Towards consistent demarcation of enterprise design domains

Marné de Vries[1[0000-0002-1715-0430]]

[1] University of Pretoria, Pretoria, South Africa
Marne.devries@up.ac.za

Abstract. This article supports the ideology that enterprise engineering (EE) could add more value if EE researchers focus on facilitating effective conversations within design teams to create a common understanding of the enterprise. One way of creating a common understanding is to define and demarcate enterprise design domains in a consistent way. Literature presents different conceptualisations for demarcating design domains, *without* using a systematic demarcation rationale. As an example, this article introduces Hoogervorst's approach and associated enterprise design domains to highlight practical difficulties when emerging *design principles* are applied to *four main design domains*, as defined by Hoogervorst. Based on the suggestion to apply the *basic system design process* to demarcate the main enterprise design domains in a *consistent way* and addressing the need for *additional* design domains, we present four alternative enterprise design domains, developed via *design science research*. We also demonstrate the usefulness of the new design domains by presenting several examples of enterprise design cycles that occur during enterprise design.

Keywords: Enterprise Design, Design Domains, Enterprise Engineering.

1 Introduction

Establishing design requirements is one of the most important elements in a design process and applied by many engineering disciplines [1; 2]. A more recent application of the design process, is to design the enterprise as an artefact, also termed enterprise engineering (EE) [3]. The enterprise engineer is mostly concerned with a holistic view of an enterprise [4] and the need to ensure enterprise-wide unity and integration [3; 5]. Yet, enterprise design is by no means simple, since enterprises rank amongst the highest in complexity, i.e. level eight on Boulding's [6] nine-level complexity scale. Even though there may be limits to formal enterprise design due to enterprise complexity, Hoogervorst [7] emphasises that the realisation of strategic intensions and successfully addressing areas of concern do not occur incidentally. Although most enterprises emerged in an ad hoc way, rather than by design [8], there is a need to govern enterprise evolution in a more systematic way [5]. Prior to governing its evolution, the enterprise design team needs to define those aspects or design domains that need to be governed. Yet, current enterprise design approaches vary in how they define different enterprise design domains/levels [4] and there is a lack of standardised terms, definitions, semantic rules and concepts to define the design domains [9].

This article explores the suggestion to use the *basic system design process* to demarcate four main enterprise design domains in a more *consistent* and *comprehensive* way. In addition, we *demonstrate the usefulness* of the newly-demarcated design domains by providing examples of several concurrent design cycles that occur during enterprise design.

The structure of the article is now discussed. Section 2 provides background on Hoogervorst's iterative enterprise design approach and associated enterprise design domains, as well as the *basic system design process* that is used as a means to demarcate enterprise design domains in a consistent way. Section 3 presents *design science research* as an appropriate research methodology for developing an *artefact*, namely a new *model* of constructs to represent enterprise design domains. Section 4 presents the *model* of constructs, as well as examples of concurrent design cycles that occur during enterprise design. Section 5 concludes with suggestions for future research.

2 Background

Section 2.1 provides background on the iterative enterprise design heuristic proposed by Hoogervorst [7]. In section 2.2, we present theory on the *basic system design process* that may be useful when demarcating design domains in a *consistent* way.

2.1 Hoogervorst's Enterprise Design Heuristic and Practical Problems

Hoogervorst [5; 7] developed an approach that is iterative, emergent, creative and non-algorithmic. His approach contrasts with big-design-up-front (waterfall) approaches of the past and support the argument that stable requirements within a changing environment is an illusion [10], whereas domain knowledge of participating individuals is also emergent [11]. Hoogervorst's [12] approach supports Lapalme's [13] belief that EE will add more value if EE researchers focus on *effective conversations* within design teams, when they have a *common understanding of the enterprise* and emerging enterprise requirements. His multi-disciplinary inquisitive approach starts with the *strategic context*, defining preliminary *design aspects*, which are translated into *areas of concern* and *requirements*. Next architecture (called *design principles*) are defined per *design domain* by domain specialists to govern enterprise evolution [7]. Conceptualisation of appropriate *design domains* ensure that *design principles* can be applied to particular *design domains*, guiding the (re-)design of enterprise constructs within the particular *design domains* to operationalise key *areas of concern* [7]. Hoogervorst [5] presents *four main design domains*:

1. The *business domain* concerns the enterprise function, "having to do with topics such as products and services, customers and the interaction/relationship with them, the economic model underlying the business, and the relationships with the environment (sales channels, market, competitors, milieu, stakeholders)"[5, p 299].

2. The *organisation domain* is part of enterprise *construction* and “concerns the internal arrangement of the enterprise, having for example to do with processes, employee behaviour, enterprise culture, management/leadership practices, and various structures and systems, such as regarding accounting, purchasing, payment, or employee evaluation” [5, p 300]. Hoogervorst applies Dietz’s [14] work to describe the *essence of the organisation domain* via aspect models. Aspect models are based on the transaction axiom, which states that the essence of enterprise operation consists of *human actor roles* that coordinate their actions around *production acts* to deliver goods and services to customers [14].
3. The *information domain* is also part of enterprise *construction* and consider aspects, such as “the structure and quality of information, the management of information (gathering, storage, distribution), and the utilisation of information [5, p 300].
4. The *technology domain,* also part of enterprise *construction*, is “essential for business, organisational and informational support”. Yet, Hoogervorst only highlights the need for *information technology* guidance [5, p 300].

A previous study already experimented with Hoogervorst’s approach, indicating difficulties during the process of sense-making, when emerging *design principles* had to be applied to Hoogervorst’s demarcated *design domains* [15]. One of the reasons is that possible ambiguity exists between the *information domain* and the *organisation domain*. Hoogervorst’s definition of the *information domain* includes constructs, such *production acts* (e.g. gathering, storage, distribution), that conceptually overlaps with the definition of the *organisation domain*. In addition, the study also highlighted that the existing *technology domain* focused on *information technology* alone, excluding other technologies that are also used to support the *organisation domain* [15]. As an example, a forklift enables/semi-automates the *production act* called *product relocation*. The study suggested that the main enterprise design domains be *redefined* in a more *consistent* way, i.e. based on the *basic system design process* [15].

2.2 The Basic System Design Process

The *basic system design process* is useful when an *object* system has to be designed within the context of a *using* system [14]. The design process has to start with knowledge about the *construction of the using system*, prior to eliciting functional requirements for a supporting *object system* [14]. Furthermore, the design process incorporates two main design activities, namely analysis (*determining requirements*) and design (*devising specifications*) [16; 17]. Even though the *analysis* activities are distinguished from *design* activities, Hoogervorst [7] emphasises that the activities are executed iteratively.

3 Research Method

This study applies *design science research* (DSR) as an appropriate research methodology to develop a *model* of constructs (see Fig. 1) to represent enterprise design domains that are based on a *consistent demarcation rationale*. In accordance with Gregor & Hevner's [18] knowledge contribution framework, the model can be considered as an *improvement*, since a new solution (*model* of constructs for representing enterprise design domains) is developed for solving a known problem. Referring to the DSR steps of Peffers et al. [19], this article focuses on the first four steps of the DSR cycle as follows:

Identify a problem: Sections 2 stated that current enterprise design approaches vary in how they define different enterprise design domains/levels [4] and there is a lack of standardised terms, definitions, semantic rules and concepts to define the design domains [9]. The Open Group [20] for instance defines four design domains (business, application, data and technology), whereas Hoogervorst [5] defines different design domains (business, organisation, information and technology). Also, a recent study that applied Hoogervorst's demarcation of design domains reported difficulties when emerging design principles had to be applied to Hoogervorst's demarcated design domains [15].

