Towards User Interface Components for Dashboard Applications on Smartphones

Marcus Homann, Vassilena Banova, Paul Oelbermann,
Holger Wittges, and Helmut Krcmar

Technische Universität München, Chair for Information Systems,
Boltzmannstr. 3, 85748 Garching, Germany
{marcus.homann,vassilena.banova,paul.oelbermann,
holger.wittges,krcmar}@in.tum.de

Abstract. Aim of this paper is to identify common functionalities of dashboard applications on smartphones. The findings are used to design reusable user interface components in order to improve the development activities for such applications. In order to identify common functionalities twelve existing dashboard applications for smartphones from different vendors were analyzed. The findings are illustrated using an UML use-case diagram. A grouping of different use-cases represents the foundation for the proposed reusable user interface components. The components are described using a structured format based on user interface patterns. The findings revealed that the analyzed dashboard applications offer similar functionalities that were categorized by ten use-cases. The groups of functionality were used to design seven user interface components that could be reused in future developments. An implementation of the proposed components for the iPhone revealed that the components are feasible.

Keywords: Mobile BI, Smartphone, Dashboard, User Interface Pattern, User Interface Component, Business Intelligence.

1 Introduction

Business intelligence applications become increasingly important as information volumes in enterprises explode. Today, driven by the increasing mobility of employees and the pervasive utilization of mobile devices in enterprise environments, the need for mobile business intelligence solutions is also growing. Mobile business intelligence (mobile BI) is the delivery of BI capabilities through mobile devices and offers a way of improving employee's productivity by using technology that employees already understand and are familiar with [1]. Recent market research shows that enterprises have already recognized the benefits of mobile BI such as operational efficiency, real-time analytics and customer responsiveness [2].

Current mobile BI solutions are mainly focused on information consumption, offering rich and interactive dashboards that generate a level of user engagement not usually seen in traditional desktop solutions [1]. During the implementation of mobile

G. Memmi and U. Blanke (Eds.): MobiCASE 2013, LNICST 130, pp. 19–32, 2014.

BI applications, different challenges related to the specific characteristics of mobile devices and distributed information consumption should be considered. The restricting effect of some of these challenges decreases over time because of technological enhancements – for example, the rapid development of wireless communication networks in the last years enabled the access of bigger amounts of data via mobile devices [3]. Nevertheless, there are other challenges, such as the limitations of data input, screen size and mobile device performance which still need to be considered as an important aspect in the scope of implementing mobile BI solutions.

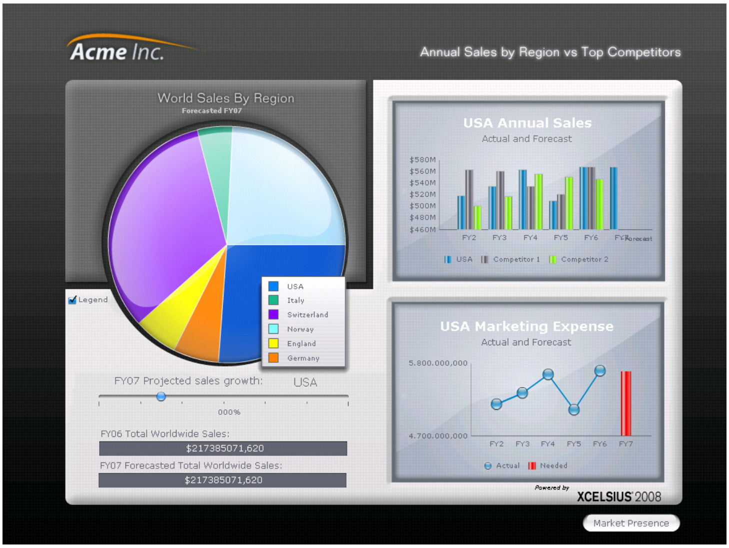
The aim of this paper is to improve the development activities for mobile BI applications by offering reusable user interface components based on different use-cases of mobile BI. To identify common functionalities and develop the reusable interface components, twelve dashboard applications for smartphones from different vendors were analyzed. The findings of this paper will offer a systematic classification of different use cases of mobile BI. Moreover, the elaborated reusable user interface components will support and simplify the implementation process of mobile dashboard applications.

The paper is structured in five sections. Section two offers an overview of the different categories of dashboards and their utilization. Section three offers a better understanding on the topic of user interface patterns and components, their definition and relevance in the scope of this paper. This section is followed by the description of the selected use cases of dashboard applications on smartphones. Subsequently, the reusable user interface components are derived from the selected use cases, offering a significant simplification of the development process of mobile dashboard applications. Finally, the last section of the paper summarizes the lessons learned as well as the outlook towards potential future research projects in the area of mobile BI.

2 Dashboards

Before we begin to describe the analysis of different use cases and common user interface patterns for mobile dashboards, a common understanding on what a dashboard application is will be introduced.

Many different understandings on what a dashboard is exist. The common aspects in all these definitions are the following: a dashboard appears on computer screens and offers visibility into key performance indicators (KPI) through simple visual graphics [4]. The understanding on dashboards in the scope of this paper is based on the definition of Few [5], who considers all these aspects and captures them in one sentence: “A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance” [5]. An example of a dashboard is shown in figure 1.



