Enterprise Architecture Software Tool Support for Small and Medium-Sized Enterprises: EASE

Research-in-Progress

Dennis Ingelbeen
Department of Management Information Systems and Operations Management, Ghent University
Dennis.Ingelbeen@UGent.be

Maxime Bernaert
Department of Management Information Systems and Operations Management, Ghent University
Maxime.Bernaert@UGent.be

Geert Poels
Department of Management Information Systems and Operations Management, Ghent University
Geert.Poels@UGent.be

ABSTRACT
In the current information society increased attention is being paid to enterprise architecture (EA) and accompanying techniques, models and frameworks. CHOOSE is an EA approach focused on and adapted to the characteristics and needs of small and medium-sized enterprises (SMEs). Though these techniques could offer significant benefits to SMEs, hardly any SME uses EA. The application and implementation of EA in general and the CHOOSE approach in particular, has proven to be a complex and challenging task. This paper describes the research-in-progress of the development of a software tool called EASE in support of the CHOOSE approach in order to maximize this disappointing rate of adoption. Furthermore, the software tool should guide the enterprise architect throughout the entire EA process and facilitate the implementation, management and maintenance of the resulting EA. A brief overview is given of the main features illustrating the added value of this research-in-progress. Finally, validation is achieved by means of multiple case studies.

Keywords
Enterprise architecture, small and medium-sized enterprise, CHOOSE, software tool support, EASE

INTRODUCTION
If you are about to build or rebuild a house, you will probably appeal to an architect to make sure the house fits your needs both structurally and functionally. The same can be said when starting, running or growing a business. An enterprise is a complex system of people, knowledge, fixed assets, projects, processes and many more brought together to fulfill a common shared vision (Lankhorst 2009). Enterprise architecture (EA) can help guide this process and consists of principles, methods and models to achieve its main objective, which is a coherent and consistent organizational design. Originally EA was focused on IT and its alignment with the business side. However, over the years the concept has grown into a much broader technique and is applied across the borders of IT. Although a lot of research is being done on EA, hardly anything is known about its use in the context of a small and medium-sized enterprise (SME) (Bhagwat and Sharma 2007). Some have pioneered in this field of study through the development of an EA technique adapted to the specific needs of this target group called CHOOSE (Bernaert and Poels 2011; Bernaert, Poels et al. 2013). The application and implementation of EA in general and the CHOOSE approach in particular, has proven to be a complex and challenging task. Though these techniques could offer significant benefits to SMEs, hardly any SME uses EA and adoption is far below par. Analysis of widely accepted adoption models has shown that software tool support could significantly contribute to solving this paradox. This paper describes the research-in-progress of the development of such a software tool in support of the CHOOSE approach called EASE (‘EA SME Environment’). This software tool guides the SME’s CEO in his function as enterprise architect throughout the EA process and facilitates the implementation, management and maintenance of the resulting EA. Validation by means of case studies provides the necessary proof of both the importance and efficacy of EASE. A pilot case study at a Belgian SME was first executed, confirming the importance and emphasizing the need for software tool support. Furthermore, four additional case studies supplement this validation process and were used to provide valuable insights and measurements for the evaluation of the efficacy of the developed software tool EASE.

In the first part of this paper, a brief overview of the key elements of the CHOOSE approach is given to provide the reader with both a background story and a proof of relevance and importance of this research-in-progress. The second part elaborates on the development process of EASE. Consecutively the importance, methodology, software tool criteria and the tool under development are discussed in this section of the paper.
The end of this second part briefly discusses the validation process of EASE. Finally, the paper ends with a succinct conclusion summarizing the key findings of this research-in-progress.

