

User Driven Service Design and Innovation Platforms

Birgitta Bergvall-Kåreborn¹ and Mikael Wiberg²

¹ Social Informatics, Luleå University of Technology, Sweden

`Birgitta.Bergvall-Kareborn@ltu.se`

² Informatics, Umeå University, Sweden

`Mikael.Wiberg@informatik.umu.se`

Abstract. Integrating consumers into the design and development process of IT is a promising strategy for companies. To stimulate this co-production of IT through end-user programming specific user platforms for innovation have been developed. In this paper we illustrate how end-user programming and configurations has been a successful practice throughout the history of the PC – from the early stationary computers in the 80s, via the development of the web, to recent toolkits for end-user configuration of mobile hardware. While this trend has been stable this paper illustrate how corresponding support for end-user programming of mobile applications is still missing. To address this need this paper presents the SATIN platform and its underlying design concepts. We view this as a contribution for advancing the development of innovation platforms and as an illustration of how concept design can help in envisioning the next step in user-driven service design.

Keywords: Concept design, Innovation Platforms, User-driven service design.

1 Introduction

Integrating consumers into the design and development process is a promising strategy for companies as they struggle with a growing individualization of consumer demands. To stimulate and facilitate user service design technical platforms for innovation have been developed. Platforms are defined as a set of user-friendly design tools that makes it easier for the user to design a product or service through a trial and error process [3, 8]. During the last years the interest in technical platforms for third-party development has increased significantly [9] and some authors even argue that that software platforms are emerging as a dominant model for software development and software-based services [7]. As Tiwana, Konsynski and Bush [7] states "Unlike traditional soft-ware development, these services leverage the expertise of a diverse developer community –with skills and an appreciation of user needs that platform owners might not posses- to creatively develop new capabilities unforeseeable by the platform's original design" (p. 675). However, within the filed of software development there is still a lack of available platforms that enables user-driven software design on a broad level. One main obstacle to achieve this is the existing toolkit languages. Taking this lack as a point of departure this paper present a

prototype of a platform for user-driven software design for non-programmers. The aim is to use this platform as a stimuli material for discussions related to the vision, form, and function of this type of platform.

2 The History of User Driven Design

There are many different approaches for defining and categorizing platforms. In this paper we use these as an inspiration/base for our categorization where we aim to illustrate in what sectors end-user programming has a history and where there are empty fields that needs exploring and development.

In our illustration of how end-user programming has been enabled through the history of personal computing we have organized this development along two axes (see the 3'4 matrix in table 1). Along the horizontal axis we trace support for end-user programming chronologically from the development of the desktop computer, via the web into contemporary developments in the mobile IT segments. Further on, we trace this development vertically along this timeline in an attempt to reflect how end-user programming has been made possible on both hardware and software. In doing so we adopt the basic structure behind this matrix from earlier work conducted by Bosch [1] in which they have demonstrated how system ecologies nowadays include several platforms (desktop, web and mobile) and how these platforms support end-user programming to different extents. Further on, as our illustration will show the trend has been quite clear. In pushing the technological envelope forward support for end-user programming has followed. Taking this logic as a point of departure our illustration will show that while the most recent developments in IT platforms brings IT to its users in mobile forms support for end-user programming has not yet been made available to the public masses of users. Accordingly, our proposal of the SATIN platform as a tool for end-user programming of mobile applications comes in timely in the current development in our field.

Table 1. End-user programming and configurations (table structure adopted after Bosch 2009)

	80s -	90s -	2000 - today
Software	From excel to Linux	Yahoo! Pipes, Scratch, Google mashup editor	Still lacking
Hardware	Peripherals, Chip, Memory, Power, Mother-board, CPU, PC	Router, blue tooth connections, (internet of things)	Configurations of mobile hardware (e.g. Arduino)
	Desktop	Web	Mobile

As illustrated in table 1. even the early version of the desktop computer presented to the public eye in the mid 80s provided support for simple end-user programming at the level of its hardware architecture. The end-user could through quite simple steps attach preferable peripherals to the computer to customize it for different personal needs. More advanced user-enabled configurations of the machine included

a range of activities from easy exchange of memory cards and chips to the more complex activity of assembling a complete PC including integration of power, motherboard, CPU, memory/ hard drive, etc. The architecture of the PC allowed for end-user configurations at the hardware level. Further on, the PC was deliberately designed to allow for end-user programming at the software level of the architecture as well. This possibility for end-user programming stretched from simple possibilities like designing scripts in Excel to more complex programming via Basic, Pascal, Linux, etc for more advanced end-users. In a similar way the development of the web has been followed by a trend of end-user programming and development – hardware wise in how end-users can easily add routers and other network hardware (including wifi, blueooth, etc) to configure the internet hardware infrastructure for different purposes. Ultimately, we view this as only the beginning of a much wider trend of end-user programming in the development of the Internet of Things. Software wise, tools (including Yahoo Pipes, Scratch, Google mashup editor) are now rapidly being developed and made available for end-users to enable people to program their own web based services. Most recently we can now witness that this trend continues as we extend IT platforms into the mobile domain. On the hardware side end-user programming has already picked up pace through open source initiatives like Arduino which enable end-users to build their own net-worked mobile devices with simple modules of electronics. However, while all of this is now constantly being developed there is still very little support available in support of end-user programming of mobile devices. While e.g. most smart phones allow for the organization of information and apps on the device there is still little support for programming or configuring the functionality of the phone. We see the SATIN platform as presented in this paper as an important contribution to this lack and as a first sign to a movement in this direction.

