An enterprise-engineering based approach to develop enterprise capacity

Andreas de Boer[1][0000-0001-5265-6931], Marné De Vries[2] [0000-0002-1715-0430]

[1, 2] University of Pretoria, Department of Industrial and Systems Engineering, South Africa
andreas@adbconsult.co.za; Marne.DeVries@up.ac.za

Abstract. Significant progress has been made in the South-African early childhood and Grade R spheres. However, South-Africa has a long way to go to meet the needs of majority of its children. Institutional capacity (IC) refers to the administrative and managerial aspects of an Early Childhood Development Centre (ECDC). Failure to build this capacity impacts the quality of services delivered to the most vulnerable children in our society. The purpose of this article is to extract knowledge from the institutional capacity knowledge area, as well as the enterprise engineering body-of-knowledge as a baseline for developing an enterprise capacity development approach (ECDA) for early childhood development centers. The ECDA should be useful to South-African ECD administrators, if they intend to develop enterprise capacity, leading to the improvement in quality of services delivered. ECDA is the main contribution of this article. As a second contribution, we validate its completeness when compared to existing IC literature, as well as its comprehensiveness in terms of eleven approach design principles. Finally, we provide a partial demonstration of ECDA's heuristic at a real-world ECDC in South Africa.

Keywords: Enterprise engineering; Institutional capacity; Early childhood development; Quality; Education service; Public sector.

1. Introduction

The number of working parents, including single-parent families and families with both parents employed is rising, creating an ever-growing need for quality child care, according to Experthub [1]. The department of social development, department of basic education, and department of health, all have the mandate to develop an integrated approach for services for children aged from birth, up to, but not including Grade R, formally classified as the early childhood phase. In South-Africa, Atmore et al. [2] classify early childhood development (ECD) centers in three distinct facility types, namely (1) public schools, (2) registered community-based ECDCs, and (3) unregistered community-based ECDCs. Public schools typically cater for grade R only, whilst community-based centers accommodate children from birth, up to and including grade R. There are approximately 25,254 centers nationally and 1,354,274 children access these centers [3]. The ECD goal is to provide developmentally-stage-appropriate quality ECD services to all infants, young children and their caregivers by 2030 [3].

Significant progress has been made in the South-African early childhood and Grade R spheres. However, South-Africa has a long way to go to meet the needs of the majority of its children [2]. Various challenges exist within the early childhood sector, including infrastructure availability, nutrition, various different ECD curricula, ECD teacher skill level, administrative and management function (also called institutional capacity), and limited government funding [2]. Of particular interest is institutional capacity, and the inability of ECDCs to execute its purpose effectively. Imbaruddin [4] defines institution as enterprise and capacity as the ability of an enterprise to pursue its objectives, indicating that intuitional capacity (IC) is a prerequisite for delivering *quality services,* as assessed by customers. IC is not a new concept in the public sector performance arena, but not well defined or researched in the ECD sector [5]. Aligned with literature, in this article, IC refers to the *administrative and management function* that is currently lacking at many ECDCs and should be realized via multiple design domains. Yet, the management function is a function that relates to multiple ECDC functions.

The provision of early childhood development services is deemed a right to all children. In addition, ECD services must be provided because they afford a foundation for good child outcomes as well as national developmental outcomes necessary to address South Africa's two key development challenges namely poverty and inequality [6]. The department of social development [6] states that the public provisioning of early childhood development thus embraces a continuum of responsibilities, and its objectives are to ensure:

- All services necessary for the optimal survival, growth, development and protection of infants and young children to their full potential are available in sufficient quantities and through a sufficient number of appropriate spaces in sufficiently close proximity so that all children have an equal opportunity to participate in or make use of the ECD services.
- All services that are provided are of a sufficiently high quality and are age- and stage-appropriate to the needs and context of the children in question to ensure universal quality outcomes for all children receiving the service.
- Early childhood development programs are appropriately designed to ensure the delivery of quality, age and stage-appropriate, and inclusive services.
- The environment, infrastructure (including ICT), and materials supporting the delivery of all early childhood development services are safe, healthy and enable the delivery of age-appropriate quality services.
- There are a sufficient number of appropriately qualified practitioners to provide age-appropriate, inclusive plus quality early childhood development services.
- Measures are implemented to ensure that the cost of the services do not preclude children living in poverty and the qualifications and number of early childhood development practitioners, including the design of the programs and services, address the needs of children with disabilities.
- Appropriate management, coordination, monitoring and evaluation systems are in place to adequately plan for, measure, monitor and improve availability, quality and equity of ECD access.

Literature indicates that the ECD environment is complex, and there is a need for fresh new thinking to evolve ECDCs. A case is made to act and think more systemically to

deliver higher quality programs and ECDC services that are sustainable over a longer period of time. IC has an impact on the quality of the ECDC, and the need is greater than ever to scale up an approach to meet increased demand of early childhood care in South-Africa. In a study conducted by Hayden [7], specific focus was placed on addressing the gap in understanding the director's role, administration, and management functions in childcare and preschool settings. It is becoming increasingly clearer that process components which make up the adult work environment, have a powerful effect upon quality care in child care centers, and that the center director plays a central role.

Atmore [8] and Van Heerden [9] indicate that community-based ECDCs lack IC, i.e. proper administrative and management systems, to meet the minimum standards set by the department of social development. Financial management of many of the community-based ECD facilities is poor, whereas more than 50% of these centers do not have many of the necessary administrative documents and structures in place [8]. The department of social development commissioned a national audit of registered ECDCs in 2013, and the scope included conditional and unregistered centers outlined in their report [10]. ECDCs across South-Africa are sub optimal, indicating that less than half of all registered centers having nothing more than staff attendance records or job descriptions [10]. This class-of-problems has already been validated via a systematic literature review (SLR) in [11].

The SLR also indicated that numerous ECD quality frameworks and IC development frameworks exist, each focusing on different performance areas and functions. We also indicated that existing frameworks, contain a mix of concerns, functions and structural/design aspects in a disparate way. Enterprise engineering (EE) could be useful to re-structure existing concepts in a consistent and useful way.

Enterprise engineering (EE) emerged as a new discipline to encourage comprehensive and consistent enterprise design [12]. Since EE is multidisciplinary, various researchers study enterprises from different perspectives, which resulted in a plethora of applicable literature and terminology, but without shared meaning [13]. The enterprise evolution contextualization model (EECM), is a metamodel for existing enterprise design approaches. Since EECM is descriptive, rather than prescriptive, approach design principles (ADPs) provide prescriptive guidance on developing new enterprise design approaches.

This article applies existing approaches, i.e. Hoogervorst's approach and existing IC approaches, guided by ADPs, to develop an enterprise capacity development approach (ECDA), assisting ECD Directors to develop EC. ECDA will enable development of all ECDA functions, but also focus on a similar scope than IC, namely the *administrative and management* function. Addressing the lack of applying IC and EE within the South African ECD sector, this study synthesizes existing knowledge by developing the ECDA.