BACKGROUND

CHOOSE: EA for SMEs

CHOOSE is an EA approach for SMEs based on the core elements of existing EA techniques following Einstein’s principle: “Everything should be made as simple as possible, but not simpler”. CHOOSE is an acronym for ‘keep Control, by means of a Holistic Overview, based on Objectives and kept Simple, of your Enterprise’. Each letter refers to one of the five criteria for an EA technique derived from Lankhorst’s definition and description of EA (Lankhorst 2009; Bernaert, Poels et al. 2013):

1. Control: EA should be usable as an instrument in controlling the complexity of the enterprise and its processes and systems.
2. Holistic Overview: EA should provide a holistic overview of the enterprise and be able to capture the essentials of the enterprise, which are the elements that are stable and do not vary across specific solutions found for the problems currently at hand.
3. Objectives: EA should facilitate the translation from corporate strategy to daily operations.
4. Suitable for its target audience: It needs to be an approach that is understood by all those involved, even when coming from different domains.
5. Enterprise: EA should enable optimization of the company as a whole instead of doing local optimization within individual domains.

The CHOOSE approach focuses on SMEs for two distinct reasons. First of all, this target group is often overlooked by the EA approaches currently available on the marketplace (see Importance New Software Tool). Secondly, the importance of SMEs in the modern economy cannot be underestimated (CHI Research Inc. 2004; European Commission 2011; Small Business Administration 2011). It is important to realize that an SME is not just a downsized large company (Welsh and White 1981). An SME operates fundamentally different from the latter, hence vindicating the need to develop an EA approach adjusted to the characteristics of this target group (criterion 4). To be able to comply with this criterion, extensive research was done with respect to the characteristics and attributes of SMEs and six well-documented criteria were identified, which can be seen as sub-criteria for criterion 4 (Suitable for its target audience) when applied to an EA approach for SMEs (Bernaert, Poels et al. 2013):

4.1. The EA approach should allow SMEs to work time-efficiently on strategic issues and challenges.
4.2. An employee with limited IT skills should be able to work seamlessly with the developed EA approach.
4.3. Few or preferably no help of external experts is required to work with the developed EA approach.
4.4. The approach should enable the company to create clear descriptions of how things are currently done in the company.
4.5. The CEO, as the central figure in SMEs, should be involved in the approach.
4.6. The expected benefits should exceed the expected costs and risks.

Figure 1: The four core dimensions of CHOOSE
Based on these criteria, the CHOOSE approach was developed. A strategic dimension (goal: why?), an active actor dimension (actor: who?), an operation dimension (operation: how?) and an object dimension (object: what?) form the core dimensions and are integrated to provide a holistic EA overview. Figure 1 gives an overview of the core dimensions and relations of the CHOOSE approach. Figure 2 gives a summarizing overview of the evaluation criteria.

To provide a frame of reference for the research-in-progress discussed in this paper, the latter can be positioned within the Design Science Research (DSR) process (Hevner, March et al. 2004; Peffers, Tuunanen et al. 2007) applied for developing the CHOOSE approach. Figure 3 gives a schematic overview of this process.

Based on the aforementioned criteria of SMEs and EA (step 1) and the knowledge gathered throughout research steps 1 to 4 on figure 3 (Bernaert and Poels 2011; Bernaert, Poels et al. 2013), general design concepts are derived for the development of EASE in support of the CHOOSE approach (confer infra EASE criteria). These are then supplemented with criteria specifically related to software tools. This combination subsequently serves as a guideline for the software development process itself (step 5). Validation of EASE by means of case studies provides valuable insights with respect to the added value of the developed software tool, the degree to which the software tool conforms to the predefined criteria and the necessary adjustments that have to be made (step 6). The steps depicted in green have been performed in previous research and form the starting point of this research-in-progress (research scope indicated in blue).