Define objectives of solution: As a solution to the general class-of-problems regarding inconsistent demarcation of design domains and *lack of consistent demarcation rationale*, section 2 suggests that the *basic system design process* is used as a means to demarcate the main enterprise design domains in a *consistent* and *comprehensive* way.

Design and develop: Based on the main concepts of the *basic system design process*, section 4 presents a newly-developed artefact, namely a *model* of constructs to present enterprise design domains.

Demonstrate: Section 4 demonstrates the usefulness of the demarcated design domains by presenting several examples of concurrent design cycles performed during enterprise (re-)design.

Although this article only focuses on the *first four steps* of the DSR cycle, we elaborate on future work in Section 5.

4 Four Main Design Domains

In this section, we present the newly-demarcated design domains, using the basic system design process. Furthermore, we introduce different constructs of Fig. 1 and motivate their inclusion to represent four main enterprise design domains.

4.1 Constructs to Represent Design Domains

***Class of Systems* Constructs.** Using the *basic system design process*, we identified an abstract *class of systems* (COSs) such that its *construction* is designed to support a using COSs. As an example, Dietz [16] identifies two enterprise COSs, the *ICT COSs* that is designed to support the *organisation COSs*. In addition, the *organisation COSs* should support the *environment COSs* or market context [16]. Other than the *organisation COSs* and *ICT COSs*, we added an *infrastructure COSs*, since Van der Meulen [15], reasoned that non-ICT technologies (e.g. facilities, utilities, machines and tooling), should also be designed to support the *organisation COSs*. Fig. 1 illustrates

COSs using light-grey-shaded rectangles and grey-shaded arrows labelled *Supports* to indicate supporting relationships between *COSs*. As indicated in Fig. 1, the *ICT COSs* may also directly *support* customers, business partners and suppliers within the environment [7]. Dietz [16] further partitions a COSs when he applies a layered nesting of systems, e.g. the *ICT COSs* may be further classified as two COSs: *software application COSs* supported by a *hardware COSs*. In addition, the *infrastructure COSs* may also incorporate sub-systems, such as *material-handling COSs* and *energy-provisioning COSs* [21]. For simplicity reasons, Fig. 1 does not illustrate further partitioning.

Approach authors differ on whether *products* should be considered as an enterprise design domain. As an example, Bernard [22] and Williams [23] include the products domain, whereas The Open Group [20] excludes products as an enterprise design domain. Although not included as an enterprise construct, we acknowledge that a *product COSs* has a significant influence on enterprise design, since the *organisation COSs* should still support the construction of products that are sold to customers, as illustrated Fig. 1 via a *Supports* arrow. Fig. 1 also indicates that the type of *product* sold to customers within the environment, may also be from the *ICT COSs*, since an enterprise may develop *software applications* as a *product* to their customers.

The *organisation COSs* at the bottom of Fig. 1, decoupled from other constructs to simplify the diagram, represents the *enterprise governance organisation* that needs to support the holistic and coherent design of all enterprise COSs and facets, but also need to be supported by *infrastructure, ICT* and *human skills & know-how.*

***Facets* Constructs.** Hoogervorst [5] believes that some *facets*, such as culture, skills and learning requirements, may not be classifiable as *systems*, since a system is defined as elements that have influencing bonds within a particular composition [14; 24]. Yet, *facets* should still be designed as part of the enterprise and some do adhere to a life cycle, i.e. from identification, concept, requirement, design, implementation, operation and decommissioning [25]. Fig. 1 depicts *facets* as light-grey-shaded cloud shapes. An example of a facet is *human skills & know-how.* An inductive analysis on existing enterprise design approaches [4] indicated that *human skills & know-how* should be designed and grown/developed intentionally.