The article is structured as follows. Section 2 presents action design research as an appropriate research methodology to develop the new artefact, called ECDA. As a theoretical foundation for developing ECDA, Section 3 provides background on method engineering, Hoogervorst's enterprise engineering approach, and general principles to develop a new approach. Section 4 presents the constructional components of ECDA, validating its comprehensiveness in section 5, and demonstrating ECDA's heuristic in section 6. Section 7 concludes on ideas for future research.

2. Research Method

Action design research (ADR) combines the strengths of two existing research paradigms, i.e. design science research and action research, facilitating an iterative process for designing and evaluating an artefact within a real-world setting [14], also suggested as an appropriate research methodology for EE-related studies [15]. We now define the four main stages of ADR and indicate their application within the context of this study:

Problem formulation: ADR studies are initiated by a practice-inspired problem and serves as an inspiration for research efforts [14]. The problem formulation, presented in section 1, indicates that community-based ECDCs lack IC, i.e. proper administrative and management systems, to meet the minimum standards set by the department of social development. This class-of-problems, validated using a systematic literature review, also features as a problem instance at a South African ECDC [11].

Building, intervention, evaluation: This stage consists of recursive cycles of building, intervention and evaluation, since an increased understanding of the organizational context also influences the selection and design of the artefact [14]. Evaluation is not separate from building. Rather, decisions about designing and reshaping the artefact and intervening in organizational work practices should be interwoven with ongoing evaluation [14]. Using a participative approach, also extracting from existing theory, i.e. method engineering, approach design principles, Hoogervorst's enterprise engineering approach, and IC approaches (presented in section 3), we *built* a new approach, namely ECDA (presented in section 4). Section 5 indicates how ECDA incorporated existing IC approaches within ECDA's heuristic. For *intervention*, we demonstrate ECDA's heuristic at a real-world ECDC, presented in section 6. As indicated in section 7, future work will further *evaluate* whether ECDA's iterative application will also improve the quality of services as ECDCs.

Reflection and learning: This stage runs parallel to the first two stages. The learning from conceptualizing a solution for a particular problem instance, is used to address a broader class-of-problems [14]. For this study, learning from the first two stages contributed towards the packaging of ECDA for addressing a class-of-problems, rather than only a single problem instance.

Formalizing of learning: The purpose of this stage is to generalize outcomes that would address a wider class-of-problems [14]. In section 7 we provide ideas for future research to further evaluate and validate the ECDA for its use within other industries.

3. Background Theory

Method engineering and situational method engineering focus on formalizing the use of methods for systems development ([16]). Even though method engineering and situational method engineering focus on formalizing methods for systems development, we believe that their concepts are also applicable to the development of an enterprise approach, and more specifically, ECDA. Formal techniques have been incorporated into situational method engineering, in particular metamodeling approaches at various scales, from full method to single fragment descriptions [16]. Multiple dimensions of

modelling exist, in particular models can be stacked in terms of their abstraction level. Metamodels provide the means of defining the rules at a higher level of abstraction, and this in essence acts as an introduction to the abstraction levels adopted in this particular study. The General Conceptual Modelling Framework, provided by Dietz and Mulder [17], assists in explaining how the ECDA was constructed.

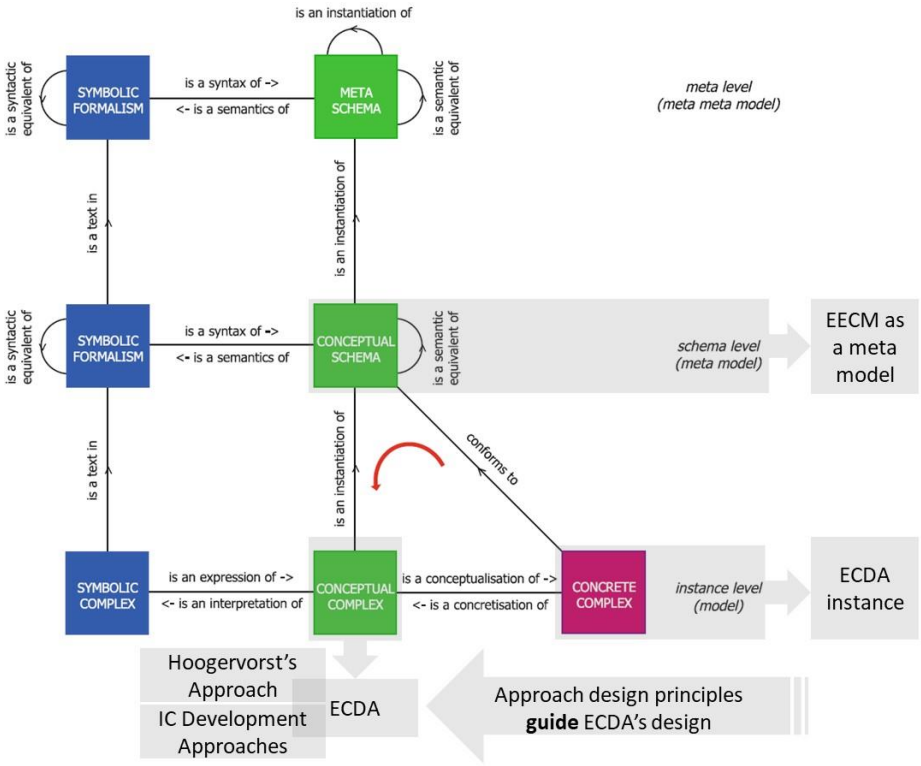


Fig. 1. The General Conceptual Modelling Framework of [17], applied to ECDA

With reference to Fig. 1, we believe that the enterprise evolution contextualisation model (EECM) is a meta model for enterprise design approaches, since EECM was developed inductively as elaborated in section 3.1. Both Hoogervorst's approach, discussed in in section 3.2, and IC development approaches, introduced in section 3.3, are *instantiations* of EECM. Using EECM as a common frame of reference, we developed ECDA, enriched by Hoogervorst's approach as well as existing IC development approaches. Since EECM is not prescriptive on developing a new enterprise design approach, we use approach design principles to provide additional guidance during ECDA's design. ECDA has to be implemented at a real-world ECDC as an ECDA instance, as indicated in Fig. 1. In section 6, we partially demonstrate ECDA's heuristic at a real-world ECDC.

3.1. EECM and Approach Design Principles

Since EE is multidisciplinary, various researchers study enterprises from different perspectives, which resulted in a plethora of applicable literature and terminology, but without shared meaning [13]. Addressing the knowledge fragmentation, the enterprise evolution contextualization model (EECM) inductively abstracted knowledge from multiple existing enterprise design approaches to create a metamodel for enterprise design approaches [13]. EECM indicates that existing enterprise design approaches consist of four main components: (1) Concept of the enterprise and paradigm of creating value; (2) Three dimensions to define the scope of evolution (design domains; concerns & constraints; and enterprise scope); (3) Supporting mechanisms & practices to enable the desired evolution across three dimensions; and (4) Approach classifiers that influence selection of appropriate mechanisms and practices [13].

