USER INTERFACES IN BUSINESS INFORMATION SYSTEMS – PROBLEMS AND SOLUTIONS

Margarita Atanasova, Anna Malinova

Abstract. Many business applications have usability problems that reduce employees’ effectiveness and performance. In this paper, important problems are presented based on the four main interested parties involved in business information systems – End-users, Developers, Software Company Owners/CEOs and Business Owners. Results from paper and article reviews, live interviews and usability test sessions with the abovementioned target groups are gathered and summarised. Adaptive and context-aware user interfaces are presented as a possible solution for most of the problems. As model-driven development emerged as a leading approach for enabling rapid, collaborative development and building adaptive UIs, four generation of model-based user interface development systems, architectures and frameworks are briefly reviewed in the third section of the article.

Key words: adaptive user interfaces, context of use, usability, business requirements, model-driven development

1. Introduction

The desire of owners of small, medium and large businesses to increase their competitiveness in emerging markets makes them demand high quality software products to help their employees carry out their daily business activities without being slowed down by a poorly designed software user interface. Many business applications have problems with usability and cause inconvenience and frustration among their users. Companies increase their awareness of how important is monitoring and improving the performance of software applications for critical

business processes. The effectiveness of the selected strategy's implementation for dealing with problems depends largely on the approach used to assess performance.

The ISO 9241-11 standard defines usability as "the extent to which a product can be used by specified users to achieve specified goals with **effectiveness**, **efficiency** and **satisfaction** in a specified context of use". There are several metrics, evaluating the usability of a product and the quality of the user experience [1, 25, 26]. The authors Tullis and Albert [26] organize dozens of metrics in six categories: performance, issue-based, self-reported, web navigation, derived, and behavioral/physiological. These indicators and their measurement can lead to change in the final product, thus improving the user experience and the usability of the software.

When making a usability test session with users, the user experience designer makes measurements of:

- How effectively users can complete a set of tasks.
- How much time is needed to complete a task.
- What mistakes are made during the execution of a task. Mistakes are useful in pointing out confusing or misleading parts of a product, process or interface.
- Efficiency – amount of effort that a user spends to complete a task.
- Learning curve of the product – whether performance improves with time.
- What prevents the user to complete a task.
- What distracts the consumer; What creates confusion for the consumer.
- What is the reason the user does not notice something.
- Why the user believes that a task is completed when it is not.
- Why user performs wrong actions.
- Why the user misinterprets the content.
- Why the user does not understand navigation.
- Positive and negative comments; Suggestions for improvements.
- Questions, expectations and confusions.
- Non-verbal actions, such as facial expressions and body language.

One of the main usability problems is the complexity of user interfaces in business information systems. Each specific business requires specialized user interface that will help users perform their daily tasks efficiently. This is one of the requirements of modern business to user interfaces [2]. A single user interface could not cope with this diversity without creating large amount of information that leads to the abovementioned complexity problem. To adapt to the specifics of the business that bought their information system and to improve the usability of their interfaces, software companies invest large resources in the development of additional screens,

once the product has been launched. This is due to the fact that during the planning and design phases developers and architects cannot provide all possible cases. That way the cost of development increases as well as the cost of maintenance of the software.

Adaptive user interfaces are subject of many studies as it is referred as a solution to some of the usability problems in the software products [10, 12, 13]. Adaptivity of the user interface depends on the specific system requirements, expectations and needs. In other words, adaptive user interface is a user interface that understands the context of use and is capable of creating response to the changes in this context [3]. This response may vary from changes in the structure of the application to changes in its behavior. The context of use represents users, tasks and equipment (hardware, software, materials), as well as the physical and social environment in which a system, product or service is used [4].

In the next section main problems of end-users, developers, company and business owners are identified. In the third section of this paper model-driven user interface development and adaptive user interfaces are presented as one of the possible solutions for these problems.

2. Identifying the Main Problems

One of the main problems in business information systems manifests itself in terms of their usability and user-friendliness. Usability is defined as the degree to which a product can be used by specified users to achieve goals effectively in specific contexts of use. [5] This definition shows that the usability of a product depends not only on the functionality of the system, but also on the circumstances (context) in which it is used. We will overview the essential problems facing the four main interested parties involved in business information systems.

