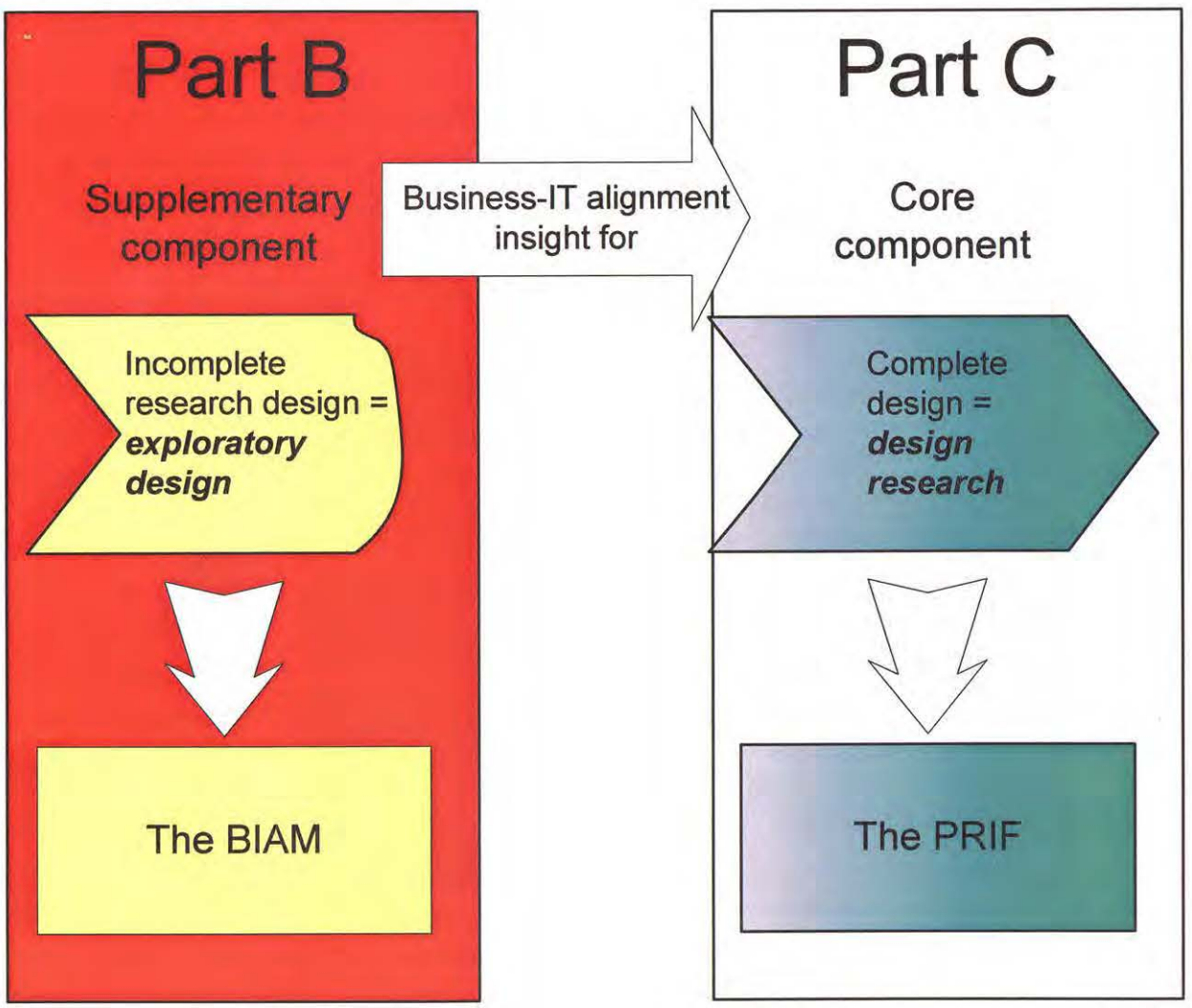
PART B: THE BUSINESS-IT ALIGNMENT MODEL (BIAM)

All we ever know is our models, but never the reality that may or may not exist behind the models. ...Our models may get closer and closer but we will never reach direct perception of reality. ~ Stephen Hawking

As stated in Chapter 2, this thesis follows a mixed methods design, with two design components: (1) a supplementary component, and (2) a core component. Since the result of the supplementary component, the BIAM, provides insight for the core component in developing the PRIF (Process Reuse Identification Framework), Part B starts with a discussion on the supplementary component.

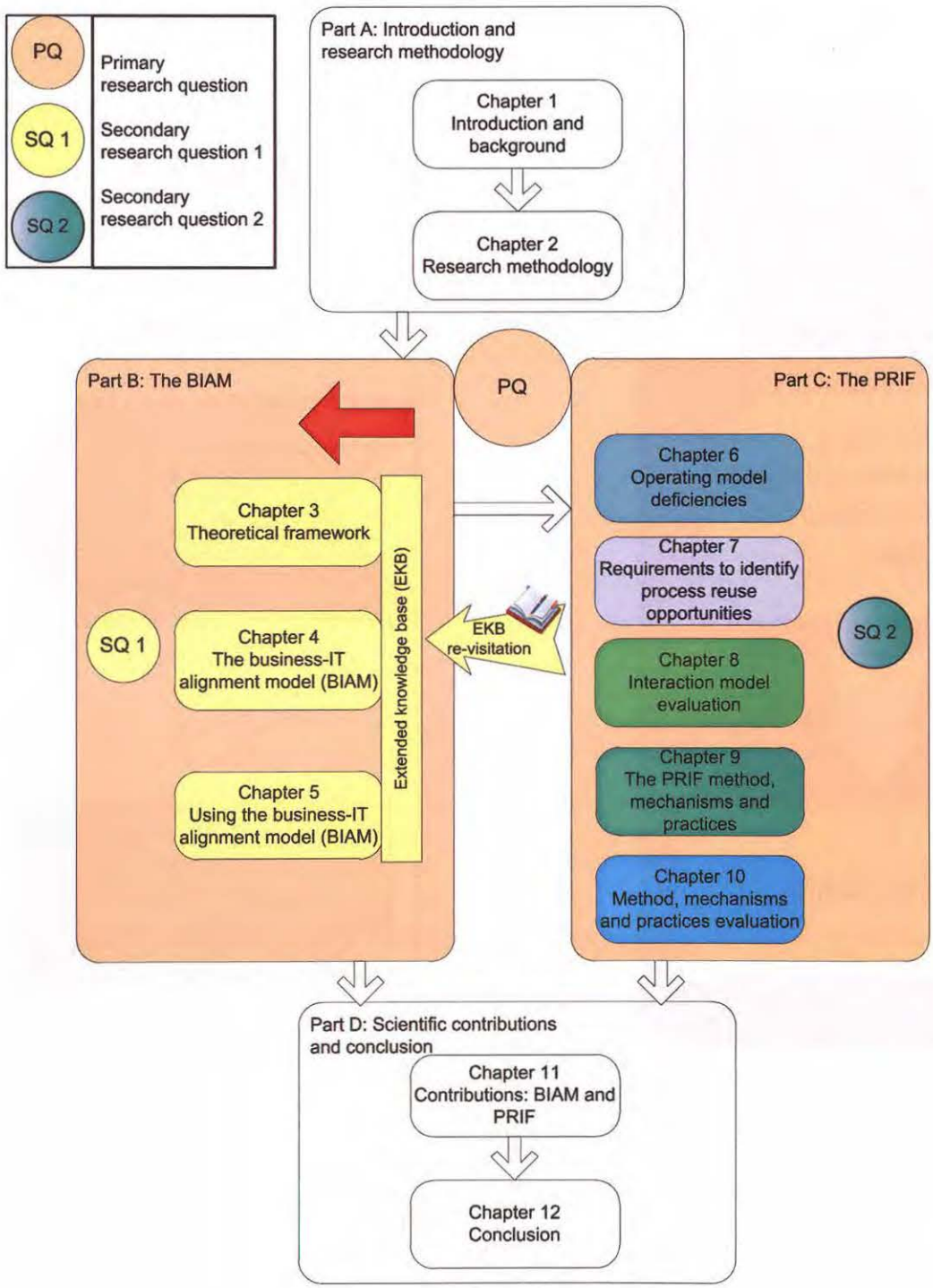


Part B answers *Research Question 1*, repeated from section 1.4:

What model is required to contextualise different business-IT alignment approaches?

Part B contains Chapters 3 to 5 to develop the BIAM (Business-IT Alignment Model), using an exploratory design, as described in sections 2.3.3 and 2.6.3.

- Chapter 3 provides theoretical background for the development of the BIAM.
- Chapter 4 applies the theoretical concepts introduced in Chapter 3 to develop the BIAM.
- Chapter 5 applies the BIAM, contextualising two alignment approaches in terms of the BIAM.



Chapter 3. Theoretical background

3.1 INTRODUCTION

This chapter starts with the theoretical background for applying exploratory design (as the supplementary component) for the development of the BIAM. The development of BIAM will answer the first research question:

What model is required to contextualise different business-IT alignment approaches?

Several authors provide definitions for *business-IT alignment*. According to Luftman and Kempaiah (2008, p. 102), business-IT alignment refers to how business and IT are "integrated, in harmony, converged, linked, fused, synthesized", whilst Wegmann, Regev, & Loison (2005, p. 1) states that business-IT alignment is the "correspondence between a set of components". Nadler & Tushman (1980, p. 40) have broadly defined business-IT fit as "the degree to which the needs, demands, goals, objectives, and/or structure of one component are consistent with the needs, demands, goals, objectives, and/or structure of another component". The latter definition provided by Nadler & Tushman is useful within the context of this thesis, as it accommodates *alignment/fit* of various components, *at various levels* within an enterprise. Many alignment approaches, however, still focus on creating *business-IT alignment*, i.e. creating consistency between the needs, demands, goals objectives, and/or structure of *business* components with the needs, demands, goals, objectives, and/or structure of *ICT* components.

According to the 2010 survey by Luftman & Ben-Zvi (2010), *business and IT alignment* has been a top concern for IT managers for almost 30 years. *Business-IT* alignment has been an important challenge in both private and public/non-profit sectors since the early 1980s (Knoll and Jarnvenpaa, 1994). There is strong evidence of a link between business-IT alignment and enterprise performance (Luftman and Kempaiah, 2007), using the alignment assessment criteria of Luftman (2003).

As stated before, Enterprise Architecture (EA) has several definitions (see section 4.3.2.1), and overlaps with other emerging disciplines (enterprise engineering and enterprise ontology). However, EA is also perceived as a *business-IT alignment* enabler (Gregor, Hart, & Martin, 2007; Ross, 2003; Sauer & Willcocks, 2004; van der Raadt, Hoorn, & van Vliet, 2005). Ballengee (2010) maintains that the penultimate purpose of EA converges around *enabling alignment at several levels*.