3 Research Methodology

The research reported in this paper follows the method of concept-driven design research. In relation to this we found good methodological support for our project since the concept-driven approach is a well-suited method when:

1. The point of departure is conceptual/theoretical rather than empirical.
2. The research furthers conceptual and theoretical explorations through hands-on design and development of artifacts.
3. The end result—that is, the final design—is optimized in relation to a specific idea, concept, or theory rather than to a specific problem, user, or a particular use context [6, p.98].

In the next section we present our concept design (the SATIN platform) followed by a description of how this method was applied in our project.

4 The Design Concept – The SATIN Platform

The SATIN platform (www.satinproject.eu) aims at offering a tool for non-programmers to create applications using modern web-tools, which are to be consumed in a mobile setting. It is based on visual programming using drag-and-drop, see figure 1. For more detailed information about the platform see Davoli and Kuenen [2].

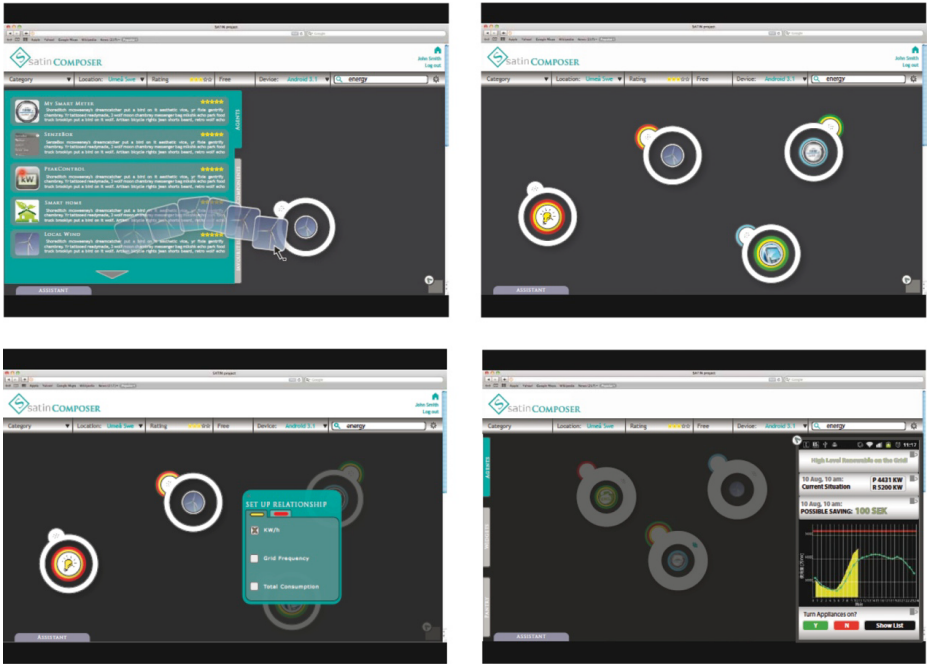


Fig. 1. The SATIN platform

From a methodological perspective we have since October 2009 worked back and forth between concept development and design of the SATIN platform. Methodologically we have 1) taken a point of departure in the concepts of user-driven service design and end-user programming 2) elaborated on these concepts through a process of concept development and design of the SATIN platform, and 3) seek to optimized the SATIN platform in terms of how it can support the idea of a platform for mobile app development for non-programmers. As to further situate our conceptual design work we have undertaken a literature study in which we have anchored the core concepts of user-driven de-sign and innovation platforms in the existing body of research.

5 Discussion and Conclusion

In this paper we have presented the concept design of the SATIN platform. We view this as an important contribution for advancing user –driven innovation for mobile

devices. We also notice how our concept-driven design of the SATIN platform also brings with it some theoretical contributions. While some authors focus on the more technical components and processes of a platform such as the codebase of software-based systems [4, 7] other take a broader view and also include knowledge, people, and relationships into the fundamental assets of a platform [5]. Our work adds to this broader view and we illustrate how our proposed design also brings with it potential not only to include people and knowledge, but also the (mobile) contexts of the people enabling anyone to form their own mobile IT environment. Accordingly, we suggest that future research on platform development should focus not only at technology and people, but also on innovation processes – whether in front of the desktop computer or out there in the field.

References

1. Bosch, J.: From Software Product Lines to Software Ecosystems. In: The 13th International Software Product Line Conference (SPLC 2009), San Francisco, CA, USA (2009)
2. Davoli, L., Kuenen, S.: New Participative Tools Require New Foundations. The Experience of Satin: programming for non-programmers. In: Crafting The Future, European Academy of Design Conference, Gothenborg, Sweden, April 17-19 (2012)
3. Franke, N., Keinz, P., Schreier, M.: Complementing Mass Customization Toolkits with User Communities: How Peer Input Improves Customer Self-Design. *Journal of Product Innovation Management* 25, 546–559 (2008)
4. Ghazawneh, A., Henfridsson, O.: Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model. *Information Systems Journal* (2012)
5. Robertson, D., Ulrich, K.: Planning for Product Platforms. *Sloan Management Review* 39, 19–31 (1998)
6. Stolterman, E., Wiberg, M.: Concept-driven Interaction Design Research. *Human Computer Interaction (HCI)* 25 (2010)
7. Tiwana, A., Konsynski, B., Bush, A.A.: Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research* 21, 675–687 (2010)
8. Von Hippel, E.: Perspective: User Toolkit for Innovation. *The Journal of Product Innovation Management* 18, 247–257 (2001)
9. Yoo, Y., Henfridsson, O., Lyytinen, K.: The New Organizing Logic of Digital innovation: An Agenda for Information Systems Research. *Information Systems Research* 21, 724–735 (2010)