Association for Information Systems

AIS Electronic Library (AISeL)

PACIS 2011 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

9 July 2011

Do End Users Accept End User Development?

Andrea Giessmann
SAP (Switzerland) Inc., Andrea.Giessmann@sap.com

Jana Ebermann
University of St.Gallen, jana.ebermann@unisg.ch

Katarina Stanoevska-Slabeva
University of St. Gallen, katarina.stanoevska@unisg.ch

ISBN: [978-1-86435-644-1]; Full paper

DO END-USERS ACCEPT END-USER DEVELOPMENT?

Andrea Giessmann, SAP Research Center St. Gallen, SAP (Switzerland) Inc., Switzerland, andrea.giessmann@sap.com

Jana Ebermann, Institute for Media and Communications Management, University of St. Gallen, Switzerland, jana.ebermann@unisg.ch

Katarina Stanoevska-Slabeva, Institute for Media and Communications Management, University of St. Gallen, Switzerland, katarina.stanoevska@unisg.ch

Abstract

Do end-users accept end-user development by using enterprise mashups? Using the technology acceptance model, this research investigates the acceptance of the FAST platform, which enables end-users to build their own application by simply drag and drop graphical building blocks onto a canvas. An evaluation workshop of 159 individuals in various countries and locations found strong support of the idea. It was revealed that perceived usefulness strongly affected the attitude towards using enterprise mashups for end-user development. In turn, perceived ease of use did not. With respect to the developed mashup platform it was found that the available content within a mashup platform is the main influencing factor on the acceptance of end-user development by using mashups.

Keywords: End-user Development, Mashups, Web Apps, Technology Acceptance Model

1 INTRODUCTION

1.1 Motivation and Problem Scope

The increasing demand for software, as for instance in products, mobile devices or on the internet, also increases the need to develop it. End-user development (EUD) is a development paradigm, which empowers end-users to design and create applications without the need for trained programmers. Since 85 to 90 percent of all employees are non-technical business users, end-user development is about taking control (Sutcliffe & Mehandjie, 2005). One way to realize EUD is to create tools that end-users are motivated to learn and use in their daily work (Fischer et al., 2004). The paper at hand leverages the paradigm of enterprise mashups in order to implement end-user development. Enterprise mashups imply a new development paradigm, which empowers end-users to create individual applications without the involvement of the IT department.

So far, there is insufficient research on the acceptance of end-user development and especially by using enterprise mashups. The goal of this research paper is to fill this gap by using the technology acceptance model (TAM) in order to investigate the acceptance of end-user development by using enterprise mashup tools. The main research objective is addressed by the following questions:

- *Research Question 1: Do end-users accept end-user development by using enterprise mashups?*
- *Research Question 2: What are the main factors influencing the acceptance of EUD tools?*
- *Research Question 3: How can end-user development tools be improved?*

In order to answer these research questions, a laboratory experiment in the form of an evaluation workshop was designed and conducted. In total, 159 participants joined the experiment and provided feedback regarding end-user development by using enterprise mashups.

1.2 Research Approach

For answering the research questions, we followed the Design Science Research (DSR) approach (Hevner et al., 2004; March & Smith, 1995). The design science approach focuses on the development of effective solutions for practical and theoretical problems by creating and evaluating IT artefacts intended to solve identified organizational problems (Hevner et al., 2004). In the desire of solving the given problem the researcher draws on the knowledge and experience found in the respective application domain, see section 2 (Hevner, 2007). To come to rigorous and relevant research results, we draw upon (Peffers et al., 2007) to specify the relevant phase of the design science research process applied. Research projects often fail to adequately evaluate the designed artefact. Due to that reason the present article focuses on the evaluation activity in the design science research cycle (Hevner, 2007). In previous research works, we have already presented the designed artefact; see (Hoyer et al., 2009; Hoyer et al., 2010; Hoyer et al., 2011).

In the first section we have described the specific research problem and shown the practical relevance. In line with (Peffers et al., 2007) first step within the design science process, problem identification and motivation, we derived the research questions guiding the paper at hand. The objectives for a solution are defined in the background section, where we have elaborated the underlying principles in end-user development, enterprise mashup paradigm as well as the technology acceptance model according to (Davis, 1989). Activity three, design and development is presented in section 3. Since the created software artefact has already been presented in previous research work, section 3 focuses on the applied study design, by means of a laboratory experiment as well as the applied research model. The demonstration and evaluation activity in the design science research cycle is the main focus of the paper at hand. The data collection is based on a survey which was part of the organized laboratory experiment and has been answered by more than 150 individuals. The results of our investigation are presented and discussed in section 4. Finally, the last section closes the paper with a brief summary, limitations of the conducted research and an outlook to further research.

2 BACKGROUND

2.1 End-User Development

End-user development is a new development paradigm, which empowers end-users to design and create without the need for trained programmers or IT departments (Sutcliffe & Mehandjie, 2005). Lieberman et al. (2006) define EUD as follows: “End-user development can be defined as a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artefact.” (Lieberman et al., 2006). While (Costabile et al., 2003) and (Spahn et al., 2008) describe an evolvement in human-computer interaction from just making systems *easy to use* towards making systems *easy to develop*.