**CHOOSE SOFTWARE TOOL SUPPORT**

**Importance Software Tool Support**

Despite the customized EA approach offered by CHOOSE with its intrinsic qualities aligned with the characteristics of SMEs, it is also very important to take the adoption of the approach into account. Techniques that are technically superior or fully customized to the needs of the user will not yield the expected benefits as long as the techniques are not effectively used in practice. To help optimize, facilitate and speed up the adoption process, one can rely on different models that explain the adoption of information systems (IS) and IS models. Of these models, the technology acceptance model (TAM) (Davis 1989) is widely used and the method evaluation model (MEM) (Moody 2003) supplements TAM to be better applicable for the evaluation of methods. TAM provides a model that helps discern external factors and their impact on the attitude, evaluation and behavior of practitioners towards the adoption of IS and IS methods, such as EA. Central determinants in this model are perceived ease of use and perceived usefulness. The conviction of the end-users that the information technology will help them better perform their job relates to the perceived usefulness. Perceived ease of use alternatively deals with the amount of effort and time needed to learn how to work with it. Both aspects influence the attitude towards the technology and subsequently the behavioral intention to use. Crucial for the adoption is that the increase in performance is perceived as being of higher influence to adoption than the effort necessary to learn the developed technology and work with it (Davis 1989).
MEM was developed as a reaction to the trend that most IS research focuses on the development of new methods rather than the evaluation of existing methods (Moody 2003). It combines the insights of the TAM with the methodological pragmatism of Rescher (1977). This model states that methods cannot be labeled as being wrong or right. Methods do not have a truth value, but a pragmatic value in the sense that they can only be evaluated based on their efficacy. The validity of a method can neither be derived inductively nor deductively but should be proven through the successful application in practice. This is called the pragmatic success of a method and is defined as the efficiency and effectiveness with which a method achieves its predefined goals. Figure 4 gives an overview of MEM and its main components. The biggest difference with TAM is the introduction of actual efficacy coming from Rescher (1977). This difference is subtle but nevertheless very important. It implies that when explaining human behavior, the subjective reality is often much more decisive than the objective reality and therefore perceived efficacy mediates the impact of actual efficacy on adoption in practice (Bernaert, Poels et al. 2013).

Based on these models, guidelines can be developed to optimize the adoption of the developed IS (Bernaert, Poels et al. 2013), in this case the EA approach CHOOSE. The development of a tool supporting the application and implementation of CHOOSE could significantly contribute to the actual efficacy, leading to a higher adoption and added value through an increase in the subjective perception of this efficacy. Hence, measuring the perceived usefulness and perceived ease of use of EASE during the validation process, will provide valuable insights with respect to the ability of this software tool support to increase adoption of the CHOOSE approach.

Next to the contribution of a tool to the adoption of an approach, research concerning the implementation and use of EA in practice stresses the complexity and need for guidance by means of tool support. In general, there are three main areas where critical problems arise in the process of EA: modeling, managing and maintaining EAs (Kaisler, Armour et al. 2005). An important driver of problems in these areas is the inherent complexity of the EA process. An enormous amount of information has to be transformed using the semantics and syntax of the modeling language. Often this information is distributed over multiple people and has to be brought together to be able to create a consistent and coherent whole. This process takes a lot of energy and is often a hotbed for errors. A tool can offer the much needed support and guidance for the development, storage and analysis of an EA (Ernst, Lankes et al. 2006). Furthermore, in many companies, enterprise architects are obliged to use existing methods and techniques from disparate functional domains, preventing them to realize the bigger picture that puts all these domains together in the required EA (Jonkers, Lankhorst et al. 2006). This drawback emphasizes the importance of an integrated tool for building, analyzing and communicating the EA to all stakeholders. Other advantages of tool support include (Lankhorst 2009):

- A tool can help standardize the semantics and syntax used during the development of the EA within a company.
- The use of a tool contributes to the construction of correct and consistent architecture artifacts by guiding the development process and through the application of mistake proofing techniques. Tools can impose rules to make sure the desired practices and guidelines are followed.
- A tool can help the enterprise architect in the use of architectural patterns. Furthermore, the reuse of certain parts of the architecture or solutions that have been developed in the past is facilitated. This contributes to the efficiency of the development process.
- Tools facilitate the comparison of alternatives by providing impact of change and quantitative analysis features.