2.1. End-user Problems

In large business information systems, there are a number of usability problems [6]:

- It's **hard finding a certain functionality** or users identify problems with navigating business applications. This problem is most often addressed by beginner users.
- **Difficult understanding or memorizing the sequence of operations** required to perform a specific business process. This problem is due to the fact that the system is not aware of specific business processes in the enterprise, thusthere is no way to successfully "lead" the user through the

process. The employee must identify himself with a series of queries, which often leads to errors. In the case described in the article cited above, to complete a certain business processes some of the employees printed and laminated sequence of actions. The solution to this problem is the system continuously providing some kind of assistance or support to let the users know that they are on the right track.

- **Problems in the execution of operations.** Users often report that the interface of operations is unnecessarily complicated. In some cases, even after the training, they printed aids describing the sequence of actions to perform an operation.
- **Limited output type**. To obtain the sample of the data they want, some users export data from the system and open the file in other applications, such as Microsoft Excel, since they cannot do the necessary actions within the business information system. Users consider this limited functionality of the system itself.
- **Lack of support when an error occurs.** One of the most common problems is when users use the system and receive incomplete or incomprehensible error message. Developers often use common messages that do not give accurate information to the user. This means that the system is not adapted to the specific error that occurred.
- **Problems with terminology**. The problem of terminology is not easily solved because each company has its own vocabulary. The use of universal terms is not a solution. However, the system developers often are not familiar with the specifics of the business in detail and use universal terms.
- **Excessive complexity of the system.**

Another study on the usability of the business applications is made by International Data Corporation (IDC) in Denmark, Norway and Sweden [7]. It includes 300 users from different companies with annual revenues exceeding $100 million. The results show that 40% of the participants **find business applications difficult to use**. Another paper indicates that business applications do not support adaptive user interface to users with different skill levels, such as beginners and experts [8].

In a research project Pierre Akiki [9] divided a control group into three categories: beginners, advanced and experts. Determination is through an anonymous questionnaire in which each user completes information upon which to determine that user's category. Then all participants are shown different versions of a component of a user interface, their task is to choose the one they like the most. The results showed significant differences between categories of users, which once again shows the need to adapt the user interface to their preferences and skill level. This would significantly increase the usability of the product.

2.2. Developer Problems

- Inability to plan all possible situations and screens during the design of the software.
- Inability to quickly and easily transform one screen to another, adapted to the specific needs of the enterprise.
- Lack of appropriate software for designing adaptive user interfaces.
- Too complicated to use and understand tools for model-based software development.
- Absence of a structured approach for designing adaptive user interfaces. There are several steps in designing such – define the task and domain model, then generate an abstract user interface (usually described by XML based language) and finally generate concrete and final user interface [10]. This approach helps in creating various screens according to the context of use more easily.
- Developers are not familiar with the specifics of the business for which they develop the system.
- The adaptation of the user interface to the context of use requires mapping pattern from the tasks model to different presentation models considering the design of the platform. Implementing ready-to-use architectural and design patterns like MVC, MVC-U, MVP is not applicable without the required modification. Architectural models are often used in combination with other models and design patterns. The relationship between them should be taken into account when applying them into a system [27]. Therefore, problem here is the need to build a large number of models before actual construction.
- Lack of time. Usually medium and large software companies work on agile methodology for software development in which at the end of each sprint[1] the team of developers have a sprint demo where they present to their manager and customers what features were developed and tested during this time. This drastically reduces the time of planning and design, usually because customers are eager to see a working prototype.
- Lack of direct contact with actual end-users of the system. As developers and designers usually work with personas[2] or the client, they have no visibility on how the system will be used. This causes problems with usability, most of which could be removed after a usability testing session

[1] Sprint – the biggest period of time in which the software requirements remain unchanged, e.g 2 weeks.
[2] Persona – a descriptive, fictional model of a certain type of user, based on data from the consumer survey [11].

with real users. In some cases, however, the project does not have budget for that, which creates the abovementioned usability problems.