Since EECM is descriptive and not prescriptive, eleven approach design principles, developed by [18] and based on EECM, provide additional guidance for approach development. Each principle is defined in terms of a statement, rationale, implications and measures. Section 5 provides a summary of the eleven principles and their application regarding ECDA.

3.2. Hoogervorst's Approach

Hoogervorst [19; 20] developed an approach that is iterative, emergent, creative and non-algorithmic. He uses a *generic system development framework* to explain his iterative approach, starting with the strategic context, defining preliminary design aspects that indicate areas of concern and requirements that need to be addressed in system design domains. His approach emphasizes the importance of design principles (also called architecture) that should be explicitly defined to guide the evolution of system design domains [20]. The design concepts included in Hoogervorst's approach are also further refined via a codebook, presented in [21].

The generic system development process (GSDP), a kernel theory, adapted from Dietz [22], facilitates design of different systems within the enterprise [20]. According to this theory, the using system (e.g. the environment) has a functional relationship with the provisioning system (e.g. the enterprise). For this particular combination of using system and provisioning system, the GSDP consists of two main phases, indicated in Fig. 2: (1) Starting with the *constructional design* of the *using system*, the designer has to perform *functional design* of the *provisioning system*; and (2) Starting with identified *functions* of the *provisioning system*, the designer has to perform *constructional design* of the *provisioning system*.

Enterprise design is complex, since multiple systems and sub-systems need to be designed, often concurrently, especially when enterprise constructs already exist. As indicated by Hoogervorst [20], enterprise designers need to define appropriate design domains for a particular enterprise. Using the GSDP as a means to demarcate design domains, De Vries [23] identifies four main design domains for enterprise design: (1) organization, (2) information, communication and technology (ICT), (3) infrastructure/facilities, and (4) human skills & know-how.

3.3. IC Development Approaches

A systematic literature review on existing IC approaches indicated that multiple frameworks already exist, each highlighting different design facets, multiple design levels and performance factors [11].

Bloom [24] describes an ECDC as a dynamic and an open social system, indicating that a systems approach for describing early childhood centers can lead towards a better understanding of the impact of change and can assist administrators to better understand the significance of their day-to-day roles and responsibilities. Bloom includes six components to enable effective ECDC design: (1) environment; (2) people; (3) structure; (4) processes; (5) culture; and (6) outcomes. Imbaruddin [4] (extended by [25]) identified three levels of capacity development: (1) the system, (2) the entity, and (3) the individual. Bergin-Seers and Breen [26] aim to close a gap in research, specific to the performance of ECDCs from a viability perspective, suggesting a performance framework that includes environmental factors, center performance, organizational factors and the leader/manager role within the center. The five dimensional framework by Grindle and Hilderbrand [27] includes a systemic method to analyze determinants of administrative competence, consisting broadly of the action environment, public sector institutional context, task network dimension, organizational dimension, and lastly human resources. Scheepers [28] describes the elements and sub-elements of IC as follows: (1) strategic leadership, (2) human resources, (3) financial management, (4) infrastructure, (5) program management, (6) process management, and (7) inter-institutional linkages. The increasing recognition of the importance of the administrator within an ECDC calls for research about this role and about the characteristics of those who assume it. The leadership framework by Nupponen [29] focuses on: (1) relational and pedagogical leadership, (2) intra and interpersonal skills, and lastly (3) education and training to master these traits.

4. Construction of ECDA

Consolidating from Hoogervorst's approach, as well as existing IC development approaches, this section presents the function (in section 4.1) and form (in section 4.2) of ECDA, following guidance from the *approach design principles* presented in section 3.1. Section 5 indicates how we validate ECDA's comprehensiveness by: (1) validating ECDA against eleven approach design principles, and (2) demonstrating how ECDA has synthesised appropriate literature to create a theory-ingrained artefact.

4.1. ECDA Introduction (function)

ECDA adopts the morphogenic enterprise paradigm from Hoogervorst [20] to address the three essential concepts that fuel and determine enterprise developments: human agency (especially employee agency), reflexivity, and reciprocity. In addition, ECDA also acknowledges that the ECDC is a social system, in accordance with Bloom [24].

Objectives and Intended Value

According to EECM [13], an enterprise design approach has to answer three questions. ECDA answers the three questions as follows:

Why should the ECDC use the ECDA to evolve? ECDA should provide constructional guidance towards the evolution of South African ECDC's, improving the *administrative and management function* associated with *enterprise functions* to increase quality of service delivery. ECDA is comprehensive for the early childhood development context, since it synthesizes knowledge from existing IC approaches, as well as an existing EE approach, i.e. Hoogervorst's approach.

What should the ECDC evolve? ECDA focuses on developing *inside-the-boundary* complexities of an ECDC (as the provisioning system) for the environment (as the using system). Four main design domains are included: (1) organisation; (2) information, communication and technology (ICT); (3) Infrastructure (i.e. facilities); and (4) human skills & know-how. ECDA also acknowledges the existence of *other facets* that evolve at an enterprise, but cannot be designed via a system development process.

How should the ECDC evolve? The ECDC will evolve by applying the ECDA, implementing a key mechanism, namely a *heuristic*.

Scope

Hoogervorst [20] denotes the system to be designed as the *provisioning system*. The *provisioning system* has a functional relationship with its environment, also called the *using system*. Using the *generic system development process* (GSDP), ECDA facilitates constructional design of the provisioning system, as indicated by Fig. 2.

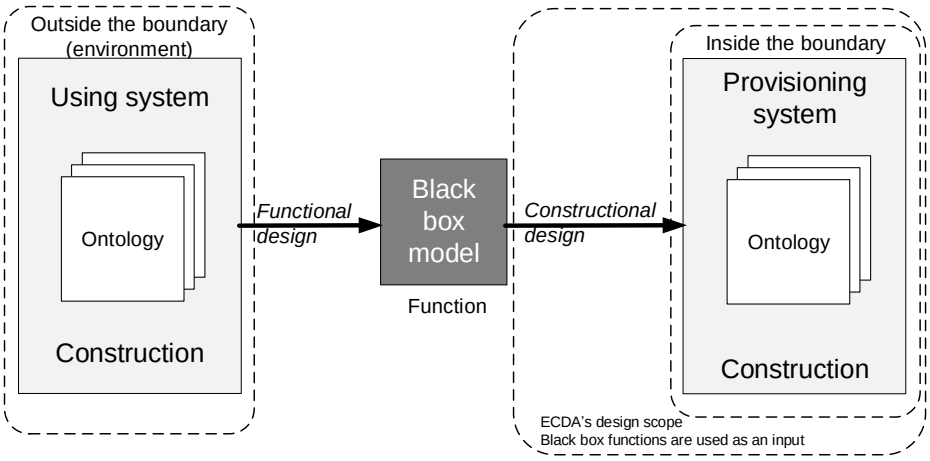


Fig. 2. The ECDA design scope, based on Hoogervorst [20, p 256]

In order to properly define the functional relationship, the wants and needs (*functions*) of the *using system* must be precisely known. In addition, the construction (white-box model) of the using system (i.e. the environment) must be known. Hoogervorst [20] defines *function* as a relationship (R) and not a system property, whereas construction is a system property.

