Enterprise Mashup vs. Service Composition: What fits to reach the next stage in End-User Development?

Boris Büchel, Till Janner, Christoph Schroth, and Volker Hoyer

SAP Research CEC St. Gallen, Blumenbergplatz 9, 9000 St. Gallen, Switzerland
{b.buechel, till.janner, christoph.schroth, volker.hoyer}@sap.com

Abstract: While the development of enterprise applications is still relegated to IT departments, we observe a high demand for situational or ad-hoc applications desired by the mass of business end-users. This huge demand cannot be fully implemented by IT departments due to cost reasons and a lack of resources. New approaches in End-User Development (EUD) may help to overcome this gap. Recent approaches emerging in this field comprise new visual composition or programming environments stemming either from Business Process Management and Service-Oriented Architectures or Enterprise Mashup and the Web 2.0 paradigm. In this work, an analysis based on an extended version of the EUD Framework of Sutcliffe et al. is conducted to identify what approach fits to reach the next stage in EUD; the empowerment of the end-users to create their individual solutions with less or even no programming skills required. The analysis indicates that both approaches support EUD to reach this next stage, but Enterprise Mashups currently seem to better satisfy the end-users' demands, especially if situational, ad-hoc, and highly individualized but not too complex solutions have to be developed.

1 Introduction and Motivation

The following scenario motivates the need for situational or ad-hoc enterprise applications of a business user: *Frank is a salesperson working for a sport nutrition manufacturer. One of his tasks is to organize promotion-events at fairs and sport events. In the past, Frank contracted professional promotion-agencies without any approval process. Things changed with his new sales area manager Steve. Steve would like to be informed about all ongoing promotion-activities and to be involved in the contracting processes. Therefore, Frank has to change his workflow adding an additional approval step. Frank first asks Matt from the IT department to implement this additional requirement to the existing enterprise application. As usual, Matt is overworked and tells Frank that he has to wait at least two months for his inquiry. Frank is bothered about this situation, and looks if he can find a solution on his own for the upcoming promotion events next month …*

This scenario demonstrates the traditional problems between IT department and business units - poor quality of support and low reaction time [Hoy08]. As a result, individual solutions for the mass of business end-users are often not being implemented at all or do not fully support the evolved business needs [CBG+07].

Most of today's software applications still lack in providing its users intuitive ways to modify or to extend them according to their ad-hoc needs (described by expressions such as *''quick and dirty,'' ''just good enough,''* and *''the perpetual beta''* [CBG+07]). New development approaches are needed to overcome theses hurdles and to involve the group of non-technical business users into the development process in order to address the long tail of their requirements and to unleash the huge potential of End-User Development (EUD) (Figure 1, left). If the mass of non-technical business users (see Figure 1, right) is enabled to actively participate in the application development, cost-effects and efficiency gains could be realized.

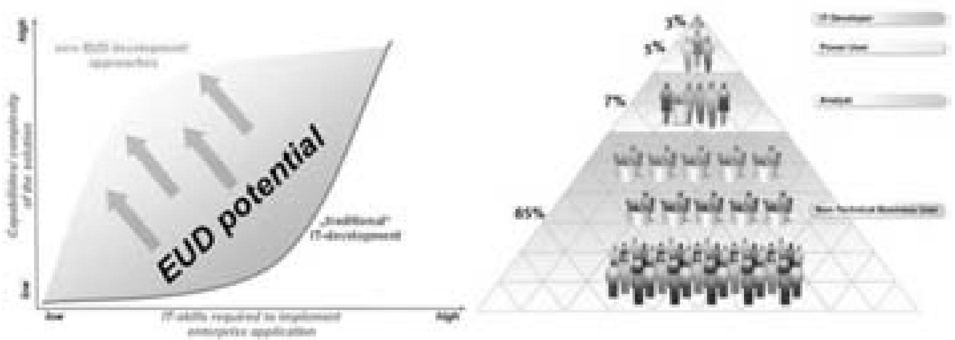

Figure 1: EUD Potential (left, own graphic) and User Classification (right, [Qui05])

