

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Khanom, Sukanya; Heimbürger, Anneli; Kärkkäinen, Tommi

Title: Icon-based language in requirements development

Year: 2012

Version: Published version

Copyright: © the Authors & MATFYZPRESS, 2012.

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Khanom, S., Heimbürger, A., & Kärkkäinen, T. (2012). Icon-based language in requirements development. In Y. Kiyoki, & T. Tokuda (Eds.), Proceedings of the 22nd European-Japanese Conference on Information Modelling and Knowledge Bases (EJC2012) (pp. 20-24). MATFYZPRESS - Univerzity Karlovy.

Icon-based Language in Requirements Development

Sukanya KHANOM¹, Anneli HEIMBÜRGER and Tommi KÄRKKÄINEN
University of Jyväskylä, Department of Mathematical Information Technology, Finland

Abstract. Most errors and misunderstandings in requirements engineering and system development owe to poor communication between users and analysts. Icon-base language is an appropriate means to decrease the difficulty of communication in multi-user environments and among different user backgrounds. It is no wonder that visualized language using icons, symbols and graphics has had a positive reception by software projects in the area of requirements engineering. The primary purpose of this paper is to sketch a tool to support requirements development in multifaceted stakeholder environments in a wiki system. The paper also introduces some new means to apply visual language in the representation of situations or activities which refer to context, in requirements development.

Keywords. Icon-based language, graphic, visualization, stakeholders, requirements development.

Introduction

Requirements engineering has become a crucial component of software development in our sophisticated technological world. It is important because software development project are frequently undertaken without a good understanding of needs and desires of stakeholders. Consequently, performing basic requirements development activities elicitation, analysis, negotiation and validation, they can sufficiently reduce risk of project failure. To support such argument, CHOAS² reveals that good requirements engineering practices contribute more than 42% towards the overall success of a project, relatively, greater than other factors. In addition, according to Emam and Koru [1], project failures and cancellations fall into the requirements engineering phase due to the changes in requirements and scopes. Most of these failures are not found until late during the project or when the system has already gone live. One of the inevitable challenges in requirements development is that there are communication problems due to the differences in language, knowledge and culture. There are various approaches to solving this communication problem. Our paper introduces one of them, together with these key concepts: icon-based language, requirements development, context, multifaceted stakeholder environments.

Icons have been developed to aid thinking. Extended usage can evolve iconic set to more stylized and ultimately abstract representations as pictures from antiquity, maps from ancient Egypt and the geometry diagrams of Euclid [2], to mention but few.

¹ Corresponding Author.

² <http://kinzz.com/resources/articles/91-project-failures-rise-study-shows>

Requirements engineering in software development typically is classified into two main categories: *requirements development* and *requirements management*. There is a multitude of sub-items associated to requirements development; requirements elicitation, requirements analysis, requirements negotiation, requirements documentation, and requirements validation.

Generally speaking, the concept of context is defined as a situation at hand or as a task a user or team is performing [3]. *Context* includes high and low context, depending on the amount of information given in a communication [4].

When dealing with *multifaceted stakeholder environments*, cross-cultural or cross-linguistic, the unavoidable issue is communication problems. Therefore, it is essential that all communications are generated clearly, ethically, consistently, completely and in a timely fashion [5]. Culture resides in the way we interact with other individuals and with our environment in different situations. Cross-culture is composed of interaction among humans, between humans and machines, and between human and environment [3, 4, 6]. In multifaceted stakeholder environments, universal language is necessary to convey common understanding among different languages and cultures [7].

The following section introduces visual language and its challenges. Section 2 highlights the potential of icons in the context of requirements development. In section 3, we describe the implementation of icon-base language in the area of requirements development. Discussion and future research is presented in Section 4.

1. Visual Language

Requirements visualization is not new. There are several studies and techniques to demonstrate both the improvement in requirements engineering by applying visualized methods as well as the advantages of graphic in software development projects to support requirements engineering. Visualization is defined as a method of forming a mental vision, image, icon or picture of something not visible or present to the sight, or of an abstraction in order to make it visible to the mind or imagination [8, 9, 10]. Many fundamental visualization techniques that are widely accepted in both business and society have long been used in requirements engineering. These techniques typically include bar graphs, charts, and hierarchical structures, which are used regularly to aggregate large amounts of information into a single representation for shared understanding and swift absorption by stakeholders. Most regulars are nevertheless about prototype, storyboards, UML use case diagrams and mock-ups. Numerous researchers emphasize that day after day many companies encounter software project failure [1, 11] even though those visualization techniques are available for supporting requirements engineering practices. One example of visual difficulties is that use case can embody complex requirements for the system therefore detailed scenarios or specific circumstances need to be provided. When using existing visualizations such as Class diagram, UML or State chart the basic technical skill is required which sometimes non-technical users are unable to understand. As consequence, there is the room space for other types of visualization like icon language that can serve to enhance communication and understanding [10].