According to (Sutcliffe & Mehandjie, 2005) EUD can be achieved by three different approaches: First, by adding simple scripting languages to existing application in order to extend and adapt applications. Second, by the programming by example approach, where end-users instruct the machine to learn from examples, see (Lieberman, 2001), and finally, by using graphical tools for end-users to design systems by interacting with graphical micro worlds. In line with (Fischer et al., 2004) we believe that EUD depends on creating tools that end-users are motivated to learn and use in their daily work. Hence, we followed the third approach by developing a platform for end-user development this work is based on, see (Hoyer et al., 2009). Consistent with (Spahn et al., 2008) we leveraged the paradigm of enterprise mashups (see the flowing sub-section) to realise end-user development.

2.2 Enterprise Mashups

“An enterprise mashup is a web-based resource that combines existing resources, be it content, data or application functionality, from more than one resource by empowering end-users to create individual information centric and situational applications.” This definition from (Hoyer & Stanoevska-Slabeva, 2009) is based on the examination of common features, which frame the nature of enterprise mashups. However, in literature, the definition of enterprise mashups is still open to debate.

The relevant architectural components of the enterprise mashup paradigm can be structured in an enterprise mashup stack comprising three main layers: (1.) Resource, (2.) Gadget and (3.) Mashup, see also (Hoyer et a., 2008; Lizcano et al., 2008). On the gadget layer we introduced the concept of virtual building block in order to create powerful gadgets for enterprise purposes. Figure 1 depicts the resulting terminology which is applied in this paper. In addition, the relevant user roles including their tasks are mapped to the different architectural terms; see also (Hoyer et al., 2009).

- *Resources & Services* contain content, data or application functionality and represent the core building blocks of enterprise mashups. They are encapsulated via well-defined public interfaces (Application Programming Interfaces, WSDL, RESTful Web Services, RSS Feeds, Atom Feeds, etc.) allowing a loose coupling of existing resources – an important feature from the SOA paradigm. Resources and Services are usually provided by enterprise systems or by external web service providers (i.e., Zoho CRM, Microsoft Bing Maps, etc.) and are created by traditional developers who are familiar with development concepts.
- *Gadgets*. The layer above contains gadgets or widgets which provide a simple user interaction mechanism abstracting from the complexity of the underlying resources. A gadget in terms of the developed platform consists of a screenflow, which again consists of various screens and allow the handling of a data or information between the screens. In analogy to resources, also input and output parameters can be defined for gadgets, which will be used at the mashup layer. A screen-flow is a composition of screens, therefore screens are key elements within our enterprise mashups architecture stack. A screen is a fully functional building block giving access to a backend services through a graphical user interface. Screens can consist of one or more forms, operators and resources. While forms are graphical user interfaces that represent data, operators aggregate, transform, filter or sort the content of the underlying resources. Business experts work

quite closely together with developers in order to provide the right content for the enterprise mashup platform.

- *Mashups*. End-users solve their business challenges by combining such visual building blocks to gadgets and deploy them to mashup platforms according to their individual business needs, thus creating a mashup. In that way end-users do have the possibility of doing visual compositions of different backend systems or services on two layers. First, by linking the in-/ outputs of a gadget on the mashup layer and second by combining graphical building blocks. While the first is called wiring, the second is called piping, whereby both ways do not require any programming skills. Finally, the end-users consume and run the created mashup scenario. If necessary, they are able to configure the mashup to some extent, e.g. (de)activation of functionalities, (re)moving gadgets, etc.