- A wide variety of system users. A system is commonly used by different types of users (beginners, advanced, experts), each with different preferences and needs.
- A wide variety of platforms over which the user interface should be integrated. This includes various methods of data input – voice, mouse, keyboard, touchpad, etc.
- Need for additional work on the automatically generated code. In most cases, after the automatic generation of code, the developers go through the code manually and rewrite some pieces of code. That makes the process of development not straightforward as described in the literature (modeling, generate an abstract user interface, generating specific and final user interface), but modeling, code generation, code correction, update the model, correcting followed errors in the model, re-generate and go through the same steps again.
- In model-based development there is no model debugging tool because the errors here are conceptual rather than bugs in the code

2.3. Software Company Owner / CEO problems

These results are summarized after interview sessions with three directors of two large software companies – one located in the USA, the other in Bulgaria. The American company is a leader in developing software for monitoring devices and reported income of $64 millions for 2014. Localized in Plovdiv, the Bulgarian company has over 200 employees. Its business is concentrated in developing business software, software components and web applications. The problems are:

- Use of large resources in the development of additional screens, once the product has been launched. This is due to the fact that during the planning stage all possible cases cannot be provided.
- **Maintainance cost increases** due to the need to support many different screens, cases and adaptations to the context of use.
- **Increase the cost of testing** the new screens and functionality required by the customer. Usually automated tests for the product are developed in the Quality Assurance department. With each new requirement of the customer the QA specialists have to write new tests. If the system is modular, old tests should be revised, and the cost and time for this is very high.
- **Clients find the software product too complex.** When creating a software product, the company may choose to sell the product as a whole or in individual modules. When the first approach is chosen, a lot of customers

find the product unnecessarily complicated, confusing them with many features that they will not use. The researched American company made it clear that the majority of customers use only 10% of the functionality of their product.

Need to quickly release the product on the market. This is the time it takes for a product to be approved, tested and ready for market. Reducing this time is especially important for the software industry, where products quickly become outdated. The need to quickly release the product on the market makes it nearly impossible for designers and developers to produce screens for different users that is why they develop single user interface for the average user.

2.4. Business Owner Problems

These results are summarized after paper and article and conducting an interview with a director of a large Bulgarian printing house which implemented an information system for personnel and tasks management.

- **Loss of funds due to the long implementation period** of the business information system. The employees in the researched printing house do not feel comfortable in updating their daily tasks in the new system, because they didn't find the system usable enough.
- **Costs of training staff to use the new system**. In the described case, the system do not have any built-in adaptation for novice employees, which is why there is a need for an additional training.
- **The high cost of purchasing a software adapted to the needs of the enterprise.** For small and medium businesses, the purchase of standard software is preferable because the price is lower.
- **A poorly designed software reduces employee productivity**, which again leads to loss of funds. The use of the product, which is a bit unintuitive or complicated to use, substantially reduces the effectiveness of the tasks in enterprises, which can result in loss of money.

As a whole business applications are often difficult to use for the average employees, costing companies millions of dollars. A single user interface that is not adapted to the needs and culture of the enterprise, reduces employee's productivity. The worst scenario is they to abandon the software product and continue to do daily business activities non-automated (manually).

4. Model-Driven User Interface Development

Adaptive user interfaces are presented as a possible solution to the usability problems in a way that the software product becomes more user-friendly and

efficient [12, 13]. Model-driven user interface development is a good way to deal with some of the aforementioned problems and challenges while reducing the effort and cost of development. This approach helps in identifying high level patterns, which allows designers to analyze and specify interactive software applications from a semantic point of view without focusing on the details of implementation [14]. Model-driven user interface development has many advantages in comparison with the traditional programming. They are not a focus of this article but can be found in literature [12, 15].

For developing adaptive user interfaces developers usually use model-driven approach [15, 16, 17]. The aim is to build a software application with well defined abstract levels, giving a way to describe different types of adaptations on the different abstraction levels.

Model-based user interface development is mentioned as early as the 1980s. In literature there are defined four MBUID generations [14, 15, 18, 19]. The first generation focused on the automatic generation of user interfaces, but does not support an integrated process for model-based development. Instead, one universal declarative model of user interface is used. Second generation MBUID systems (1995–2000) upgrade this model with other models which help in the semantic description of the user interface. In this generation task, dialog and presentation models are defined. Third generation systems (2000–2004) are focused on the development of the UIs for the diversity of platforms. Current, fourth generation (2004 – till now), focuses on the development of user interfaces, which are sensitive to the context of use and applicable to the wide range of available platforms.

One of the more notable projects from the past is an architecture for developing interactive applications called TRIDENT which first introduced the concept of abstract and concrete interactive objects [20]. The project MOBI-D [21] provides developers with a set of tools supporting the entire process of development of user interfaces – from extraction and modeling of tasks and mappings between them, through their integration into the presentation layer to the user testing.

