Demonstrating Approach Design Principles during the Development of a DEMO-based Enterprise Engineering Approach

Thomas van der Meulen[1,2], Marné de Vries[1] and Aurona Gerber[1,3]
[1]*Department of Industrial Engineering, University of Pretoria, Pretoria, South Africa*
[2] *Engineering Department, Bertie van Zyl (Pty) Ltd, Pretoria, South Africa*
[3] *CAIR, CSIR Meraka as well as Department of Informatics, University of Pretoria, Pretoria, South Africa*

Keywords: Enterprise Engineering, Design Principles, Approach Design.

Abstract: Enterprise engineering (EE) aims to address several phenomena in the evolution of an enterprise. One prominent phenomenon is the inability of the enterprise as a complex socio-technical system to adapt to rapidly-changing environments. In response to this phenomenon, many enterprise design approaches (with their own methodologies, frameworks, and modelling languages) emerged, but with little empirical evidence about their effectiveness. Furthermore, research indicates that multiple enterprise design approaches are used concurrently in industry, with each approach focusing on a sub-set of stakeholder concerns. The proliferating design approaches do not necessarily explicate their conditional use in terms of contextual prerequisites and demarcated design scope; and this also impairs their evaluation. Previous work suggested eleven design principles that would guide approach designers when they design or enhance an enterprise design approach. The design principles ensure that researchers contribute to the systematic growth of the EE knowledge base. This article provides a demonstration of the eleven principles during the development of a DEMO-based enterprise engineering approach, as well as a discussion to reflect on the usefulness of the principles.

1 INTRODUCTION

Enterprise engineering (EE) studies enterprises from an engineering perspective (Albani and Dietz, 2010) and addresses the need for a comprehensive view of the enterprise (Giachetti, 2010; Hoogervorst, 2009; Kappelman, 2010; Van Tonder and Roodt, 2008). A recent study focused on defining the domain of the EE discipline — i.e., identifying the phenomena and core problems of interest that require research within the discipline (De Vries, Gerber, and Van der Merwe, 2015). Survey participants of the study highlighted a prominent phenomenon, namely the inability of an enterprise, as a complex socio-technical system, to adapt to rapidly-changing environments (De Vries, Gerber*, et al.*, 2015). The research also highlighted the need for appropriate *methodologies* and *architecture description* (using typologies and modelling languages). Yet, numerous approaches (including methodologies, frameworks, and modelling languages) already exist (De Vries, Van der Merwe, and Gerber, 2015). From a practical viewpoint, enterprises usually apply a combination of methodologies to cover several design domains (Blowers, 2012). The diversity of modelling languages is not surprising if we recognise that enterprise stakeholders have different concerns, which need to be addressed via multiple enterprise design approaches to deal with the full richness of the real world (Espejo and Reyses, 2011; Mingers and Brocklesby, 1997).

The proliferation of enterprise design approaches however impairs the systematic growth of the EE knowledge base, since many design approaches do not necessarily explicate their conditional use, contextual prerequisites, and demarcated design scope. Drawing from existing theory on theoretical enterprise design approaches (De Vries, Van der Merwe*, et al.*, 2015) and eight components for *design theory* in IS (Gregor and Jones, 2007), design research was used to develop eleven *approach design principles* (ADPs) to guide the approach developer (De Vries, 2016). Although the study applied a focus group discussion to validate the principles, the principles had not been applied to a real scenario. Additional demonstration was needed to further evaluate the usefulness of the eleven ADPs.

Meulen, T., Vries, M. and Gerber, A.
Demonstrating Approach Design Principles during the Development of a DEMO-based Enterprise Engineering Approach.
DOI: 10.5220/0006382204710482
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 3*, pages 471-482
ISBN: 978-989-758-249-3

The purpose of this article is to demonstrate the use of a generic set of ADPs during the development of a new approach, called the DEMO-based enterprise engineering approach (DEEA), which had been applied at a real enterprise. In addition, we discuss the usefulness of the ADPs.

The article is structured as follow: section 2 provides background on a real enterprise and its need for a new enterprise design approach, as well as a summary of eleven ADPs that should guide the development of new enterprise design approaches. Section 3 presents the research methodology that was used to evaluate the ADPs by demonstrating their use. The demonstration of the ADPs and interview results are discussed in section 4, followed by a summary of the results and opportunities for ADP improvement in section 5. We conclude the article in section 6.

2 BACKGROUND

The main approach developer of DEEA identified the need to guide enterprise evolution at an agricultural enterprise, called ZZ2. Management at ZZ2 encourages an open systems approach (evident on the ZZ2 web site (www.zz2.biz, Vision), which encourages farmers or other role players to continuously adapt plans as necessitated by the dynamic agricultural environment. Based on discussions with employees, it was evident that ad hoc and unintegrated designs created frustration, since the current ad hoc approach failed to identify and address enterprise design concerns in a holistic way (Van der Meulen, 2017). The new approach had to improve the *quality of design*, which according to Gause & Weinberg's (1989), can be defined as the extent of identifying and addressing relevant stakeholder requirements for the to-be system.

