Bringing Usability to Industrial Control Systems

Marcus Reul
RWTH Aachen University
52056 Aachen, Germany
marcus.reul@rwth-aachen.de

CHI 2009, April 4 – 9, 2009, Boston, MA, USA
ACM 978-1-60558-246-7/09/04.

Abstract
Within my ongoing work at a manufacturer for industrial test systems, I want to show how domain-specific HCI design patterns can be introduced into an existing software development process. This paper describes the findings from the first step of my work – which is a contextual inquiry in the domain of test automation systems. Moreover, it gives an outlook to the further proceedings of my activities.

Keywords
Usability Guidelines, Industrial Control Software, HCI Design Patterns, Style Guides.

ACM Classification Keywords
H.5.2 User Interfaces: Theory and methods, H.5.2 User Interfaces: Style guides, H.5.2 User Interfaces: User-centered design.

Introduction
Current Industrial Plants suffers frequently from usability problems of their control software. A poor user interface often leads to frustrated users and lowers the productivity of the employees who must work with the system.

Common consumer products and websites have often a much better usability than industrial systems. For example, products from Apple and Philips are often praised for their good usability.

As explanation for this discrepancy, the differences in the software development processes can be mentioned. While manufacturer for industrial systems use mostly conventional process models (e.g., the waterfall model or the V-model), some companies of the consumer marked (e.g., Apple) have applied iterative process models, which are known to improve the usability of the resulting products [5,12].

Despite the application of these process models can improve the usability of the products, they are not widely accepted in the industry. One reason is that these models do not fit easily in the established development processes and quality management systems. Finally, because the results of these efforts do not appear immediately and qualities like usability are difficult to measure, it is not easy to justify these rearrangements of the development process in a cost-benefit calculation to the management.

Instead of restructuring the entire development process, there exist also the possibility to train the developers and make them sensitive for usability issues.

Usability guidelines are a common approach to deliver the knowledge about good user interface design to the developers. There exist a wide variety of usability guidelines but many of them are hard to apply in real development projects. Many of these guidelines are too abstract. Hence, developers have problems to recognize in which situation a specific guideline should be applied.

My approach is to create a set of concrete, domain-oriented human-computer interaction (HCI) design patterns [4] for the domain of test systems. These patterns should help the developers and product managers to make design decisions for the user interface.

During my work, I want to examine how these HCI design patterns can be integrated into the existing development process, which problems occur thereby and how they can be solved. To explore this method in practice, I collaborate with a manufacturer of industrial test systems.

Related Work

One prominent set of usability guidelines are the eight golden rules of interface design by Ben Shneiderman [13]. These are fundamental usability guidelines but because of their abstractness they are very hard to implement for developers who are generally not usability experts.

Usability guidelines can be classified by their abstraction level. Deborah J. Mayhew has proposed a differentiation between [11]:

- Platform guidelines
- Corporate guidelines
- Product family guidelines
- Product guidelines

The Apple Human Interface Guidelines [1] are a popular example for platform guidelines. These guidelines are created to assist the developers in developing applications for the Mac OS X operating system that have a consistent and familiar look and behaviour. They describe appearance and behaviour of standard user interface elements (e.g., windows, buttons...) and mechanisms (e.g., drag and drop). These guidelines also include layout rules for the user interface elements. Microsoft has developed its own user interface guidelines for its operating system [15].

While platform guidelines are still very abstract and general, product guidelines are highly specific and contain many product-unique characteristics. Because of their concreteness, they are easier to apply for the developers but only useful for the specific product, they were created for.

Product-family related usability guidelines are applicable for an entire product family but are still very concrete and domain-specific.

Other guidelines are device-related instead of platform-related. These guidelines go beyond system frontiers and express design principles that are valid for an entire class of devices. Devices have their very own characteristics that also reflect on the way, users are interacting with them. Examples are guidelines for mobile devices [7] or touch screens [8]. Mobile devices for example have small screen sizes, no pointing device, and limited computing power as common properties. These characteristics have major effects on how a good user interface for these devices should be designed. Proven approaches to solve reappearing problems that rely on such device-specific properties can be expressed in device-related user interface guidelines.

The International Organization for Standards (ISO) has published its own standards for creating usable user interfaces [10]. For example: ISO 9241-10 focuses on dialogue principles related to the ergonomic design of the dialogue between the user and the interactive system. ISO 9241-151 contains guides for the creation of usable web sites and web applications. ISO 9241-12 contains guides for the arrangement and presentation of data on a screen.

Beside usability guidelines, there exists another methodology for transferring usability knowledge. Human-computer interaction (HCI) design patterns capture common user interface design problems together with a proven solution in a consistent format.

A pattern can reference on other patterns so that an entire pattern language can be created [4]. Prominent pattern collection are Jenifer Tidwell’s patterns for effective interaction design [14] or Douglas K. van Duyne’s patterns for web sites [6]. Another example for the versatility of HCI design patterns are Staffan Björk’s patterns for game design [3].

J. Tidwell’s patterns are very general and universally usable for nearly all kinds of graphical user interfaces. They describe popular interaction principles and specify in which situation they should be used.

D.K. Duyne’s patterns for web sites is a collection of principles and methodologies for web design. This compilation helps web developers to find the right interaction method for a certain task.

Scott Henninger has done research in the area of organization specific user interface guidelines [9]. He arguments that usability guidelines need to be augmented with context-specific guidelines and examples to fully unleash their power. Examples are important to illustrate the different application possibilities. His studies have shown that examples were referenced more frequently by the developers than the guidelines itself.

Furthermore, he has created methods and tools for developing and managing user interface guidelines inside an organization. This includes rules for selecting appropriate guidelines by means of project characteristics and methods for enhancing and refining existing guidelines and patterns during their lifetime.