***Design Cycle* Constructs.** In Fig. 1, the dark-shaded rectangles with incoming and outgoing arrows represent the *basic system design process* that should be followed when designing an *object/provisioning* COSs/facet within the context of its *using* COSs/facet. Broken arrow-lines (labelled *Iterative* in Fig. 1) emphasise the iterative nature of analysis and design activities. The double-directed arrow-line on the right-hand side of Fig. 1 signifies the concurrent identification of *areas of concern/interest* that need to be addressed via enterprise design. Keeping the diagram simple, we did not illustrate all enterprise design cycles, but rather used grey-shaded arrows (labelled *Supports*) to indicate prominent relationships between constructs that would require iterative design cycles.

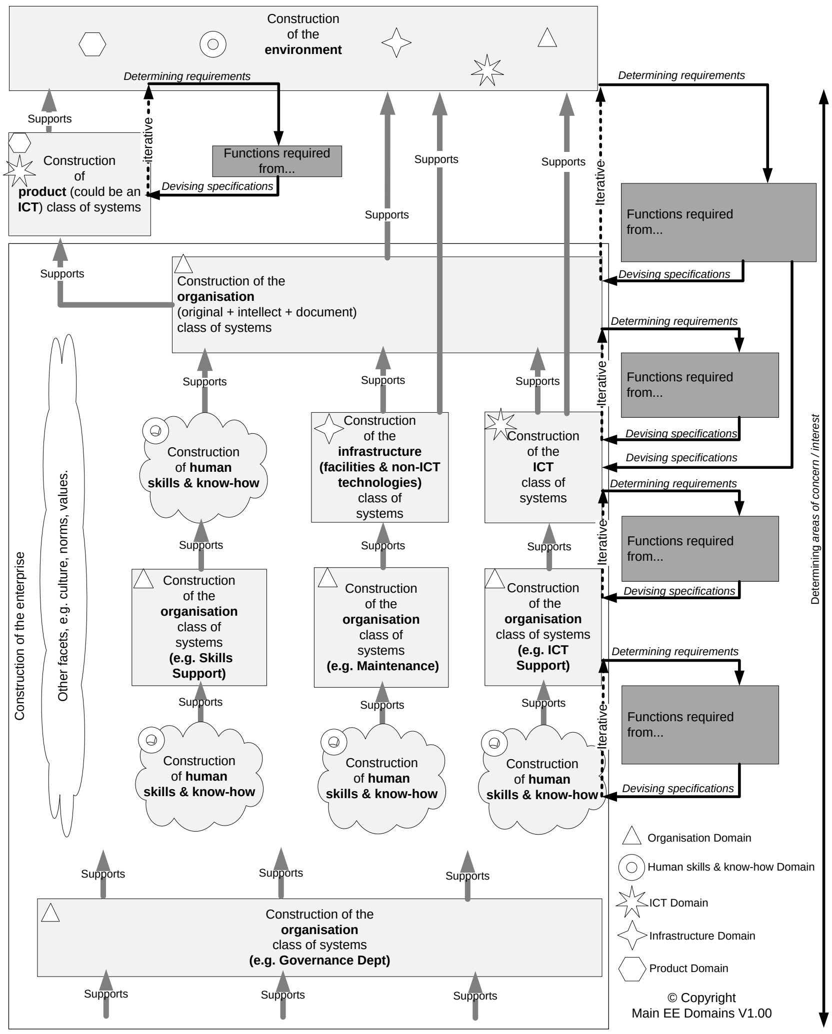

Fig. 1. Model of constructs to represent enterprise design domains

***Design Domain* Constructs.** We already acknowledged different *COSs* and *facets* that need to be designed due to their support/provisioning relationships. Fig. 1 also illustrates that COSs or facets have multiple supporting relationships. Although we duplicated some COSs to highlight different supportive relationships, small icons on Fig. 1 represents similar COSs/facets, i.e. *design domains*. The triangle icon represents the *organisation design domain*, the donut represents the *human skills & know-how design domain*, the 4-point star represents the *infrastructure design domain*, the 7-point star represents the *ICT design domain* and the hexagon represents the *product design domain*. As indicated before, the *product design domain* is currently not considered to be an *enterprise design domain*.