Using [20], we provide two examples within the ECDC context:

F (function): Using system (need, purpose) R Provisioning system (properties)
F (child caregiving): Child (need, purpose) R ECDC (properties)
F (child caregiving): Parent (need, purpose) R ECDC (properties)

Child caregiving as the *function* in the two examples, refers to the operating function of child-caregiving, as well as its management, that should be enabled through various design domains within the ECDC, delivering on various stakeholders' needs and purposes.

Role Players and Users

ECDA will be useful to enterprise engineers as well as design teams. ECDC administrators will find ECDA useful to develop IC in order to improve quality of services delivered. The main user of ECDA will be the early childhood development director/administrator, typically the role accountable for quality of the ECDC operation.

Prerequisites for Using ECDA

The following are identified as prerequisites for using the ECDA: (1) Existence of a problem/deficiency related to the *administration and management* of one or more functions at the ECDC; (2) Need and desire for change clearly established; (3) Buy-in from the director; (4) ECDC functions have already been determined from the environmental (using system) context; and (5) the director is aligned with ECDA's concept of the enterprise, i.e. the morphogenic paradigm defined by Hoogervorst [20] and the social system paradigm presented by Bloom [24].

4.2. ECDA Mechanisms and Practices (form)

The ECDA adopts a *heuristic* indicated in Fig. 3, using multiple enterprise functions (f_1, $f_2 \ldots f_n$) as main input to perform four main activities via multiple cycles.

A *function* is defined as the utility or capability that must be addressed via enterprise design. Conversely, the enterprise, its design domains or constructs, must operationalize one or more functions [21]. The function should be specified using an adjective(s) + noun, also associating the function with the *entire enterprise* or a *particular design domain or construct*, indicating how an input should be transformed into an output [21].

Next, we present the *heuristic's* four activities in more detail. We used alphabetic letters as a quick reference to the activities, but the alphabetic sequence is not an indication of execution sequence. Once enterprise functions have been identified, the heuristic may either start with A or B. As indicated in Fig. 3, representations of constructional design exist for *current designs* as well as *future designs*. Activities that are associated with *future design* are grey-shaded.

A: Execute construction design cycles for selected design domains

Design domains are those aspects of an enterprise that approach authors deem important/necessary for design [13]. De Vries [23] suggests that design domains are demarcated in a consistent way, using the generic system development process. ECDA

adopts the design domains as described by De Vries [23] as a means to represent an ECDC's constructional design.

As discussed in [23], design domains cannot all be classified as systems. Using the definition provided by [17], a homogeneous system consists of elements that are of a similar kind. When human beings are considered to be social elements, then an organization system's construction can be defined by its kernel elements, boundary elements, environmental elements and structural bonds between elements [17]. Depending on the analyst's purposeful demarcation of a system boundary [30], an organization system thus includes human beings as kernel elements (within the boundary), boundary elements (on the boundary) and environmental elements (outside the boundary). Humans that form part of an organization's construction have structural bonds, since they collaborate to produce new production facts. Likewise, other systems also exist within the enterprise, but their elements are of a different kind. Information, communication and technology (ICT) are constructed from hardware and software elements, whereas infrastructure (i.e. facilities) are constructed from building-construction elements. The enterprise thus consists of multiple sub-systems, where each of the sub-systems need to be designed, using the *generic system development process* (GSDP). The GSDP that was also illustrated in Fig. 2, starts with the construction of a *using system* to derive black box functions for a *provisioning system*.

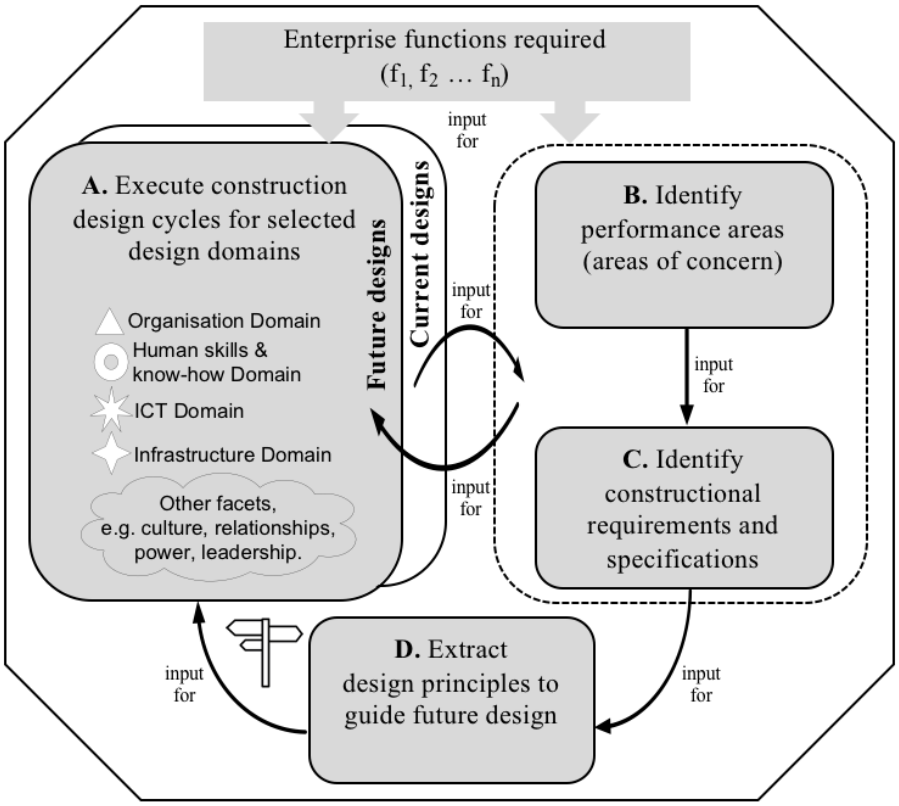


Fig. 3. ECDA's heuristic within the GSDP, based on Hoogervorst [20, p 256]

Fig. 4 provides a simplified view from [23] to illustrate how the GSDP was used, starting from the *using system* (i.e. *the environmental context*) to design the provisioning system (i.e. the *enterprise*). The GSDP is also used to illustrate how multiple enterprise sub-systems are developed concurrently. Each *support* arrow in Fig. 4 represents an iterative GSDP that exists between a using system and a provisioning system.