Van der Meulen (2017) considered a number of enterprise design approaches to replace the ad hoc nature of enterprise design with a more holistic, comprehensive and integrated design process. Hoogervorst's (2009) approach was selected as a good candidate, due to its holistic way of eliciting stakeholder concerns and related design principles that would govern the evolution of enterprise design domains. In addition, requirements are translated into constructional specifications for the organisation domain in the form of DEMO models, which are, according to Dietz (2006), coherent, comprehensive, consistent and concise.

2.1 Need for an Enhanced Approach

At the start of the study, Hoogervorst's (2009) approach was considered as an appropriate approach for application at ZZ2. Yet, during the implementation of the approach, especially the *generic enterprise development framework* (GEDF) of Hoogervorst (2016), practical implementation tools and supporting software were absent. A study was initiated, using action design research and guidelines from Sein *et al.* (2011), to develop the DEEA (Van der Meulen, 2017). The developer of DEEA also applied the ADPs of De Vries (2016) to explicate DEEA's conditional use in terms of *contextual prerequisites* and *demarcated design scope*.

2.2 Approach Design Principles

Drawing from existing theory on theoretical enterprise *design approaches* (De Vries, Van der Merwe*, et al.*, 2015) and eight components for *design theory* in IS (Gregor *et al.*, 2007), design research was used to develop eleven *approach design principles* (ADPs).

The term *design approach* features in almost every principle statement and relates to an enterprise design approach, designed by an approach developer, with the intent of guiding the evolution of an enterprise system or sub-system.

We distinguish between an *approach* and a *methodology* by referring to the four-tier inheritance structure defined by Iivari, Hirschheim, and Klein (2001), i.e. paradigms, approaches, methodologies and techniques.

Furthermore, De Vries (2016) states that an enterprise *design approach* is based on a particular *conceptualisation* of the enterprise; is designed to *create value* for the enterprise; focuses on designing *particular design domains* or sub-systems; highlights particular *concerns* that may be neglected in other approaches; and incorporates several *mechanisms and practices* that will enable value-creation for the enterprise.

This section summarises the principles as defined by De Vries (2016) in terms of a *statement* and *rationale*. The *rationale* is stated in terms of the main focus of the ADPs, namely to guide the approach developer in explicating the *conditional use* of a newly-developed approach in terms of its *contextual prerequisites* and *demarcated design scope* (De Vries, 2016).

The practical components of each ADP (i.e. the ADP *implications* and *measures*) are only presented in section 4 when we demonstrate their use during the development of a new enterprise design approach, called DEEA.

Principle A: Explicit Concept of the Enterprise

Statement: A design approach should indicate how an enterprise is perceived or *conceptualised*.
Rationale: Different analogies are used to conceptualise the enterprise, such as machines, biological systems, and psychic prisons (De Vries, Van der Merwe*, et al.*, 2015), which may also differ from one industry to the next. By explicating the enterprise conceptualisation(s) the approach author also acknowledges the limitations of a particular conceptualisation, as indicated by Morgan (2006).

Principle B: Explicit Phenomenon

Statement: A design approach should provide evidence for a *phenomenon* or *class-of-problems*, i.e. similar kinds of problems.
Rationale: Phenomena "that are not fully understood cannot be properly addressed and improved" (Hoogervorst, 2016, p. 49). Although numerous approaches may already exist to address a well-understood phenomenon, there is still a lack of knowledge about approach performance (De Vries, Gerber*, et al.*, 2015).

Principle C: Explicit Paradigm of Value Creation

Statement: A design approach should state a *paradigm of value-creation* as a testable proposition for addressing an existing *phenomenon* or *class-of-problems.*
Rationale: Creating testable propositions for existing and new approaches provides a starting point to extend the existing EE knowledge base. From a practitioner's perspective, the value-creation paradigm clarifies the intended value of the approach.

Principle D: Explicit Means (Ways) of Demarcating and Representing Design Scope

Statement: A design approach should clearly define and motivate the way to demarcate design scope (*enterprise scope*, *design domains,* and *concerns/requirements*) relevant to the approach.
Rationale: Demarcation of design scope (*enterprise scope, design domains, concerns/requirements*) is contextual and depends on the intentions (*paradigm of value-creation*) of the observer/ analyst (Espejo *et al.*, 2011; Giachetti, 2010). Thus, acknowledging that demarcation of design scope will inevitably differ from one approach author to the next, the approach author should stipulate the method or theoretical grounds used for demarcation, which encourages reflection about the demarcation rigour.

We clarify definitions for *enterprise scope*, *design domains* and *concerns*.

The *enterprise scope* provides a dimension to demarcate the scope of design for a specific approach, by referring to the *internal structures* of the enterprise — e.g., business units, lines of business, departments, programmes and projects — and the *external legal entities*, such as government, partners and suppliers.

The demarcation of *design domains* depends on the approach author's intentions or *paradigm of value-creation*. For example, Hoogervorst (2009, p. 134) maintains that the demarcation/delineation of domains reveals "functional or constructional *system facets* for which design activities are required".

Concerns acknowledge a third way of demarcating enterprise design scope. Approach authors highlight particular *concerns* or *areas of concern/requirements* that should be addressed during enterprise design. For example, Parmenter (2010) highlights six areas of concern for enterprise design: financial asset utilisation, operational performance, customer satisfaction, employee satisfaction, community engagement, and learning and growing.

Principle E: Well-demarcated and Well-defended Design Scope

Statement: A design approach should define and defend the intended design scope to achieve the intended *value-creation*.
Rationale: The new approach should demarcate an appropriate scope to achieve the intended value-creation, and relate to existing theory that focuses on a similar scope.