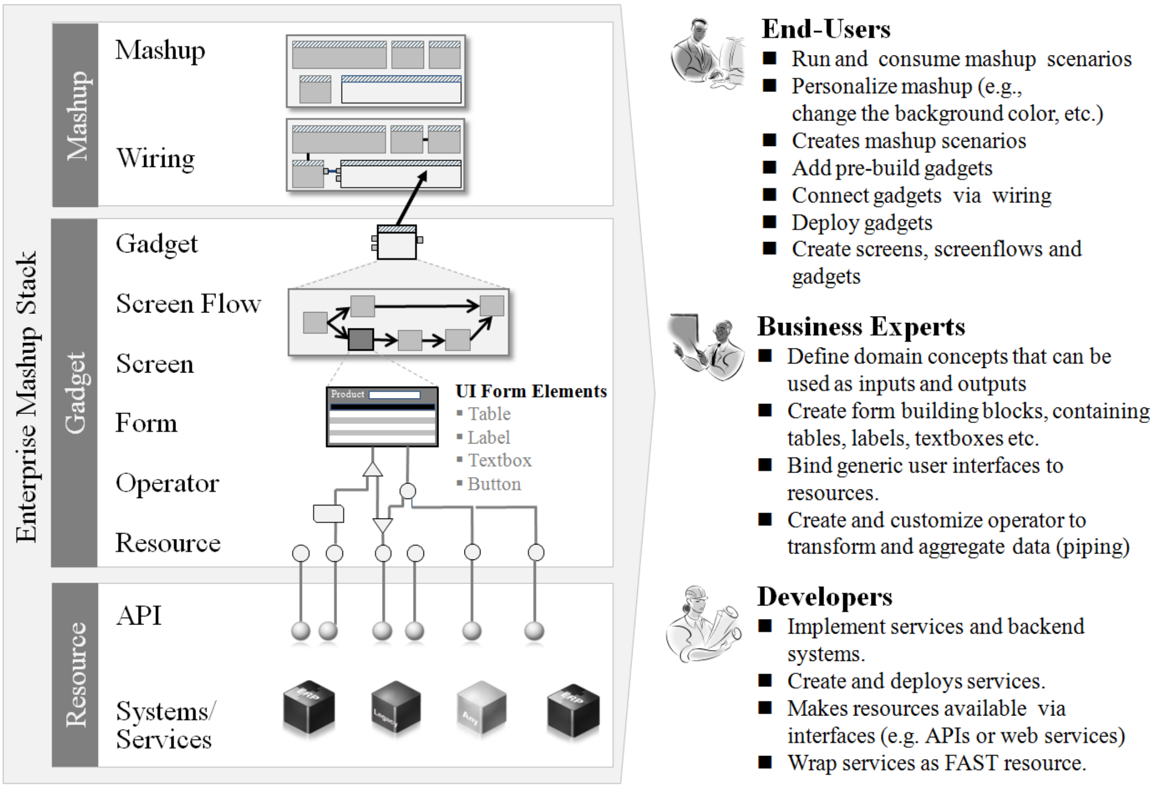

Figure 1. Enterprise Mashup Development Layers, Terminology and User Roles.

In summary, the composition principle of the resource layer of traditional SOA environments is transferred to the user interface level where the end-users are empowered to create ad-hoc enterprise-class application.

2.3 Technology Acceptance Model

The technology acceptance model was first introduced by Davis in 1989 and is for the main part based on the work of Aijzen and Fishbein on reasoned action and planned behaviour, see (Ajzen & Fishbein, 1977; Davis, 1989). TAM was developed in the light of concerns that people do not use information systems which have been provided to them (Davis, 1989; Davis et al., 1989). Since its introduction, TAM was further developed in TAM2 (Venkatesh & Davis, 2000), the unified theory of acceptance and use of technology (UTAUT) (Venkatesh et al., 2003), and TAM3 (Venkatesh & Bala, 2008). TAM and its successive models have received considerable attention in Information Technology acceptance research and have become widely accepted.

These Models posit that user acceptance is determined by two key beliefs, namely perceived usefulness (PU) and perceived ease of use (PEOU) (Venkatesh & Morris, 2000). Perceived usefulness is defined as “the degree to which a person believes that using a particular system would enhance his or her job performance” (Davis, 1989), while perceived ease of use indicates “the degree to which a person believes that using a particular system would be free of effort" (Davis, 1989). Both, PU and PEOU are supposed to be indicators for the attitude toward using (ATU) information systems (Davis, 1993).

3 STUDY DESIGN

3.1 Laboratory Experiment

The laboratory experiment in the form of an evaluation workshop was designed in line with (Creswell, 2009) and utilized the existing mashup platform FAST. For more information regarding the platform, please see (Hoyer et al., 2009) or visit the website http://fast.morfeo-project.eu. First, a short introductory presentation explained the principles of end-user development as well as the enterprise mashup paradigm to the participants. Second, the participants watched a video tutorial of almost 10 minutes. The video demonstrated the features of the mashup platform in a simple scenario. The participants did not receive a dedicated training session. Third, the participants had to implement a real life scenario by using the FAST platform. In order to achieve this, the participants had to build and deploy at least one new gadget. Meaning they had to identify relevant building blocks, create new screens and screenflows, build the corresponding gadgets and finally deploy the newly develop gadget to a desired platform like for instance iGoogle or EzWeb. Further details of the evaluation task are available at https://sites.google.com/site/fastchallenges/, see challenge one to five. In the fourth and final step, the participants' feedback was collected. The collection of feedback was conducted by means of a questionnaire. Each participant was asked to fill out a questionnaire in order to provide his or her individual opinion and impression. The designed questionnaire was based on the research model, presented in the following sub-section of the paper at hand.

The experiment took place in several different countries including Spain, Ireland, Germany, the Netherlands and Switzerland and contained different locations as for instance small and medium enterprises, large enterprises, research establishments and universities. Interested employees, researcher and students were invited to participate in the experiment by attending a local evaluation workshop. At each location, the same experiment was carried out and the experiment participants were provided with the same materials as well as all necessary equipment (laptop and internet connection).

The evaluation results are presented and discussed in detail within the following chapter. In total, 159 participants joined the experiment and provided feedback regarding end-user development by using enterprise mashups and in particular the FAST platform. 137 of participants have completed the online questionnaire. 74 percent of the participants are aged 19-29 years, 23 percent are aged between 30 and 39 years and 3 percent were older than 40 years. The results of the question "How do you rate your programming skills" show an almost equal distribution between beginners and experts among the participants, see also figure 2.

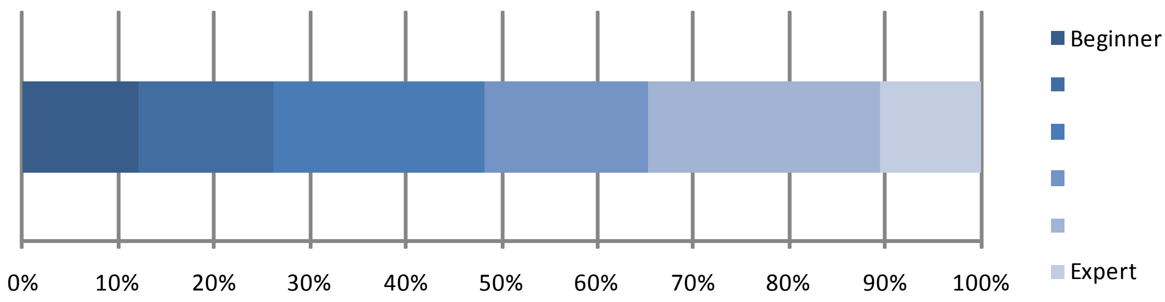

Figure 2. Programming skills of participants

3.2 Research Model

In order to examine not only the acceptance of end-user development by using enterprise mashups, but also to investigate which aspects influence the acceptance most, this study utilized an extended version of the TAM, see figure 3. In accordance with (Davis, 1989), (Davis, 1993) and (Venkatesh & Bala, 2008), computer self-efficacy (CSE) plays an important role to the successful implementation of systems and consequently the acceptance of technologies as it directly influence perceived ease of Use and indirectly affect perceived usefulness and consequently the attitude towards using.