Fig. 4 illustrates two of the *support* arrows, highlighted in black, with the following interpretation: (1) The *enterprise* (as provisioning system) supports *the environmental context* (as using system); and (2) The *ICT sub-system* (as provisioning system) supports the *organization sub-system* (as using system). Explaining the last-mentioned *supports* arrow in terms of the GSDP, illustrated right next to the *supports* arrow in Fig. 4, the construction of the *organization sub-system* is used as a starting point to perform *functional design* of the *ICT sub-system*. Then, the *functions* of the *ICT sub-system* are used as input to perform *constructional design* of the *ICT sub-system*.

As indicated in [23], the notion of *system* alone is not sufficient to describe the enterprise, its construction and its behavioral complexities. An enterprise consists of many *facets*, such as human skills and know-how, culture, relationships, power and leadership [20]. Fig. 4 represents *facets* with cloud-constructs. We believe that some of these facets may also be classified as design domains when it is possible to follow the GSDP to design a future version of the *facet*. Hence, we believe that *human skills & know-how* needs to support the *organization sub-system* and should therefore be designable. Yet, we acknowledge that the GSDP is less useful when other facets, such as culture and power need to be "designed".

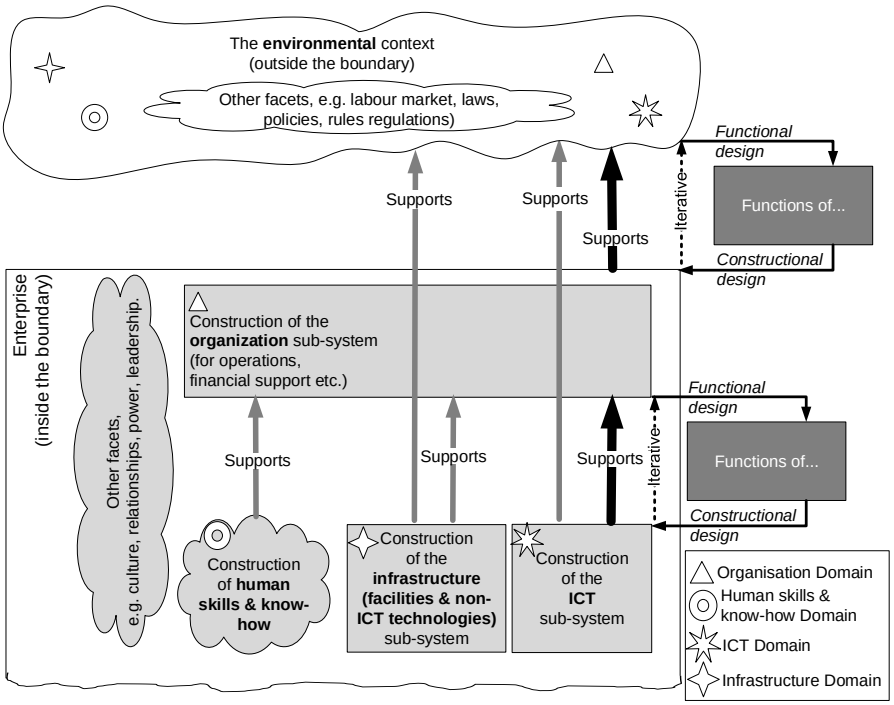


Fig. 4 The main EE domains, based on [23]

Fig. 4 includes several grey-shaded constructs to indicate the envisaged design scope for ECDA, including organization, ICT, infrastructure, human skills & know-how, and other facets. Next, we provide ECDA's interpretation and means for representing the four design domains:

(a) Organization. Dietz and Mulder [17] define the o*rganization* of an enterprise as a *social system*, i.e. actor roles, implemented by human beings, form relationships due to collaboration to produce production facts. ECDA adopts Dietz and Mulder's [17] four aspect models to represent the essence of enterprise operation in a coherent, comprehensive, consistent and concise way.

(b) ICT. Software applications, databases and ICT hardware are included [17]. ICT can be designed in the context of different using systems, such as construction of the organization, or construction of the environment. Hoogervorst [20] describes IT design aspects as the use of IT systems and their infrastructural characteristics. ECDA is not prescriptive on suggesting models for representing ICT constructs.

(c) Infrastructure. Facilities and other non-ICT technologies that support actor roles and their production acts are included. Enterprises within different industries may require different representations of infrastructure, based on the type of production acts that should be supported [23]. ECDA is not prescriptive on suggesting models for representing infrastructure constructs.

(d) Human skills and know how - Human skills & know-how constitutes human abilities and skills required when executing production acts, as well as coordination acts [23]. 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. The three-level capacity development approach of Imbaruddin [4] identifies the *individual (level 3)* as the skills, experience and knowledge that allow each person to perform. Some of these are acquired formally, through education and training, whereas others come informally, through doing and observing. Bloom [24] refers to the people component as a psychosocial subsystem, meaning the interrelation of social factors and individual thought and behavior. The description of this component includes elements such as values, attitudes, motivation, morale, and personal behavior of each individual.

B: Identify performance areas (areas of concern)

Areas of concern are generic characteristics that the black-box or white-box enterprise properties must manifest [20]. De Vries [21] states that due to the negative connotation to *concerns*, *performance areas* should rather be used. In this context, a *performance area* is a generic characteristic of an enterprise that must be addressed via enterprise design. A *design domain* must operationalize one or more *performance areas*. The performance area must be stated in terms of a variable, that can increase (improve) or decrease (deteriorate) [21].

Within this step, in consultation with ECDA's main user, performance areas or concerns are documented, e.g. internal efficiency, fiscal viability, or quality of caregiving.

C: Identify constructional requirements and specifications

The *constructional requirements* express certain wants and needs that the system construction must fulfil in view of the intended black box properties as well as the performance areas [20].

As indicated by [21], it is difficult to distinguish between *constructional requirements* and *design principles* (used in the next activity, i.e. activity D). Both provide *guidance* on how design of design domains or their embedded constructs must proceed. Usually a constructional requirement is defined for a narrow design scope, i.e. designing one particular construct, such as a software application. If a constructional requirement is generic in nature and applicable to a larger design scope, such as the entire ICT domain, the constructional requirement is transformed into a design principle, as indicated in activity D.

Constructional requirements will be defined by ECDA's main user. The requirements have to be associated with the *performance areas* that were identified and effected through design cycles when (re-)designing the design domains.

Constructional requirements need to be stated in a prescriptive format, using the words/phrases such as *should*, *must* or *may not*. The phrase "must be" is useful to indicate that the prescription needs to be verifiable [21].

D: Extract design principles

White box system properties result from the system's construction. Guidance for constructional design is informed by constructional architecture, also called *design principles* [20].