Principle F: Representations of Design Scope

Statement: A design approach should clearly define and motivate notation standards that are used to adequately describe/represent the design scope.
Rationale: Multiple languages and notation standards already exist to represent different perspectives or design domains of the enterprise. Yet existing notation standards are usually based on a particular notion about the nature of the enterprise and its sub-systems. The approach author should defend why the selected language and notation standard is appropriate within his/her conceptualisation of the enterprise. A new conceptualisation or notion of the enterprise may require deviation from existing notation standards. As an example, the Business Process Modelling Notation (BPMN) standard may be used to depict the business organisation domain. Yet, Dietz (2006) criticizes BPMN for being too

implementation-specific and motivates the need for representing the *essence* of enterprise operation by suggesting a new notation standard, i.e. the Design and Engineering Methodology for Organisations (DEMO).

Principle G: Approach Form and Function

Statement: A design approach should clearly define the constructs and features of the approach.
Rationale: The basic design process for any artefact, including a new *design approach*, requires both functional design and constructional design (Dietz and Hoogervorst, 2007). For a *design approach*, the *paradigm of value creation* represents its function, whereas constructional parts represent the form. An artefact, such as a new *design approach*, can only be *maintained and tailored* in future if both its function and construction/form is known (Dietz *et al.*, 2007). The four components of EECM provide a meta-model of typical constructs of existing design approaches. The third component (*mechanisms and practices*) also highlights ten different kinds of constructs that may form part of an approach.

Principle H: Justificatory knowledge

Statement: A design approach must provide explanatory knowledge that links the *paradigm of value-creation* with its constructional components.
Rationale: The justificatory knowledge provides an explanation of why an artefact is constructed as it is, and why it works. Pointers to *some* kernel theories would provide researchers and practitioners with information that would be useful when comparing or combining approaches. It may also be possible that new theories are formulated, which cannot be traced to kernel theories.

Principle I: Approach Mutability

Statement: A design approach should clearly state possibilities for tailoring the approach, within the pre-defined design scope.
Rationale: Since the design approach may not have been demonstrated for multiple instances within the pre-defined design scope, the designer needs to identify possibilities for tailoring the approach. Design approaches need to address the dynamic nature of the enterprise and its environment, which is also a key concern within the EE discipline (De Vries, Gerber, *et al.*, 2015).

Principle J: Principles of implementation (conditional)

Statement: A design approach may incorporate guidance for implementing the approach.
Rationale: This principle is *conditional*, since the designer needs to consider the pre-defined design scope and decide whether additional advice would add value, e.g. additional advice may be required if the approach has been designed for the health industry. Even for other industries, the designer may need to provide additional advice to indicate how the approach should be used in specific contexts.

Principle K: Expository Instantiation (Optional)

Statement: A design approach may incorporate an instantiation.
Rationale: This principle is *optional*. A realistic implementation of an approach contributes to the identification of potential problems in its design, also demonstrating its worth. Even though the designer should have implemented the approach to evaluate its utility, the implementation results do not necessarily form part of the *construction* of the design approach.

3 RESEARCH METHODOLOGY

Design research acknowledges the development of *principles* as a valid form of knowledge contribution (Hevner et al., 2004). Previous research applied the first three phases of the design research (DR) cycle of Peffers, Tuunanen, Rothenberger, and Chatterjee (2008), i.e. (1) *identify the problem*, (2) *define the objectives* of the solution, and (3) *develop the solution*, which was a set of eleven ADPs. The ADPs were also *validated via a focus group discussion*.

This article focuses on the subsequent steps of Peffer *et al.'s* (2008) DR cycle, namely (4) *demonstration*, i.e. finding a suitable context for demonstrating the ADPs, and (5) *evaluation*, i.e. evaluating the usefulness of the ADPs.

In terms of *demonstration*, an approach developer, who was *not the developer of the ADPs*, applied the ADPs when he developed a new approach, called DEEA. He documented the application of the ADPs as part of a Masters dissertation. For *evaluation*, the ADP's developer followed the guidelines of Marshall and Rossman (2011) to perform an in-depth 3-hour interview with the *user* of the ADPs. The interview was structured to inquire about every ADP, especially how the *implications* and *measures* were interpreted and applied by the APD user. The interview allowed for additional inquiry where the ADP user applied a different interpretation than was intended (see Section 4.4). Finally, the interview feedback, presented per ADP, was sent to the interviewee for validation and additional inputs.

The next section presents DEEA in terms of the eleven ADPs, i.e. demonstrating how the ADP's *implications* and *measures* were applied during the development of DEEA. In addition, we provide feedback per ADP regarding the use of its *implications* and *measures*, based on the interview feedback from the interviewee.

4 DEMONSTRATION OF PRINCIPLES

This section presents each principle in terms of the ADP *implications* and *measures*, followed by their *application* during the development of DEEA and *interview feedback* from the approach developer. During the interview feedback, the approach developer had to reflect on the usefulness of the particular ADP, highlighting opportunities for further improvement.

4.1 Principle A: Explicit Concept of the Enterprise

Implications:

- Provide a description of the enterprise, using analogies.
- Provide a motivation, also referring to *supporting theory*, for using the particular enterprise conceptualisation.