A large number of theoretical EA frameworks exist; each has its own alignment focus/intent and possible application within a specific industry or type of enterprise. Examples include the Zachman Framework (Zachman, 1987) or the Open Group Architecture Framework (TOGAF) (The Open Group, 2009). Previous studies however fail to compare existing EA frameworks in terms of alignment intent, scope and means. Although Schekkerman (2004) provided a

descriptive comparison between various EA frameworks, and Sessions (2007) compared four prominent EA frameworks/methodologies with one another based on twelve (12) measurement criteria, an alignment-contextualisation model did not exist. An alignment-contextualisation model would be useful if an existing alignment approach (e.g. the *foundation for execution* approach of Ross et al. (2006)) required enhancement from another alignment approach. Therefore, there was a need to contextualise *numerous theoretical approaches* (some being associated with EA frameworks) in terms of *business-IT alignment* by answering three questions:

1. *Why* should the enterprise use the proposed approach to align?
2. *What* should the enterprise align?
3. *How* should the enterprise align?

Some authors delivered major contributions within the domain of business-IT alignment developing very specific frameworks, such as the Zachman Framework (Zachman, 1987) or the Open Group Architecture Framework (TOGAF) (The Open Group, 2009). Since this study focuses on an alignment perspective, and many frameworks and methodologies also enable alignment at several levels (Ballengee, 2010), this thesis uses the term *approach* to refer to the various frameworks and methodologies. As an example, reference is made to the Zachman *approach*, rather than the Zachman framework, highlighting the *alignment aspects*.

This chapter starts with definitions and perspectives on two complementary concepts, alignment and governance, in section 3.2. Section 3.3 introduces four prominent business-IT alignment approaches (the Zachman approach, the Open Group approach, the OMB approach, and the Gartner approach), followed by two less prominent alignment approaches (the *foundation for execution* approach, and the *essence of operation approach*). Section 3.4 briefly discusses eight *other alignment approaches* as secondary data sources for this thesis. The chapter concludes in section 3.5.

3.2 ALIGNMENT AND GOVERNANCE

Alignment, according to Hoogervorst (2009), refers to a certain *state*, which can only be attained through intentional *activities*. One of the key reasons for elusive alignment, is that executives tend to look for one silver bullet that will enhance alignment, whereas enterprises need to address many alignment components concurrently (Luftman & Ben-Zvi, 2010). Incremental IT developments for instance, occur collaboratively, iteratively, and concurrently with other enterprise developments. A larger scope of alignment inquiry could thus contribute to better alignment. Hoogervorst (2009) therefore presents alignment on two levels of scope, *business-IT alignment* versus *enterprise alignment* (see definitions in section 1.2.3).

With reference to Figure 17, *business-IT alignment* and *IT governance* are closely related. Hoogervorst (2009) distinguishes *between corporate governance*, *enterprise governance* and *IT governance*.

Corporate governance is defined as the "totality of internal structures and systems, as well as external rules and regulation, for internal control and risk management that ensures that enterprises exercise their responsibilities towards shareholders effectively and adequately" (Hoogervorst, 2009, p. 155). According to Hoogervorst (2009, p. 187), corporate governance focuses on compliance (financial reporting and internal control). However, he reasons that compliance requirements could only be satisfied as a result of enterprise design and the design of the ICT system, based on considerations such as process excellence, quality, efficiencies and security. Therefore, *enterprise governance* and *IT governance* are prerequisites for compliance.

IT governance is the competence used (the *how*) for continuously creating a business-IT alignment *state*. *IT governance,* as defined by Hoogervorst (2009, p. 221) concerns the integration of skills, knowledge and technology for providing unified and integrated attention for IT development in:

1. establishing IT strategic initiatives,
2. developing IT architecture,
3. designing IT systems,
4. defining a portfolio of subsequent IT projects to implement designs, and
5. implementing IT projects (Hoogervorst, 2009, p. 221).

With reference to Figure 17*, enterprise governance* is the complement of IT governance, but within a wider context creating an enterprise alignment *state*. Comparable to the definition of IT governance, enterprise governance concerns the integrated attention for:

1. developing strategy (establishing strategic choices, initiatives, areas of concern and their related objectives),
2. developing enterprise architecture to guide enterprise design,
3. designing the enterprise,
4. defining the portfolio of subsequent projects, and
5. implementing the projects" (Hoogervorst, 2009, p. 316).

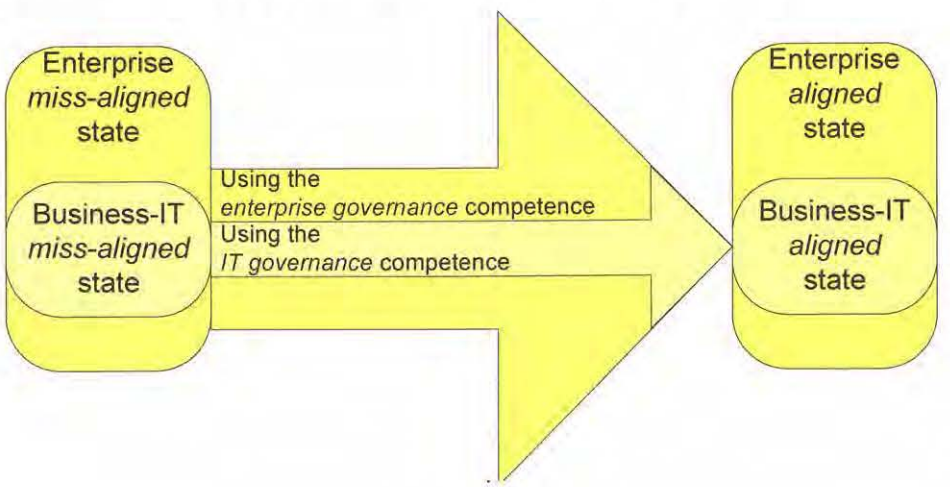


Figure 17: Using IT governance and enterprise governance to enact alignment

Although a number of business-IT alignment approaches exist, each with its own alignment paradigm, alignment scope, alignment mechanisms and practices, Hoogervorst (2009) maintains that miss-alignments can only be addressed from the perspective of the *enterprise as a whole*. The introduction of new information systems not only involve new hardware and software, but also require synchronisation with changes in jobs, skills, management and organisation (Laudon & Laudon, 1998; Martin, 1995). A mechanism is therefore required to understand the whole enterprise and all its components - not only focusing on the business and ICT components.

Since this study intends to develop a mechanism to understand the *components of the business-organisation*, as related to the *ICT components*, the theoretical foundations of current business-IT alignment approaches are abstracted. The theoretical foundations, creating common grounds for conceptual understanding, are:

1. Systems theory (section 3.2.1).
2. Systems engineering and basic system design process (section 3.2.2).
3. Three schools of thought on aligning the enterprise (section 3.2.3).
4. The ISO/IEC/IEEE 42010 standard (section 3.2.4).

Later in this thesis, the theoretical foundations are used in combination with a set of six alignment approaches, to develop a Business-IT Alignment Model (BIAM). Chapter 4 (section 4.3.1 and Figure 46) also provides an indication of how each of the following theoretical sections contributed towards the construction of the BIAM.

3.2.1 Systems theory

Since alignment concerns various components of an *enterprise*, systems theory is discussed as a means to create a common conceptual understanding of an *enterprise as a system* (see section 1.2.1 for a definition of the enterprise as a system).

Various definitions exist for describing a system; Jackson (2003, p. 3) defines a system as "a complex whole, a functioning of which depends on its parts and the interaction between these parts". Others extend the systems definition, stating that the parts are connected to perform a unique function that could *not be performed by the parts alone* (Boardman & Sauser, 2008; Gharajedaghi, 2006; Giachetti, 2010; Maier & Rechtin, 2002). Dietz (2006) emphasizes that the interacting parts or sub-systems *influence* each other. If the parts do not have an interacting effect, the parts merely form an aggregate.

Giachetti (2010) maintains that an appreciation of *typical system properties* contribute towards the analysis and design of systems. The discussion of several alignment approaches (see sections 3.3 and 3.4) related to this study, also refers to *typical system properties* and the means to accommodate the *system properties* during enterprise alignment. A list of *typical system properties* include (Giachetti, 2010):

1. *System boundaries.* A system boundary defines what is part of the system and what is not. The boundary is arbitrary, because it depends on the intentions and aims of the observer.
2. *Sub-systems.* Sub-systems are part of another system, but also systems in their own right. The viewpoint of the observer/analyst determines the boundary of a sub-system. Hitchins (2003) recommends that a sub-system should be defined such that the intra-relationships (relationships between parts of the sub-system) should be more than the interrelationships (relationships between parts and other sub-systems). In accordance with Hitchin's viewpoint, a functional structuring of an *enterprise* may define sub-systems, such as marketing, sales and manufacturing. As will be indicated in Part C (Chapter 8) of this thesis, the confinement created by a *sub-system boundary*, usually have adverse implications on streamlining/measuring end-to-end processes.
3. *Holism/complementation.* Holism/complementation is the idea that a system reveals emergent properties and behaviour that one cannot attribute to any one of its parts. For example, the *emergent* property, *performance* of an enterprise, cannot be attributed to a single part of the enterprise (e.g. marketing, operations, logistics etc.). Holism contrasts with reductionism, which decomposes a system into its parts and studies each part individually. Following a holistic approach requires one to focus on the relationships between the parts to understand how the interaction of the parts contributes to the emergent properties.
4. *Open versus closed.* An open system interacts with its environment, whereas a closed system does not interact with its environment. Enterprises as open systems need to observe their environment and perform dynamic adjustment of its system components to remain in a steady state.
5. *Purposefulness.* Purposeful systems have goals and motivations, but also the free will to change their goals. The enterprise, for instance has a mission statement (goal), whereas its employees also have their own goals and motivations. Understanding the purpose of the enterprise requires a deep understanding of the rationale that explains its actions. The rationale also depends on the environment, business culture and social culture.
6. *Feedback and control vs. dynamic interactions.* The field of *cybernetics* conceptualises the feedback and self-regulation mechanisms of a system. In an enterprise, management need to control the enterprise system. Managers usually use *performance measurements* as a feedback mechanism to control the enterprise. Performance measures may however be in conflict, which could lead to counterintuitive behaviour when management implements control actions. However, the basis of an open system model is the *dynamic interactions* of the components, rather than focusing on feedback (Hitchins, 2003). Enterprises change over time. They need to continuously adapt to their environment.
7. *Complexity.* If a system has a large number of parts, the system is *complicated*. The large number of parts makes it difficult to understand, but it is understandable to the skilled designer of the system. *Complexity*, however, occurs when a large number of parts exist, and the interaction between the parts creates *unpredictable behaviour*. According to

Gharajedaghi (2006) complexity inhibits our understanding of cause-and-effect relationships. Complexity leads to counterintuitive behaviour, e.g. actions intended to produce a certain outcome may generate opposite results. The theory of system dynamics developed by Forrester (1968) aims to model the interrelationships between system parts to predict *system behaviour*.

8. *Equifinality*. Enterprises exhibit the property of equifinality, which means that the system can accomplish its objectives with different inputs and different internal processes to produce outputs. Equifinality implies that there is no single best way to reach a goal. In addition, a best practice in one enterprise may not be transferable to another enterprise due to different cultures.

The list of *typical system properties* is referenced in upcoming sections to discuss different ways of addressing the *typical system properties* of an enterprise.