Although the aforementioned research confirms the importance of tool support, these findings cannot simply be extrapolated to the environment of SMEs and the importance of tool support for the implementation of the CHOOSE approach. However, case studies performed by Bernaert and Callaert (upcoming paper) confirm the need for tool support for the development of EAs in SMEs. During these case studies, the CHOOSE technique was applied in six SMEs by means of simple whiteboards.
and post-its. During four of these case studies, this technique was supplemented with an initial version of EASE. Comparing both clearly showed the added value of having access to a software tool supporting the EA process. Figure 5 shows a small fraction of the resulting EA and pinpoints the importance of a tool for the development, storage and analysis of the EA artifacts, since the use of post-its created an unmanageable EA model.

![Figure 5: Partial EA artifact of a Belgian SME](image)

**Importance New Software Tool**

Most tools available on the marketplace are based on EA frameworks. The Zachman framework is one of the best-known and is often used as a descriptive framework through which EA artifacts can be categorized (Zachman International 2011). Another well-known framework and method is The Open Group Architecture Framework (TOGAF) (The Open Group 2009). TOGAF is considered to have a broader application than Zachman and provides more guidance for the development of the EA. Both frameworks provide support on a high level of abstraction and primarily help to decide which business and technological domains to incorporate in the EA but they provide little assistance in creating the architectural artifacts themselves (Jonkers, Lankhorst et al. 2006). Nevertheless, there are software tools available that help the enterprise architect in the creation of these artifacts. For example, Objectiver is a requirements engineering tool that allows the end-user to draw a part of or the entire EA on a canvas (Respect-IT 2010). However, this software tool was not developed for this specific purpose and it is based on the metamodel of KAOS, a goal-oriented software requirements engineering approach. To be applicable for the modeling of EA artifacts, some significant changes should be made. Furthermore, the focus of this and other available software tools is not on SMEs, which leads to the incorporation of unnecessary and potentially confusing features while other important features may not be available at all because they are not valued by the broader end-user base.

In general, research has identified some common weaknesses of the available software tools (Ernst, Lankes et al. 2006):

- Most software tools do not support automatic visualization of data. Furthermore the semantics of these visualizations are often only defined in vague terms. A lot of these tools are plain drawing tools in which the end-users have to draw the EA on a canvas themselves. However, a variety of important functionalities, like the reuse of model components, cannot be supported by these kinds of tools (Braun and Winter 2005).
- Most software tools come with a predefined metamodel. Due to the considerable differentiation between companies, these metamodels ought to be adjusted before the architect can start developing the EA. These standard metamodels are either too small to be able to capture the entire EA or too big, impeding the ease of use and readability of the resulting model.
- Most software tools provide a metamodel but lack a method for developing the EA. The absence of a method can slow down the development process and can lead to the misinterpretation and wrongful application of the concept of the metamodel.

Based on these insights, it is safe to say that the development of a software tool adjusted to the specific needs of SMEs, based on the CHOOSE approach and incorporating these general weaknesses could substantially improve the added value of EA for SMEs.

**Methodology**

A solid methodology for the development of the software tool is indispensable and contributes to the academic and practical value of this research-in-progress. A methodology was derived from three well known methodological frameworks:

- **Conceptual Framework for the Methodological Soundness of Requirements Engineering (RE) Papers** (Wieringa and Heerkens 2006): This framework proposes a set of criteria based on which the methodological soundness of RE papers can be evaluated. As the RE domain shows substantial affinity with this research-in-progress, valuable insights were obtained from this framework.
• **Design Science Research Methodology for Information Systems Research** (Hevner, March et al. 2004; Peffers, Tuunanen et al. 2007): This framework proposes a methodology for conducting design science research (DSR) in the Information Systems (IS) academic discipline. DSR aims at designing artifacts (e.g., concepts, models, methods, instantiations of these) that embed scientific knowledge about problems and their solutions. A software tool can be seen as an instantiation of a method, which in case of a modeling method is based on an underlying metamodel and concepts, through which the method can be evaluated (March and Smith 1995). Ergo, it might be a good idea to incorporate the guidelines of this framework in the methodology for developing the tool as a research artifact.

• **Software Process Models** (Boehm 1988; Sommerville 1996): A software process is an abstract collection of activities and information necessary for the development of a specific software system. Each organization has its own software process but usually these individual approaches follow some generic abstract model. These are called software process models and they provide insight in the different steps from concept up to the finished software system. By analyzing and comparing the different types of models, some valuable insights were obtained.