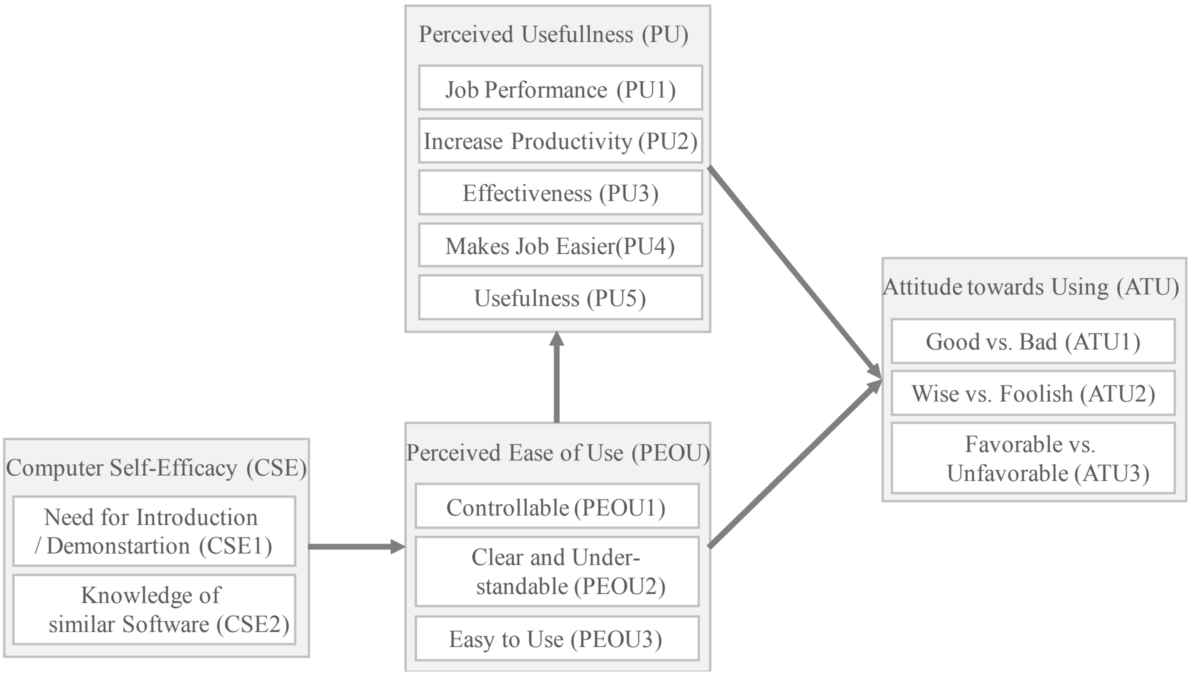

Figure 3. Research model based on (Davis, 1989)

Several empirical studies indicate that computer self-efficacy is important to the successful implementation of systems in organizations, (Compeau & Higgins, 1995a; 1995b; Venkatesh & Bala, 2008). CSE refers to the degree to which an individual believes that he or she has the ability to perform a specific task/job using the computer and was measured by using two items adapted from (Compeau & Higgins, 1995a). The first variable (CSE1) records whether end-users could use the FAST platform for end-user development if someone showed them how to do it first (Need for Introduction / Demonstration). While CSE2 records, if the use of similar software would be sufficient in order to realise EUD by using the FAST platform.

Perceived usefulness and perceived ease of use are measured using a five-point Likert scale. PEOU is characterised by three measures: Controllable (PEOU1), i.e. how easy it is to get the FAST platform to do what the user wants it to do. PEOU2, clear and understandable, describes how clear and understandable the interaction with the FAST platform is for end-user. PEOU3 indicates how easy to use end-users perceive the FAST platform. Perceived usefulness is described by the following five items introduced in (Davis, 1989): job performance (PU1), i.e. the extent on how much the development of enterprise mashups would improve end-users job performance; job productivity (PU2); effectiveness on the job (PU3); PU4 indicates to what extent it`s easier to do daily work by using enterprise mashups and PU5 how useful end-users perceive the use of end-user development tools like the FAST platform.

Attitude towards using is defined as "the degree of evaluative affect that an individual associates with using the target system in his or her job" (Ajzen & Fishbein, 1977). ATU was measured using three standard 7- point semantic differential rating scales as suggested by (Ajzen & Fishbein, 1980) and used by (Davis, 1993). The operationalization of ATU was carried out by the question "All things considered, me using the FAST platform in my job is: Good-Bad (ATU1); Wise-Foolish (ATU2); and Favorable-Unfavorable (ATU3)" on a seven point scale with midpoint label "neutral".

4 RESULTS AND DISCUSSION

4.1 Results

Dimensionality

After creating the model, dimensionality has to be analysed. The figure and the theoretical implications indicate unidimensionality. "Unidimensionality can be defined as the existence of one latent trait or construct underlying a set of measures" (Anderson, 1987). To verify unidimensionality, a factor analysis is performed at the beginning of the methodological approach (Anderson & Gerbing, 1988; Netemeyer et al., 2003).