FY07 Projected sales growth: USA

000%

FY06 Total Worldwide Sales:
$217385071,620

FY07 Forecasted Total Worldwide Sales:
$217385071,620

Fig. 1. Example of a dashboard [6]

Dashboards are commonly used by decision makers and have therefore the peculiarity of executive information systems. They are often seen as an addition to Online Analytical Processing (OLAP) Tools, offering more structured and easier to comprehend visualization of relevant information [6]. Dashboards can be categorized according to different aspects such as the type of information they visualize, the way they display information to serve a particular purpose and – related to this purpose – the target user group of the dashboard. In the scope of this paper we categorize different dashboard types according to their target group of use. Based on this categorization perspective three general dashboard types can be defined [6]:

1. **Strategic dashboards:** Target group of strategic dashboards are decision makers who need to get a quick overview of the current situation within the company. Strategic dashboards do not provide all the detailed information needed to make complex decisions, but focus on key figures which can help identify opportunities for further and more detailed analysis. Therefore they should offer a simple representation of the most important key performance indicators.
2. **Analytical dashboards:** Target group of analytical dashboards are mainly analysts or controllers. The data is here more detailed and more complex context can be displayed. Analytical dashboards provide information which enables the analysis of trends or even root cause analysis by drawing comparisons across multiple variables over time. To enable users to understand trends and why certain things are happening in an organization, analytical dashboards facilitate interactions with the data allowing the

increase of detail of the analyzed data, as well as time-dependent comparisons. Therefore, an analytical dashboard usually contains much more information and can be more complex than strategic and operational dashboards.

3. **Operational dashboards:** Target group of operational dashboards are users in operational business, who use detailed business process and transactional data to complete business tasks or to monitor real time operations. Moreover, operational dashboards are used to alert users, in case of deviations from the norm or critical states in operational data. Therefore, operational dashboards need to be updated frequently or even in real time. They contain less information than strategic and analytical dashboards to enable a clear visualization of critical operational data.

These dashboard categories also apply for mobile BI, yet there are some additional aspects to be considered during the implementation process in comparison to a desktop version of a dashboard. Dashboards for mobile BI users should provide relevant information that is easily processed on mobile devices. Moreover the smaller screen size of mobile devices and the lack of input devices (e.g. keyboard, mouse) should also be considered during the implementation of mobile dashboards. Since mobile BI user are viewing information on a small screen and do not have access to traditional interfaces, the dashboard's operational performance is one of the most relevant concerns to be addressed in the scope of developing mobile BI applications.

The consideration of these aspects during the development process of mobile BI applications is essential and therefore different guidelines addressing specific issues exist. The aim of this paper is to elaborate reusable user interface patterns and therefore the next section discusses the concept of user interface patterns and its use in the scope of a software implementation process.

3 User Interface Patterns and Components

Collected design knowledge about UIs is usually provided in form of guidelines, principles or patterns [7]. Principles such as the eight golden rules of Shneiderman [8] have a generic character, while guidelines are often platform specific [9]. Both tend to be difficult to use during the design process, because they suggest absolute validity [7]. On the contrary, UI patterns explicitly focus on context and are thus problem related [7]. They tell the designer when, how and why the provided solution is applicable [7].

The concept of design patterns was originated in urban architecture by Christopher Alexander [10], adapted to object-oriented software design by Beck and Cunningham [11] and became popular through the book of Gamma et al. [12]. In general, patterns are structured descriptions of an invariant solution to a recurrent problem in a certain context [13]. They follow a three-part rule which expresses the relation between a certain context, a problem and a solution [10]. The interest in patterns in Human-Computer Interaction dates back to 1994 [14]. According to a CHI workshop in 2000 an UI pattern "captures the essence of a successful solution to a recurring usability problem in interactive systems" [14]. Therefore patterns can be seen as descriptions of best practices which capture common solutions to design tensions and are thus by definition not novel [15].

A collection of UI patterns is often referred to as pattern catalogue or pattern language [9, 13]. Several UI pattern catalogues are focused on designing websites. However, there are also UI pattern collections available that deal with the design of smartphone applications, like [16, 17]. These patterns usually have a general orientation and are therefore suitable for a variety of application types. In contrast, domain-specific patterns focus on a specific application domain. The lack of general applicability of domain-specific patterns is affiliated with an easier handling and increased development productivity in the focused application domain [18, 19]. In software engineering, software components represent reusable program code fragments that could be used in various applications. Their usage generally increases the development productivity and improves the quality of the resulting applications. However, the identification of reusable components is challenging [20]. There is always a trade-off between usefulness on the one side and costs and quality on the other side [20]. Transferred to our context we understand user interface components as reusable building blocks to implement the user interface of dashboard applications on smartphones.

Similar to general software components the challenge is to identify a suitable degree of reusability with these components. In our approach we try to address this challenge by identifying recurring functionalities first. These are used to implement UI components afterwards.