In addition to the *typical system properties*, Dietz (2006) states that two different notions exist for understanding a system: (1) the constructional system notion (see-section 3.2.1.1), which is required to understand the *structure/construction* of a system, and (2) the teleological/functional system notion, which is required to use and *control the system* (see-section 3.2.1.2). Both Dietz (2006) and Hoogervorst (2009) emphasise the *constructional* system notion in their alignment approaches, stating that one needs to have a deep understanding of how an enterprise is constructed prior to requirement elicitation for supporting information systems. The different notions of a system are re-visited when discussing the *essence of operation approach* of Dietz in sections 3.3.6 and 8.2.

3.2.1.1 The constructional system notion

This section applies the *typical system property* regarding *system boundary* discussed above (section 3.2.1) to provide and understanding of the constructional notion of a system. Bunge (1979) uses the *system boundary* property to distinguish between different constructs of a system (as illustrated in Figure 18). Due to a logical/physical *system boundary*, a system consists of a:

- *composition* (parts of the some category, i.e. physical, social, biological etc.),
- *environment* (parts of the same category, but not within the *boundary of the system*), and
- *structure* (a set of influencing bonds between the parts within the *boundary*, and between them and the parts in the environment).

Dietz (2006) added another construct, namely that a system has a definite *production* output (the parts within the *boundary* produce things that are delivered to the parts in the environment). Although not mentioned by Dietz, Hitchins (2003) also highlights that every part or system has a definite *capacity*, which influences *production* output. Capacity is however, an implementation issue, and thus not required for the ontological/essential view of a system.

Applying the constructs of Figure 18 to an enterprise, the *composition* of the enterprise as a social system would be social individuals; the *environment* would be parts of the same category

(social individuals) directly linked to the compositional parts, but outside the boundary; whereas the *structure* would be the mutual influencing relations among the system parts (i.e. individuals within the boundary and certain individuals outside the boundary). The *production* would be goods and/or services that are delivered to the environment.

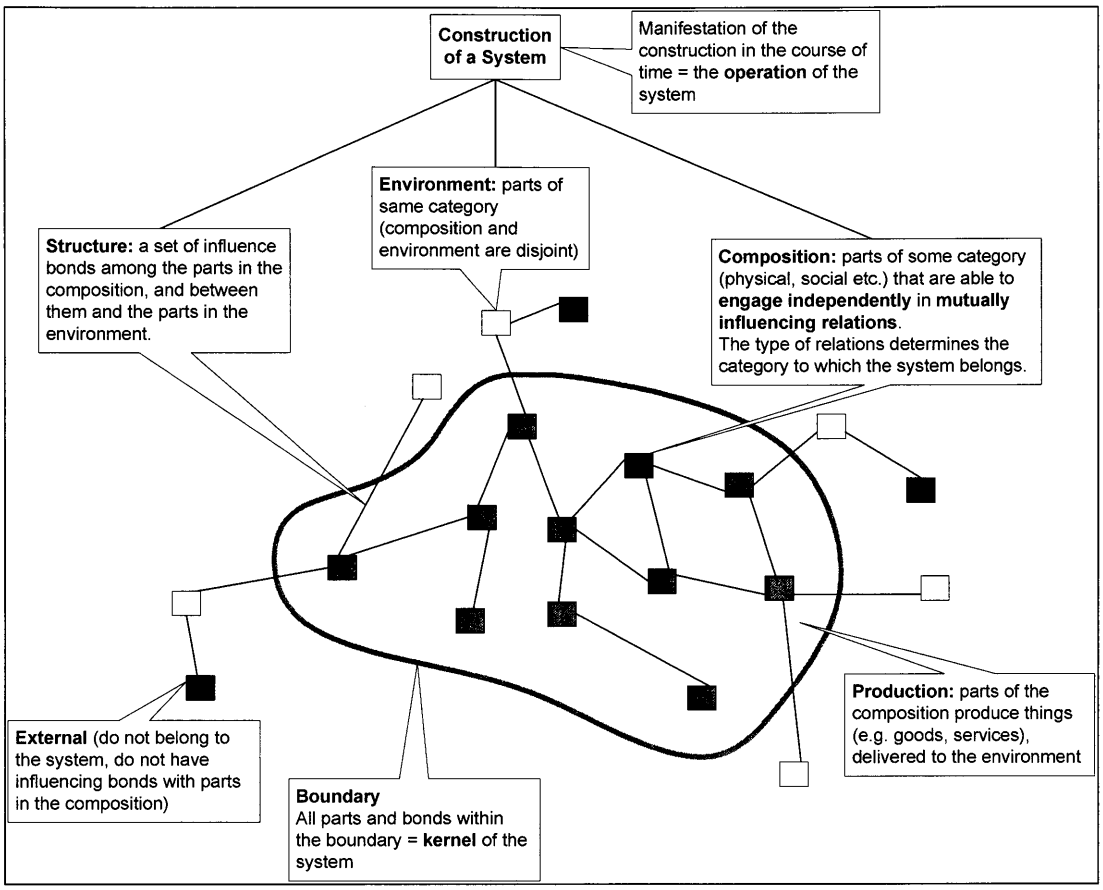


Figure 18: The structure/ontology of a system, based on Dietz (2006)

The constructional notion of the enterprise as a system (as depicted in the previous paragraph) needs to be communicated using appropriate representations. Dietz (2006) suggests the use of *white box models* to provide a conceptualisation of the constructional notion of a system. *White-box models* are used for building or changing/maintaining a system and the dominant type of model in all engineering sciences. An example of a *white box model* is the constructional decomposition model (i.e. bill-of-material) of a car (the car being the system), e.g. a car consists of a chassis, wheels, motor and lamps (Dietz, 2006).

The constructional notion of the enterprise as a system, represented *by white box models,* is thus required to understand how an enterprise is *constructed* and used by the *enterprise designer/engineer* as to build/maintain the enterprise. Only a few alignment approaches emphasise the constructional notion of a system, as highlighted later during the discussion on different alignment approaches.

In addition to the constructional notion of the enterprise, it is also necessary to understand the teleological notion of a system, which is concerned with the *function* and *behaviour* of the

system. The subsequent section therefore provides more theory on the teleological notion of a system.

3.2.1.2 The teleological system notion

Evidence of teleology, of purpose/goal-seeking behaviour in enterprises are unmistakable (Hitchins, 2003). An understanding of the behaviour of a system would allow managers to control the system and it is thus the dominant notion used by managers. A number of alignment approaches emphasise the teleological notion of a system (e.g. the Gharajedaghi approach), as highlighted later during the discussion of different alignment approaches. This section provides the teleological notion of a system and re-visits some of the *typical system properties* discussed earlier in section 3.2.1.

Management is usually concerned with the functions of an enterprise and how control of the input variables has an effect on output variables (Dietz, 2006). A *typical system property* emphasised with the teleological system notion, is that of system *feedback and control*. Managers of enterprises typically use performance measurement to gain *feedback and control* over enterprise behaviour.

The teleological notion of the enterprise as a system needs to be communicated using appropriate representations. *Black-box models* are typically used to conceptualise the functions and behaviours of the system *without knowing the detail construction and operation* of the system. An example of a *black box model* is the functional decomposition model of a car (the car being the system), e.g. a car consists of a lightning system, power system, steering system and brake system. *Black box models* are not useful to an engineer when maintaining the system (Dietz, 2006). Examples of black *box models* that describe enterprise behaviour include: process flowcharts and cause-and-effect diagrams, e.g. the sistemigrams of Boardman & Sauser (2008).

3.2.2 Systems engineering and the basic system design process

The previous section (section 3.2.1) on systems theory provided theory to conceptualise the enterprise as a system. i.e. revealing *typical system properties*, and understanding the enterprise from both a *constructional* viewpoint and a *teleological* viewpoint. This section introduces systems engineering and the basic *system design process* to delineate the process required for the development of any system. The purpose is to demonstrate how the *design process* is used as a *vehicle* to align systems with one another, ensuring that the needs, demands, goals, objectives, and/or structure of one system are consistent with the needs, goals, objectives, and/or structure of another system. The *design process* is for example evident in the Zachman approach (Zachman, 2009a) (see section 3.3.1) where Zachman refers to the *process of reification*, which gradually transforms system requirements to implementations. The *essence of operation* approach of Dietz (2006) (see section 3.3.6) also refers to the *design process* as a systematic process for aligning business with IT.

The International Council of Systems Engineering (INCOSE) (2004) defines *systems engineering* as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem".

One of the essential mechanisms of *systems engineering* is the basic *system design process*, depicted in Figure 19.

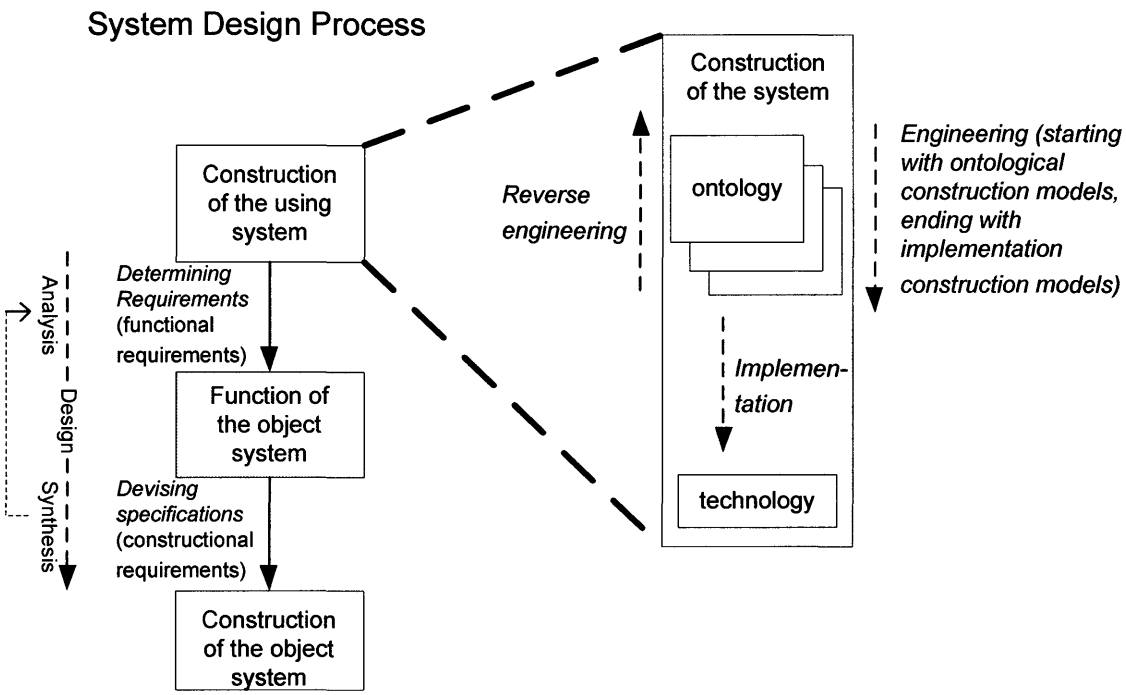


Figure 19: The basic system design process, based on Dietz (2006) and Hoogervorst (2009)

Every system that needs to be designed follows a generic *design process* that incorporates two systems: the *using system* and the *object system*. The object system is used by the using system. As an example, the object system could be an ICT system that needs to be designed and is used by the using system, the enterprise. The first design phase (see Figure 19, *Determining requirements*) involves the definition of the required function of the object system (the function is represented by a *black box model*). The function can only be determined in terms of the construction of the using system. The second design phase (see Figure 19, *Devising specifications*) starts with the function of the object system and concludes with the construction of the object system. Hoogervorst (2009) renames *specifications* as *constructional requirements,* that relate to the constructional design of a system. Dietz (2006) also explains that *design* (Figure 19, *Design* arrow) is the iterative alternation between *analysis* (Figure 19, *Analysis)* and *design* (Figure 19, *Design)*, i.e. design is not a one-way process.

