

# End-User-Development and Evolution of Web Applications: The WebComposition EUD Approach

Olexiy Chudnovskyy and Martin Gaedke

Chemnitz University of Technology  
olexiy.chudnovskyy@cs.tu-chemnitz.de,  
martin.gaedke@cs.tu-chemnitz.de

**Abstract.** End-user-development (EUD) has been a field of study for more than 30 years already. The results are visible – users, who have no or only little programming skills, have become active creators of Web applications, developing new tools to meet their situational needs, sharing them with colleagues and combining them into more complex solutions. Recent trends, like maturation of cloud computing, mass customization and changing demographics resulted in even higher demand for flexible, feature-rich and extensible platforms for end-user development. While the potential of involving end-users into developing task is extremely high, a clear need for new systematic methods has emerged, which would take both the new technological opportunities but also risks resulting from non-professional application development into account. In this paper, we present our research towards systematic, end-user-oriented Web application development. We identify relevant research challenges; derive requirements on EUD-oriented development process and show how the WebComposition approach can be extended to support end-users during the whole life-cycle of Web applications - from requirements elicitation to evolution.

## 1 Introduction

The opportunities of end-user-development have become much greater in the last years. Not only technological foundation has advanced, but also social conditions have changed [1]. Nowadays end-users are much more skilled in dealing with IT. Motivated by ubiquitous internet access and pervasiveness of mobile devices providing rich user experience, users have become active shapers of the Web [2]. Many end-user-oriented platforms and new development paradigms have emerged. One of the most promising are so called mashups [3], i.e. Web applications, which let end-users combine data, application logic and UI coming from different sources on the Web.

The EUD trend has also moved to enterprises. The new generation of workforce is willing and able to create tools for their everyday needs, avoiding long-lasting feature requests to IT departments. In addition, end-users are better domain experts and know exactly how software should support their tasks, even if they can't always explain it properly. Giving end-users an opportunity to develop their own solutions can save significant costs to the company and unburden IT department with numerous feature requests.

Though the potential of EUD is extremely high, it also bears some risks and in worst case can have negative impact on the whole enterprise. The recent Gartner report on so called “citizen development” predicts that at least a third of businesses without formalised citizen developer governance policies will encounter substantial data, process integrity and security vulnerabilities [4]. One of the main problems identified is an inadequate application life-cycle management, resulting in plenty almost identical applications exposing many security vulnerabilities or performance drawbacks. Another problem is that if application becomes unmanageable, it is delegated to IT department, making its maintenance and evolution a cost-inefficient and time-consuming burden. Finally, end-users are simply not aware about best practises and governance rules to be applied during the application development process.

In this work, we address the above problems and propose a new method for end-user-development and evolution of Web applications. We build our work upon the WebComposition approach [5] and adapt its process model, tools and methods to the peculiarities of the EUD domain. The rest of the paper is structured as following: Section 2 gives an overview of the tackled research challenges. In section 3 we introduce the WebComposition approach and show how it can be extended towards the EUD domain. Finally, we conclude the paper and give an outlook into our future work.

## 2 Research Challenges

Consider the following example scenario, illustrating common problems and difficulties, which end-users face while developing new Web applications.

John is an employee of a medium-sized enterprise and works often with customer-relationship-management (CRM) systems, which help him in finding information about customers, performing data analysis or contacting them through various communication channels. John uses the system in his everyday work, but soon he discovers that many tasks are performed inefficiently – for example, the data should be collected from different areas of the system and can't be merged into one holistic view. John performs also many data lookup activities to find customer-related data on the Web. He can't enrich the system with other data sources rather than those foreseen by developers. Furthermore, he wants some system functionality to be available in other contexts, e.g. on desktop or embedded in his favourite portal application.

John decides to create his own application and takes the composition tool offered by the company IT department. The first hurdle John faces is the missing knowledge on how to start building an application. Most of the tools follow a bottom-up approach, giving end-users a toolkit to compose data, application logic or user interfaces. The requirements specification phase and corresponding modelling tools haven't gained much attention so far and lead to a unsystematic and time-consuming development process [6]. John starts to play with the given toolbox and encounters next problems - this time in finding relevant data sources or parts of application he should use. The building blocks are usually not fine-grained enough to be shared between applications or lack adequate semantic descriptions to be found among plenty

of available artefacts. John creates his own data specifications, defines presentation and navigation aspects of the new application. The procedure takes him quite some time due to the missing expertise in Web application development. Unwittingly John doesn't consider security and performance aspects as his main purpose is to fulfil the current needs without taking possible future reuse of artefacts or application evolution into account. Having created several applications, John uses them quite often but later on encounters incompatibility problems if his applications should be integrated with the ones developed by his colleagues.