As shown in Table 1 below, there are 4 relevant factors (eigenvalue >1) that match with the theoretical concepts and do not cross load. Within each dimension the items exceed factor loadings of 0.52. Unidimensionality can be confirmed. Additionally, the Kaiser-Meyer-Olkin criterion (KMO) as measure of sample adequacy amounts to 0.893. This implies that the correlations between the items are suitable for a factor analysis, respectively in a second step for the structural equation model. (Backhaus, 2008; Netemeyer et al., 2003)

Rotated Component Matrixa				
	Component			
	1	**2**	**3**	**4**
PU1	.859			
PU2	.876			
PU3	.842			
PU4	.889			
PU5	.837			
PEOU1		.860		
PEOU2		.902		
PEOU3		.832		
CSE1				.915
CSE2				.918
ATU1			.687	
ATU2			.809	
ATU3			.815	
Extraction Method: Principal Component Analysis.[1] Rotation Method: Varimax with Kaiser Normalization.[2]				
a. Rotation converged in 6 iterations.				

Table 1. Factor analysis

Reliability

Composite reliability reflects the intern consistency of a measurement of a construct (Netemeyer et al., 2003) "The most widely used internal consistency reliability coefficient is Cronbach`s (1951) coefficient alpha. […] A more stringent test of internal structure/stability involves assessing the amount of variance captured by a construct´s measure in relation to the amount of variance due to measurement error - the average variance extracted (AVE). By using a combination of the criteria above […] scales can be developed in an efficient manner without sacrificing internal consistency" (Netemeyer et al., 2003). Table 2 below illustrates the results of AVE and cronbach´s alpha.

[1] "The central idea of principal component analysis (PCA) is to reduce the dimensionality of data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in data set" (Joliffe, 2002)
[2] "To make factors more interpretable (and item retention and delation more meaningful), factors are 'rotated' after extraction. […] VARIMAX is the most common form of orthogonal rotation for EFA [explorative factor analysis] and will show simple structure in most cases." (Netemeyer et al., 2003)

AVE				Cronbach´s alpha			
Constructs (including variables)							
Perceived Usefulness (PU)	*Perceived Ease of Use (Peou)*	*Computer Self-Efficacy (Cse)*	*Attitude towards Using (Atu)*	*Perceived Usefulness (PU)*	*Perceived Ease of Use (Peou)*	*Computer Self-Efficacy (Cse)*	*Attitude towards Using (Atu)*
PU1-PU5	PEOU1, PEOU2, PEOU3	CSE1, CSE2	ATU1-ATU2	PU1-PU5	PEOU1, PEOU2, PEOU3	CSE1, CSE2	ATU1-ATU2
0.81	0.78	0.71	0.70	0.96	0.91	0.83	0.89

Table 2. Composite Reliability

Following Netemeyer, Bearden and Sharma (2003) Cronbach´s Alpha could be accepted with a value higher than 0.7. "A rigorous level of .50 or above has been advocated for AVE" (Netemeyer et al., 2003). Therefore reliability can be supported for the applied constructs.

Validity

"Discriminant validity is the degree to which measures of distinct concepts differ. This means that measures of different concepts should share little common variance (in a relative sense) and that a too high covariation casts doubt on the uniqueness of the measures and/or the concepts" (Bagozzi & Phillips, 1982). Comparing the results of AVE with the squared regression weights (Fornell-Larcker-Kriteria) all constructs are valid (Fornell & Larcker, 1981). Therefore the average variance shared with the corresponding indicator variables should be higher than the variance shared with other constructs.

Analysis of standardized estimates

After proving the structure and relations between items and dimensions, the structural equation model is implemented in Amos. The measures should indicate the implications for the FAST platform. In Figure 4 the standardized estimates of the model are presented.

- Computer self-efficacy has a positive impact on perceived ease of use'.
- Perceived ease of use influences attitude towards using by a value of 0.22.

The highest influences are the following:

- Perceived ease of use and perceived usefulness amounts to a value of 0.51.
- The relations between perceived usefulness and attitude towards Using gain a value of 0.68.

All values respectively relationships are significant ($p<0.05$).

	Estimate
Perceived Ease of Use (PEOU)	.070
Perceived Usefulness (PU)	.256
Attitude towards Using (ATU)	.665
PU2	.851
ATU3	.588
ATU2	.633
ATU1	.866
PEOU3	.744
PEOU2	.890
PU3	.798
PEOU1	.701
PU5	.761

	Estimate
PU4	.827
PU1	.842
CSE2	.739
CSE1	.686

Table 3. Squared Multiple Correlations

The value of squared multiple correlations coefficients (Table 3) indicates that computer self-efficacy could explain 7 percent of the variance of the dimension perceived ease of use. PEOU defines 26 percent of the variance of the dimension perceived usefulness. Both perceived ease of use and perceived usefulness explicate 67 percent of the variance of attitude towards using (Backhaus, 2008).