Existing constructional requirements, identified in activity C, will be used during activity D to identify requirements that are generic and useful to guide future development of applicable design domains. We believe that general design principles may also be extracted from IC literature.

5. Validating ECDA's Comprehensiveness

The SLR in [11] indicated that numerous IC development approaches exist, each focusing on different performance areas and functions in a disparate way, as noted in section 1. We include these learnings and contributions within ECDA. Fig. 5, a different representation of ECDA's heuristic, indicates how ECDA incorporated existing IC development approaches, previously discussed in section 3.3. The construction design cycle (activity A) adopts an iterative process of suggesting the (re-)design of constructs. This step encapsulates various elements of IC approaches, demonstrating ECDA's comprehensiveness.

ICDA	EE Main design domains	IC within ECD			Other IC approaches		
		Social system Bloom [24]	**Performance framework** Bergin-Seers and Breen [26]	**Leadership framework** Nupponen [29]	**Three levels of capacity development** Imbaruddin [4]	**Five dimensional framework** Grindle and Hildebrand [27]	**IC elements** Scheepers [28]
Activity A	Execute construction design cycles for selected design domains						
	Environment	X	X		X	X	X
	Enterprise boundary				X		
	Organisation sub system (Operations)	X	X			X	X
	Organisation sub system (Financial support)	X					X
	Organisation sub system (Skills support)		X	X			
	Organisation sub system (Human support)					X	X
	Organisation sub system (Maintenance)	X					
	ICT sub system	X					X
	Human skills and know-how sub system	X		X	X		X
	Infrastructure sub system	X					X
Activity B	Identify performance areas (areas of concern)						
		X	X	X		X	
Activity C	Identify constructional requirements and specifications						
Activity D	Extract design principles to guide future design						

Fig. 5 ECDA's heuristic, synthesizing existing IC approaches

In Table 1 we also validate the comprehensiveness of ECDA as an enterprise design approach, indicating that ECDA addresses the eleven *approach design principles* from [18].

Table 1. Validating ECDA against the ADPs

Approach Design Principle	**Applied to ECDA**
Principle A - Explicit concept of the enterprise: A design approach should indicate how an enterprise is perceived or conceptualized.	The ECDC is perceived as social system. Also, it is perceived as a living organism defined within a morphogenic paradigm (see section 4.1).
Principle B – Explicit phenomenon: A design approach should provide evidence for a phenomenon or class-of-problems, i.e. similar kinds of problems.	Atmore et al. [2] state that various challenges exist within the early childhood sector, among those are IC identified as a class-of-problems (see section 1).
Principle C – Explicit paradigm of value-creation: A design approach should state a paradigm of value-creation as a testable proposition for addressing an existing phenomenon or class-of-problems.	The ECDA, through a heuristic, enables ECD directors to effectively transition from functional requirements to constructional design in order to develop IC.
Principle D - Explicit means (ways) of demarcating and representing design scope: 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.	ECDA do not demarcate new design domains, but adopts those described in section 4.2 as means to represent the ECDC' constructional design (activity A of ECDA's heuristic).
Principle E – Well-demarcated and well-defended design scope: A design approach should define and defend the intended	The ECDA uses the *generic system development process* (GSDP) to facilitate constructional

Approach Design Principle	Applied to ECDA
design scope to achieve the intended value-creation.	design of the provisioning system (refer to Fig. 2).
Principle F – Representations of design scope: A design approach should clearly define and motivate notation standards that are used to adequately describe/represent the design scope.	The organization domain adopts DEMO aspect models [17] as the notation standard, whilst ECDA is not prescriptive for the infrastructure and ICT domains. The human skills & know how domain will be represented by curriculum vitae.
Principle G - Approach form and function: A design approach should clearly define the constructs and features of the approach.	The ECDA's function is described in section 4.1 and its form (i.e. a heuristic), is presented in section 4.2.
Principle H: Justificatory knowledge: A design approach must provide explanatory knowledge that links the paradigm of value-creation with its constructional components.	As indicated in Fig. 1, the ECDA is a theory-ingrained artefact, guided by approach design principles (discussed in section 3.1), informed by Hoogervorst's approach (presented in section 3.2), as well as existing IC development approaches (introduced in section 3.3).
Principle I – Approach mutability: A design approach should clearly state possibilities for tailoring the approach, within the pre-defined design scope.	The ECDA may be applied to a different sector, industry or operational context than ECD.
Principle J – Principles of implementations (conditional): A design approach may incorporate guidance for implementing the approach.	Partially: The demonstration of ECDA's heuristic in section 6 provides some guidance in the form of questions per activity (see Table 2).
Principle K – Expository instantiation (optional): A design approach may incorporate an instantiation.	Partially: An instantiation of ECDA's heuristic is included in section 6. The instantiation is not comprehensive to cover all the design domains.

As indicated in Table 1, the current version of ECDA addresses the eleven principles, except for principles J and K that are only partially addressed. Suggestions for comprehensive demonstrations of ECDA are discussed in section 7.

6. Application of ECDA's Heuristic

In this section, we demonstrate ECDA's heuristic in accordance with section 4.2. Ideally, holistic design requires identification of multiple *functions*, *multiple performance areas* that need to be identified for the entire enterprise and all its *design domains and facets* [20]. In addition, *design principles* need to guide the design of the entire enterprise [20]. ECDA's heuristic supports a holistic approach, but for the purpose of this article, we only focused on a single function, i.e. *child caregiving*, to demonstrate a single cycle of ECDA's heuristic. Table 2 presents ECDA's heuristic on the left-hand side and its application at a real-world ECDC at the right-hand side. We also grey-shaded the parts of ECDA's heuristic that were demonstrated.

As indicated in Fig. 3, ECDA's heuristic requires the main user to select a function as main input to perform the four main activities. The function *child caregiving* was

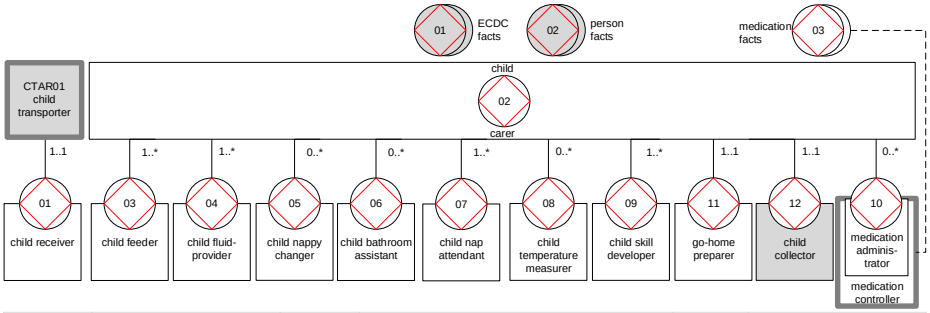
selected to demonstrate the heuristic, since its *management and administration* is currently inadequate due to inefficient ICT support in providing timeous feedback to management when new production facts come into existence. Inadequate management has a detrimental effect on one of the performance areas, i.e. *quality of caregiving*.