4 Use-Cases of Existing Dashboard Applications on Smartphones

There are already several dashboard applications for smartphones available on the market. For our analysis we used a market overview of mobile dashboard applications from Gartner [21]. This report provides an overview of the most important vendors and their corresponding applications. Some vendors decided to implement native applications that are specialized on one particular mobile operating system like iOS or Android. Others use web technologies in order to implement cross-platform applications. This aspect is important for our analysis, because native applications can generally use a more extensive collection of UI elements. The report of Gartner revealed that the most dashboard applications are currently implemented as native iPhone applications. Therefore we decided to focus our analysis on the iPhone applications mentioned in the Gartner report. In summary, our analysis comprised twelve iPhone applications. Table 1 shows an overview of the analyzed applications, including vendor name, application name and version number.

Table 1. Analyzed dashboard applications for the iPhone

Vendor	Application Name	Application Version
Actuate	BIRT Mobile Viewer	3.2
Enterprise Signal	SurfBI	3.4
Extended Results	PushBI	3.5.1
Information Builders	Mobile Faves	2.2.0
Jaspersoft	JasperMobile	1.1.1
MeLLmo	Roambi Analytics Visualizer	4.4.3
Microstrategy	Microstrategy Mobile for iPhone	9.2.1.2.8
Oracle	Oracle Business Intelligence Mobile	11.1.1.5.0
QlikTech	QlikView	4.0.4
SAP	SAP BusinessObjects Explorer	4.0.8
SAP	SAP BusinessObjects Mobile	4.2.5
Yellowfin	YellowfinBI	6.12

In our analysis we examined the listed applications according their provided functionalities. Our results revealed that the analyzed applications offer a similar set of functionalities. The identified functionalities are illustrated using an UML use-case diagram in figure 2.

A large part of the identified functionalities is related to dashboard elements. According to [6] a dashboard element is a building block of a dashboard that visualizes a certain piece of information. In the analyzed dashboard applications the available dashboard elements are often presented in a single-column table (Use-Case: List Dashboard Elements).

Sometimes it is possible to sort the list according a certain criteria like in alphabetic order (Use-Case: Sort Dashboard Elements). In many cases a text based search for a certain dashboard element is provided (Use-Case: Search for Dashboard Element). In order to view a certain dashboard element the corresponding dashboard element must be selected within the table (Use-Case: View Dashboard Element). For the sake of organizing the dashboard elements some applications offer the possibility to manage a set of often used dashboard elements as so called favorites (Use-Case: Manage favorite Dashboard Elements). The visualized data is often received from a backend system such as a business intelligence server. Therefore, many of the analyzed applications offer the opportunity to update the displayed data by pressing a button or perform the update activities automatically (Use-Case: Show metadata of Dashboard Element). The most applications provide a set of parameters to maintain the access information to the associated backend system (Use-Case: Manage Backend Servers). Some applications provide the possibility to add comments to a dashboard element (Use-Case: Manage Comments). In several applications it is possible to share a dashboard element. Possible opportunities are via e-mail or as message using the multimedia messaging service (MMS) (Use-Case: Share Dashboard Element). Due to the limited space on smartphone displays some applications provide additional information about the dashboard element through separate screens or pop-up windows. Examples for this type of information are author or creation date of the dashboard element.

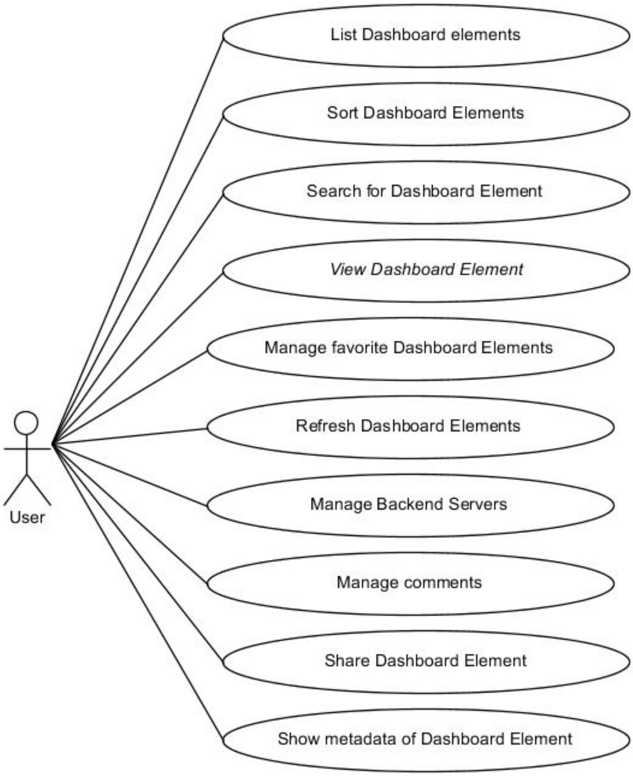
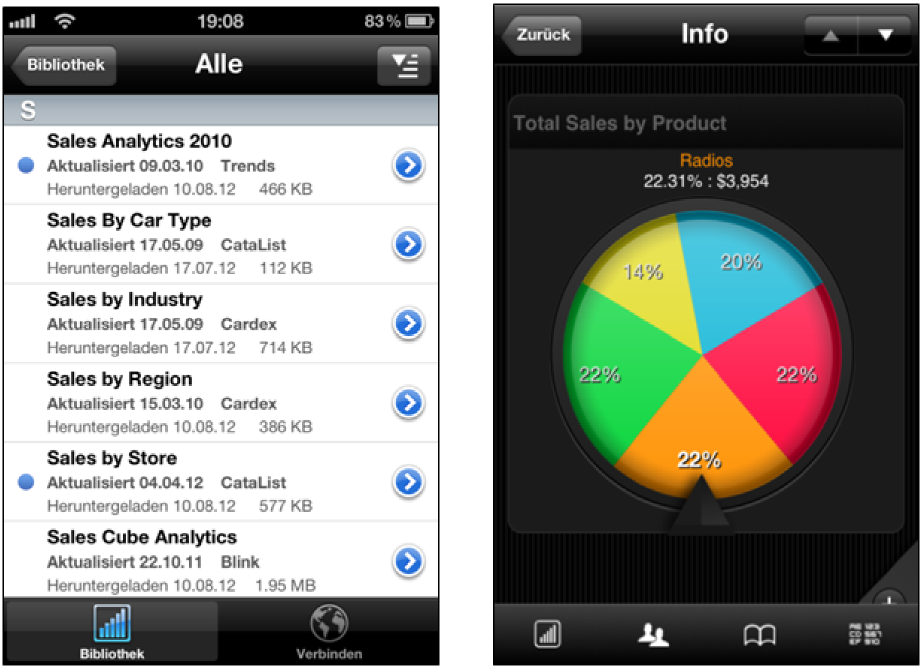