Through the consolidation of the relevant and valuable aspects of the aforementioned models and frameworks, a customized methodology was created for the elaboration of this research-in-progress. Figure 6 gives a schematic overview of this. Indicated in blue (bottom three lines) are the relevant elements of each of the underlying approaches and in red (top line) the resulting methodology used in this research. Throughout the development process of the software tool, insights in the underlying approaches can be used to maximize the academic and practical value of this research.

![Figure 6: Methodology](image)

**EASE Criteria**

The first step in the development process consists of an extensive literature review with the main objective of translating the criteria of EA and SMEs (figure 2), from research step 1 in figure 3, together with knowledge of the CHOOSE metamodel and method into general guidelines for tool support. In figure 3, this is depicted by research step 5. These guidelines for software tool support are primarily derived from knowledge gathered by means of literature from the following fields:

- Adoption models for IS and IS models.
- Problem areas of EA implementation, management and maintenance.
- Weaknesses of current (software) tools.
- CHOOSE metamodel and method.

This results in the following design objectives for EASE, reflecting research steps 1 through 4 of figure 3:

1. **Simplicity**: The tool ought to be easy to learn/use and should provide a user-friendly interface through which the end-user can access all available features. Moreover, it allows the dispersed information to be gathered from the different people involved in the EA process through interaction with a standard interface. The latter helps improve the agility and flexibility of the software tool and therefore contributes to its added value. Furthermore, the simplicity should significantly reduce the threshold for an enterprise-wide implementation (criteria 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 5).

2. **Efficiency**: The input of the end-user should be kept to a bare minimum. In particular, this means that the time, effort and costs to use EASE have to be kept as low as possible. Hence, the software tool should allow employees to work time-efficiently. Furthermore, the simplicity criterion (confer supra) contributes to the development of a software tool with a steep learning curve, reducing the costs for education and adaption significantly (criteria 4.1, 4.6).
3. **Effectiveness**: Given the minimal input, the software tool must maximize the added value of the output. Therefore, EASE must be very flexible and transformation of the data should provide valuable information for a variety of viewpoints. Omitting irrelevant data reduces the complexity of the model significantly, allowing the end-user to keep a holistic overview. Incorporation of various constraints can help enforce generally accepted guidelines and best practices, contributing to a consistent and coherent end result (criteria 1, 2, 3, 4, 4.4, 4.6).

4. **Business oriented**: The level of IT and EA knowledge in the environment of SMEs is quite limited. To maximize adoption, it is of utter importance that the software tool communicates using the language of the end user, in case of SMEs the business language (criteria 4.2, 4.3, 4.5, 5).

5. **Completeness**: EASE must provide guidance throughout the entire EA process. Furthermore, it is important to keep the general weaknesses in mind when developing the software tool to prevent making similar mistakes and provide a solution to the needs of the marketplace (criteria 2, 4.4, 5).

These general objectives were supplemented with criteria specifically related to software tool support and development such as adaptability, modularity, GUI design and flexibility. However, these criteria are more technical in nature and will not be discussed in detail in this paper.

**Tool Development**

By means of a literature review of EA in practice (Kaisler, Armour et al. 2005; Ernst, Lankes et al. 2006; Lankhorst 2009) and by applying the CHOOSE approach in practice in multiple case studies, the following problem areas were identified:

- **Input**: This step implies transforming the available knowledge and information into the EA through the application of the syntax and semantics provided by the CHOOSE approach. The enormous amount of information, its distributed and dynamic character and the interdependencies between the different concepts make this a time-consuming and often very complex process in need of guidance by means of tool support. Moreover, software tool support can impose constraints that contribute to the compliance with predefined guidelines, best practices and the rules of the metamodel in contrast with the previously mentioned methods in which the user is not restricted in the actions (s)he can take.