Table 2. Scope of demonstrating ECDA's heuristic

ECDA's heuristic – holistic scope	**ECDA's application at ECDC**
Functions ($f_1, f_2 \ldots f_n$) as input.	Single function (f_1): *child caregiving*.
B. Identify performance areas (areas of concern).	Single performance area of concern: *quality of caregiving.*
A. Execute construction design cycles for selected design domains – **current design.**	See below.
Organization domain: What is the *current design* of the organization domain? Is it effective in terms of *performance areas*?	The *current design* for the function *child caregiving*, using the Cooperation Model (CM) to represent the *current design* (see Fig. 6). We believe that the *essence* of the current operations is effective in terms of *quality of caregiving*. The problem is that operations are not well supported by ICT.
ICT domain: What is the *current design* of the ICT domain? Is it effective in terms of *performance areas*? Is it effective in supporting the organization domain? If not, what *functions* are needed from the ICT domain?	The ICT domain is currently under-represented by the ECDC. It does not support the organization domain and has a detrimental effect on the *quality of caregiving*. Given the essential design of the child caregiving function, depicted in Fig. 6, the following functions are needed from ICT: • Fact maintenance, i.e. creating, reading, updating and deleting facts associated with child reception, feeding, providing fluids, nappy changing, bathroom assisting, nap attending, temperature measuring, structured skill-development, and go-home preparation. • On-time reporting to the director, highlighting problems, e.g. push notifications where needed. • Daily electronic reporting to parents.
Infrastructure domain: What is the *current design* of the infrastructure domain? Is it effective in terms of *performance areas*? Is it effective in supporting the organization domain? If not, what *functions* are needed for the infrastructure domain? **Excluded for demonstration.**	
Human skills & know-how domain: What are the *current* human skills & know-how? Is it effective in terms of *performance areas*? Is it effective in supporting the organization domain? If not, what changes are needed? **Excluded for demonstration.**	
Other facets: What are the *current* facets? Are they effective in terms of *performance areas*? If not, what changes are needed? **Excluded for demonstration.**	
C. Identify constructional requirements and specifications.	See below.
Organization domain: What *constructional requirements* should be addressed by the *future*	Since the current organizational design for *child caregiving* is sufficient, there is no need to identify constructional requirements for the future design of the organisation.

ECDA's heuristic – holistic scope	ECDA's application at ECDC
design of the organization domain?	
ICT domain: What *constructional requirements* should be addressed by the *future design* of the ICT domain?	Constructional requirements for an ICT solution specify that the solution: • Must be cloud-based. • Must be easy to use. • Must be accessed via single sign-on using fingerprint-identification. • Must be available 100% of the time, accessible on- or off-line. In event of being off-line, data will be uploaded as soon as connectivity is restored. • Must be easily accessible to the primary caregiver, such as a hand-held device. • Must be accessible to multiple users in real time, e.g. parents, the director and caregiver. • Must be the core communication interface between the ECDC and parents.
Infrastructure domain: What *constructional requirements* should be addressed by the *future design* of the infrastructure domain? **Excluded for demonstration.**	
Human skills & know-how domain: The concept of *constructional requirements* is NOT applicable to human skills & know-how. **Excluded for demonstration.**	
Other facets: The concept of *constructional requirements* is NOT applicable to other facets. **Excluded for demonstration.**	
D. Extract design principles to guide future design.	From the constructional requirements that were identified for the ICT solution (Activity C), the following are generic for the ICT domain: • Must be cloud-based. • Must be easy to use.
A. Execute construction design cycles for selected design domains – **future design.**	See below.
Organization domain: What *future design* of the organization domain will address identified *constructional requirements*?	Future design will be the same as the current design, i.e. the essential operations, as depicted in Fig. 6 also represent the future design.
ICT domain: What *future design* of the ICT domain will address organization-supporting *functions* and identified *constructional requirements*?	Although not detailed here, alternative constructs will be compared against the required *functions* and *constructional requirements*. It is possible that existing software solutions exist that may be bought off-the-shelf. Alternatively, a new software application will have to be developed.
Infrastructure domain: What *future design* of the infrastructure domain will address organization-supporting *functions* and identified *constructional requirements*? **Excluded for demonstration.**	
Human skills & know-how domain: What *future* human skills & know-how will support the organization domain? **Excluded for demonstration.**	

ECDA's heuristic – holistic scope	ECDA's application at ECDC
Other facets: What should be the arrangement of future facets? **Excluded for demonstration.**	



transaction ID	transaction kind	product ID	product kind	executor ID	executor role
TK01	child reception	PK01	[child] **on** [date] is received	AR01	child receiver
TK02	child caring	PK02	child caring **for** [date] **is** done	AR02	child carer
TK03	feeding	PK03	**the** feeding **for** [child] **on** [time] **is** done	AR03	child feeder
TK04	providing fluids	PK04	**the** fluid **for** [child] **on** [time] **is** provided	AR04	child fluid-provider
TK05	nappy changing	PK05	[child] **on** [time] **is** nappy-changed	AR05	child nappy changer
TK06	bathroom assisting	PK06	[child] **on** [time] **is** bathroom-assisted	AR06	child bathroom assistant
TK07	nap attending	PK07	[child] **on** [date] **is** nap-attended	AR07	child nap attendant
TK08	temperature measuring	PK08	**the** temperature **for** [child] **on** [time] **is** measured	AR08	child temperature measurer
TK09	structured skill-development	PK09	**the** skill **of** [child] **on** [date] **is** developed	AR09	child skill developer
TK10	medication administering	PK10	**the** medication **for** [child] **on** [time] **is** administered	AR10	medication administrator
TK11	go-home preparation	PK11	[child] on [date] **is** prepared to go	AR11	go-home preparer
TK12	child collection	PK12	[child] on [date] **is** collected	AR12	child collector

Fig. 6 The ECDC's Cooperation Model, represented by the Coordination Structure Diagram and Transactor Product Table

The demonstration of ECDA's heuristic in Table 2 excluded theory from IC, even though Fig. 5 provided a mapping to existing IC literature. During ECDA's further development and refinement, we believe that existing IC literature (as mapped in Fig. 5) will be useful to further shape ECDA.

7. Conclusion and Future Research

IC is defined as the ability of an enterprise to pursue its objectives, and is therefore a prerequisite for delivering quality services. IC is not a new concept in the public sector performance arena, but not well defined or researched in the ECD sector.

ECDCs across South-Africa are sub optimal, indicating that *less than half* of all registered centers having nothing more than staff attendance records or job descriptions. Various solutions, frameworks and approaches exist, but none are integrated or constructed in a manner to guide administrators on *how* to develop IC, let alone inform the (re)design of constructs in order to improve quality of services delivered. Thus, ECDA is constructed to *develop* IC that is useful to ECD directors or administrators when they need to improve quality of services.