The presented scenario reveals several problems and research challenges, which will be tackled in this work:

- How end-users can be supported during the whole Web application life-cycle including requirements engineering and evolution?
- Which concepts and development paradigms are appropriate for EUD? Which expertise can be expected from end-users nowadays?
- Which interactions between IT experts and end-users are needed to guarantee a required quality of produced software artefacts and compliance to company policies?
- How development expertise and application artefacts can be shared among end-users? Which artefacts can be reused during the application life-cycle?
- How models and tools can be adapted to establish a systematic and efficient development process? Which EUD-platforms offer the best trade-off in simplicity and expressive power?

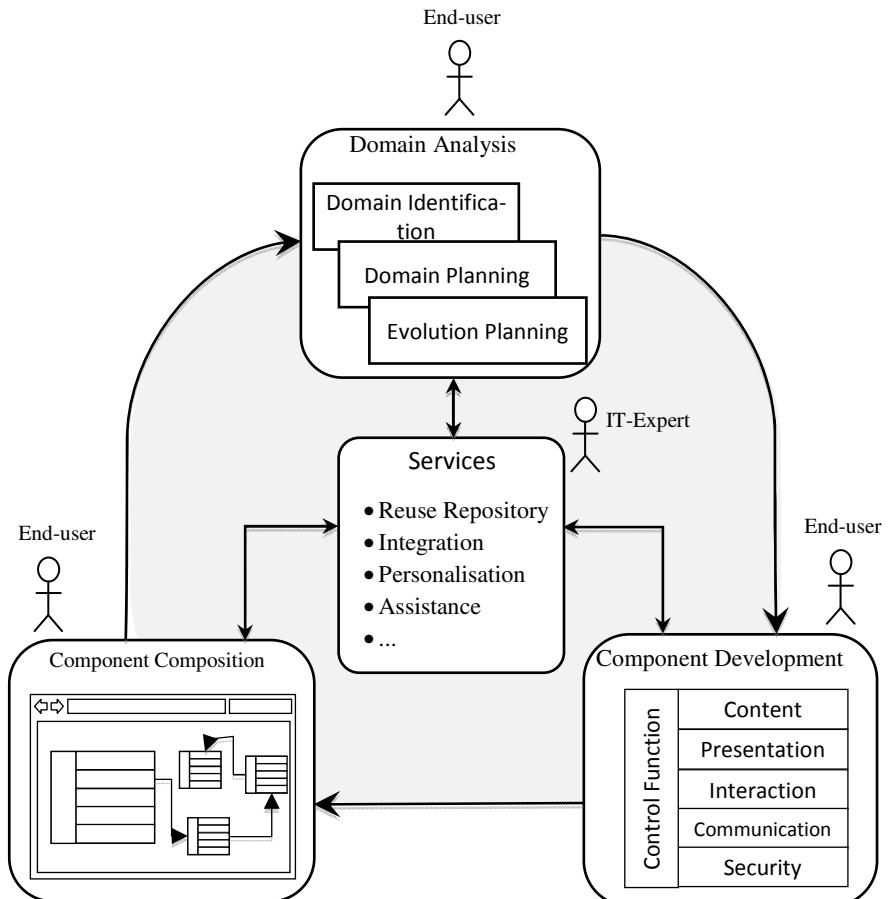
In the following section, we present our proposed extension of the WebComposition approach, which is adapted to tackle the aforementioned problems.

### **3 Research Methodology and Approach**

The WebComposition approach was first introduced in 1997 as an object-oriented approach for development and evolution of Web applications [7]. The Web-based solution is seen as a set of fine-grained application components, which deal with different perspectives of a Web application, such as content [8], distributed system and architecture [9] or user interface experience perspectives [10]. The WebComposition process model focuses on systematic development and reuse of application artefacts, taking continuous application evolution into account. To take the peculiarities of EUD domain into account, we propose an extension of the WebComposition approach, illustrated in Figure 1.

The WebComposition approach considers a Web-based solution as a set of areas or so called "domains", which are continuously evolving and extending the original application. The first step within the development process is the domain analysis, which is focused on identification and description of envisioned application domains. Already existing domains are analysed with respect to their further evolution. The planning process is supported by the so called WebComposition Reuse Repository, con-

taining artefacts, which can be reused and adapted to the current needs. In our work we are going to identify a set of artefacts specific to EUD, which can be produced during domain analysis and reused later on. These could be, for example, conceptual domain descriptions, user-interface-mockups, or user stories. Furthermore a set of services from IT experts as well as tools to support the domain analysis step will be elaborated.