Measures: No additional measures.

Principle Application

Since DEEA was based on Hoogervorst's approach, his conceptualisation of the enterprise was also adopted, namely that the enterprise resembles an organised complexity, which should be purposefully designed. DEEA also adopted the notion that an enterprise is a heterogeneous system that consists of many sub-systems in accordance with Dietz (2006). Furthermore, the sub-systems are often constructed as to support one another, e.g. ICT systems are constructed to support the operation of the enterprise.

Interview Feedback

The interviewee indicated that the ADP *implications* were sufficient. However, he commented that one of the prerequisites for applying a particular approach is that management had to appreciate the enterprise conceptualisation, which would ensure buy-in and active participation. As an example, management at ZZ2 also adopted a systems notion of the enterprise. More so, they used a similar conceptualisation of an enterprise object system that needs to be constructed in support of a using system. Hoogervorst's conceptualisation of the enterprise correlated with ZZ2's paradigm of an organised complexity, which would also allow buy-in from ZZ2.

Furthermore, an entomologist at ZZ2 referred to Aristotle's belief that even nature is teleological, since nature serves ends (see Lloyd (1968)), which compares well with the notion of an object system that serves a using system.

4.2 Principle B: Explicit Phenomenon

Implications:

- Produce *sufficient* evidence that an existing *phenomenon* or *class-of-problems* exists, but that it is *inadequately addressed* by existing theory or theory application.

Measures: No additional measures.

Principle Application

The study was initiated by identifying a *problem instance* at ZZ2, which was then validated as a *class-of-problems* from a theoretical viewpoint. The *problem instance* at ZZ2 is that the enterprise developed in an ad hoc way to accommodate the dynamic environment of the agricultural industry. An industrial engineer that practiced in an agricultural environment for more than nine years stated: "I do not even want to begin thinking about putting this company into circles and rectangles". Due to its dynamic nature, the agricultural enterprise cannot be represented with the same rigour than a nuclear power station or a financial department.

This *problem instance* features as a *class-of-problems* in literature. Since the farming industry is dependent on many variables such as temperature, rainfall, wind, soil diseases and other natural or political factors, it becomes a very complex environment to implement system standardisation or design techniques with the intent of gaining more control (Nuthall, 2011). Geng, Ren, and Wang (2007) state that it is difficult to implement IT solutions in the agricultural environment, since there is a lack of standard measures. In addition, produce is diverse and difficult to handle. Wolfert et al. (2010) state that agricultural companies need to constantly innovate to be able to survive in the complicated and dynamic agri-food environment.

According to the developer of DEEA, current theory existed to provide some design guidance within a dynamic agricultural environment. Whereas systems engineering was too generic to provide design guidance for an enterprise, Hoogervorst's (2009, 2016) approach acknowledges enterprise dynamics and focuses on holistic enterprise design and representation of organisation construction in a coherent, comprehensive, consistent and concise way (Van der Meulen, 2017). Yet, an application of Hoogervorst's approach indicated practical gaps. An approach was required to facilitate ease-of-understanding when eliciting enterprise requirements from employees with an agricultural background. In addition, requirements and enterprise designs had to be traceable and documented in an unambiguous way.

Interview Feedback

Although the approach developer adhered to the ADP *implications*, the *implications* focuses too much on the identification of a *class-of-problems*. The implications should rather focus on the identification of a *problem instance*, prior to its generalisation as a *class-of-problems*.

4.3 Principle C: Explicit Paradigm of Value Creation

Implications:

- State the intended *paradigm of value-creation* in terms of a testable proposition, which may be in the form 'If approach X is instantiated, then it will achieve the intended value, or it will be better in some way than other approaches'.

Measures: No additional measures.

Principle Application

If DEEM is instantiated, then it will:

- Generate a concept of the enterprise in an integrated manner for a particular design scope of interest,
- Address stakeholder concerns from a holistic point of view,
- Ensure ease-of-understanding for employees with an agricultural background during enterprise requirements elicitation, and
- Represent enterprise requirements and conceptual to-be construction in an unambiguous way.

Interview Feedback

The ADP *implications* are sufficient.

4.4 Principle D: Explicit Means (Ways) of Demarcating and Representing Design Scope

Implications:

- Define the way to demarcate design domains (e.g., using systems theory or demarcation heuristics).
- Define the way to demarcate concerns per design domain (e.g., using generic areas of concerns from theory or using heuristics to identify concerns).

Measures: No additional measures.

Principle Application

- *Way to demarcate design domains:* DEEA applied the basic systems design process delineated in Dietz (2006) to define domains that have supportive relationships, e.g. the information domain (i.e. the object system) supports the organisation domain (i.e. the using system).
- *Way to demarcate concerns per design domain:* DEEA applied Hoogervorst's (2016) heuristics, embedded in the *generic system development framework*, to elicit areas of concern and enterprise requirements.

Interview Feedback

Although the approach developer was able to define the means of demarcation, as stipulated under *Principle Application*, he initially defined DEEA's design domains, rather than their *means* of demarcation. The ADP should explain the phrase "means of demarcating".