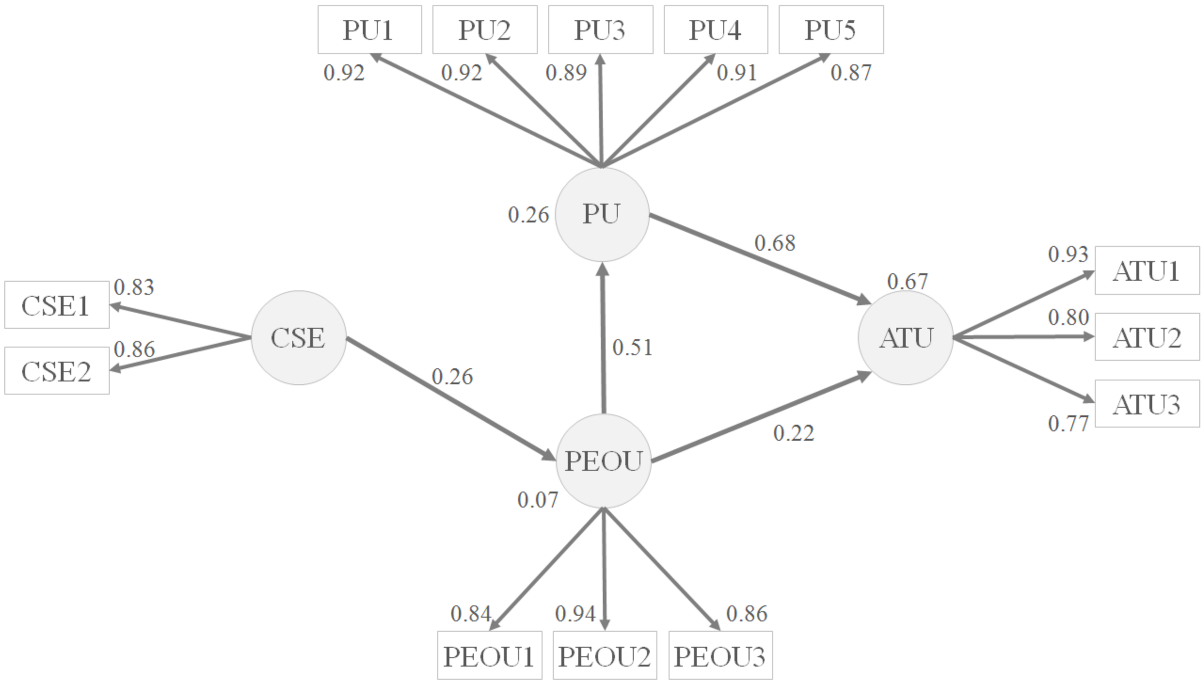

Figure 4. Standardized Estimates

One additional adaption within the model has been made. There were four outlier within the analysis of the modification indices of our model. Modification indices show how much the chi square decreases, if a link between 2 variables, dimensions or residuals would be made. Therefore we linked the four suggested residuals. (Bollen, 1989)

Quality Criteria

To rate the results, the quality criteria are relevant. In Table 4, there are the most relevant indices with their criteria for good model fits. To avoid misleading interpretations, several indices have to be evaluated (Homburg, 2006; Hu & Bentler, 1998).

Index	**χ2**	**GFI**	**AGFI**	**CFI**	**NFI**	**RMSEA**
Criterion	≥ 2.5*df	≥.90	≥.90	≥.90	≥.90	≥.05

Table 4. Criteria for model-fit indices (Baumgartner, 1996; Cudeck & Browne, 1983; Hatcher, 1994; Homburg, 2006; Hu & Bentler, 1998; Marsh & Hocevar, 1985; Wheaton et al., 1977)

The values of model fit in Table 5 are obtained from AMOS output. The GFI of our model is 0.942 (see Table 5). This means that 94.2 percent of the former total variance could be explained through the structure of the model (Backhaus, 2008). All values of AGFI, NFI and CFI are clearly higher than 0.9, what suggests a good model fit.

Model	χ2/Df	GFI	AGFI	NFI Delta1	CFI	RMSEA
Default model	1.012	.942	.910	.963	1.000	.009
Saturated model		1.000		1.000	1.000	.376
Independence model		.240	.113	.000	.000	

Table 5. Model Fit

The value of RMSEA has to be 0.05 or smaller. With RMSEA=0.009 there is a strong indication for a good model fit. (Steiger, 1990; Steiger & Lind, 1980)

4.2 Discussion

More than 60 percent of the participants do have a positive attitude towards using enterprise mashups, represented by the FAST platform for end-user development; see Table 6 as well as Figure 5. The table must be interpreted as follows: 0 means extremely or slightly bad, foolish or unfavourable, whereby one of the three variables can also be evaluated with neither. 1 means undecided, while alternatively one of the three variables can be evaluated different. 2 means extremely to slightly good, wise and favourable, whereby one of three variables can be evaluated with neither or was not evaluated at all.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	.00	11	7.9	8.4	8.4
	1.00	36	25.9	27.5	35.9
	2.00	84	60.4	64.1	100.0
	Total	131	94.2	100.0	
Missing	System	8	5.8		
Total		139	100.0		

Table 6. Attitude towards Using

5.8 percent of the participants did not answer the questions regarding attitude towards using consistently. Of the 94.2 percent, who answered the questions consistently, 64.1 percent are satisfied or very satisfied with the FAST platform. 27.5 percent are undecided and only 8,4 percent are slightly or extremely dissatisfied.