Other authors present universal, declarative language called MARIA, which supports high-level abstractions and provides developers with a mechanism for generating different versions of the user interface at runtime of the application [22]. Using MIRA framework developers can define and implement semantic interfaces for web applications [23]. Semantic interfaces are defined as those who can understand and adapt to the presented data. In addition to this work Bertti and Schwabe present MIRA as JavaScript based library, which uses models that can describe responsive context-aware user interface. The only disadvantage of MIRA is that the library documentation is written only in Portuguese. The framework code is

open-source, uploaded on GitHub. Cedar Studio [15] and CIAT-GUI [24] generate user interfaces by describing patterns of tasks, abstract, concrete and final user interface. Cedar Studio provides designers a dragging functionality in the final UI for rearraging UI elements, but for the moment it is not a sufficient factor to achieve a competitive market appearance of the user interface.

5. Conclusion

In this article, we define four main categories of problems based on the main interested parties involved in business information systems – End-user, Developer, Software Company Owner/CEO and Business Owner. For the average employee business applications are often too complex and difficult to use. One of the main problems is also the amount of time they need to learn how to use the application. Chief executive officers define that some of their problems are caused by the long-period needed for the implementation of business information system which results in loss of funds. In conclusion, a single user interface that is not adapted to the needs and culture of the enterprise reduces employee's productivity. One of the main solutions to most of the problems is adaptive context-aware user interfaces. A system with easy-to-use, context-sensitive and intuitive user interface tailored to the needs and skills of the user could be used in its full capacity, thereby multiply the performance in the enterprise.

Acknowledgements

This work is partially supported by the IT15-FMIIT-004 project of the Scientific Fund of the University of Plovdiv "Paisii Hilendarski", Bulgaria.

References

[1] Seffah, A, Donyaee, M., Kline, R., Padda, H., Usability measurement and metrics: A consolidated model, *Springer Science + Business Media*, Inc. 2006, Software Qual J (2006) 14: 159–178, DOI 10.1007/s11219-006-7600-8.

[2] Atanasova, M., Malinova, A., Krushkov, H., Analysing the requirements of the modern business towards user interfaces, *International Journal "Information Theories & Applications" (IJ ITA)*, 2016, 359 – 375, ISSN 1310-0513 (printed).

[3] Fonseca, J., Model-Based UI XG Final Report, W3C Incubator Group Report 04, May 2010.

[4] Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: Context of use description.

[5] Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Document published on: 1998-03-19. Edition: 1

[6] Topi, H., Lucas, W., Babaian, T., Identifying Usability Issues with an ERP Implementation, *In Proceedings of the 7th International Conference on Enterprise Information Systems*, Miami, USA: SciTePress, pp. 128–133, 2005.

[7] Lykkegaard, B., Elbak, A., IDC – Document at a Glance – LC52T, International Data Corporation (IDC), 2011.

[8] Singh, A., Wesson, J., Evaluation Criteria for Assessing the Usability of ERP Systems, *In Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, Emfuleni, South Africa: ACM, pp. 87–95, 2009.

[9] Akiki, P., Bandara, A., Yu, Y., Integrating adaptive user interface capabilities in enterprise applications, Proceedings of the 36th International Conference on Software Engineering, pp. 712–723, 2014, ISBN: 978-1-4503-2756-5.

[10] Menkhaus, G., Pree, W., A Hybrid Approach to Adaptive User Interface Generation, Software Research Lab, Department of Computer Science, University of Salzburg, Austria, *Journal of Computing and Information Technology* – CIT 10, pp. 171–179, 2002.

[11] Marshall, R., Design and evaluation: End users, user datasets and personas, *Applied Ergonomics*, Vol. 46, Part B, pp. 311–317, 2015.

[12] Reinecke, K., Bernstein, A., Minder, P., MOCCA – A System That Learns and Recommends Visual Preferences Based on Cultural Similarity, International Conference on Intelligent User Interfaces (IUI), 2011.

[13] Akiki, A., Bandara, A., Yu, Y., Engineering Adaptive Model-Driven User Interfaces, IEEE Transactions on Software Engineering, 2016.