Fig. 2. Identified functionalities of dashboard applications on smartphones

Figure 3 illustrates an example of the use-cases List Dashboard Elements (left screen) and View Dashboard Element (right screen) using the application Roambi Analytics Visualizer as an example.



Sales Analytics 2010
Aktualisiert 09.03.10 Trends
Heruntergeladen 10.08.12 466 KB

Sales By Car Type
Aktualisiert 17.05.09 CataList
Heruntergeladen 17.07.12 112 KB

Sales by Industry
Aktualisiert 17.05.09 Cardex
Heruntergeladen 17.07.12 714 KB

Sales by Region
Aktualisiert 15.03.10 Cardex
Heruntergeladen 10.08.12 386 KB

Sales by Store
Aktualisiert 04.04.12 CataList
Heruntergeladen 10.08.12 577 KB

Sales Cube Analytics
Aktualisiert 22.10.11 Blink
Heruntergeladen 10.08.12 1.95 MB

Fig. 3. Exemplary Use Cases using the application Roambi Analytics Visualizer as example

5 Implementing Dashboard User Interface Components for the iPhone

The proposed UI components are aimed to be reusable solutions for the previously identified use-cases of dashboard applications on smartphones. Overall we implemented six UI components. In this section we give an overview of the developed UI components and describe one component, the strategic & operational dashboard component, in more detail.

5.1 Implementing User Interface Components for the iPhone

Due to the different programming languages and libraries that are used to implement smartphone applications it is necessary to focus on a specific mobile platform. The iPhone was selected because it has been already used to identify our use-cases described in the previous section. The reason for choosing the iPhone was the fact that most of the dashboard applications were available for the iPhone when compared with other platforms. Applications for iOS are developed using the programming language Objective-C. The user interface is implemented with a set of programming libraries called Cocoa Touch. The used development environment is called Xcode.

All components can be integrated into new applications using a provided set of programming interfaces.

5.2 Overview of the Implemented Dashboard Components

In order to identify the functionality of the implemented UI components we used our previously proposed use-cases as foundation. Afterwards we analyzed, which use-cases have a semantic relationship with each other. This resulted in six UI components that are illustrated using an UML package diagram in figure 3.