Engineering (used in the narrow sense of the term, contrary to its use in systems engineering) entails the process during which constructional models (*white box models*) are produced (see Figure 19, *Engineering*). Engineers systematically produce a series of *ontological* construction models (e.g. construction models that are *implementation-independent*) and end with

implementation construction models, i.e. models that could be linked to technology means (Dietz, 2006).

This section discussed *systems engineering* and the *basic system design process* as vehicles to align different systems with one another. The next section presents different schools of thought that exist in the enterprise architecture community. The rationale is that alignment approach authors differ in their worldview and perception/focus on alignment value-creation.

3.2.3 Three schools of thought on aligning the enterprise

Lapalme (2011) states that the debates on enterprise architecture may be traced back to different schools of thought that exist in the enterprise architecture community. He suggests the use of three schools of thought to create common grounds in our understanding of the different value-propositions offered by enterprise architecture authors.

Lapalme provides a hypothesis that three schools of thought exist (see Table 8):

1. enterprise IT architecting (EIT),
2. enterprise integrating (E), and
3. enterprise ecological adaptation (EiE).

The taxonomy of *three schools of thought* is not meant to be exhaustive and should be viewed as 'ideal' types, i.e. author(s) typically do not fit perfectly in one school, but rather gravitate towards one (Lapalme, 2011). Also, Hoogervorst (2009, p. 120) states that the understanding and designing of enterprises lies in avoiding the either-or scheme by combining the structural-functionalistic perspective (evident in EIT and E) with the interpretative perspective (evident in EiE).

Table 8: A sub-set of qualifiers for the three schools of thought, based on Lapalme (2011)

Enterprise IT architecting (EIT)	**Enterprise integrating (E)**	**Enterprise ecological adaptation (EiE)**
Scope		
Enterprise wide IT platform (EIT). All components (software, hardware, etc.) of the enterprise IT assets.	**Enterprise (E).** The enterprise as a socio-cultural-techno-economic system; hence ALL the facets of the enterprise are considered – the enterprise IT assets being one facet.	**Enterprise-in-environment (EiE).** Includes the previous scope but adds the environment of the enterprise as a key component as well as the bidirectional relationship and transactions between the latter and its environment.

Enterprise IT architecting (EIT)	Enterprise integrating (E)	Enterprise ecological adaptation (EiE)
Purposes (value-creation paradigm)		
Effective enterprise strategy execution and operation through IT-Business alignment. The purpose is to enhance business strategy execution and operations. The primary means to this end is the aligning of the business and IT strategies so that the proper IT capabilities are developed to support current and future business needs.	**Effective enterprise strategy implementation through execution coherency.** The purpose is effective enterprise strategy implementation. The primary means to this end is designing the various facets of the enterprise (governance structures, IT capabilities, remuneration policies, work design, etc.) to maximise coherency between them and minimise contradictions.	**Innovation and adaptation through organisational learning.** The purpose is organisational innovation and adaptation. The primary means is the fostering of organisational learning by designing the various facets of the enterprise (governance structures, IT capabilities, remuneration policies, work design, etc.) as to maximise organisational learning throughout the enterprise.
Motto		
"EA as the glue between business and IT".	"EA as the link between strategy and execution".	"EA as the means for organisational innovation and sustainability".
Principles and Assumptions		
• Reductionism. • Business strategies and objectives are provided by the business and are correct. • Independent design of organisational dimensions. • Disinterest in none-IT dimensions.	• Holism. • Business strategies and objectives are provided by the business and are correct. • Environment as something to manage. • Joint design of all organisational dimensions.	• Holism. • System-in-environment coevolution. • Environment can be changed. • Joint design of all organisational dimensions.

One of the key differentiators between the three schools of thought is the *scope* of alignment. According to Table 8 (*Scope* qualifier), EIT authors emphasise alignment of components related to the enterprise IT assets, whereas the E authors consider alignment of all facets of the enterprise (IT assets being one asset). The EiE authors expand the extent of alignment even further by adding the *environment* as an alignment component. Since Lapalme defines an enterprise as a composition of socio/cultural/techno/economic parts, the environment (according

to Bunge (1979)) refers to parts of the same category (social/cultural/technical/economic parts), but not within the composition of the enterprise. When the *scope* of alignment increases, different purposes, mottos, principles and assumptions apply. Since EIT focuses on the IT assets, a *reductionist* paradigm may be appropriate, i.e. decomposing technical systems into parts. However, extending the alignment *scope* to include social, cultural, technical and economic parts requires a *holistic* paradigm (*holism* being a typical property of a system, as defined in section 3.2.1). According to the *holistic* paradigm, the emergent properties and behaviour of the enterprise cannot be attributed to the parts alone.

Section 4.3.2.1 re-visits the different schools of thought of Lapalme and provides a motivation for developing a Business-IT Alignment Model (BIAM) in accordance with the motto of the first school of thought (EIT). The next section presents the ISO/IEC/IEEE 42010 standard to provide common grounds for representing different facets of the enterprise. The purpose is to apply existing theory on *architecture description* (embedded in the ISO/IEC/IEEE 42010 standard) during the construction of BIAM.

3.2.4 The ISO/IEC/IEEE 42010 standard

The ISO/IEC JTC 1/SC 7 committee (2011) produced an *architecture description* standard (for systems and software engineering) to create common grounds (a conceptual model) for architecture description. Dictionary.com (n.d.) defines a *metamodel* as "the components of a conceptual model, process, or system". The architecture description could thus also be classified as a *metamodel,* i.e. *components of the conceptual model of an enterprise architecture description.* Figure 20 portrays the *metamodel*, using conventions for class diagrams defined in [ISO/IEC 19501] (see Appendix D for class diagram notation standards). Table 9 provides definitions for the elements in Figure 20.

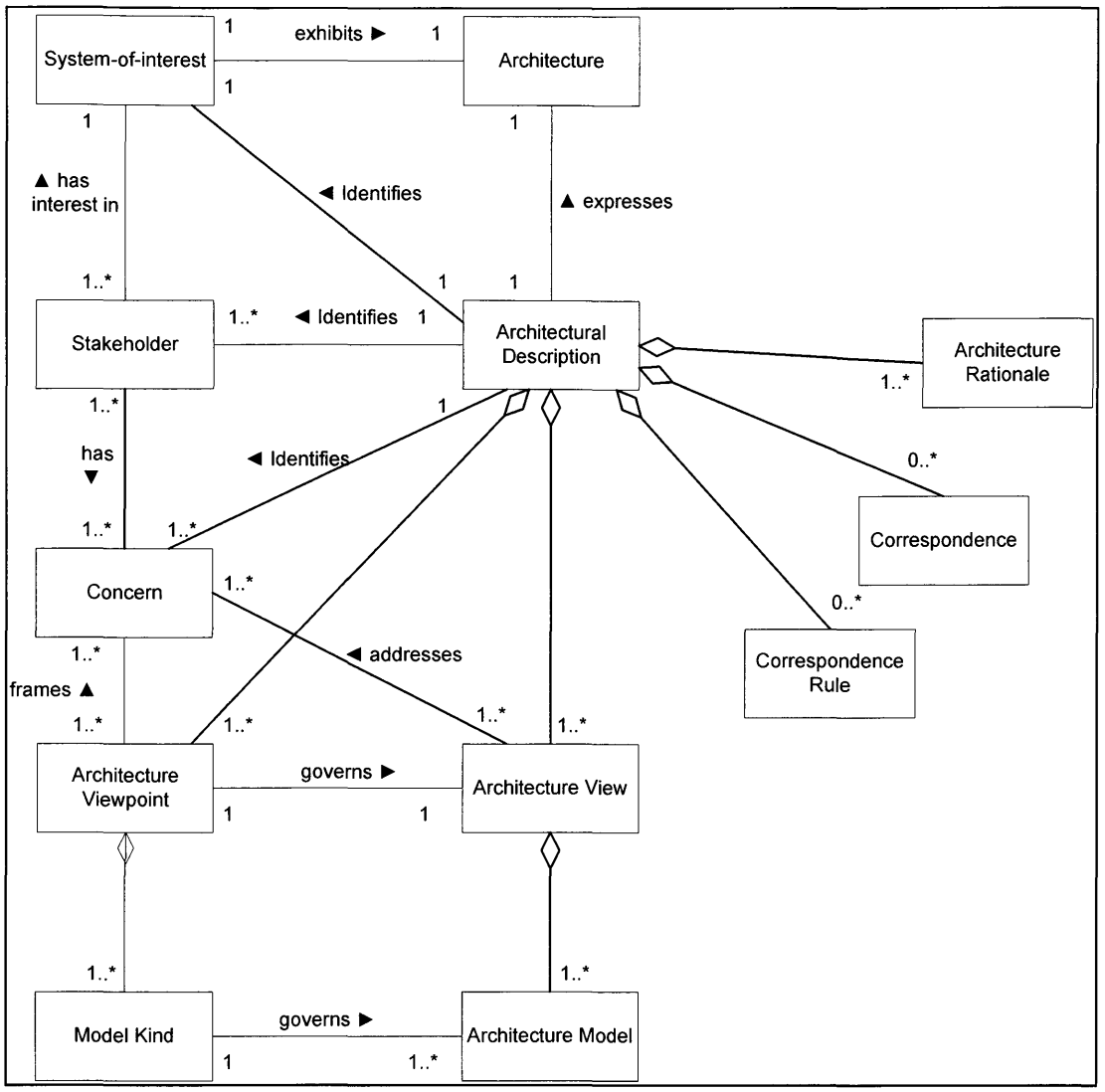


Figure 20: Metamodel of an architecture description, based on ISO/IEC JTC 1/SC 7 committee (2011, p. 5)

Table 9: Definitions of architecture description, based on ISO/IEC/IEEE 42010 standard, based on ISO/IEC JTC 1/SC 7 committee (2011)

Metamodel components	**Description and Use**
Architecture	The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. The term *architecture* conveys the essence or fundamentals of the system.
Architecture description	A *work product* used to express an architecture. Example: an *architecture description* is developed for enterprise ABC.
Architecture model	An *architecture model* is a *work product*; its subject is determined by its *model*

Metamodel components	Description and Use
	kind. Example: if an architecture is developed for the enterprise ABC and the *model kind* 'class diagram' is used, then the *architecture model* is a class diagram depicting knowledge of enterprise ABC.
Architecture view and viewpoint	*Viewpoint* refers to the conventions for expressing an *architecture* with respect to a set of *concerns*. A *viewpoint* is a way of looking at systems; a *view* is the result of applying a viewpoint to a particular system-of-interest. Each architecture *view* needs to represent the whole system from the perspective of the system *concerns* framed by its governing *viewpoint*. Example: ArchiMate (a modelling language) defines eighteen viewpoints, which results from using a matrix of six layers of concerns and 3 aspects of concerns.
Concern	Any topic of interest pertaining to the system. The *stakeholders* of a system hold these *concerns*. A concern pertains to any influence on a system in its environment including: developmental, technological, business, operational, organisational, political, economic, legal, regulatory, ecological and social influences.
Correspondence and correspondence rule	*Correspondences* are used to express relations between architecture description elements. They can for instance be used to express consistency, traceability, composition, refinement and model transformation. A correspondence rule expresses a constraint to be enforced on a correspondence. Example: Consider two viewpoints, *hardware* and *software* components. A correspondence rule relating the two is: R1: Every *software* element, *ei*, as defined by software components needs to execute on one or more platforms, *pj*, as defined by *hardware*.
Model kind	Conventions for a type of modelling. Examples: data flow diagrams, class diagrams, organisation charts.
Stakeholder	Individual, team, organisation, or classes thereof, having an interest in a system.
System and system-of-interest	Entities whose architectures are of interest. The entities encompass, but are not limited to, entities within the domains of: • *systems* (as described in [ISO/IEC 15288]) that are "man-made and may be configured with one or more of the following: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g. operator instructions), facilities, materials and naturally

Metamodel components	Description and Use
	occurring entities"; • *software products and services* (as described in [ISO/IEC 12207]; • *software-intensive systems* (as described in [IEEE Std 1471™:2000]) as "any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole" to encompass "individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, while enterprises, and other aggregations of interest".
Work product (not on Figure 20)	A *work product* is understood as an "artefact associated with the execution of a process" [ISO/IEC 15504-1:2004, 3.55].