**Fig. 1.** The WebComposition process model for end-user- development of Web applications

The next phase focuses on specification and development of building blocks required to implement the envisioned domain. One identifies already existing components and creates the missing ones. In our work we will refine the component model originally defined in the WebComposition approach to take security, system integration and distribution requirements into account. We believe that components would provide much higher value both in terms of reusability and efficiency if appropriate

packaging and distribution aspects are considered [11]. We will identify required IT-expert services, which would assist end-user in the development tasks, e.g. by providing consistency or policy compliance checks. A well-balanced cooperation between skilled and unskilled developers is needed to guarantee an efficient and maintainable development. We will develop a dedicated environment, which would let end-users implement the aforementioned components and access the identified IT-expert services.

The last step of the development cycle is the assembly of identified components into one solution, which would satisfy the requirements of the envisioned domain. Similar to the previous step, the composition process should happen in an end-user-friendly way, i.e. hiding the complexity of underlying technologies and providing assistance in such tasks as dependency resolution or establishing of component interoperability. We envision a dedicated run-time environment, which would enable graphical composition and provide end-user assistance facilities, e.g. recommendation and automatic composition engine. We will build upon well-adopted Web mashup technologies [3] but additionally cover security and user-experience aspects.

The working plan of our research is focused on tackling the research challenges stated in Section 2. In particular, we will continue with the following steps:

- Detailed requirements elicitation and use case definition for EUD-oriented process model and supporting tools.
- Profound state-of-the-art analysis of principles, methods and tools aiming to support EUD. We are going to cover the Web mashups domain, visual programming techniques, EUD suits as well as Web Engineering approaches.
- Specification of a EUD approach based on WebComposition process model, which would result in appropriate communication between different stakeholders and efficient development process.
- Development of a dedicated platform, supporting different phases of adapted WebComposition approach. End-user-friendly models, tools and run-time environments will be developed in this step.
- An evaluation of the proposed approach within several case studies. Applicability of tools and efficiency of the introduced process will be analysed.

## 4 Conclusions and Outlook

As recent studies showed, the risk of security vulnerabilities and poor maintainability of end-user-created products becomes high, if the development happens unsystematically and is not supported by IT departments. In this paper we presented a dedicated process model, tools and techniques to avoid the aforementioned risks. We described how WebComposition approach can be adapted towards the needs of EUD domain and elaborated our plan for the future research.

## References

- [1] Sarner, A.: How to Determine Levels of Engagement for Generation Virtual (2008)
- [2] O'Reilly, T.: What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software (2005), <http://oreilly.com/web2/archive/what-is-web-20.html>
- [3] Chudnovskyy, O., Nestler, T., Gaedke, M., Daniel, F., Ignacio, J.: End-User-Oriented Telco Mashups: The OMELETTE Approach. In: World Wide Web 2012 (2012)
- [4] Finley, I., Knipp, E.: Citizen Developers Are Poised to Grow (2011)
- [5] Gaedke, M., Rehse, J.: Supporting compositional reuse in component-based Web engineering. In: Proceedings of the 2000 ACM Symposium on Applied, pp. 927–933 (2000)
- [6] Robles Luna, E., Rossi, G., Garrigós, I.: WebSpec: a visual language for specifying interaction and navigation requirements in web applications. Requirements Engineering 16(4), 297–321 (2011)
- [7] Gellersen, H.-W., Wicke, R., Gaedke, M.: WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. In: Electronic Proc. of The 6th International WWW Conference (1997)
- [8] Chudnovskyy, O., Gaedke, M.: Development of Web 2.0 Applications using WebComposition/Data Grid Service. In: The Second International Conferences on Advanced Service Computing (Service Computation 2010), pp. 55–61 (2010)
- [9] Gaedke, M., Meinecke, J.: The Web as an Application Platform. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) Web Engineering Modelling and Implementing Web Applications, pp. 33–45. Springer, Berlin (2008)
- [10] Chudnovskyy, O., Brandt, S., Gaedke, M.: Integrating Human-Services using WebComposition/UIX. In: ACMIFIPUSENIX 12th International Middleware Conference (2011)
- [11] Spahn, M., Wulf, V.: End-User Development of Enterprise Widgets. In: Pipek, V., Rossen, M.B., de Ruyter, B., Wulf, V. (eds.) IS-EUD 2009. LNCS, vol. 5435, pp. 106–125. Springer, Heidelberg (2009)