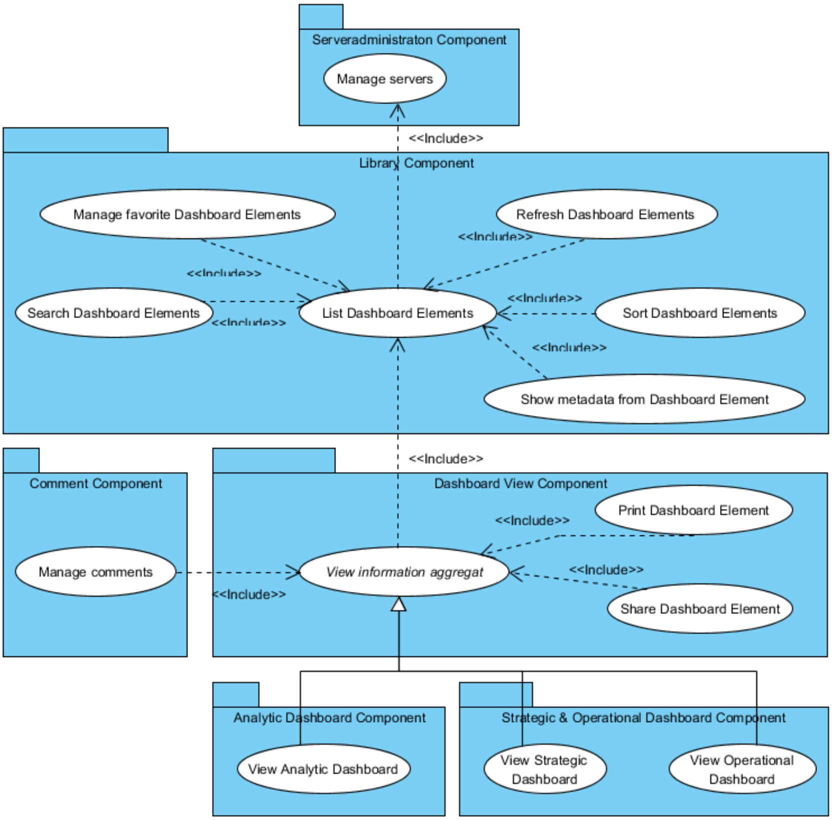
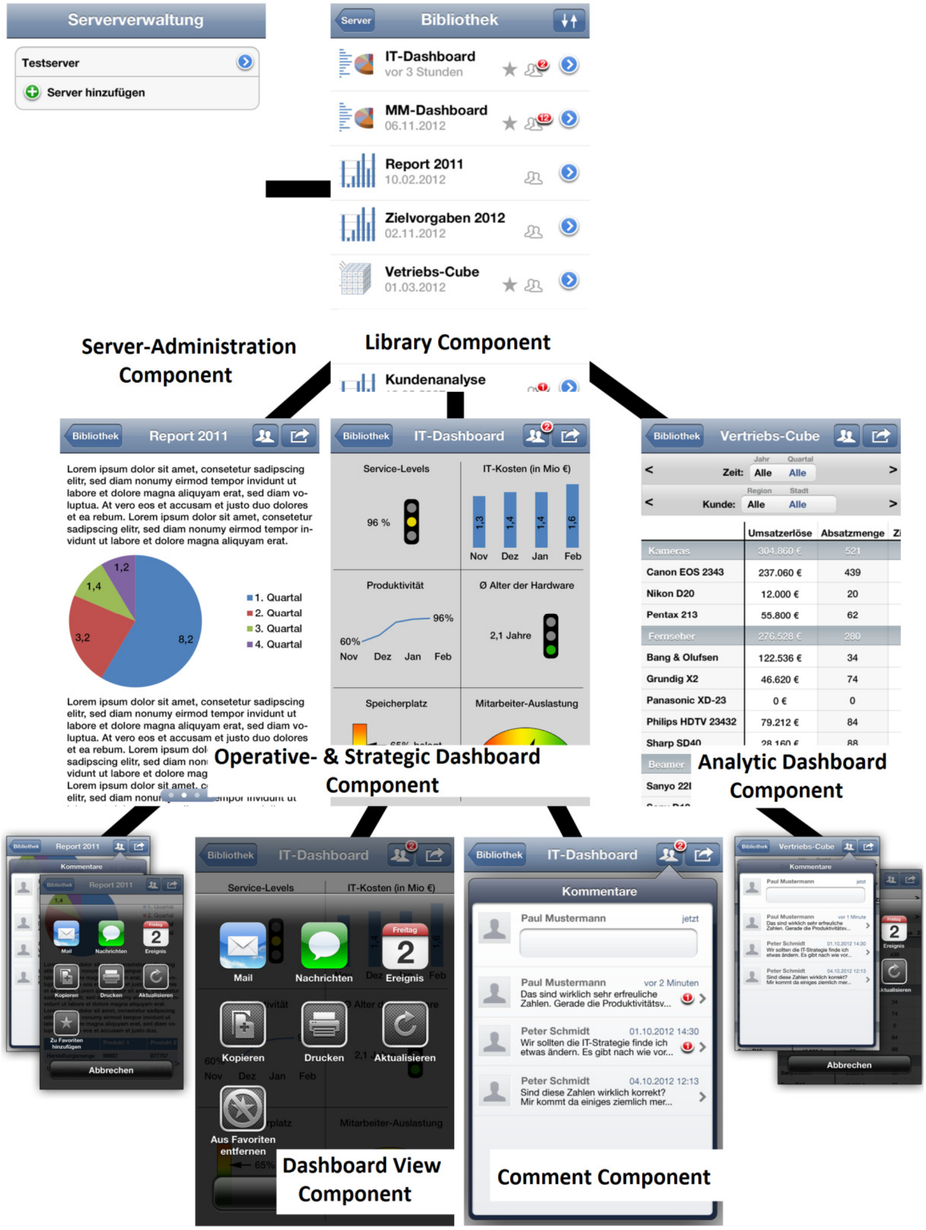


Fig. 4. Developed UI Components (Source: Own illustration)

The *Server-Administration Component* is responsible for the configuration of the backend connections. Configurable parameters are usually the network of a business intelligence server, used network port, used network protocol, username and password. The component offers the possibility to manage the connection data of several business intelligence servers.