Recently, new EUD approaches and development methods are emerging in research and industry practice. Most prominently mentioned are approaches stemming either from Business Process Management (BPM) and Service-Oriented Architectures (SOA) or Enterprise Mashup and the Web 2.0 paradigm. Especially Enterprise Mashups, a new generation of Web-based applications, seem to adequately fulfill the individual and heterogeneous requirements of end-users [Hoy08]. To shorten the traditional development process of design, implement, compile, test, and run, these new breed of applications are developed by non-professional programmers, often in a non-formal, iterative, and collaborative way by assembling existing building blocks [CBG+07]. We continue with our motivational case: *...We assume that Frank is motivated and starts to search for a quick solution to add the additional approval step to his work process, even without the help of the IT department. He remembers Nick, a young colleague from marketing, who showed him a quick and easy-to-use Mashups solution to aggregate all sport news and events for the whole team last summer. Frank decides to ask Nick at lunch if he has a "just good enough" solution for his requirements. Nick is quite optimistic and after lunch they are able to use the Web-based Enterprise Mashup tool to create a new Mashup application which includes the required approval process.*

The scenario indicates a huge demand for situational applications to solve ad-hoc business requirements. Further, it describes the goals of EUD to empower Frank and Nick to develop their own solution without support of the IT department. The main objective of this work is to develop a framework that helps to analyze the mentioned new EUD approaches, to classify them, and to provide decision guidance between them. For our work, we leverage the well known EUD framework by Suttcliff et al. [SLM03].

We extend the framework towards a more specific version which brings the methods and concepts of EUD, Enterprise Mashup/ Web 2.0, and BPM/ SOA/ service composition together. Related research used for our work cannot be presented due to space limitations, but the most important references used during our work were: [LPW06], in the context of EUD, [SJ07], [RD08] with regards to BPM/ SOA and service composition, and [Hoy08], [HSSJS08] in the field of Enterprise Mashups/ Web2.0.

2 The Analysis Framework

This chapter is devoted to present a framework for the analysis and classification of new EUD approaches. The EUD-Framework of Sutcliffe et al. [SLM03] provides the basis of our analysis framework. We now extend and refine this foundational framework by introducing a set of criteria for each of the original frameworks' dimensions with regard to the two lightweight composition approaches in scope of this work and their usefulness to support EUD. **Scope dimension**: describes the range of EUD problems that environments can support, ranging from task-/ or domain-specific to general purpose and is specified towards our analysis by the following criteria:

EUD Attribute	**Definition**
Learnability	The degree to which the EUD system supports the learning effort of end-users; support of "gentle slope" [MCLM90] as a concept to empower users with different skill levels.
Ease of understanding	The ease with which EUD system can be understood. According to [SJ07] Web 2.0 and SOA have gained momentum because of their notion of complexity hiding and reuse. Complex products will not satisfy their users because of their larger learning burden [SLM03].
Functionality	The capabilities of the system and the range of problems that can be solved; the general applicability of the EUD system (requires more user expertise and results in a longer learning curve [SLM03]) or its restriction to a specific domain.

Table 1: The attributes of the "Scope" dimension

Representation and means of communication with the user refers to the nature of the expressions used for programming and the interaction between EUD systems and end-users: they range from specific to abstract and are specified by the following criteria:

EUD Attribute	**Definition**
Visual nature	The degree to which information is represented graphically, i.e. as icons, diagrams, graphs, etc. [KHA97]. According to Sutcliffe et al., the motivation of a user can be increased by using readily available EUD applications [SLM03]. E.g. concept of repositories, where reusable building blocks are stored and published.
Paradigm support	The degree to which the EUD system supports its intended programming paradigm [KHA97].
Level of the language	Inverse measure of the amount of details (given by user) in order to achieve the desired result [Shu88]. The level of languages for the visual representation can range from procedural, over declarative, to object-oriented languages.

Table 2: The attributes of the "Representation" dimension

System Initiative: describes the assistance capabilities of the EUD system from passive systems to the provision of intelligent assistance (i. e., EUD system is able to guess users' programming intentions by watching examples).

EUD Attribute	Definition
Community support	Besides system initiative, a kind of social creativity can be leveraged by integrating communities to harness the collective wisdom of the crowd.
Semantic support	The modularization and huge variety of fine grained services leads to the challenge to find the right service according to the user's context. Semantic support of EUD tools is required to support the user in navigating through the growing Internet of Services [SJ07].
System support	Help and support functionality offered by the EUD system. The degree of user actions needed to specify a solution.

Table 3: The attributes of the "Initiative" dimension.