The deviation between icon-tool derived from this development and other sorts of implemented icons like iOS SDK is the sound of its purpose and usability. This icon-tool aims to distribute widespread receptions in requirements engineering taxonomies

and requirements development activities among receivers as those of the other public icons. For instance, icons appear in human life mainly as communication means and they range from icons that are used to operate devices, through icons that appear in public places, airport, hotel, maps and traffic signs. Many of these icons are intended to provide the same information everywhere, visually speaking across languages. Thus, Icon-based language can be recognized as a part of mapmaking communication [3, 4, 9, 10]. On the contrary, iOS SDK (Software Development Kit), formally iPhone SDK, is a native application for iOS allowing developers to make application for the iPhone and iPod Touch.

2. Icons in the Context of Requirements Development

Requirements development process plays an essential role in software development. Therefore, universal and practical methods are needed to make it as simple as possible and to permit all parties understand the purpose. An effective method to communicate and transfer common perception among multiple users in requirements development is icon/graphic.

Intelligent icons, which should be user-friendly, informal and interactive, enable us to receive a better understanding of stakeholders' needs and move us from the technical domain, in which many developers are most comfortable, into the real-world problem domain. We can take advantage of the unique characteristics of icons to construct requirements taxonomies to help drive elicitation design, validation and negotiation. To improve the structure of the requirements categories and requirements development activities, icon-base provides, instead of itemized individual requirements, a sequence of actions between the system and the user. It does a better job in encouraging the users of the system, in a sequential fashion, to accomplish their goals with alternatives and exceptions [12]. Figure 1 depicts two different requirements development activities in multicultural forms. People in one culture are frequently unable to understand another culture therefore icons are capable of reducing impediments connected to misunderstandings arisen due to cross-cultural background environments and providing a common virtual board that can be shared by distributed stakeholders.

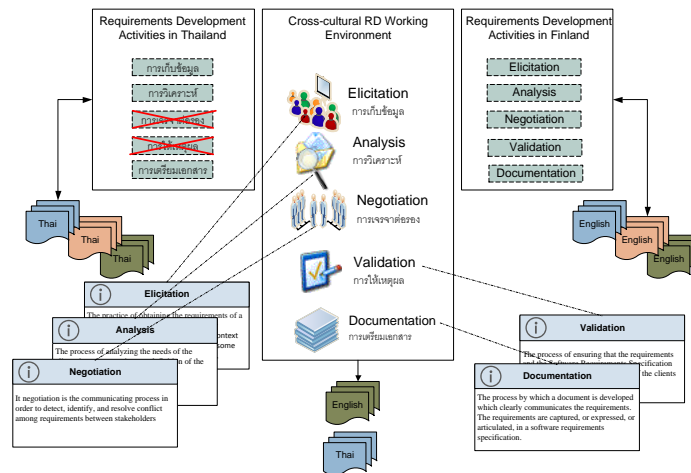


Figure 1. Examples of icons which represent the contexts of requirements development.

3. Towards Icon-Tool Implementation

Wikis as base software for the implementation are a great way to conduct the initial stage of requirements development process. They provide a means by which multiple stakeholders can work together to define requirements in an easy-to-access and easy-to-document manner, where collaboration is not only supported but encouraged by the nature of the technology itself. Figure 2 illustrates example icons used to represent each requirement type. The icons have embedded meaning in its representation. For example, when selecting of User Requirements, multi-subclass will be given by browsing information from wiki database. The demonstration offers link to requirements gathering with default information corresponding to a chosen requirement type such as requirements classification and requirements ID. The key driver of using wiki software in implementation icon-tool is because wiki provides easy page linking to reduce redundancy by making it easier to link content than to copy a page. Moreover, it supports historical page capture which strongly broads users for requirements tractability on a per-document [13, 14]. Specifically, it offers the way to handle and indicate misunderstandings and both expressed and unexpressed conflicts.

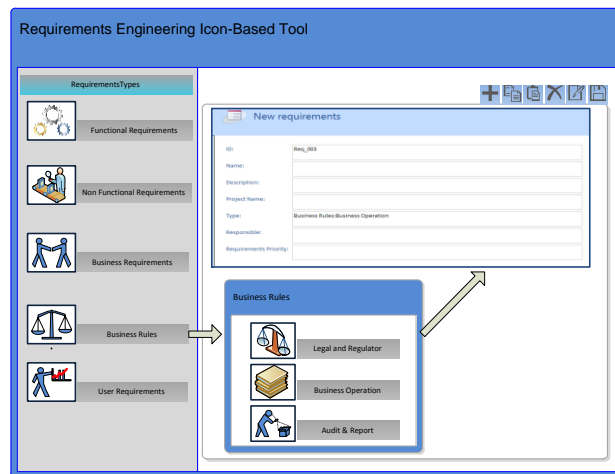


Figure 2. An example of icon-tool set as a framework in the context of requirements development in wiki.