[14] Meixner, G., Paternò, F. & Vanderdonckt, J., Past, Present, and Future of Model-Based User Interface Development, pp. 2–11, 2011.

[15] Akiki, A., Bandara, A., Yu, Y., Adaptive Model-Driven User Interface Development Systems, ACM Computing Surveys, ACM, 47 (1), pp. 64:1–64:33, 2014.

[16] Criado, J., Vicente-Chicote, C., Padilla, N., Iribarne, L., A Model-Driven Approach to Graphical User Interface Runtime Adaptation, Workshops and Symposia at MoDELS 2010, Olso, Norway, October 3–8, 2010.

[17] Taktak, H., Riahi, I, Moussa, F., A Model Driven Approach for Adaptive User Interfaces Specification: User, Task and Environment Impact, ACHI 2016: The Ninth International Conference on Advances in Computer-Human Interactions, CRISTAL Laboratory, National School of Computer Sciences, Manouba University, Tunisia.

[18] Schlungbaum, E., Model-Based User Interface Software Tools – Current State of Declarative Models, Technical Report, 96–30, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 1996.

[19] Da Silva, P., User Interface Declarative Models and Development Environments: A Survey, *Proc. of the 7th International Conference on Design*, Specification, and Verification of Interactive Systems, pp. 207–226, 2000.

[20] Vanderdonckt J, Bodart F., Encapsulating knowledge for intelligent automatic interaction objects selection. InterCHI'93. ACM Press, 1993.

[21] Puerta, A., Maulsby, D., MOBI-D: a model based development environment for user centered design, Atlanta (EE.UU.), CHI'97, 1997.

[22] Paternó, F., Santoro, C., Spano, L. and D. Maria, A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments, *ACM Transactions on Computer-Human Interaction*, 16 (4), 2009.

[23] Bertti, E., Schwabe, D., MIRA: A Model-Driven Framework for Semantic Interfaces for Web Applications, In book: Web Engineering, DOI: 10.1007/978-3-319-38791-8_3, pp.40–58, 2016.

[24] Molina, A., Giraldo, W., Gallardo, J., Redondo, M., Ortega, M., García, G., CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems, *Advances in Engineering Software*, 52, 10–29, 2012.

[25] Usability Metrics by Jakob, 2001, https://www.nngroup.com/articles/usability-metrics/

[26] Tulli, T., Albert, W., *Measuring the User Experience – Analyzing, and Presenting Usability Metrics*, Imprint: Morgan Kaufmanne, 2008, ISBN: 9780080558264.

[27] Hristov, Hr., Krushkov, Hr., Model-View-Controller Architecture Helping in Teaching of Web Technologies, *Mathematics and Informatics Journal of Education Research*, Sofia, Year LVI, Vol. 4, 2013, pp. 368–383. (in Bulgarian)

Faculty of Mathematics and Informatics
Plovdiv University
236 Bulgaria Blvd,
Plovdiv 4003, Bulgaria
E-mail: margi_hk@yahoo.co.uk, malinova@uni-plovdiv.bg

ПРОБЛЕМИ И РЕШЕНИЯ ПРИ ПОТРЕБИТЕЛСКИТЕ ИНТЕРФЕЙСИ В БИЗНЕС ИНФОМРАЦИОННИТЕ СИСТЕМИ

Маргарита Атанасова, Анна Малинова

Резюме. Голяма част от бизнес приложенията имат проблеми с лекотата на употреба, което намалява ефективността на служителите и води до загуба на средства на компанията, интегрирала софтуерния продукт. В настоящата статия са разгледани съществените проблеми, които срещат четирите основни страни, имащи отношение към бизнес информационните системи – потребителите на системата, разработчиците, собствениците на софтуерни компании и собствениците на бизнес. Обобщени са резултатите от литературно проучване, както и от направени интервюта с целевите групи. В третата част на статията адаптивните потребителски интерфейси са представени като едно възможно решение на проблемите с лекотата на употреба, повишаване ползваемостта на софтуерните продукти, подобряване производителността на разработчиците и намаляване загубата на средства от страна на бизнеса. Разгледани са примерни системи, архитектури и работни рамки за моделно-базирана разработка на потребителски интерфейси като възможни решения на част от проблемите, посочени в статията.

Ключови думи: адаптивни потребителски интерфейси, контекст на употреба, ползваемост, бизнес изисквания, моделно-базирана разработка