Based on the *architecture description*, the ISO/IEC/IEEE 42010 also incorporates an *architecture framework* and *architecture description language*. Since both the *architecture framework* and *architecture description language* are used later in section 4.3.2.3, both concepts are defined according to the ISO/IEC/IEEE 42010 definition.

The ISO/IEC/IEEE 42010 standard (ISO/IEC JTC 1/SC 7 committee, 2011, p. 26) defines an *architecture framework* as a "means of defining existing and future architecture frameworks in a uniform manner to promote sharing of information about systems, architectures and techniques for architecture description" (see Figure 21). The ISO/IEC/IEEE 42010 standard states that although the current standard does not define all framework elements (e.g. prescriptions and relationships, process requirements, life cycle connections and documentation formats), the potential for standardisation exists.

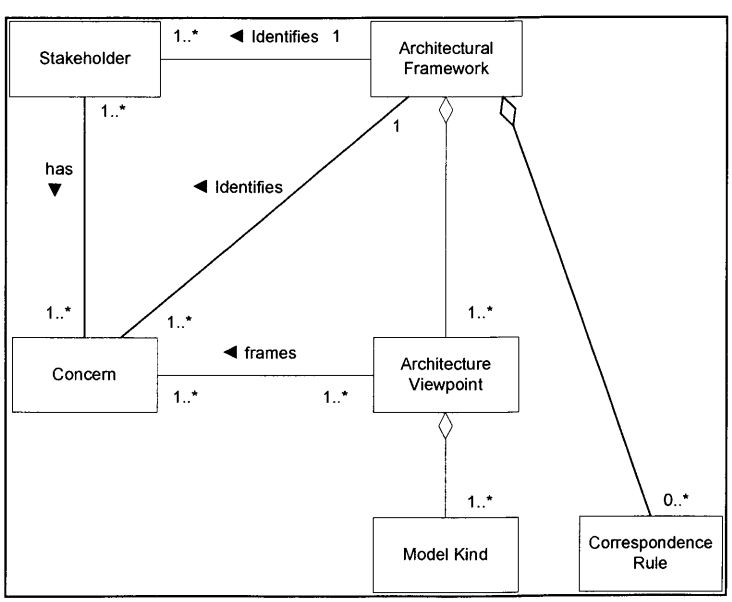


Figure 21: Metamodel of an architecture framework, based on ISO/IEC JTC 1/SC 7 committee (2011, p. 10)

The ISO/IEC/IEEE 42010 standard (ISO/IEC JTC 1/SC 7 committee, 2011, p. 26) defines an *architecture description language* (ADL) as "any language for use in an architecture description. An ADL can be used by one or more viewpoints to frame identified system concerns within an architecture description".

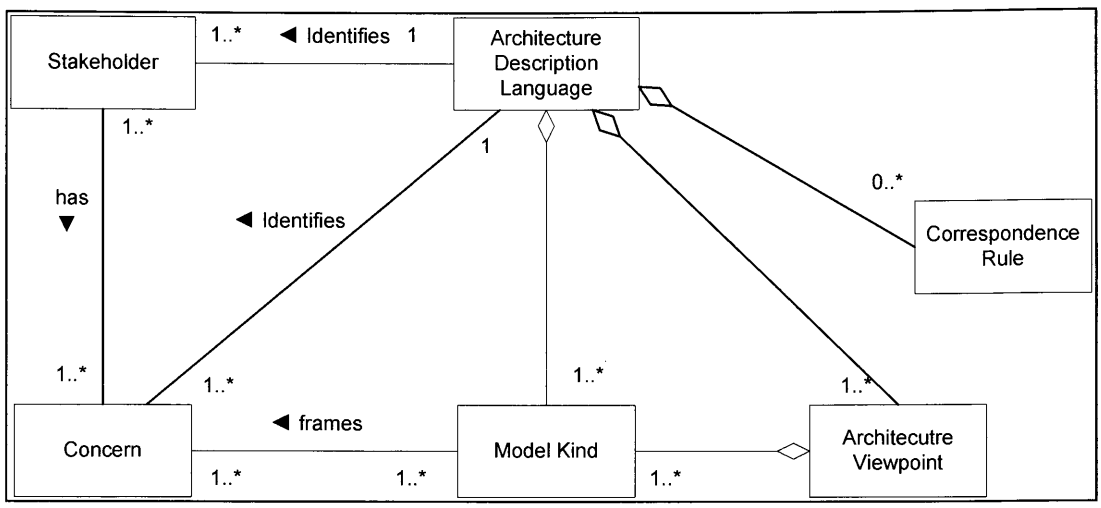


Figure 22: Metamodel of an architecture description language, based on ISO/IEC JTC 1/SC 7 committee (2011, p. 11)

This section introduced the standard for *architecture description* developed by the ISO/IEC JTC 1/SC 7 committee (2011), also using elements of the complete *architecture description* to define *architecture frameworks* and *architecture description languages.* Later, section 4.3.2.3 applies the standards provided on *architecture description*, *architecture frameworks* and *architecture description languages* during the construction of a component (*alignment mechanisms and practices*) of the Business-IT Alignment Model (BIAM).

3.3 ALIGNMENT APPROACHES

This section provides a rationale for introducing six alignment approaches that are relevant to this study. Later, the Business-IT Alignment Model (BIAM) is used as a common reference model for contextualising four of the six alignment approaches in terms of business-IT alignment.

Although a large number of theoretical EA frameworks exist, each with an aim to induce business-IT alignment at an enterprise, Sessions (2007) states that many EA frameworks/methodologies have appeared and disappeared. According to Sessions (2007), 90% of the field however, uses one of four frameworks/methodologies: the Zachman Framework, the Open Group Architecture Framework (TOGAF), the Federal Enterprise Architecture (FEA) and the Gartner Methodology. Figure 23 depicts historic events in the development of the prominent EA frameworks/methodologies. Although TOGAF is increasingly considered to be the de facto standard way of working for the development and deployment of modern IT systems in enterprises (Dietz & Hoogervorst, 2011), several other alignment approaches emerged, each providing a different perspective on alignment value-creation. A

recent study performed by OVUM (Blowers, 2012), for instance indicated that the Pragmatic EA Framework and Essential Project also increased in popularity.

This thesis acknowledges the *four prominent* alignment approaches listed by Sessions (2007) and their contribution towards to the construction of the BIAM (later in section 4.2). In addition, two less prominent alignment approaches are introduced (the *foundation for execution* approach and the *essence of operation* approach) since both are used during the construction of the PRIF (Process Reuse Identification Framework) in Part C.

The purpose of section 3.3 is merely to *introduce* the six alignment approaches to the reader. Further contextualisation and comparison between the approaches will only be possible, once a common Business-IT Alignment Model (BIAM) is used. In Chapter 5, two of the six alignment approaches are re-visited (the Zachman approach and the Open Group approach) in demonstrating business-IT contextualisation using BIAM. In Chapters 7 and 8, another two of the six alignment approaches are re-visited (the *foundation for execution* approach and *essence of operation* approach) to further demonstrate business-IT contextualisation using BIAM.

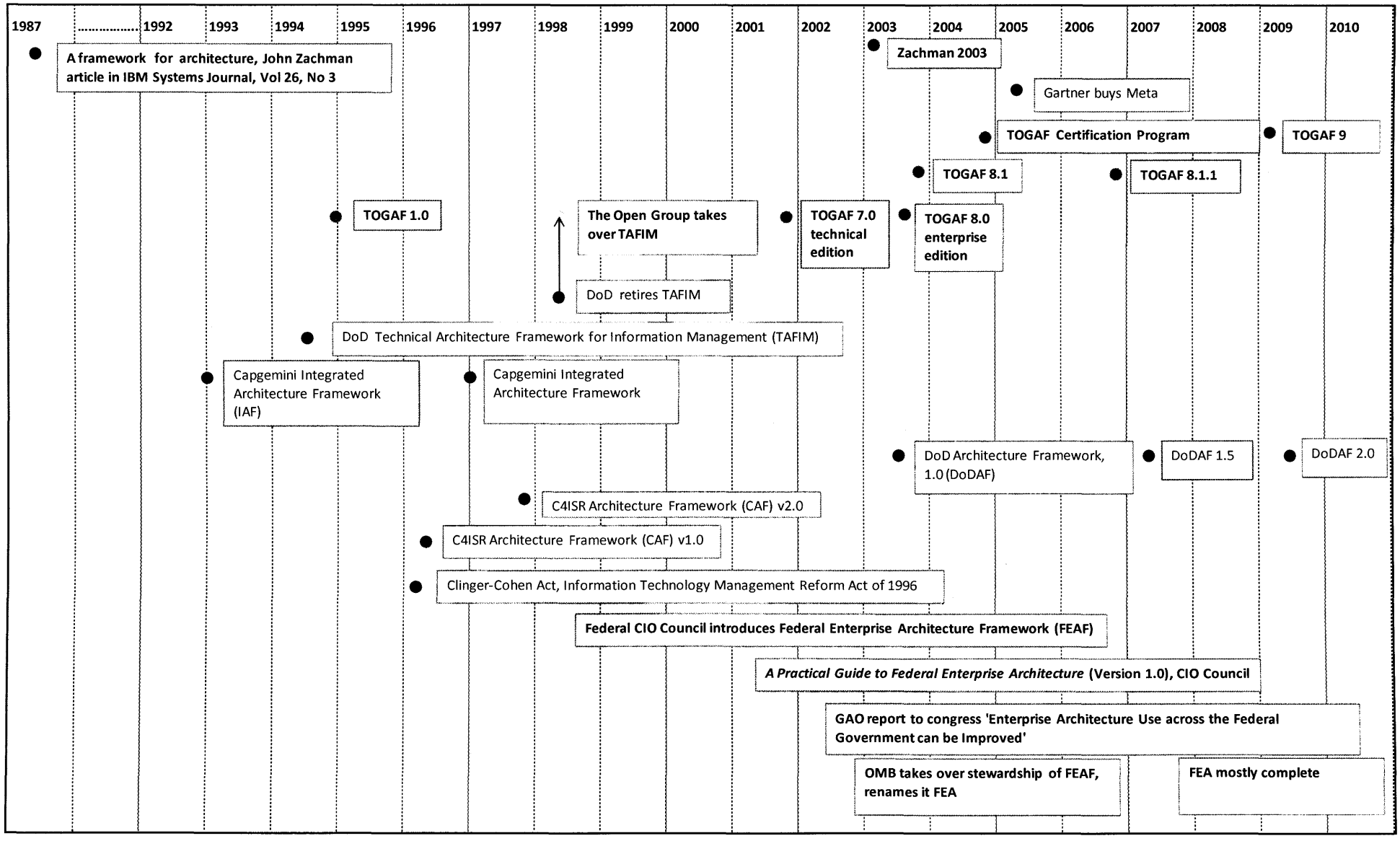


Figure 23: Timeline of enterprise architectures, based on Bespoke Systems (2012)