- **Storage**: Once modeled, the architecture has to be stored to be able to access, adjust or analyze it in the near future. In the absence of tool support, data is often stored physically, for example through post-its on whiteboards, or in the best case in an in-house developed database. It is quite clear that this is a suboptimal solution, significantly reducing the user-friendliness and added value of the EA.

- **Adjust data**: In society today, companies are operating in highly dynamic environments in which changes occur on a daily basis. Consequently, they require their systems, including their EA, to be able to respond to this need for agility and flexibility. Software tool support allows quick and effortless adaptation of the architectural artifacts, hence contributing to this important business need.

- **Retrieving data**: Whether data ought to be adjusted or information is needed to perform a specific analysis, it is necessary to find and collect those specific elements of the EA that will help the end-user to do so. Software tool support can significantly reduce the effort required to perform this process of retrieving data, contributing to the added value of EA.

- **Analysis**: Dependent on the stakeholder involved, the degree of relevance of different aspects of the EA varies significantly. Ergo, it is very important to be able to reduce the complexity inherent to EA by omitting irrelevant data and varying the level of detail according to the background, knowledge and purpose of the end-user by means of different viewpoints. Furthermore, analysis of the EA can identify and resolve any inconsistencies and other irregularities.

In absence of tool support, the effort required in each of the aforementioned problem areas in terms of time, cost and energy would increase exponentially with the amount of data stored in the EA. Lankhorst identifies three categories of tools in support of the EA process (Lankhorst 2009). Each of these categories can be linked to one or more of the five problem areas discussed above:

1. **Modeling and design**: This category supports the problem area with respect to input.
2. **Reporting and publication**: This category offers a solution for the problem areas with respect to the retrieval and analysis of data.
3. **Storage and retrieval**: This category tackles the storage, retrieval and adjustment of data problem areas.

Hence, a complete and valuable software tool has to offer a solution in support of each of these problem areas. Consequently EASE has to integrate the three different categories into one umbrella tool offering end-user support for the entire end-to-end EA process.
Based on these insights, EASE was developed around three main functionalities: input, adjust and output (figure 7 left). The software tool was developed using Java as programming language and MS Access as database management system (DBMS). In the remainder of this paper, a brief overview will be given of the main features of the software tool EASE. Furthermore, a short description of the programming challenges yet to be tackled are provided. Finally, this paper ends with a concluding remark with respect to validation.

**Input**

The different elements and their interdependencies have to be modeled and stored by means of the concepts and relationships of the CHOOSE approach provided to the end-users through a user-friendly and highly intuitive interface. First, the end-user chooses the concept that (s)he wants to add and the interface provides a window that asks for the necessary information and gives an overview of all the possible relationships with other entities within the EA. The right part of figure 7 illustrates the input of a goal using the software tool under development.

Moreover, a variety of constraints are imposed to ensure a consistent and coherent EA. The storage in the underlying database occurs automatically and is oblivious to the perception of the business user. Hence, this separation of concern ensures that in case the DBMS no longer meets the requirements, it can easily be replaced without noticeable difference for the service consumers.

**Adjust**

There are a number of ways through which the end-user can look up and potentially adjust entities, their attributes and their relationships within the EA. The straightforward way is just by searching for a specific entity by means of a search functionality provided by the software tool as depicted in figure 8.
The software tool also provides the business user with the functionality to represent the entities per dimension of the CHOOSE approach in a tree structure. Allowing the end-user to drill down through the entities, starting at the highest level of abstraction, hence providing a holistic overview of the overall structure of the entities of one of the four core dimensions (figure 9 left). Furthermore, by clicking on a specific entity, represented by a node of the tree, all connected elements are displayed following the visual representation of figure 1. By consistently applying this structure, the end-user always knows where to find information concerning a specific dimension, contributing to the user-friendliness and efficiency of the software tool.