Although there exist a big variety of user interface guidelines, reports about their application in practice are very rare.

Domain Analysis

To develop usability guidelines for a specific domain, a profound knowledge of the field and the users is indispensable.

Industrial control systems are used to command and monitor a plant. These systems are generally controlled by trained users. Frequently, the systems have to fulfill safety and reliability standards. These systems are often placed in a noisy environment inside a big factory floor.

In most cases, the test systems are used to test a prototype of a new developed product or an established product that has to be optimized. In some other cases the test systems are used to check whether a certain product meets some specific requirements or standards.

To get a deeper understanding of the users and their tasks, I have met some users in their working environment and talked with them about their job. This interviewing method is called “contextual inquiry” and was presented in [2].

In the company where I have done these studies, there exist two major user groups of automated test systems: test operators and project engineers. Until now, I have met five test operators and one project engineer. Further interviews with project engineers are planned.

The test operators conduct the test system during the actual test. They control the plant and monitor various parameters. During automated tests, they supervise the system and handle the test results. They are responsible for a trouble-free execution of the test. Most test operators have a profound knowledge of the concerning test object. They can predict the behaviour of test object, which is necessary to act appropriate in safety-critical situations.

An improperly handling can cause a defect of the test object (which is in many cases an expensive prototype that is hard to replace). In special cases it even can compromise the safety of the persons who are working right next to the test-cell.

The project engineers supervise the project for which they are responsible. They create the test schedules and manage the resources that are necessary for the

project execution. The project engineers have a profound scientific knowledge of the test object and create the test schedules based on sophisticated mathematic and statistic models. Therefore they use established mathematic engineering tools. During the actual execution of the test, they work hand in hand with the test operators.

During the contextual inquiry, I was able to deduct some requirements that are important for the interviewed user groups.

For the test operators, it is important to have:

- at any time the full control over the system.
- all important values directly in the field of view.
- clear information about the system's state
- the ability of customizing the screen according to their own needs and the current test configuration.
- the possibility to see the trend of a measured value.

The project engineers have other requirements:

- being able to create and manage test schedules easily
- easy transfer of the data between test automation system and supporting mathematic tools

The contextual inquiry was also done to identify problems in the existing software. One big issue of the current system is that some actions can only be done if the system is in a specific mode. However, the switch between these modes is very time consuming (in special cases it takes up to 30 minutes), which is very frustrating to the user.

This is an example of an usability problem in the existing software. However, such problems could be avoided by applying the right usability guidelines or patterns during the development process.

Work Outline

As a next step, I will gather different existing guidelines (like [1,10,15,16]) and use them as a starting point for the creation of the design patterns for test automation systems.

I will select a set of guidelines and patterns that are appropriate for the concerning domain. Based on this, I will create some domain-specific HCI design patterns. Each pattern will contain some concrete examples, which show how the pattern can be applied in a user interface of the field. To assure that the selected patterns are relevant for user interfaces of test systems, I will talk to experts from the domain and discuss the patterns with them.

The applicability of these patterns will be tested within a real software development project. Therefore the patterns will be delivered to the developers and will be applied by them during an ongoing software development project without the additional help of a usability expert.

Afterwards, I will conduct some interviews with the developers to find out if problems occurred during the application of these patterns and how they could be solved. The results of these interviews can be used for further improvements of the design patterns.

An expert review of the created software will show if the developers have applied the HCI design patterns in a manner that improves the usability.

Acknowledgements

Special thanks to my research advisors Prof. Dr. Jan Borchers and Jonathan Diehl for their support and recommendations for improvement. Furthermore, I thank all responsible persons and interview partners at the manufacturer for test systems – especially my advisor.

This work was funded in part by the German B-IT Foundation.

References

[1] Apple Human Interface Guidelines. http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/index.html Apple Inc., 2008.

[2] Beyer, H. & Holtzblatt, K. Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, San Francisco, CA, USA, 1998.

[3] Björk, S., Lundgren, S. & Holopainen, J. Game Design Patterns. In Copier, M. & Raessens, J. Level Up Proceedings of Digital Games Research Conference Utrecht, The Netherlands, 2003.

[4] Borchers, J. A pattern approach to interaction design. Wiley, West Sussex, UK, 2001.

[5] Dix, A., Finlay, J., Abowd G., and Beale, R. Human-Computer Interaction. Pearson, Essex, UK, 2004.

[6] Duyne, D.K.V., Landay, J., and Hong, J.I. The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley, Boston, MA, USA, 2002.

[7] Gong, J. & Tarasewich, P. Guidelines for Handheld Mobile Device Interface Design. Proc. of the Decision Sciences Institute (DSI) Annual Meeting, pp. 3751-3756, Boston, 2004.

[8] Haywood, A. & Reynolds, R. Touchscreen: Usability Guidelines. Serco Usability Services, London, UK, 2008.

[9] Henninger, S. Creating organization-specific usability guidelines. CHI '97 extended abstracts on Human factors in computing systems. ACM, New York, NY, USA, 1997.

[10] ISO 9241: Ergonomic requirements for office work with visual display terminals. International Organization for Standardization, 1998.

[11] Mayhew, D.J. The usability engineering lifecycle: A Practitioner's Handbook for User Interface Design. Morgan Kaufmann, San Francisco, CA, USA, 1999.

[12] Nielsen, J. Usability Engineering. Morgan Kaufmann, San Francisco, CA, USA, 1994.

[13] Shneiderman, B. Designing the User Interface. Addison-Wesley, Boston, MA, USA, 2005.

[14] Tidwell, J. Designing Interfaces. O'Reilly, Sebastopol, CA, USA, 2006.

[15] Windows User Experience Interaction Guidelines. http://msdn.microsoft.com/en-us/library/aa511258.aspx, Microsoft Corporation, 2008.