IT-Dashboard
vor 3 Stunden

MM-Dashboard
06.11.2012

Report 2011
10.02.2012

Zielvorgaben 2012
02.11.2012

Vetriebs-Cube
01.03.2012

Kundenanalyse

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Lorem ipsum dol sadipscing elitr, sed diam non vidunt ut labore et dolore mag Lorem ipsum dolor sit amet, c elitr, sed diam nonu empor invidunt ut

Zeit: Jahr Alle Quartal Alle

Kunde: Region Alle Stadt Alle

	Umsatzerlöse	Absatzmenge	Zi
Kameras	304.860 €	521	
Canon EOS 2343	237.060 €	439	
Nikon D20	12.000 €	20	
Pentax 213	55.800 €	62	
Fernseher	276.528 €	280	
Bang & Olufsen	122.536 €	34	
Grundig X2	46.620 €	74	
Panasonic XD-23	0 €	0	
Philips HDTV 23432	79.212 €	84	
Sharp SD40	28.160 €	88	
Beamer			
Sanyo 22I			

Paul Mustermann jetzt

Paul Mustermann vor 2 Minuten
Das sind wirklich sehr erfreuliche Zahlen. Gerade die Produktivitätsv...

Peter Schmidt 01.10.2012 14:30
Wir sollten die IT-Strategie finde ich etwas ändern. Es gibt nach wie vor...

Peter Schmidt 04.10.2012 12:13
Sind diese Zahlen wirklich korrekt? Mir kommt da einiges ziemlich mer...

Fig. 5. Overview of the implemented UI components (Source: own illustration)

The *Library Component* manages the available dashboard elements. This is done within a single-column table. Each row of the table represents a selectable dashboard element. It is search for a specific dashboard element through keywords and to sort the list of available dashboard elements through attributes like used chart type or the name of the dashboard element.

The *Dashboard View* component is responsible for displaying the selected dashboard element. Depending on which type of dashboard element is selected, Analytical Dashboard Component or the Strategic & Operational Dashboard component is used to display the dashboard element. Thus, the Dashboard View Component is an abstract container for visualizing dashboard elements that provides common functionalities like share dashboard elements as pictures through e-mail or as short message or send the picture of a selected dashboard element to the printer.

The *Comment component* offers the possibility for the user to add some additional information to a dashboard element. An example could be additional explanations for the dashboard element and its visualized figures.

The *Analytical Dashboard* component aims to support data analysts, who need more sophisticated interaction possibilities like drill-down activities for selected datasets. This component is described in more detail in the next section of this paper.

The *Strategic & Operational Dashboard* component implements dashboard elements that support strategic decisions and operational activities. Although, both tasks are different, the used dashboard elements are pretty similar. The data is visualized with simple, easy to understand diagrams and in some cases enhanced with additional descriptions in text form. Sophisticated interaction possibilities like in the Analytical Dashboard component are not possible.

Figure 4 illustrates an overview of the user interface of the implemented components.

5.3 Example: Analytic Dashboard Component

In order to describe our implemented dashboard components we used the format of UI patterns to describe the components in a structured way. We also used the Apple's OS X documentation sets to generate an Xcode documentation that describes the usage of our components more detailed from a programmer's perspective. In the following we describe one of our components, the Analytical Dashboard Component, using the UI pattern format:

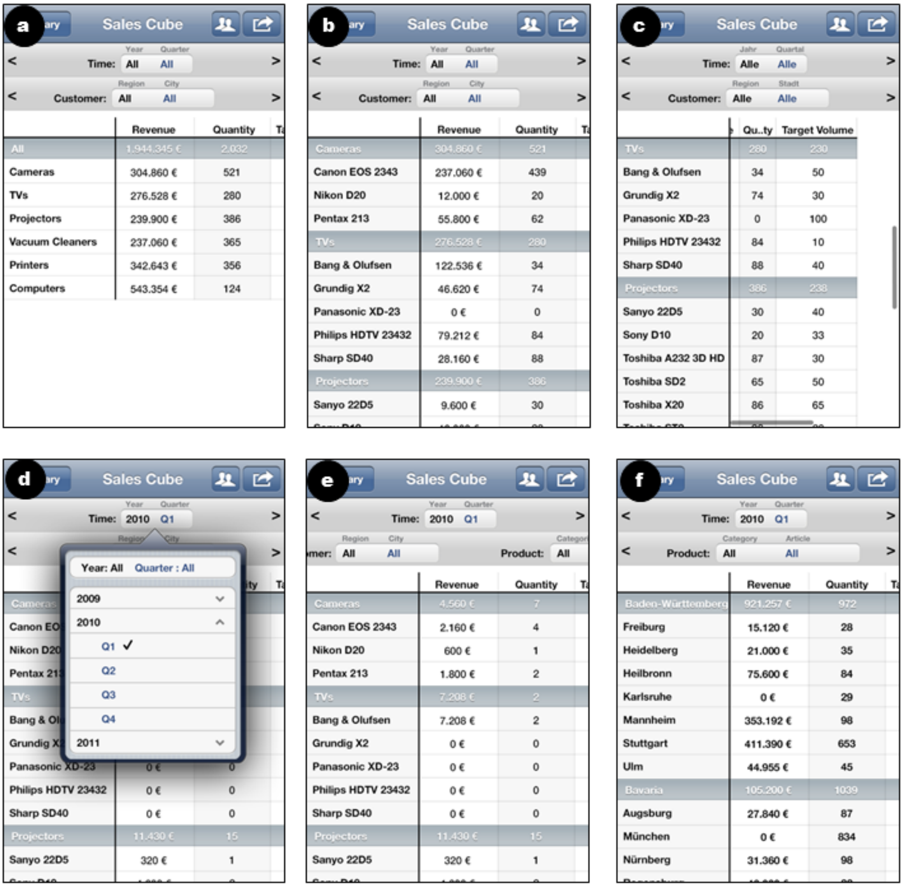
Name: Analytic Dashboard Component

What: The Analytical Dashboard component pattern can be used to perform sophisticated interactions with a data cube. It is possible to navigate within the predefined data space with the help of multiple interaction possibilities that are known from Business Intelligence applications like drill-down and roll-up, slicing and dicing or rotating the data cube [22].

Use When: The component can be used in analytical dashboard elements, in order to support the data analysis with the mentioned interaction possibilities on a selected data cube.

Why: Data cubes are well-known in business intelligence as well as the mentioned operations drill-down, roll-up, slicing and dicing, etc. The Analytical Dashboard component implements a data cube using a table UI element to display one dimension and additional superordinate row elements to navigate to the other dimensions of the cube.

Examples: Figure 5 illustrates examples of the offered interaction possibilities of the Analytical Dashboard component. The exemplary data cube has the dimensions time, customer and product. The table shows the revenues, sales and target volumes for individual products and product groups. Using a zoom gesture on the view shown in figure 5a the user can perform a drill-down. This results in a more detailed data. In the resulting view in figure 5b a pinch gesture can be used to perform a roll-up again.



a

Time: Year All Quarter All
Customer: Region All City All

	Revenue	Quantity	Ta
All	1.944.345 €	2.032	
Cameras	304.860 €	521	
TVs	276.528 €	280	
Projectors	239.900 €	386	
Vacuum Cleaners	237.060 €	365	
Printers	342.643 €	356	
Computers	543.354 €	124	

b

Time: Year All Quarter All
Customer: Region All City All

	Revenue	Quantity	Ta
Cameras	304.860 €	521	
Canon EOS 2343	237.060 €	439	
Nikon D20	12.000 €	20	
Pentax 213	55.800 €	62	
TVs	276.528 €	280	
Bang & Olufsen	122.536 €	34	
Grundig X2	46.620 €	74	
Panasonic XD-23	0 €	0	
Philips HDTV 23432	79.212 €	84	
Sharp SD40	28.160 €	88	
Projectors	239.900 €	386	
Sanyo 22D5	9.600 €	30	

c

Time: Jahr Alle Quartal Alle
Customer: Region Alle Stadt Alle

	Qu..ty	Target Volume
TVs	280	230
Bang & Olufsen	34	50
Grundig X2	74	30
Panasonic XD-23	0	100
Philips HDTV 23432	84	10
Sharp SD40	88	40
Projectors	386	238
Sanyo 22D5	30	40
Sony D10	20	33
Toshiba A232 3D HD	87	30
Toshiba SD2	65	50
Toshiba X20	86	65

d

Time: Year 2010 Quarter Q1

Year: All Quarter : All
2009
2010
Q1 ✓
Q2
Q3
Q4
2011

		ity	Ta
Cameras			
Canon EO			
Nikon D20			
Pentax 21			
TVs			
Bang & Ol			
Grundig X			
Panasonic XD-23	0 €	0	
Philips HDTV 23432	0 €	0	
Sharp SD40	0 €	0	
Projectors	11.430 €	15	
Sanyo 22D5	320 €	1	

e

Time: Year 2010 Quarter Q1
mer: Region All City All Product: Categor All

	Revenue	Quantity	Ta
Cameras	4.560 €	7	
Canon EOS 2343	2.160 €	4	
Nikon D20	600 €	1	
Pentax 213	1.800 €	2	
TVs	7.208 €	2	
Bang & Olufsen	7.208 €	2	
Grundig X2	0 €	0	
Panasonic XD-23	0 €	0	
Philips HDTV 23432	0 €	0	
Sharp SD40	0 €	0	
Projectors	11.430 €	15	
Sanyo 22D5	320 €	1	

f

Time: Year 2010 Quarter Q1
Product: Category All Article All

	Revenue	Quantity	Ta
Baden-Württemberg	921.257 €	972	
Freiburg	15.120 €	28	
Heidelberg	21.000 €	35	
Heilbronn	75.600 €	84	
Karlsruhe	0 €	29	
Mannheim	353.192 €	98	
Stuttgart	411.390 €	653	
Ulm	44.955 €	45	
Bavaria	105.200 €	1039	
Augsburg	27.840 €	87	
München	0 €	834	
Nürnberg	31.360 €	98	

Fig. 6. Exemplary operations implemented within the Analytical Dashboard component

Besides, there are some basic functionalities available. With swipe gestures the user can scroll horizontally or vertically trough the table (figure 5c). By tapping and holding on the left and right cell margins of the first row and subsequently swiping to the left or right, the user can change the column size. A double tap brings him back to the optimal column width. In addition, the user can also perform a slicing on multiple dimensions, which is called dicing [22]. To perform a slicing operation the user has to type on the field of the corresponding dimension first. This opens a pop-over dialog (figure 5d). Afterwards the user can choose how to slice the data cube. Figure 5e shows the result of the slicing process. By horizontal swiping on the dimensions shown at the top, the user can rotate the data cube. If the user pulls the customer dimension to the left, it is replaced by the remaining product dimension (figure 5f).