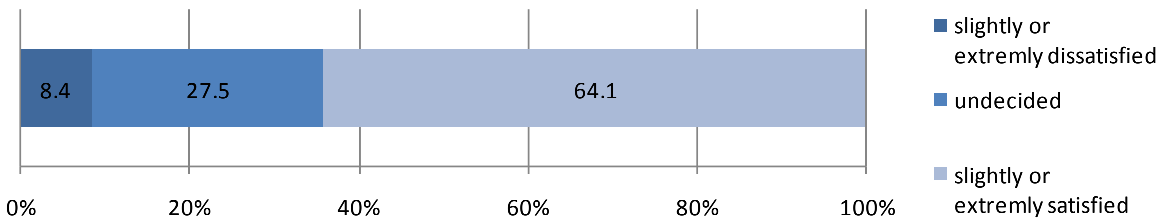

Figure 5. Attitude towards Using

It was shown that computer self-efficacy has a positive, but very low impact on perceived ease of use and PEOU influences the attitude towards using only by a value of 0.22. Hence, in contrast to (Fischer et al., 2004) it cannot be confirmed that only highly motivated end-users would utilize EUD tools. On the contrary, it can be confirmed that even end-users with no special training on the tool are able to create their own applications by using enterprise mashups resp. the FAST platform.

In comparison with perceived usefulness, perceived ease of use has only a small influence on the acceptance of end-user development by using mashups. In other words, improving the ease of use of EUD tools like the FAST platform, by for example improving usability aspects, would not have a big impact on the acceptance of such tools. On the other hand, perceived usefulness affects ATU with a value of 0.68, and therefore has a much bigger impact on the acceptance than perceived ease of use. In this case, it would be more effective to improve the perceived usefulness of EUD tools as its ease of use.

In line with the relevance cycle according to (Hevner, 2007) the elaborated design feedback for the FAST platform, our build and designed artifact, is to improve the perceived usefulness of the platform. In case of the FAST platform, this implies to have more and most notable, more relevant content within the platform. Content in the sense of the FAST platform means building blocks like, screens, resources, operators and forms. Primarily the available resources and services within the EUD tool do have the biggest impact on the acceptance of the platform. As a result it would be desirable, that end-users are empowered to integrate backend systems and services which are of interest within their working domain. The result could be a component, which automatically creates new resource building blocks based on a given application programming interface (API) or even a url (uniform resource locator).

5 CONCLUSION AND OUTLOOK

The purpose of this investigation was to evaluate and validate the idea of end-user development by using enterprise mashups. In order to achieve this, we followed the design science methodology. After defining the main terms related to the acceptance of end-user development by using enterprise mashups, we presented our study design by means of laboratory experiment as well as the underlying research model.

It has been shown, that the resulting model is unidimensional, reliable and valid. The result of our investigation is that more than 60 percent of the participants do have a positive attitude towards end-user development by using enterprise mashups. It was revealed that perceived usefulness strongly affected the attitude towards using. In turn, computer self-efficacy and perceived ease of use did not. With respect to the developed EUD tool it was found that the available content within the mashup platforms is the main influencing factor on the acceptance of end-user development by using mashups. Further development should focus on the automatically integration of backend systems and services by simply providing an API or even an URL. In this way, end-users are empowered to integrate even resources, which are very specific to a certain field of application.

However, there also exist some limitations of our conducted research. To begin with, the sample population is almost completely under the age of 40 years, thus, potential differences with respect to end-users age might not be considered. Second, due to the immediate development stage of the FAST platform and the time limitations of the evaluation workshops, the evaluation was limited to small application domains.

ACKNOWLEDGEMENTS

The authors would like to thank Bastiaan de Visser and Frédéric Noyer for their work in preparing and conducting the evaluation workshops. This work is supported in part by the European Commission under the first call of its Seventh Framework Program (FAST STREP Project, grant INFSO-ICT-216048).

References

Ajzen, I., & Fishbein, M. (1977). Attitude-behavior relations: A theoretical analysis and review of empirical research. *Psychological Bulletin*, *84*(5), 888-918.

Ajzen, I., & Fishbein, M. (1980). *Understanding Attitudes and Predicting Social Behavior*. Englewood Cliffs: Prentice Hall.