Summing up, the extended framework reflects the characteristics of the two EUD approaches in scope of this work and can serve as an evaluation tool to classify and to compare lightweight service composition approaches and tools. We now briefly apply the extended EUD framework as we continue our motivational case presented in the introduction of this paper: *We assume that the use of a visual service composition environment could be an alternative to Franks Enterprise Mashup solution. In this specific situation, our framework can be applied as follows to decide whether an Enterprise Mashup or a service composition tool is the appropriate choice for Frank: To add the additional process step, Frank and Nick have to request access to the visual service composition tool which is managed by the IT department due to governance reasons. After waiting some days highlighting the urgency, they receive the permission and try to adapt the business process. Unfortunately, Nick has no experience in BPMN and is not able to change the process with the service composition tool as the specification of various constraints for the process step represents a barrier. Consequently, Frank has to wait for the IT department. After three months, Matt solves the problem without writing one line of code. The model driven approach of the service composition tool allows for adapting and deploying the new solution as well.*

Enterprise Mashups make use of limiting the functionality and hiding complexity depending on the user's context. Due to Franks and Nicks low programming skill level, Enterprise Mashups would support their first attempts in a better way. Other use cases, which depend on governance processes, might be better implemented via service composition tools. For the creation or adaptation of applications to react to ad-hoc requirements (see Figure 1), Enterprise Mashups seem to be the environment of choice, especially with regards to the generation of users who are used to Internet technology and are accustomed to find out things on their own (as Nick does in our case).

3 Conclusion and Outlook

In this work, we developed an analysis framework allowing to compare and to classify new lightweight composition approaches with regard to their usefulness for EUD. We built upon the existing EUD framework of Sutcliffe et al. [SLM03] and extended it with more specific attributes. A first use case shows that both approaches have a huge potential to advance EUD and to involve the mass of end-users more actively in creating and adapting applications that fulfill their individual needs. In this context, service composition approaches seem better to address the upper part of the presented user pyramid (see Figure 1), as analyst and technical skills are required. Nevertheless, Enterprise Mashups currently seem to be a better choice for EUD if situational, ad-hoc, and highly individualized but not too complex solutions have to be developed. Future work will comprise the further development of our framework. While being involved in the EU funded project FAST (http://fast.morfeo-project.eu), which aims at developing an Enterprise Mashup tool, various use case scenarios can be developed and evaluated on the basis of this work.

References

[CBG$^+$07] Luba Cherbakov, Andrew Bravery, Brian D. Goodman, Aroop Pandya, and John Baggett. Changing the corporate IT development model: Tapping the power of grassroots computing. *IBM SYSTEMS JOURNAL*, 46(4):743, 2007.

[Hoy08] Volker Hoyer. The Changing Role of IT Departments in Enterprise Mashup Environments. Submitted to the 2nd International Workshop on Web APIs and Service Mashups, Pending, 2008.

[HSSJS08] Volker Hoyer, Katarina Stanoesvka-Slabeva, Till Janner, and Christoph Schroth. Enterprise Mashups: Design Principles towards the Long Tail of User Needs. In *SCC '08: Proceedings of the 2008 IEEE International Conference on Services Computing*

[KHA97] James D. Kiper, Elizabeth Howard, and Chuck Ames. Criteria for Evaluation of Visual Programming Languages. *Journal of Visual Languages and Computing*, 8(2):175–192, 1997.

[LPW06] Henry Lieberman, Fabio Paternò, and Volker Wulf. *End User Development (Human-Computer Interaction Series)*. Springer-Verlag New York, Inc., 2006.

[MCLM90] Allan MacLean, Kathleen Carter, Lennart Lövstrand, and Thomas Moran. User-tailorable systems: pressing the issues with buttons. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pages 175–182. ACM New York, NY, USA, 1990.

[Qui05] Kevin Quinn. Not Everyone Who Drives a Car Fixes It Themselves -Strategic Information Infrastructure. http://www.dmreview.com/news/1041222-1.html, November 2005.

[RD08] Pradip Roychowdhury and Diptiman Dasgupta. Take advantage of Web 2.0 for next-generation BPM 2.0. *http://www.ibm.com/developerworks/webservices/library/ws-web2bpm2/index.html*, 2008.

[SJ07] Christoph Schroth and Till Janner. Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IT Professional*, 9(3):36–41, 2007.

[SLM03] Alistair Sutcliffe, Darren Lee, and Nikolay Mhandjiew. Contributions, Costs and Prospects for End-user Development. Human Computer Interaction - International Proceedings, Lawrence Erlbaum Associates, New Jersey, 2003.