4.2 Domain-related Design Cycles

Organisation Domain. Dietz [14] defines the *organisation COSs* as social systems, i.e. *actor roles*, implemented by human beings, form relationships due to their interactions and communications when they perform *production acts*. Dietz [14] suggests *aspect models* that represent the *essence of enterprise operation* in a coherent, comprehensive, consistent and concise way.

Following the *basic design process*, *organisation COSs* have to be designed within the context of its using COSs. According to Fig. 1, the *organisation COSs* shown as the top-most construct within the rectangle labelled *Construction of the enterprise,* can be designed within the context of different using COSs, i.e. (1) the *environment*, which encapsulates multiple COSs, and (2) *product COSs*. Fig. 1 indicates that the *organisation COSs* also feature in support of other COSs, e.g. supporting the (1) construction of *human skills & know how*, (2) the *infrastructure COSs*, and (3) the *ICT COSs*.

When we consider, as an example, the design cycle that starts with the *construction of the environment* in Fig. 1, the enterprise design team considers the *construction of the environment*, i.e. *possible* products/services, other enterprises (markets, suppliers, partners, competitors, government institutions), citizens, channels, legislation, infrastructure, ICT and possible revenue to *determine requirements* and *functions* of the enterprise, which could be summarised in an *identify statement*. The identify statement provides meaning or sense-giving to enterprise stakeholders and may change over time [26]. An example of an identity statement for a department at a tertiary education institution is: *Encouraging blended learning, the engineering department offers tertiary education within the discipline of engineering, as well as quality research outputs*. Ostewalder [27] suggests that feasible *functions* of an enterprise can be specified in the form of a *business model canvass* in terms of a value proposition, key partners, key activities, key resources, customer relationships, customer segments, channels, cost structure and revenue streams. Based on the identified *functions*, the enterprise design team needs to *devise specifications* for *constructing the provisioning organisation COSs*, i.e. identifying and organising *actor roles* to perform appropriate *production acts*. During enterprise implementation, *production acts* and associated *actor roles* are usually grouped into departments, such as infrastructure, human resources, technology development, procurement, inbound logistics, operations, outbound logistics, marketing & sales and customer service [28].

Strategies for organising *production acts* may be influenced by several broad *areas of concern/interest* and even incidental concerns, which requires human sense-making and re-structuring to deal with the concerns [26]. Typical *areas of concern/interest* include profit, process excellence, customer orientation, and employee involvement.

ICT Domain. The ICT domain incorporates software applications, databases and ICT hardware [14]. Different representations are used to communicate ICT designs, such as *unified modelling language* (UML) models [29] and *wire-framing* models [30].

According to Fig. 1, an ICT system can be designed within the context of different using systems, i.e. (1) the construction of the *organisation COSs* and, (2) the construction of the *environment*.

When we consider, as an example, the design cycle that starts with the *construction of the organisation COSs* in Fig. 1, the enterprise design team considers the *construction of the organisation COSs,* i.e. the existing *actor roles* and their *production acts*, to *determine requirements* and *functions* of supporting *ICT COSs*. Based on the identified *functions*, the enterprise design team needs to *devise specifications* for the *construction of the provisioning ICT COSs*, i.e. providing *software applications, databases and hardware.*

Other than *functional concerns* to support the *construction of the using COSs*, typical *areas of concern*/*interest* for ICT COSs include interoperability, scalability, security and user friendliness.

Infrastructure Domain. Infrastructure entails facilities and other non-ICT technologies that support *actor roles* and their *production acts*. Enterprises within different industries may require different *representations* of infrastructure, based on the type of *production acts* that should be supported. The educational industry, for instance, may apply web-based *3D interactive campus models* to visualize learning facilities.

According to Fig. 1, the infrastructure COSs can be designed within the context of different using COSs, i.e. (1) the *construction of the organisation COSs* and (2) the *construction of the environment*.