4. Discussion and Future Research

Effective and efficient iconic artifacts are potentially able to reduce misconceptions and gaps in understanding by presenting many aspects of the requirements process. During the requirements phases, visualization can help in understanding the context for requirements development and provide the groundwork necessary for any requirements development process. To this we should add the importance of the development of visualization in promoting elicitation, negotiation, documentation, verification and validation in requirements development tasks.

Future research will focus on the construction of a set of icons providing a fundamental perception among multifaceted stakeholders especially, multicultural practitioners. The starting milestone for establishing an icon-tool project is to represent

a set of icons in requirements engineering categories to address an ease of understanding and adoption. Furthermore, future research desires can also be kinds of requirements development activities and point of views identified to be supported by icons. The main contributions are as follow.

- The set of icons are built to support requirements development works and mitigate the communication gaps among stakeholders. The concept of icon sets will be developed based on both theoretical research and industrial survey.
- The identification of icons' characteristics to help clarify requirements development activities and hence enabling to transfer enough understanding for all stakeholders.
- One of the most valuable features is to distribute collaborative environment encouraging multifaceted stakeholders to brainstorming, elicitation, or validation the knowledge involved in requirements engineering projects.

The empirical evaluation of the developed tool intends to perform in two phases: an initial phase to obtain feedback for further improvement, and the second phase to repeat evaluation after proceeding modification. In an evaluation process, there are two practitioner groups students in the Requirements Engineering course (ITKS452) given by the university of Jyväskylä and in some software companies. Test tasks and interviews will be used as empirical evaluation methods.

References

- [1] Emam, E. K. and Koru G. A. A. Replicated Survey of IT Software Project Failure, IEEE Computer Society, 2008.
- [2] Ware, C. Information Visualization Perception for design, Second edition, 2005.
- [3] Heimbürger, A., Kiyoki, Y. and Ylikotila, T. Communication Across Cultures in the Context of Multicultural Software Development. Reports of the Department of Mathematical Information Technology. Series C. Software and Computational Engineering. No. C 1/2011. 64 p. 2011.
- [4] Heimbürger, A., Kiyoki, Y. and Kohtala, S. Intelligent Icons for Cross-Cultural Knowledge Searching. In: Henno, J., Kiyoki, Y., Tokuda, T. and Yoshida, N. (Eds.) Proceedings of the 21st European-Japanese Conference on Information Modelling and Knowledge Bases, Tallinn, Estonia, June 6 - 10, 2011. Tallinn University of Technology Press, Tallinn, Estonia. Pp. 84 - 96. 2011.
- [5] Laplante P. A. Requirements Engineering for Software and System. Auerbach Publication. 2009.
- [6] Heimbürger, A., Kiyoki Y., Kärkkäinen T., Gilman E., Kyoung-Sook K. and Yoshida N. "On Context Modelling in Systems and Applications Development," In: Heimbürger A., Kiyoki Y., Tokuda T., Jaakkola H. and Yoshida N. (Eds.). Frontiers in Artificial Intelligence and Applications, Vol. 225 Information Modelling and Knowledge Bases XXII. Amsterdam: IOS Press. pp. 396-412, 2011.
- [7] Aykin, N. Usability and Internationalization of Information Technology, 2005.
- [8] Teyseyre, A. R. 3D Requirements Visualization, Journal of Computer Science and Technology, Vol. 3, No. 2, pp.45-51, 2003.
- [9] Gotel, O., Marchese, F. T., and Morris, S. J. On Requirements Visualization, Second International Workshop on Requirements Engineering Visualization (rev'07), 2007.
- [10] Cooperr, J. R., Lee, S.-W, Gandhi, R. A. and Gotel,O. Requirements Engineering Visualization: A Survey on the State-of-the-Art, Fourth International Workshop on Requirements Engineering Visualization (rev'09), 2009.
- [11] Glass L. R. IT Failure Rates-70% or 10-15%, IEEE Computer Society, 2005.
- [12] Hull, E., Jackson, K. and Dick, J. Requirement Engineering. Second Edition. Springer 2005.
- [13] Fitriani, S., Datcu, D. and Rothkrantz, J.M. Human Communication Based on Icons in Crisis, pp. 57-66, 2007.
- [14] Abeti, L., Ciancarini, P and Moretti, R. Wiki-based Requirements Management for Business Process Reengineering, IEEE ICSC'09 Workshop, pp. 1-11, 2009.