4.5 Principle E: Well-Demarcated and Well-defended Design Scope

Implications:

- Identify the overall *scope* of the approach — i.e., focusing on *inside-the-boundary* complexities versus *outside-the-boundary* complexities.
- Identify the scope of the approach in terms of *design domains* that will be addressed by the approach.
- Identify the scope of the approach in terms of areas of *concern* for different *stakeholders* that will be addressed.
- Use additional ways of scoping to define the context for which the approach is intended — e.g., a specific industry.

- Indicate the actual *implementation* scope used to demonstrate the approach.
- Identify the role players, especially the main *users* of the approach.

Measures:

- Is the scope described in such a way as to relate to existing theory that covers a similar design scope?
- Is the scope described in such a way as to know whether an application context falls within the scope of the approach?

Principle Application

- *Overall scope of the approach*: DEEA focuses on *inside-the-boundary* complexities rather than *outside-the-boundary* complexities.
- *Design domains* scope: DEEA provides a method to design three design domains, i.e. organisation, information and infrastructure. The rationale and demarcation of the designs are discussed in Van der Meulen (2017).
- *Areas of concern*: DEEA is not prescriptive on the areas of concern that should be addressed, but rather applies the *generic system development framework* to identify enterprise-specific areas of concern.
- *Industry:* DEEA should be applicable to different industries, but has only been evaluated within the agricultural industry.
- *Other means of scope demarcation:* DEEA applies the system life cycle model of Kossiakoff *et al.* (2011) and only incorporates the concept development stage. In future, DEEA may have to be extended for the other stages as well.
- *Role players:* DEEA should be used by enterprise engineers and their design teams. Although other stakeholders may contribute towards the identification of areas of concerns and requirements, they are not users of DEEA.

Interview Feedback

The ADP implications and measures are sufficient.

4.6 Principle F: Representations of Design Scope

Implications:

- Define notation standards that are used to describe *design domains*.
- Motivate any deviation from existing standards.

Measures: No additional measures.

Principle Application

DEEA applied the following notation standards per design domain:

- Organisation design domain: DEMO's aspect models.
- Information design domain: Entity relationship models.
- Infrastructure design domain: Process flows with concept technologies and wire frames to model concept interfaces and concept technologies.

Interview Feedback

The ADP implications are sufficient.

4.7 Principle G: Approach Form and Function

Implications:

- Define the overall structure and organisation of the approach.
- Define the *mechanisms and practices* explicitly, stating their form (conceptual parts) and function, ensuring that their interpretation is clear.
- Define how the *mechanisms and practices* are related to one another.
- Define the roles involved when using the *mechanisms and practices*.

Measures:

- Are the structures described comprehensively enough to be useful and transferable?
- Did you validate the interpretation of the *mechanisms and practices* (their form and function) with appropriate research participants?
- For each *mechanism* or set of *practices*, is it clear which roles are involved?

Principle Application

- *Structure and organisation of the approach*: DEEA is organised into two main parts, *introduction* and *constructs*. The *introduction* part describes the function, scope, users and prerequisites of DEEA, whereas the *constructs* part presents the mechanisms and practices that form part of DEEA.
- *Mechanisms and practices*: DEEA incorporates a method that consists of three main steps, illustrated in Figure 1: (1) Analyse Needs, (2) Determine Starting Point, and (3) Design a Design Domain. The documented DEEA also presents additional mechanisms, namely a code

book, requirements database, DEMO aspect models, entity relationships models, flow diagrams and wire frames (See Van der Meulen (2017).

- *Relationships of mechanisms and practices*: The relationships of the mechanisms and practices are graphically portrayed in Van der Meulen (2017).
  Roles involved when using the mechanisms and practices: DEEA is not prescriptive on the roles that are involved, since it has not been tested with a dedicated EE team. Yet, DEEA could be further enhanced in future to prescribe roles.

Interview Feedback

- The ADP implications and measures are sufficient. However, more guidance should be provided to ensure that the structure addresses the ADPs.

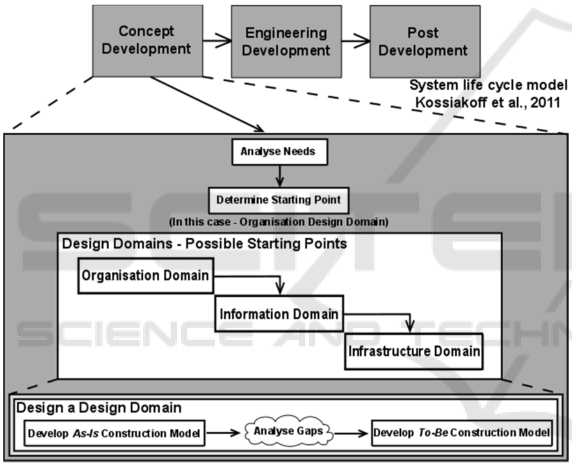
Concept Development
Engineering Development
Post Development
System life cycle model Kossiakoff et al., 2011
Analyse Needs
Determine Starting Point
(In this case - Organisation Design Domain)
Design Domains - Possible Starting Points
Organisation Domain
Information Domain
Infrastructure Domain
Design a Design Domain
Develop *As-Is* Construction Model
Analyse Gaps
Develop *To-Be* Construction Model

Figure 1: Reduced illustration of DEEA.

4.8 Principle H: Justificatory Knowledge

Implications:
Define kernel theories on which the approach and its components are based, and on how they are related to different components of the approach.
Measures: No additional measures.