Anderson, J. (1987). On the Assessment of Unidimensional Measurement: Internal and External Consistency, and Overall Consistency Criteria. *Journal of Marketing Research*, *24*(4), 432 - 437.
Anderson, J. C., & Gerbing, D. W. (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin*, *103*(3), 411-423.
Backhaus, K. (2008). *Multivariate Analysemethoden Eine anwendungsorientierte Einführung* (12th ed., p. 830 S.). Springer: Berlin.
Bagozzi, R. P., & Phillips, L. W. (1982). Representing and testing organizational theories: A holistic construal. *Administrative Science Quarterly*, *27*(3), 459--489.
Baumgartner, H. (1996). Applications of structural equation modeling in marketing and consumer research: A review. *International Journal of Research in Marketing*, *13*(2), 139 - 162.
Bollen, K. (1989). A new incremental fit index for general structural equation models. *Sociological Methods & Research*, *17*(3), 303-316.
Compeau, D. R., & Higgins, C. A. (1995a). Computer Self-Efficacy: Development of a Measure and Initial Test. *MIS Quarterly*, *19*(2), 189-211.
Compeau, D. R., & Higgins, C. A. (1995b). Application of Social Cognitive Theory to Training for Computer Skills. *Information Systems Research*, *6*(2), 118-143.
Costabile, M. F., Fogli, D., Fresta, G., Mussio, P., Piccinno, A., Informatica, D., et al. (2003). Building Environments for End-User Development and Tailoring. *Proceedings of IEEE Symposium on Human Centric Computing Languages and Environments.* (pp. 31- 38). Washington, USA: IEEE Computer Society.
Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (3rd ed.). Los Angeles: Sage Publications, Inc.
Cudeck, R., & Browne, M. (1983). Cross-Validation Of Covariance Structures. *Multivariate Behavioral Research*, *18*(2), 147-167. Psychology Press.
Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, *13*(3), 319 - 340.
Davis, F. D. (1993). User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies*, *38*(3), 475-487.
Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, *35*(8), 982-1003.
Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., & Mehandjiev, N. (2004). Meta-design: a manifesto for end-user development. *Communications of the ACM*, *47*(9), 33-37.
Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of marketing research*, *18*(1), 39-50.
Hatcher, L. (1994). *A Step-by-Step Approach to Using the SAS System for Factor Analysis and Structural Equation Modeling* (p. 608). SAS Publishing.
Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information System*, *19*(2), 87-92.
Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.
Homburg, C. (2006). Die Kausalanalyse in der empirischen betriebswirtschaftlichen Forschung - Problemfelder und Anwendungsempfehlungen. *DBW - Die Betriebswirtschaft / Business Administration Review (BARev)*, *66*(6), 727 - 748.
Hoyer, V., Fuchsloch, A., & Janner, T. (2010). Implementing Cross-Organizational Business Processes with Enterprise Mashups : Hype or Reality ? *Proceedings of the Sixteenth Americas Conference on Information Systems* (pp. 1 - 11). Lima, Peru: AIS Electronic Library (AISeL).
Hoyer, V., Janner, T., Delchev, I., Fuchsloch, A., López, J., Ortega, S., et al. (2009). The FAST Platform : An Open and Semantically- Enriched Platform for Designing Multi-channel and Enterprise-Class Gadgets. In L. Baresi, C. Chi, & J. Suzuki (Eds.), *Proceedings of the 7th International Joint Conference on Service-Oriented Computing* (pp. 316-330). Stockholm, Sweden: Springer Verlag.
Hoyer, V., Stanoesvka-Slabeva, K., Janner, T., & Schroth, C. (2008). Enterprise Mashups: Design Principles towards the Long Tail of User Needs. *Proceedings of IEEE International Conference on Services Computing* (pp. 601-602). Washington, USA: IEEE Computer Society.

Hoyer, V., & Stanoevska-Slabeva, K. (2009). Towards A Reference Model for Grassroots Enterprise Mashup Environments. *Proceedings of 17th European Conference on Information Systems (ECIS)* (pp. 1-13). Verona, Italy: AIS Electronic Library (AISeL).

Hoyer, V., Stanoevska-Slabeva, K., Kramer, S., & Giessmann, A. (2011). What are the Business Benefits of Enterprise Mashups? *Proceedings of 44th Hawaii International Conference on System Science (HICSS)* (p. 10). Kauai, Hawaii.

Hu, L.-T., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model misspecification. *Psychological Methods*, *3*(4), 424-453.

Joliffe, I. T. (2002). *Principle Component Analysis* (Springer S., p. 487). New York, Berlin, Heidelberg: Springer.

Lieberman, H. (2001). *Your wish is my command: programming by example* (1st ed.). San Francisco, California: Morgan Kaufmann Publishers.

Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006). End-User Development: An Emerging Paradigm. In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End-User Development* (pp. 1-8). Dordrecht, Netherlands: Springer.

Lizcano, D., Soriano, J., Reyes, M., & Hierro, J. J. (2008). EzWeb/FAST: Reporting on a Successful Mashup-Based Solution for Developing and Deploying Composite Applications in the Upcoming "Ubiquitous SOA". *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08)* (pp. 488-495). Washington, DC, USA: IEEE Computer Society.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251-266.

Marsh, H. W., & Hocevar, D. (1985). Application of confirmatory factor analysis to the study of self-concept: First- and higher order factor models and their invariance across groups. *Psychological Bulletin*, *97*(3), 562-582.

Netemeyer, R. G., Bearden, W. O., & Sharma, S. (2003). *Scaling procedures: Issues and applications*. California, USA: Sage Pubn Inc.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45-77. M. E. Sharpe, Inc.

Spahn, M., Dörner, C., & Wulf, V. (2008). End User Development: Approaches Towards A Flexible Software Design. *Proceedings of the 16th European Conference on Information Systems (ECIS)* (pp. 1-12). Galway, Ireland: AIS Electronic Library (AISeL).

Steiger, James. (1990). Structural Model Evaluation and Modification: An Interval Estimation Approach. *Multivariate Behavioral Research*, *25*(2), 173-180. Psychology Press.

Steiger, Jh, & Lind, J. (1980). Statistically based tests for the number of common factors. *Presented at the Annual Meeting of the Psychometric Society*. Iowa City, Iowa.

Sutcliffe, A., & Mehandjie, N. (2005). End-user development. *Communications of the ACM*, *47*(9), 31-32.

Venkatesh, V., & Bala, H. (2008). Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences*, *39*(2), 273–315.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: four longitudinal field studies. *Management Science*, *46*(2), 186-204.

Venkatesh, V., & Morris, M. G. (2000). Why Don't Men Ever Stop to Ask for Directions? Gender, Social Influence, and Their Role in Technology Acceptance and Usage Behavior. *MIS Quarterly*, *24*(1), 115-139.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, *27*(3), 425-478.

Wheaton, B., Muthen, B., Alwin, D., & Summers, G. (1977). Assessing reliability and stability in panel models. *Sociological methodology*, *8*, 84-136.