When we consider, as an example, the design cycle that starts with the *construction of the organisation COSs* in Fig. 1, the enterprise design team considers the *construction of the organisation COSs,* i.e. the existing *production acts*, to *determine requirements* and *functions* of *provisioning infrastructure COSs*. Yet, according to Tompkins et al. [21], many other *organisation implementation decisions* also affect the provisioning infrastructure, such as packaging, service levels for spares, and delivery times. Based on the identified *functions*, the enterprise design team needs to *devise specifications* for the *construction of the provisioning infrastructure COSs*, i.e. providing *facilities, such as offices, factories and warehouses.*

Other than *functional concerns* to support the construction of the using COSs, typical *areas of concern/interest* for infrastructure include space utilisation, flexibility, upgradability, environmental friendliness, reliability, security, noise levels, vibrations, lighting, air quality and work space.

Human Skills & Know-how Domain. Human skills & know-how constitutes human *abilities and skills* required when executing *production acts*, as well as *coordination acts*. As indicated in Fig. 1, *human skills and know-how* cannot be defined as a *class of systems*, since there are no interactive parts, but need to be developed in support of the *organisation COSs*. Skills and know-how are often represented in *curricula vitae*.

When we consider, as an example, the first cycle that starts with the *construction of the organisation COSs* in Fig. 1, the enterprise design team considers the *construction of the organisation COSs*, i.e. the existing *production acts*, to *determine requirements* and *functions* of provisioning *human skills & know-how*. Based on the identified *functions*, the enterprise design team needs to *devise specifications* for required contextual knowledge, experience, skills and working styles (e.g. perseverance, stress resistance and self-control) to perform coordination acts and production acts.

Other than the *functional concerns* to support the construction of the using COSs, typical *areas of concern/interest* that should also be incorporated during design include *dynamic expansion of relevant knowledge.* Enterprises should not only encourage expansion of *skills and know-how* via formal training programmes, but also encourage facilitation of invisible learning environments for lifelong learning [31].

5 Conclusions and Future Research

Current enterprise design approaches vary in how they define different enterprise design domains/levels [4] and there is a lack of standardised terms, definitions, semantic rules and concepts to define the design domains [9]. Although the existing EE body of knowledge is mostly encapsulated in enterprise design approaches, many of the existing approaches do not provide a consistent demarcation rationale for their associated design domains, which impairs approach comparison [4].

Based on the ideology that EE could add more value if enterprise design teams create a common understanding of the enterprise [13; 32], the suggestion to apply the *basic system design process* to demarcate enterprise design domains in a *consistent way* and expanding the enterprise design scope by adding appropriate design domains [15], we developed a new *model* of constructs to represent enterprise design domains. In addition, we demonstrated the usefulness of the design domains by presenting several examples of enterprise design cycles that incorporate the design of object/provisioning COSs within the context of a using COSs.

Since this article only focused on the first four steps of the DSR cycle, future work is required to further demonstrate and evaluate the four main enterprise design domains. We suggest that a study, similar to [15], should be performed to apply the newly-demarcated design domains in association with the approach of Hoogervorst. Future research also needs to evaluate whether the newly-demarcated design domains are useful when combined with *other* enterprise design approaches.

References

1 Dym, C. L., Little, P.: Engineering design 3rd edn. John Wiley & Sons, New York (2009).
2 Eggert, R. J.: Engineering design 2nd edn. High Peak Press, Idaho (2010).
3 Dietz, J. L. G., Hoogervorst, J. A. P., Albani, A., Aveiro, D., Babkin, E., Barjis, J., Caetano, A., Huyments, P., Iijima, J., Van Kervel, S. J. H., Mulder, H., Op't Land, M., Proper, H. A., Sanz, J., Terlouw, L., Tribolet, J., Verelst, J., Winter, R.: The discipline of enterprise engineering. International Journal of Organisation Design and Engineering, 3(1), 86-114 (2013).
4 De Vries, M., Van der Merwe, A., Gerber, A.: Extending the enterprise evolution contextualisation model. Enterprise Information Systems, 11(6), 787-827 (2017).
5 Hoogervorst, J. A. P.: Enterprise governance and enterprise engineering. Springer, Diemen (2009).
6 Boulding, K. E.: General systems theory: the skeleton of science. Management Science, 2, 197-207 (1956).
7 Hoogervorst, J. A. P.: Applying foundational insights to enterprise governance and enterprise engineering. Springer, Berlin-Heidelberg (To be published).