Principle Application

The components of DEEA are based on existing theoretical works, which are discussed in Van der Meulen (2017). Prominent works include: the system life cycle model of Kossiakof et al. (2011), the generic system development framework of Hoogervorst (2016) and DEMO of Dietz, as presented in Perinforma (2015). DEMO has its roots in the *theory of communicative action* of Habermas (1984), the *speech act theory* of Searle (1969) and *systemic ontology* of Bunge (1979).

Interview feedback

The ADP implications are sufficient.

4.9 Principle I: Approach Mutability

Implications:
Define foreseeable changes — i.e., approach constructs that will change, and the kinds of change that would be required.
Measures:

- Are conditions for changes described?
- Is it clear which parts could possibly change and to what they could change?
- Are possibilities for tailoring the approach defined to enable extension of the approach in future?

Principle Application

DEEA provides various scenarios for enterprise design, as well as the associated method tailoring. In addition, advice is offered regarding DEEA's extension beyond the concept development stage of Kossiakof's life cycle development model (Van der Meulen, 2017).

Interview feedback

The ADP implications are sufficient.

4.10 Principle J: Principles of Implementation (Conditional)

Implications:

- Define tailoring advice.
- Define advice regarding introduction into real-life settings.

Measures:
Does the advice for implementing the approach cover different settings within the scope, or is it at least clear about the scope to which it applies?

Principle Application

DEEA does not provide additional advice on implementation, since the discussion on alternative tailoring scenarios was deemed sufficient.

Interview feedback

The interviewee commented that ADP I and J seem to overlap. An alternative interpretation of ADP J is that an approach *user manual* should be constructed to guide the approach user and improve the usability of the newly-designed approach.

4.11 Principle K: Expository Instantiation (Optional)

Implications:

- Report on a real-life implementation of the approach.

Measures:

- Does the implementation cover a case within the scope, and is it covering the main *mechanisms and practices* of the approach?
- Is the implementation specific enough to be illustrative?

Principle Application

A real-life demonstration of DEEA is presented in Van der Meulen (2017) to design the concept for a to-be post-harvest system. The demonstration covers a case in scope, covers all mechanisms and practices and provides a realistic demonstration of DEEA.

Interview feedback

The ADP implications and measures are sufficient.

5 RESULTS SUMMARY

According to the interviewee, the ADPs were useful in terms of their *implications* and *measures*. Yet, he indicated that 5 of the 11 principles could be further enhanced, namely ADPs A, B, D, G and J. Section 5.1 summarises the interview results, whereas section 5.2 presents opportunities for further enhancement of the ADPs.

5.1 Summary of Interview Results

Principle A: Explicit Concept of the Enterprise

The approach developer indicated that the ADPs do not highlight the necessity for stating prerequisites for using a particular approach.

Principle B: Explicit Phenomenon

The ADP *implications* should focus on the identification of a *problem instance*, prior to its generalisation as a *class-of-problems*.

Principle D: Explicit Means (Ways) of Demarcating and Representing Design Scope

The approach developer initially defined DEEA's design domains, rather than their *means* of demarcation. The ADP should explain the phrase "means of demarcating".

Principle G: Approach Form and Function

More guidance should be provided to ensure that the approach structure addresses the ADPs.

Principle J: Principles of implementation (conditional)

ADP I and J seem to overlap. An alternative interpretation of ADP J is that an approach *user manual* should be constructed to guide the approach user and improve the usability of the newly-designed approach.

5.2 Opportunities for Improvement

Based on the feedback from the approach developer, the following suggestions are made to improve the ADPs.

Principle A: Explicit Concept of the Enterprise

Refer to ADP E and ADP G, which address the approach developer's feedback.

Principle B: Explicit Phenomenon

Implications should focus on the identification of a *problem instance*, prior to its generalisation as a *class-of-problems*. The implications may be adapted as follows:

Implications:

- Produce evidence that a *problem instance* exists at a real enterprise. The problem instance should be an enterprise engineering type of problem.
- Produce *sufficient* evidence that the *problem instance* also features as an existing *phenomenon* or *class-of-problems* in literature, but that it is *inadequately addressed* by existing theory or theory application.

Measures: No additional measures.

Principle D: Explicit Means (Ways) of Demarcating and Representing Design Scope

The ADP should explain the phrase "means of demarcating".

Implications:

- Define the *way to demarcate* design domains. As an example, the *basic systems design process* delineated in Dietz (2006) may be used as the means to define design domains *that have supporting relationships*, e.g. the ICT domain/class-of-systems supports the organisation domain/class-of-systems.
- Define the *way to demarcate* concerns per design domain (e.g., using generic areas of

concerns from theory or using heuristics to identify concerns). As an example, Hoogervorst (2016) provides a *heuristic*, linked to the *generic system development framework*, to elicit enterprise-specific areas of concern and enterprise requirements.

Measures: No additional measures.

Principle E: Well-demarcated and Well-defended Design Scope and Using Scope

The principle description should be extended to include the phrase “and using scope”. In addition, the implications should also be adapted to reflect the extension.