3.3.1 The Zachman approach

Zachman (1996), often called the farther of enterprise architecture, developed the *Zachman Framework for Enterprise Architecture* (six by six matrix presented in Figure 24) that provides a logical structure for classifying and organising the descriptive representations that are significant to the management of the enterprise and the development of enterprise systems. The *Zachman Framework for Enterprise Architecture* is an *enterprise ontology*; ontology being "a theory of the existence of a structured set of essential components of an object for which explicit expression is necessary (or even mandatory) for designing, operating and changing the object" (Zachman, 2009a, p. 15).

According to Zachman (2012) the six by six matrix depicts six communication interrogatives (what, how, when, who, where and why) as *columns* and six reification transformations (scope contexts, business concepts, system logic, technology physics, tool components, and operations instances) as *rows*. The reification process is similar to the *design process* of systems engineering, which gradually transforms system requirements to implementations (see section 3.2.2. on the *design process*).

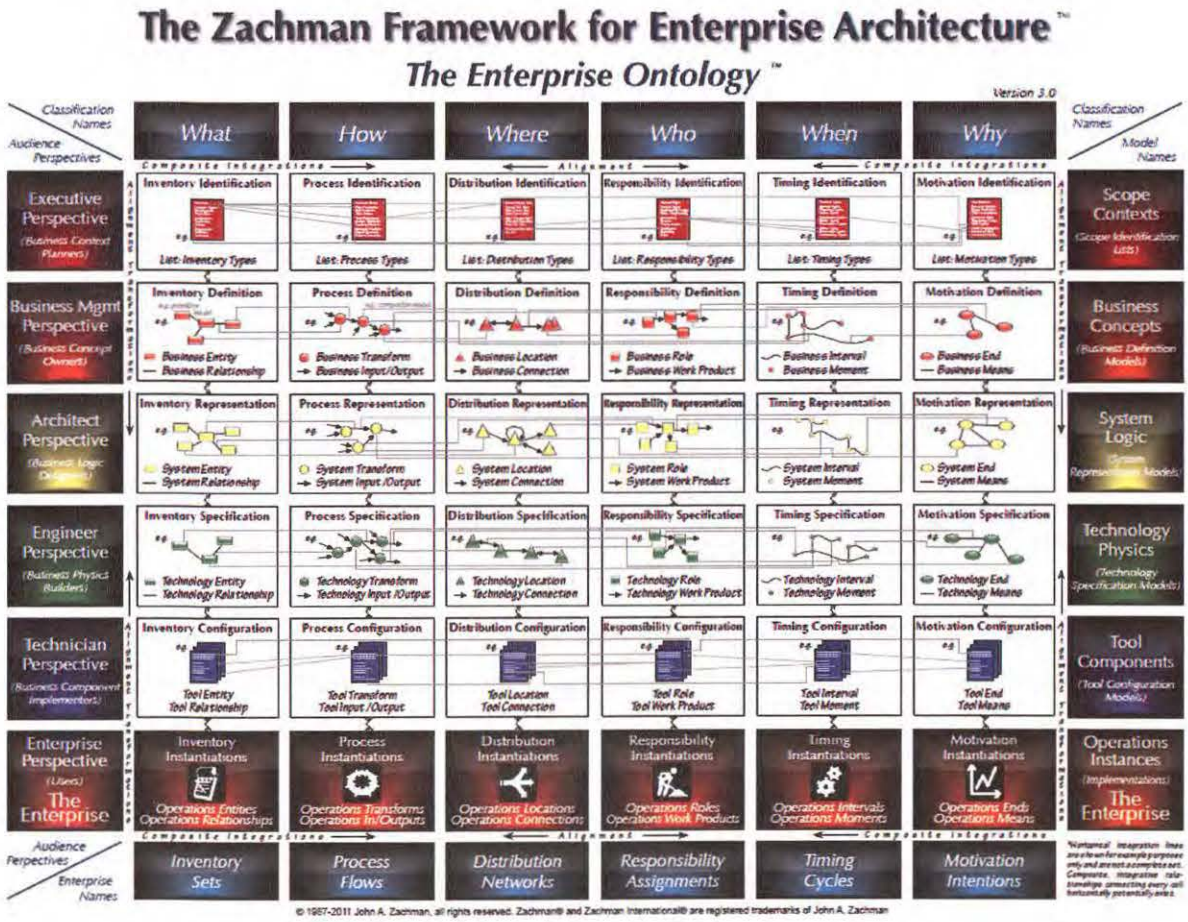


Figure 24: The Zachman Enterprise Framework, Version 3.0, a direct copy (Zachman, 2012)

The six communication interrogatives that appear as column names in Figure 24 are translated into *enterprise names* (column descriptions at the bottom of the Zachman Framework). Each communication interrogative can thus be *translated* into enterprise terminology as follows:

- What: Inventory Sets
- How: Process Flows
- Where: Distribution Networks
- Who: Responsibility Assignments
- When: Timing Cycles
- Why: Motivation Intentions

The six reification transformations that appear as rows and named by the right-hand side of Figure 24, are associated with *model names* (given in brackets next to the reification descriptions). The reification transformations concern enterprise-related *audience perspectives* (depicted as row names on the left-hand side of Figure 24). Each reification transformation thus *relate to* an audience perspective as follows:

- Scope Contexts (Scope Identification Lists): Executive Perspective (Business Context Planners)
- Business Concepts (Business Definition Models): Business Management Perspective (Business Concept Owners)
- System Logic (System Representation Models): Architect Perspective (Business Logic Designers)
- Technology Physics (Technology Specification Models): Engineer Perspective (Business Physics Builders)
- Tool Components (Tool Configuration Models): Technician Perspective (Business Component Implementers)
- Operations Instances (Implementations): Enterprise Perspective (Users)

The Zachman Framework differentiates between *abstractions* (general qualities or characteristics, apart from concrete realities, specific objects, or actual instances (Locke, 2009a)) and concrete *instantiations*. The top five rows represent *abstractions*, whereas the sixth row represents concrete *instantiations*. The intersections of the six columns with six rows produce thirty-six (36) cells, each described by its own *model*. The thirty-six models are also called *primitive models*, as each model represents the intersection of only one column with one row.

Concerning the *primitive models*, Zachman (2009a) maintains that enterprise designers should start with the explication of *primitive models* as the essential building blocks of the enterprise, to ensure re-usability of the building blocks in future enterprise designs. Once primitive building blocks have been defined via primitive models, a systematic transformation and integration of the primitive models are required. A systematic transformation of primitive models within a *single column* is called *vertical integration*, whereas the systematic integration between primitive

models within a *single row* is called *horizontal integration* (Locke, 2009a). The following two examples further demonstrate the difference between *vertical integration* and *horizontal integration*:

Figure 25 represents an example of *vertical integration* and how models (abstractions), based on entity relationship modelling notation standards (see Appendix D), within the first column (*What: inventory sets*) are gradually transformed via the reification process to transform *entities* into implemented *tables* on a database. *Vertical integration* ensures that no discontinuity exists between the various rows, i.e. ensuring consistency with requirements.

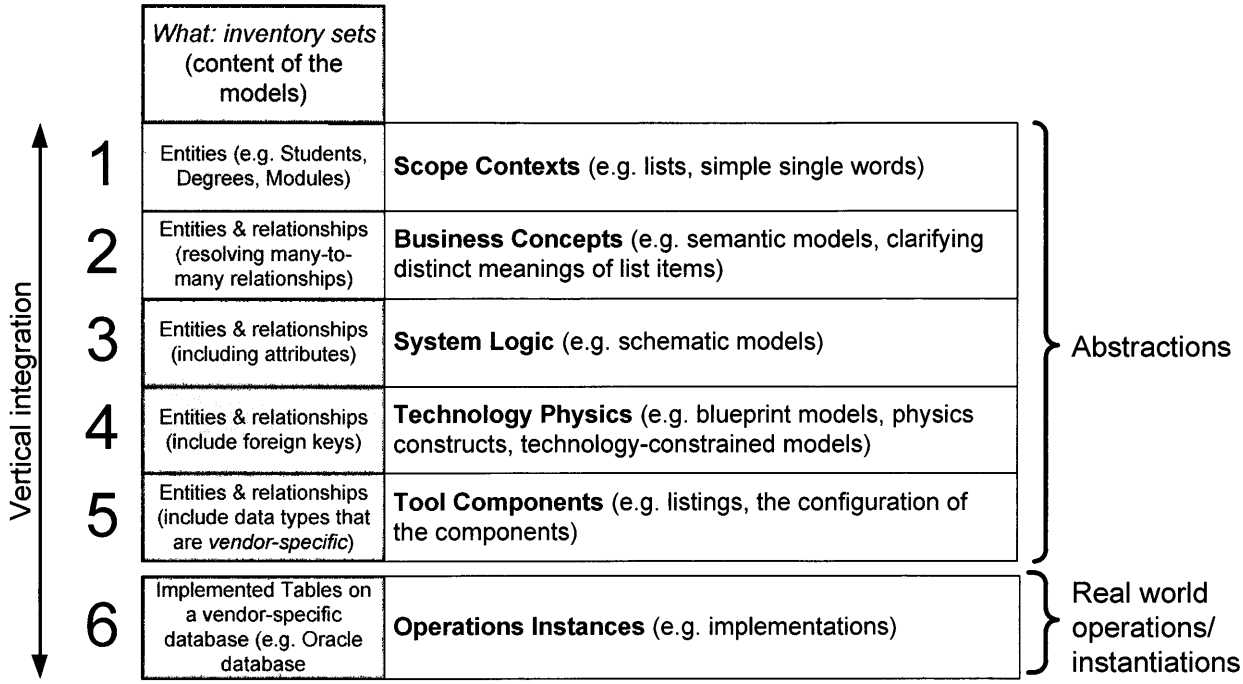


Figure 25: Example of vertical integration, based on Locke (2009a)

Figure 26 represents examples of *horizontal integration,* i.e. integrating models from different columns, but within a single row. When *primitive models* (models within separate cells) are combined, *composite models* are created, e.g. a CRUD (create, read, update, delete) matrix maps *business entities* to *business transformations*/processes, i.e. combining the first two columns (*what* and *how*) into a single model. Another example is the RACI (responsible, accountable, concerned, informed) matrix that maps *business transformations*/processes to *business roles,* i.e. combining the second and fourth columns (*how* and *who*) into a single model. *Horizontal integration* ensures that no discontinuity exists between different kinds of models from one column to the next.

