May 15–21, 2005, St. Louis, Missouri, USA, p. 136-145.
4. Hodgins, J., Bruckman, A., Hemp, P., Ondrejka, C. & Vinge, V. (2007). The Potential of End-User Programmable Worlds: Present and Future. Panel SIGGRAPH '07: ACM SIGGRAPH 2007 pan-els.
5. Ruthruff, J. R. & Burnett, M. (2005). Six challenges in supporting end-user debugging. ACM SIG-SOFT Software Engineering Notes, Volume 30 Issue 4, p. 1-5
6. McDonald, M. & Rogers, B. (1998). Key Account Management – Learning from supplier and customer perspectives. Oxford: Butterworth Heinemann.
7. Meyer, R. M. & Masterson, T. (2000). Towards a better visual programming language: critiquing prograph's control structures. The Journal of Computing in Small Colleges, 15(5):181–193
8. Millman, A. F. & Wilson, K. J. (1995). From Key Account Selling to Key Account Management, Journal of Marketing Practice: Applied Marketing Science, 1 (1), 9 -21.
9. Mørch, A. I., Stevens, G., Won, M., Klann, M., Dittrich, Y., & Wulf, V. (2004). Component-Based Technologoies for End-User Development. Communications of the ACM September 2004/Vol. 47, No. 9 p. 59-62.
10. Myers, B., Burnett, M. M., Wiedenbeck, S. & Ko, A. J. (2007): End User Software Engineering: CHI'2007 Special Interest Group Meeting. CHI 2007, San Jose, California, USA.
11. Pendse, N. (1998): What is OLAP?, The OLAP Report, 1998. http://www.olapreport.com/fasmi.htm (visited: 13. 03. 2008)
12. Scaffidi, C., Shaw, M. & Myers, B. (2005). An Approach for Categorizing End User Programmers to Guide Software Engineering Research. First Workshop on EndUser Software Engineering (WEUSE I). May 21, 2005, Saint Louis, Missouri.
13. Scaffidi, C. (2007). A Data Model to Support End User Software Engineering. 29th International Conference on Software Engineering, ICSE'07 Companion.
14. Sidow, H. D. (2007). Key Account Management. Landsberg am Lech: mi-Fachverlag.
15. Trolltech. (2008). http://trolltech.com/. (visited: 11.03.2008)