Implications:

- Identify the overall *scope* of the approach — i.e., focusing on *inside-the-boundary* complexities versus *outside-the-boundary* complexities.
- Identify the scope of the approach in terms of *design domains* that will be addressed by the approach.
- Identify the scope of the approach in terms of areas of *concern* for different *stakeholders* that will be addressed.
- Use additional ways of scoping to define the context for which the approach is intended — e.g., a specific industry.
- Indicate the actual *implementation* scope used to demonstrate the approach.
- Identify the role players, especially the main *users* of the approach.
- Specify the using scope by defining *prerequisites*. An example of a prerequisite is that the approach should only be considered for implementation if management has a similar conceptualisation of the enterprise than delineated in the approach.

Measures:

- Is the scope described in such a way as to relate to existing theory that covers a similar design scope?
- Is the scope described in such a way as to know whether an application context falls within the scope of the approach?

Principle G: Approach Form and Function

We suggest that ADP G should mandate inclusion of an approach component that explicitly states *prerequisites* for applying the newly-developed approach, also providing an example of a prerequisite. In addition, more guidance should be provided to ensure that the approach structure addresses the ADPs. The implications and measures may be adapted as follows:

Implications:

- Define the overall structure and organisation of the approach and ensure to address the relevant ADPs.
- A suggested structure is depicted in Table 1, also relating to the ADPs that should be addressed. The principles that are omitted (H and K), should not be incorporated as structural parts of the newly-designed approach.
- The overall structure should incorporate the approach *function* (i.e. *Introduction* part) and *form* (i.e. *Mechanisms and practices* part).

Table 1: Suggested structure for a design approach.

Structural parts	**ADPs**
Introduction	
Objectives and intended value	A, B, C
Scope	D, E, F
Users	E
Prerequisites	E
Mechanisms and practices	G
Approach tailoring	I
User guidance (conditional)	J

- Define the *mechanisms and practices* explicitly, which encapsulates the approach *form*. The *mechanisms and practices* should also be detailed in terms of their *form* (conceptual parts) and *function*, ensuring that their interpretation is clear.
- Ensure that *mechanisms and practices* are sufficient to address the objectives and intended value of the approach.
- Define how the *mechanisms and practices* are related to one another.
- Define the roles involved when using the *mechanisms and practices*.

Measures:

- Are the structures described comprehensively enough to be useful and transferable?
- Did you validate the interpretation of the *mechanisms and practices* (their form and function) with appropriate research participants?
- For each *mechanism* or set of *practices*, is it clear which roles are involved?

Principle J: Principles of implementation (conditional)

We suggest that ADP J highlights additional advice for implementation, rather than providing advice on tailoring the approach. The implications may be adapted as follows:

Implications:

- Define advice regarding introduction into real-life settings.
- Compile a *user manual* for using the newly-designed approach, including examples.

Measures:

Does the advice for implementing the approach cover different settings within the scope, or is it at least clear about the scope to which it applies?

6 DISCUSSIONS AND CONCLUSIONS

Previous work applied design research to develop eleven ADPs which needed additional *demonstration* and *evaluation* (De Vries, 2016). The ADPs were developed to guide the approach developer in explicating the *conditional use* of a newly-developed approach in terms of its *contextual prerequisites* and *demarcated design scope* (De Vries, 2016). The ADPs thus primarily have an academic value to encourage systematic growth of the EE knowledge base.

This article presented a *demonstration* of the ADPs, since they were used during the development of a new approach, called the DEMO-based enterprise engineering approach (DEEA). In addition, we *evaluated* the usefulness of the ADPs, using an in-depth interview to inquire about every ADP, especially how the ADP user interpreted and applied the ADP *implications* and *measures*.

The ADPs currently have an *academic focus*, ensuring that new/enhanced enterprise design approaches are explicated in terms of their demarcated design scope, contextual prerequisites and their conditional use. In terms of the stated *academic focus*, the evaluation feedback was positive and useful to suggest a number of opportunities for extending the current ADPs. However, when an approach developer identifies a phenomenon/problem that may be addressed by developing a new *design approach,* the ADPs alone will not provide sufficient guidance for the development endeavour. Thus, although the approach developer of DEEA applied the ADPs during approach development, he also required additional guidance, e.g. using guidelines from Sein *et al.* (2011) on action design research, to develop DEEA.

Future applications of the *extended ADPs* will further increase the rigour of the ADPs. It is possible that developers of new approaches, such as ambidextrous BPM, will identify valid reasons for adapting the existing ADPs or identifying additional ADPs.

De Vries and Berger (2016) also provide additional guidance on appropriate research methods for enterprise approach design, highlighting *action design research*, whereas Venable *et al.* (2016) provide guidance for *evaluating* design science research.

ACKNOWLEDGEMENTS

The authors would like to thank the management of ZZ2 for the opportunity to demonstrate and use DEEA within the enterprise and their consent for documenting the findings. We would like to convey our gratitude towards Johannes Grobler who supported the project for redesigning the post-harvest system from the start.

REFERENCES

Albani, A., Dietz, J. L. G. 2010. Preface. In Albani, A. and Dietz, J. L. G. (Eds.), *Advances in Enterprise Engineering IV*. Berlin, Heidelberg: Springer.

Blowers, M. 2012. Hybrid enterprise architecture frameworks are in the majority. Retrieved from http://ovum.com/2012/03/22/hybrid-enterprise-architecture-frameworks-are-in-the-majority/

Bunge, M. A. 1979. *Treatise on basic philosophy, Vol. 4: a world of systems*. Dortrecht: Reidel Publishing Company.