6 Outlook

In this paper, twelve selected mobile BI applications for the iPhone have been analyzed with the aim of identifying common functionalities. The analysis revealed ten use cases that were implemented across different applications. These use cases were grouped to UI components using their semantic relationship. Afterwards, the proposed components were implemented using the iPhone development environment Xcode, the programming language Objective-C as well as the necessary programming libraries like Cocoa Touch.

The implemented UI components can be used to implement new mobile BI applications. They can be parameterized through offered programming interfaces. The components are described using the format of UI patterns. These structured descriptions allow understanding their purpose. Additionally, the implemented components are also described using Apple OS X documentation sets, in order to describe their usage from a programmer's perspective.

Currently, the components are only implemented for the iPhone. Possible steps in the future could comprise the implementation of the elaborated reusable components also for other popular mobile platforms, like Android, BlackBerry OS or Windows Phone. Moreover, the implementation for tablet PC's like the iPad could be beneficial, although the larger form factor of tablet PC's should be considered in this case. In addition, HTML5-based programming libraries such as jQuery Mobile or Sencha Touch, which are becoming increasingly powerful, could be used to facilitate the implementation of cross-platform mobile BI components.

The usage of the proposed reusable components is currently done within the source code of the mobile BI application. For this purpose, programming skills are required. A useful future extension seems to be the development of a tool that avoids this prerequisite. Such a tool could enable non-programmers to develop or generate mobile BI applications for their purpose based on their own specification.

References

1. Tapadinhas, J.: Innovation Insight: Mobile BI Innovation Expands Business Analytics Boundaries. Gartner Research (2012)
2. Bensberg, F.: Mobile Business Intelligence. In: Bauer, H., Bryant, M., Dirks, T. (eds.) Erfolgsfaktoren des Mobile Marketing, pp. 71–87. Springer, Heidelberg (2009)
3. Zhu, X., Huang, Y.: A Framework for Mobile Business Intelligence Based on 3G Communication Environment. In: Jin, D., Lin, S. (eds.) Advances in FCCS, Vol. 2. AISC, vol. 160, pp. 75–81. Springer, Heidelberg (2012)
4. Rivard, K.C.: Doug: Are You Drowning in BI Reports? Using Analytical Dashboards to Cut Through the Clutter. DM Review 14, 26 (2004)
5. Few, S.: Information Dashboard Design. The Effective Visual Communication of Data. O'Reilly Media, Inc., Sebastopol (2006)
6. Hecht, S., Schmidl, J., Krcmar, H.: Xcelsius: Dashboarding mit SAP Business Objects. Galileo Press, Bonn (2010)
7. van Welie, M., van der Veer, G.C.: Pattern languages in interaction design: Structure and organization. In: Proceedings of Interact, Zuerich, Switserland (2003)
8. Shneiderman, B., Plaisant, C.: Designing the user interface: strategies for effective human-computer interaction. Addison-Wesley, Boston (2010)
9. Borchers, J.: A pattern approach to interaction design. Wiley, Chichester (2001)
10. Alexander, C., Ishikawa, S., Silverstein, M.: A pattern language: towns, buildings, construction. Oxford University Press, New York (1977)
11. Beck, K., Cunningham, W.: Using Pattern Languages for Object-Oriented Programs. In: Workshop on Specification and Design for Object-Oriented Programming, Orlando, FL, USA (1987)
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, New Jersey (1995)
13. Dearden, A., Finlay, J.: Pattern Languages in HCI: A critical review. Human Computer Interaction 21, 49–102 (2006)
14. Rijken, D.: The Timeless Way.. the design of meaning. SIGCHI Bulleting 6 (1994)
15. Tidwell, J.: Designing interfaces. O'Reilly, Sebastopol (2011)
16. Neil, T.: Mobile Design Pattern Gallery. O'Reilly, Sebastopol (2012)
17. Hoober, S., Berkman, E.: Designing Mobile Interfaces. O'Reilly, Beijing (2011)
18. Homann, M., Wittges, H., Krcmar, H.: Towards user interface patterns for ERP applications on smartphones. In: Abramowicz, W. (ed.) BIS 2013. LNBIP, vol. 157, pp. 14–25. Springer, Heidelberg (2013)
19. Merrnik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Computing Surveys 37, 29 (2005)
20. Caldiera, G., Basili, V.R.: Identifying and qualifying reusable software components. IEEE Computer 24, 61–70 (1991)
21. Gartner: Who's Who in Mobile BI. Gartner Research (2011)
22. Hecht, S., Jörg, S., Krcmar, H.: Xcelsius: Dashboarding mit SAP Business Objects. Galileo Press, Bonn (2010)