Metamodels provided the means of defining the rules at a higher level of abstraction, and this in essence acted as an introduction to the abstraction levels adopted in this study. The general conceptual modelling framework was used to explain how the ECDA was constructed as an instantiation of EECM, a metamodel for enterprise design approaches.

ECDA as a theory-ingrained artefact was guided by (1) approach design principles that were derived from EECM, (2) Hoogervorst's approach, and (3) IC development approaches. Through synthesis, it is shown and proven that the plathora of existing solutions and frameworks were effectively integrated within ECDA's heuristic. The heuristic should enable ECDA's main user to systematically (re-)design certain enterprise design domains in order to have an impact on problematic performance areas.

We *demonstrated* ECDA's heuristic within a real-world ECDC, starting with the organisation domain's *current design* and specifying the ICT domain's *future design* to address inefficiencies related to the *administration and management* of the function *child caregiving*. Within its existing theoretical structure, we need to further develop ECDA iteratively and in a participative way to ensure that it is useful within a real-world ECDC context. In future, ECDA could also be tested within a different sector or industry to test its suitability, robustness, mutability and scalability.

In closing, albeit significant progress has been made in the South-African early childhood and Grade R spheres, ECDA as a theory-ingrained artefact has the ability to develop IC, and thus improve the *quality of services* delivered. ECDA is poised to not only have a contribution to the educational domain, but could have a societal impact for the majority of South-Africa's children.

References

1 Experthub: How do I start a child services business?, https://www.experthub.info/launch/start-up-industry-specific-qas/how-do-i-start-a-child-services-business/, last accessed 2018/15 December

2 Atmore, E., Van Niekerk, L. J., Ashley-Cooper, M.: Challenges facing the early childhood development sector in South Africa. S. Afr. J. of Child. Educ., **2**(1), 120-139 (2012). doi.org/10.4102/sajce.v2i1.25

3 UNICEF: Draft national integrated early childhood development policy. Report, (2015)

4 Imbaruddin, A. (2003). Understanding institutional capacity of local government agencies in indonesia. Australian National University

5 Assefa, T. (2014). Institutional capacity of local government in implementing the national policy on ethiopian women (NPEW): the case of Dilla town. Dilla University

6 Department of Social Development: Introduction and background to the policy. In: National integrated early childhood development policy. pp. 18-28. Government Printers, Pretoria (2015). https://www.gov.za/sites/default/files/gcis_document/201610/national-integrated-ecd-policy-web-version-final-01-08-2016a.pdf

7 Hayden, J.: Directors of early childhood services: experience, preparedness and selection. J. of Australian Res. in Early Child. Educ., **1**, 49-61 (1997)

8 Atmore, E.: Early childhood development in South Africa – progress since the end of apartheid. Int. J. of Early Years Educ., **21**(2-3), 152-162 (2013). 10.1080/09669760.2013.832941

9 Van Heerden, J.: Quality in South African early learning centres: Mothers' and teachers' views and understanding. S. Afr. J. of Child. Educ., **6**(1), 1-11 (2016). doi.org/10.4102/sajce.v6i1x.423

10 Department of Social Development: Audit of early childhood development (ECD) centres - national report. Report, Department of Social Development (2014)

11 De Boer, A. C., De Vries, M.: A framework to embed institutional capacity at an early childhood development centre. SAIIEneXXXt Proceedings, 30th September – 2nd October 2019, 613-627 (2019)
12 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. Int. J. of Org. Des. and Eng., **3**(1), 86-114 (2013). doi: 10.1504/IJODE.2013.053669
13 De Vries, M., Van der Merwe, A., Gerber, A.: Extending the enterprise evolution contextualisation model. Ent. Inf. Syst., **11**(6), 787-827 (2017). doi.org/10.1080/17517575.2015.1090629
14 Sein, M., Henfredsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. MIS Q., **35**(1), 37-56 (2011). doi: 10.2307/23043488
15 De Vries, M., Berger, S.: An action design research approach within enterprise engineering. Systematic Practice and Action Research, **30**(2), 187-207 (2017). DOI: 10.1007/s11213-016-9390-7
16 Henderson-Sellers, B., Ralyte, J.: Situational method engineering: state-of-the-art review. J. of Univers. Comput. Sci., **16**(3), 424-278 (2010)
17 Dietz, J. L. G., Mulder, H., B, F.: Enterprise ontology: A human-centric approach to understanding the essence of organisation. Springer International Publishing AG (2020)
18 De Vries, M.: Guiding the development of enterprise design approaches. S. Afr. J. of Ind. Eng., **27**(3), 12-22 (2016). doi.org/10.7166/27-3-1621
19 Hoogervorst, J. A. P.: Foundations of enterprise governance and enterprise engineering - presenting the employee-centric theory of organisation. Springer, Berlin Heidelberg (2018)
20 Hoogervorst, J. A. P.: Practicing enterprise governance and enterprise engineering - applying the employee-centric theory of organization. Springer, Berlin Heidelberg (2018)
21 De Vries, M.: Reducing ambiguity during enterprise design. Requir. Eng., **25**(2), 231-251 (2020). 10.1007/s00766-019-00320-1
22 Dietz, J. L. G.: Enterprise ontology. Springer, Berlin (2006)
23 De Vries, M.: Towards consistent demarcation of enterprise design domains. In: De Cesare, S., Frank, U. (eds.) Advances in Conceptual Modeling. pp. 91-100. Springer, Switzerland (2017). doi.org/10.1007/978-3-319-70625-2_9
24 Bloom, P. J.: Child care centers as organizations: A social systems perspective. In: Child and Youth Care Forum. Vol. 20, pp. 313-333. Springer (1991). DOI: 10.1007/BF00757061
25 Davis, A., Lemma, T.: Capacity development: A UNDP Primer, https://www.undp.org/content/undp/en/home/librarypage/capacity-building/capacity-development-a-undp-primer.html, last accessed 2020/15 April
26 Bergin-Seers, S., Breen, J.: The performance of long day care centres in rural and remote areas. Australian J. of Early Child., **27**(1), 24-33 (2002). DOI: 10.1177/183693910202700106
27 Grindle, M. S., Hilderbrand, M. E.: Building sustainable capacity in the public sector: what can be done? Public Adm. and Dev., **15**(5), 441-463 (1995). doi.org/10.1002/pad.4230150502
28 Scheepers, L. A. (2015). An institutional capacity model of municipalities in South Africa. Stellenbosch University, Stellenbosch
29 Nupponen, H.: Framework for developing leadership skills in child care centres in Queensland, Australia. Contemp. Issues in Early Child., **7**(2), 146-161 (2006). doi.org/10.2304/ciec.2006.7.2.146
30 Giachetti, R. E.: Design of enterprise systems. CRC Press, Boca Raton (2010)