De Vries, M. 2016. Guiding the development of enterprise design approaches. *South African Journal of Industrial Engineering, 27*(3), 12-22. Retrieved from http://sajie.journals.ac.za/pub

De Vries, M., Berger, S. 2016. An action design research approach within enterprise engineering. *Systematic Practice and Action Research*. doi:10.1007/s11213-016-9390-7

De Vries, M., Gerber, A., Van der Merwe, A. 2015. The enterprise engineering domain. In Aveiro, D., Pergl, R., and Valenta, M. (Eds.), *Advances in Enterprise Engineering IX* (pp. 47-63). Switzerland: Springer.

De Vries, M., Van der Merwe, A., Gerber, A. 2015. Extending the enterprise evolution contextualisation model. *Enterprise Information Systems, doi: 10.1080/17517575.2015.1090629, http://www.tandfonline.com/doi/pdf/10.1080/17517575.2015.1090629.*

Dietz, J. L. G. 2006. *Enterprise ontology*. Berlin: Springer.

Dietz, J. L. G., Hoogervorst, J. A. P. 2007. Enterprise ontology and enterprise architecture - how to let them evolve into effective complementary notions. *GEAO Journal of Enterprise Architecture, 1.*

Espejo, R., Reyses, A. 2011. *Organisational systems*. Berlin Heidelberg: Springer-Verlag.

Gause, D. C., Weinberg, G. M. 1989. *Exploring requirements: quality before design*. New York: Dorset House Pub.

Geng, S., Ren, T.-z., Wang, M.-h. 2007. Technology and Infrastructure Considerations for E-Commerce in Chinese Agriculture. *Agricultural Sciences in China, 6*(1), 1-10. doi:http://dx.doi.org/10.1016/S1671-2927(07)60010-8

Giachetti, R. E. 2010. *Design of enterprise systems*. Boca Raton: CRC Press.

Gregor, S., Jones, D. 2007. The anatomy of a design theory. *J Assoc Inf Syst, 8*(5), 312-335.

Habermas, J. 1984. *The theory of communicative action: Reason and the realization of society*. Boston: Beacon.

Hevner, A. R., March, S. T., Park, J., Ram, S. 2004. Design science in information systems research. *MIS Quarterly, 28*(1), 75-105. Retrieved from http://wise.vub.ac.be/thesis_info/design_science.pdf

Hoogervorst, J. A. P. 2009. *Enterprise governance and enterprise engineering*. Diemen: Springer.

Hoogervorst, J. A. P. 2016. *On the realization of strategic success: A paradigm shif needed: Enterprise governance and enterprise engineering as essential concepts*. Antwerp: Antwerp Management School.

Iivari, J., Hirschheim, R., Klein, H. K. 2001. A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information systems, 17*(3), 179-218.

Kappelman, L. A. 2010. *The SIM guide to enterprise architecture*. Boca Raton: CRC Press.

Kossiakoff, A., Weet, W. N., Seymour, S., Biemer, S. M. 2011. *Systems engineering principles and practice* (2nd ed.). New Jersey: John Wiley & Sons.

Lloyd, G. E. R. 1968. *Aristotle: the Growth and Structure of his Thought*. Cambridge: Cambridge University Press.

Marshall, C., Rossman, G. B. 2011. *Designing qualitative research* (5^{th} ed.). California: Sage Publications.

Mingers, J., Brocklesby, J. 1997. Multimethodology: towards a framework for mixing methodologies. *Omega, 25*(5), 489-509.

Morgan, G. 2006. *Images of organisations*. Thousand Oaks: Sage Publications.

Nuthall, P. L. 2011. *Farm business management: analysis of farming systems*: CABI.

Parmenter, D. 2010. *Key performance indicators: developing, implementing and using winning KPI's* (2nd ed.). Hoboken: John Wiley & Sons.

Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S. 2008. A design science research methodology for information systems research. *Journal of MIS, 24*(3), 45-77.

Perinforma, A. P. C. 2015. *The essence of organisation 2.0*. Netherlands: Sapio.

Searle, J. R. 1969. *Speech acts, an essay in the philosophy of language*. Cambridge MA: Cambridge University Press.

Sein, M., Henfredsson, O., Purao, S., Rossi, M., Lindgren, R. 2011. Action design research. *Management Information Systems Quarterly, 35*(1), 37-56.

Van der Meulen, T. (2017). *Towards a Useful DEMO-based Enterprise Engineering Methodology, Demonstrated at an Agricultural Enterprise. Unpublished.*, University of Pretoria.

Van Tonder, C. L., Roodt, G. 2008. *Organisation development: theory and practice*. Pretoria: Van Schaik Publishers.

Venable, J., Pries-Heje, J., Baskerville, R. 2016. FEDS: A framework for evaluation in design science research. *European Journal of Information Systems, 25*(1), 77-89. Retrieved from http://www.palgrave-journals.com/ejis/journal/v25/n1/full/ejis201436a.html

Wolfert, J., Verdouw, C. N., Verloop, C. M., Beulens, A. J. M. 2010. Organizing information integration in agri-food—A method based on a service-oriented architecture and living lab approach. *Computers and Electronics in Agriculture, 70*(2), 389-405. doi:http://dx.doi.org/10.1016/j.compag.2009.07.015