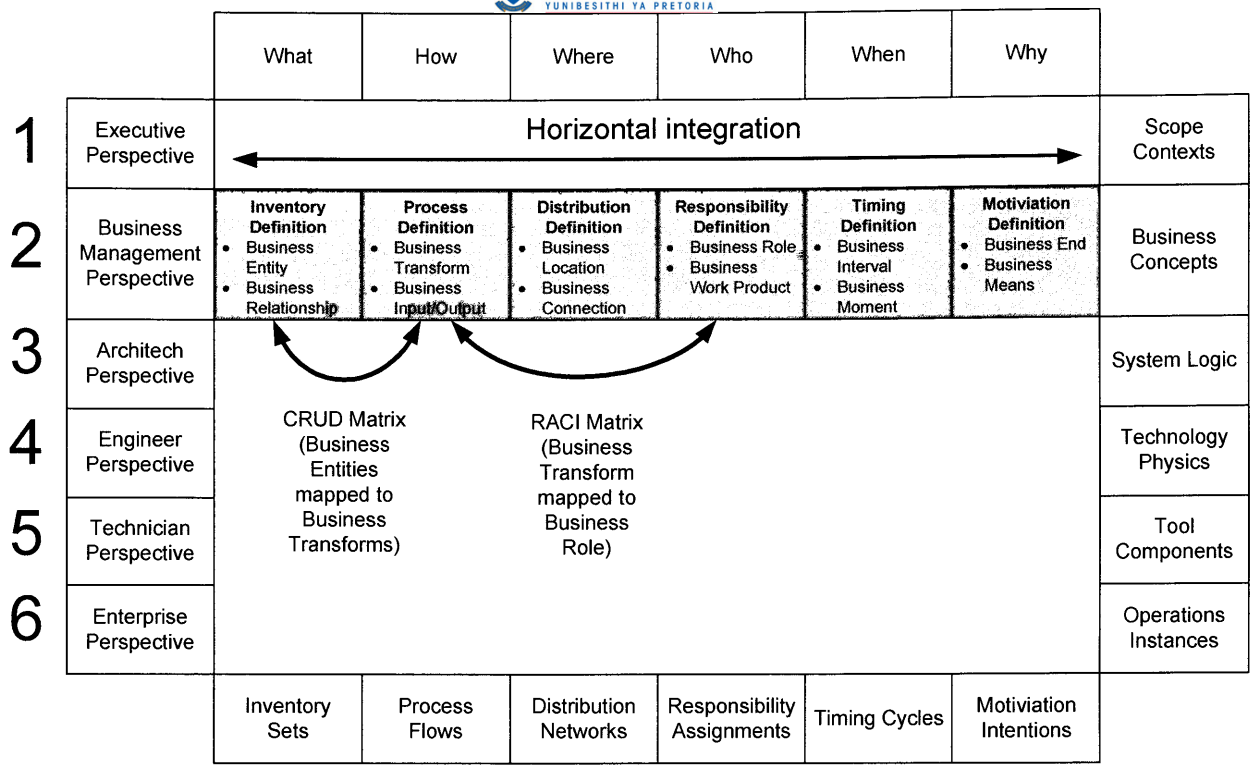


Figure 26: Example of horizontal integration, based on Locke (2009a)

Numerous developers of EA models were inspired by Zachman and applied one or more enterprise representation dimensions to describe the enterprise as a complex object. Examples include the Extended Enterprise Architecture Framework (E2AF) (Schekkerman, 2004), Integrated Architecture Famework (IAF) (Capgemini, 2007), the Federal Enterprise Architecture (FEA) (OMB, 2007b), the Gartner Enterprise Architecture Framework (GEAF) (Gartner, 2008a, 2008b) and the EA3 Cube (Bernard, 2005).

The Zachman approach is primarily concerned with creating *consistency and alignment* across the individual rows and columns on the Zachman Framework. Although the Zachman approach was only introduced in this section, section 5.2 re-visits the Zachman approach, but within the context of the Business-IT Alignment Model (BIAM), which will be defined in section 4.3.

3.3.2 The Open Group approach

TOGAF (The Open Group Architecture Framework), owned by the Open Group, became best known for its Architecture Development Method (ADM), which is an architectural process/methodology, rather than an architectural framework (Giachetti, 2010). The ADM consists of ten phases (see Figure 27), including:

1. *Preliminary.* This phase defines the *capabilities for doing architecture work*, i.e. defining the "where, what, why, who and how we do architecture". Main aspects include: defining the scope of the enterprise concerned with architecture work; key drivers and elements in the enterprise context; requirements for architecture work; architecture principles, frameworks to be used; the relationships between management frameworks; and an evaluation of enterprise architecture maturity.

2. *Phase A. Architecture vision.* This phase defines the *scope of the architecture* effort and the constraints that must be dealt with. Main aspects include: gaining recognition, endorsement and commitment from management; identification of relevant stakeholders, their concerns and objectives; definition of key business requirements and constraints that must be addressed; formulation of the value proposition/offering that demonstrates a response to the requirements and constraints; articulation of a comprehensive plan for doing architecture work; securing formal approval; and understanding the impact on other enterprise architecture development projects.
3. *Phase B. Business architecture*. This phase *defines the baseline and target business architectures*, which is a prerequisite for architecture work in any other domain (data, application and technology). Main aspects include: developing the baseline and target business architectures; analysing the gaps between the baseline and target architectures; developing architecture viewpoints for specific stakeholders to demonstrate that stakeholder concerns are addressed; selecting and using relevant tools and techniques for constructing the required viewpoints.
4. *Phase C. Information systems architecture.* This phase defines the *target data and/or application architectures* that would support the target business architecture. Main aspects include: developing baseline and target data and/or application architectures; and analysing gaps between the baseline and target architectures.
5. *Phase D. Technology architecture.* This phase maps the data and/or application components (defined in Phase C) to a set of technology components, representing required software and hardware components.
6. *Phase E. Opportunities and solutions.* This phase provides a logical grouping of IT activities into *project work packages* within the IT portfolio and other portfolios that are dependent upon IT. Main aspects include: assessing the feasibility to implement changes at the enterprise; deriving transition architectures that deliver continuous and incremental business value; and gaining consensus on an implementation/migration strategy.
7. *Phase F. Migration planning.* This phase creates a *viable implementation/migration plan* in co-operation with the portfolio and project managers. Main aspects include: assessing dependencies, costs and benefits of the various migration projects and their prioritisation; negotiating contracts for implementation projects; and monitoring the detailed implementation/migration projects in accordance with the transition architectures defined in Phase E.
8. *Phase G. Implementation governance*. This phase *governs and manages the contract* for implementing and deploying the solution(s). Main aspects include: performing appropriate governance functions while the solution is implemented and deployed; ensuring conformance to pre-defined architecture; ensuring conformance of the deployed solution with the target architecture; and mobilising supporting operations to underpin the future working lifetime of the deployed solution.
9. *Phase H. Architecture change management.* This phase *manages changes to the architecture* in a consistent way. Main aspects include: establishing an architecture

change management process for the new enterprise architecture baseline; supporting the implemented enterprise architecture as a dynamic architecture; and assessing the performance of the new architecture and make recommendations for change.

10. *Requirements management*. This phase interacts with phases A to H and denotes the dynamic process of identifying, storing and *managing the supply of enterprise architecture change requirements* (The Open Group, 2009).

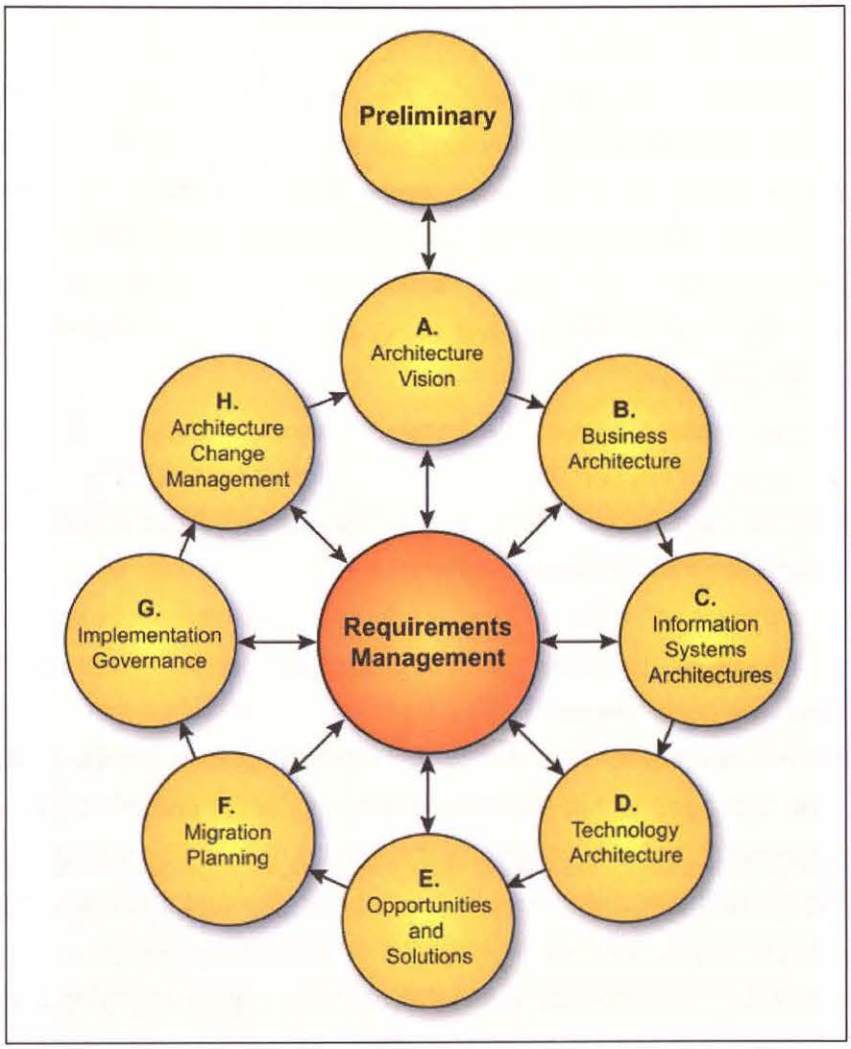


Figure 27: TOGAF ADM Cycle, a direct copy (The Open Group, 2009, p. 54)

Published in February 2009, TOGAF 9.0 incorporated major document structural changes compared to TOGAF 8.1.1. The new structure highlight seven main parts and their relationships (see Figure 28):

- *Part I: Introduction* (not shown on Figure 28). High-level introduction to key concepts, definitions of terms, release notes, and the TOGAF approach in general.
- *Part II: Architecture development method (ADM).* The step-by-step approach to develop an enterprise architecture.
- *Part III: ADM guidelines and techniques.* The set of guidelines and techniques that are available for use when using TOGAF and the TOGAF ADM.