8 Giachetti, R. E.: Design of enterprise systems. CRC Press, Boca Raton (2010).
9 Von Rosing, M., Laurier, W.: An introduction to the business ontology. International Journal of Conceptual Structures and Smart Applications, 3(1), 20-41 (2015).
10 Proper, H. A., Lankhorst, M. M.: Enterprise architecture: towards essential sensemaking. Enterprise Modelling and Information Systems Architectures, 9(1), 5-21 (2014).
11 Markus, M. L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Q.uarterly, 26(3), 179-212 (2002).
12 Hoogervorst, J. A. P.: The imperative of employee-centric organising and its implications for enterprise engineering. Journal of Organizational Design and Enterprise Engineering(January), 1-16 (2017).
13 Lapalme, J.: Organizations (and organizing) are a technology that humans know very little about. Organizational Design and Enterprise Engineering, 1(1), 27-31 (2017).
14 Dietz, J. L. G.: Enterprise ontology. Springer, Berlin (2006).
15 Van der Meulen, T. (2017). Towards a useful DEMO-based enterprise engineering methodology, demonstrated at an agricultural enterprise. Unpublished., University of Pretoria.
16 Dietz, J. L. G., Albani, A.: Basic notions regarding business processes and supporting information systems. Requirements Engineering, 10, 175-183 (2005).
17 Dietz, J. L. G., Hoogervorst, J. A. P.: Enterprise ontology and enterprise architecture - how to let them evolve into effective complementary notions. GEAO Journal of Enterprise Architecture, 1, (2007).
18 Gregor, S., Hevner, A.: Positioning and presenting design science research for maximum impact. MIS Quarterly, 37(2), 337-355 (2013).
19 Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. Journal of MIS, 24(3), 45-77 (2008).
20 The Open Group: TOGAF 9.1, http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html, last accessed 2017/06/17.
21 Tompkins, J. A., White, J. A., Bozer, Y. A., Tanchoco, J. M. A.: Facilities planning 4 edn. John Wiley & Sons,, USA (2010).
22 Bernard, S. A.: An Introduction to Enterprise Architecture EA3 2nd edn. Authorhouse, Bloomington (2005).
23 Williams, T. J.: A reference model for computer integrated manufacturing (CIM). Purdue Research Foundation, West Lafayette (1991).
24 Jackson, M. C.: Systems thinking. Wiley, Chichester (2003).
25 Bernus, P., Nemes, L., Schmidt, G.: Handbook on enterprise architecture. Springer, Berlin (2003).
26 Weick, K. E., Sutcliffe, K. M., Obstfeld, D.: Organising and the process of sensemaking. Organisation science, 16(4), 409-421 (2005).
27 Ostewalder, A., Pigneur, Y., Clark, T.: Business model generation 1st edn. Wiley, Hoboken, NY (2010).
28 Porter, M. E.: Competitive advantage: creating and sustaining superior performance. The Free Press, New York (2004).
29 Theuerkorn, F.: Lightweight enterprise architectures. Auerbach Publications, New York (2005).
30 Garrett, J. J.: The elements of user experience: user-centered design for the web and beyond 2nd edn. New Riders Press, Berkeley (2011).
31 Marsick, V. J., Watkins, K. E.: Informal and indidental learning. New directions for adult and continuing education, 89, 25-34 (2001).
32 Hoogervorst, J. A. P.: Foundational insights for enterprise governance and enterprise engineering - presenting the employee-centric theory of organisation. Springer, Berlin-Heidelberg (To be published).