![Figure 9: Tree structure overview (left) & Focused architectural overview (right)](image)

**Output**

The output part of the software tool is currently still under development. A brief description will be given of those features that are currently operational. The focused architectural overview provides the business user with the possibility to zoom in on a specific entity of the EA including its immediate environment, offering an automatic visualization functionality that allows the creation of a plethora of viewpoints. This feature tackles one of the general weaknesses of the current tool landscape as described earlier. For the development of a clear, understandable and effective visualization, insights were obtained by means of an extensive literature review with respect to visual perception and the syntax of modeling tools (Lankhorst 2009; Moody 2009). The right part of figure 9 gives an example of the focused architectural overview feature.

Case studies have revealed that MS Excel remains an important tool within the environment of SMEs for the communication, transformation and analysis of data (Osadnik and Landryova 2011). Hence, incorporating a functionality allowing the export of data from the Access to the Excel environment, proved to improve the added value of the software tool. The data can either be exported in the form of lists as they are stored in the database or they can be transformed into a meaningful representation. An example of the latter is the combination of data of the actor and operation dimension resulting in the construction of a RACI chart (ISACA 2012).

Currently still under development are an as-is/to-be analysis and an impact-of-change analysis functionality. The former should allow starting from the as-is situation and adjusting the EA to incorporate the changes necessary to achieve a desired future state. Comparison of both models identifies the steps that ought to be undertaken to accomplish this transformation. The latter can be used to investigate the impact of a specific change on the different dimensions and the EA as a whole.

**Validation**

**Approach**

As stated by MEM, the validity of a method should be proven through the successful application in practice. Hence, a first pretest stressed the importance of tool support and provided us with a preliminary feedback on tool functionalities. Subsequently CHOOSE was implemented in four Belgian SMEs, which will be further elaborated here. These companies were submitted to the situation in which software tool support was initially not available, but later in the EA process they
were given the beta version of EASE to support them in managing and maintaining their EA. Therefore, these companies can clearly judge the added value of EASE as they have been exposed to both situations without and with software tool support.

Main Results

By means of a survey, the CEOs of each of the participating companies were asked to evaluate the perceived usefulness and perceived ease of use of EASE through the adapted six-item scales of Davis (1989) for measuring both variables (figure 10). On average the companies perceive it as more than slightly likely that EASE has a high degree of usefulness (PU) and ease of use (PEU). Furthermore variability is low which means that EASE scores consistently well throughout the four case studies. EASE, however, is not yet fully developed. Nevertheless, these results confirm its potential of improving the adoption of CHOOSE through an increase in the PU and PEU.

![Table](Figure 10: Evaluation perceived usefulness and perceived ease of use)

Besides the perceived efficacy, feedback was obtained with respect to the evaluation of the CHOOSE (figure 2) and EASE (cf. section ‘EASE Criteria’) criteria. Criteria 4.5 and 4.6 were not evaluated, since the CEO was always involved (4.5) and the adoption was already assessed in figure 10 (4.6). In general, the CEOs positively evaluated the contribution of EASE to and the compliance of EASE with the predefined criteria (figure 11). Especially the reduction of the inherent complexity of an EA and the improved overview are considered valuable assets of EASE. The contribution of EASE to criterion 5 (EA 5) is less straightforward. A potential explanation for this is that EASE has a limited impact on the scope on which CHOOSE is applied and CHOOSE has to be used in combination with other approaches (e.g., business process modeling languages) in order to see every detail of the enterprise.

![Table](Figure 11: Evaluation CHOOSE and EASE criteria)

CONCLUSION

This research-in-progress has investigated the need for a software tool in support of the implementation of EA in the environment of SMEs as pioneered by the CHOOSE approach. Both literature review and case studies have confirmed this need and the paper presented a software tool under development (EASE) in support of this need based on criteria for EA, criteria for SMEs and criteria for tool support. An overview of the main features of EASE was given and an initial validation by means of four case studies has confirmed the potential of the software tool in increasing the adoption of CHOOSE and providing the much needed guidance and support. Furthermore, EASE has reached its goals through the contribution to and the compliance with the predefined criteria. Nevertheless, EASE is still under development and the case studies have identified multiple improvement paths to be tackled. Further research with respect to additional valuable functionalities is required and continuous fine-tuning will contribute to the overall added value of EASE in support of CHOOSE.
REFERENCES