- *Part IV: Architecture content framework.* A description of the TOGAF content framework, which includes a structured model for architectural artefacts. The part also include re-usable architecture building blocks and an overview of typical architecture deliverables.
- *Part V: Enterprise continuum & tools.* Appropriate taxonomies and tools for categorising the outputs of architecture activity within an enterprise.
- *Part VI: TOGAF reference models.* A selection of reference models, including the TOGAF foundation architecture, and the integrated information infrastructure reference model (III-RM).
- *Part VII: Architecture capability framework.* Content about the organisation, processes, skills, roles and responsibilities required for establishing and operating an architecture function within an enterprise.

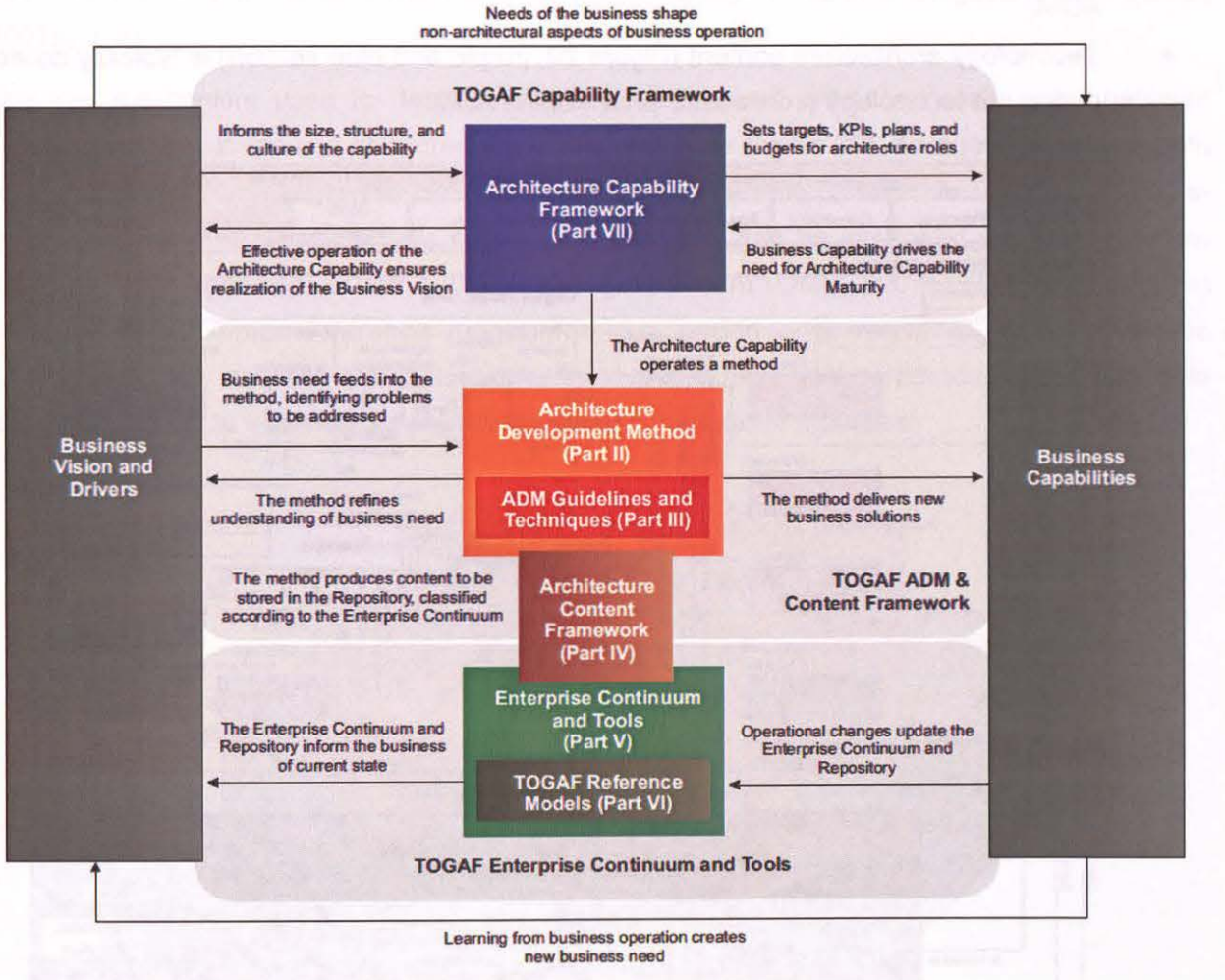


Figure 28: Structure of the TOGAF document, a direct copy (The Open Group, 2009, p. 4)

The *architecture development method* (ADM) (Figure 28, Part II) is used in combination with *ADM guidelines and techniques* (Figure 28, Part III) and the *architecture content framework* (Figure 28, Part IV) in delivering new business solutions. The *architecture content framework* "provides a structural model for architectural content" and may also be substituted with other frameworks, such as the Zachman Framework (The Open Group, 2009, p. 361). Contrary to the intention of the Zachman Framework to create an enterprise ontology, the *architecture content*

framework defines a set of entities to enable consistent, complete and traceable *capturing of architectural concepts*. In fostering its use, in combination with the ADM, the *architecture content framework* is structured to highlight correlation with the ADM phases. A detailed representation of the architecture content framework, called the *content metamodel* (see Figure 29) demonstrates the correlation between content and ADM phases:

- *Architecture principles, vision, requirements, and roadmap* content (Figure 29, pink section) is typically collected in the preliminary and architecture vision phases of the ADM.
- *Business architecture* content (Figure 29, yellow section) is typically collected during the business architecture phase of the ADM.
- *Data architecture* and *application architecture* content (Figure 29, purple and light-green sections) is typically collected during the information systems architecture phase of the ADM.
- *Technology architecture* content (Figure 29, purple and blue section) is typically collected during the technology architecture phase of the ADM.

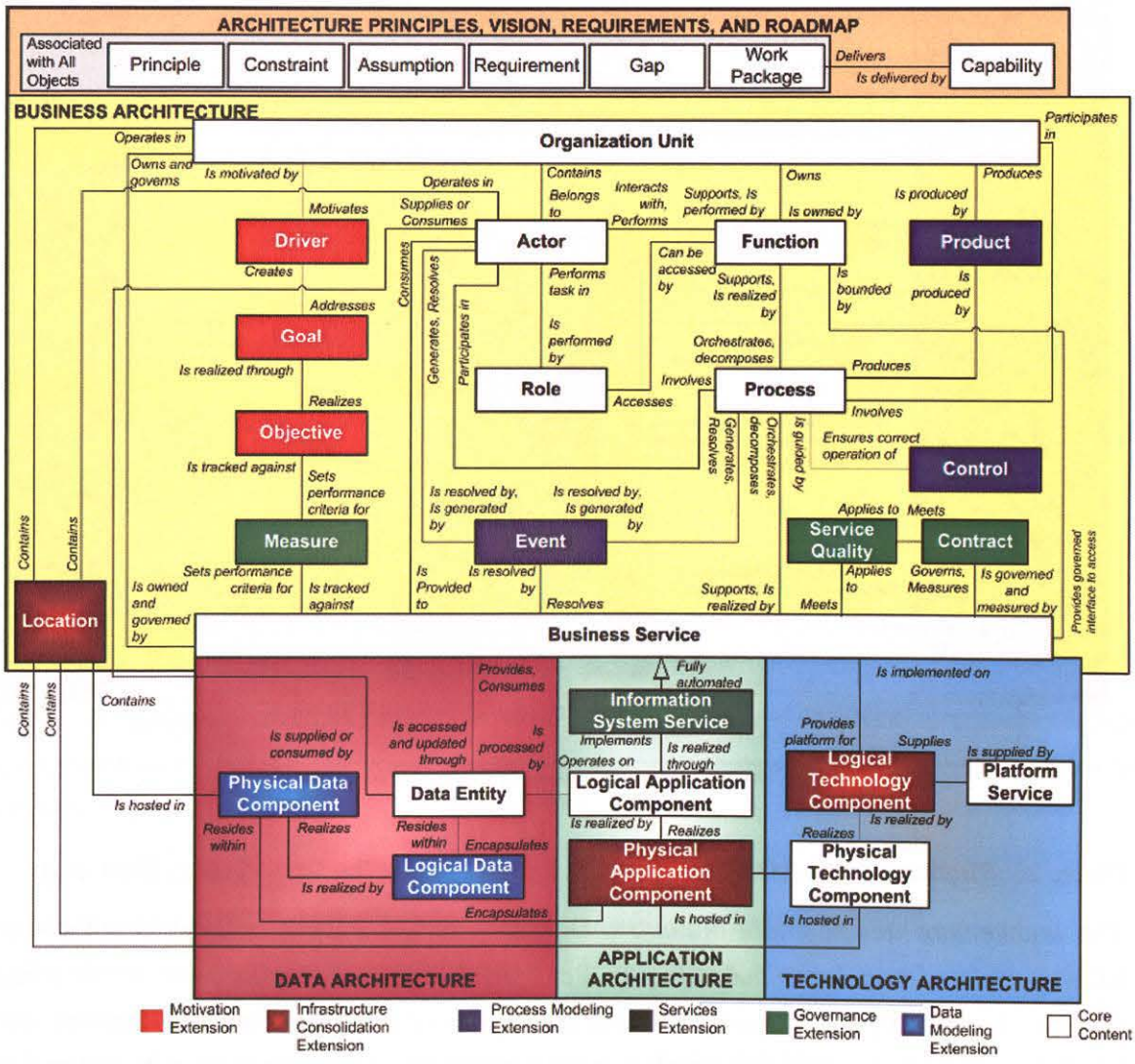


Figure 29: Relationships between entities in the content metamodel, a direct copy (The Open Group, 2009, p. 379)

The Open Group approach as presented in TOGAF is primarily concerned with creating an *alignment methodology* for designing/changing the enterprise. In this section, the Open Group approach was introduced to the reader, but will be re-visited in section 5.3, after defining the Business-IT Alignment Model (BIAM) in section 4.3.

3.3.3 The OMB approach

The Federal Enterprise Architecture (FEA) evolved from the Federal Enterprise Architecture Framework (FEAF) as the latest attempt made by the U.S. government to unite their agencies and functions under a common EA (OMB, 2007b). The FEA Program Management Office (FEAPMO) maintains that FEA provides the Office of Management and Budget (OMB) and federal agencies with "a common language and framework to describe and analyse IT investments, enhance collaboration and ultimately transform the federal government" (OMB, 2007b, p. 4).

The key mechanism used to describe the architecture and enhance collaboration between federal agencies, is the use of segment architectures (see Figure 30). An agency contains both core and mission area segments and business service segments. Enterprise services are cross-cutting services that span multiple segments. Segments can be leveraged within an agency, across several agencies, or the entire federal government (OMB, 2007b). The OMB (2007a) also provides common reference models for (e.g. performance reference model, business reference model, service component reference model, technical reference model and data reference model) to enhance collaboration between the federal agencies.

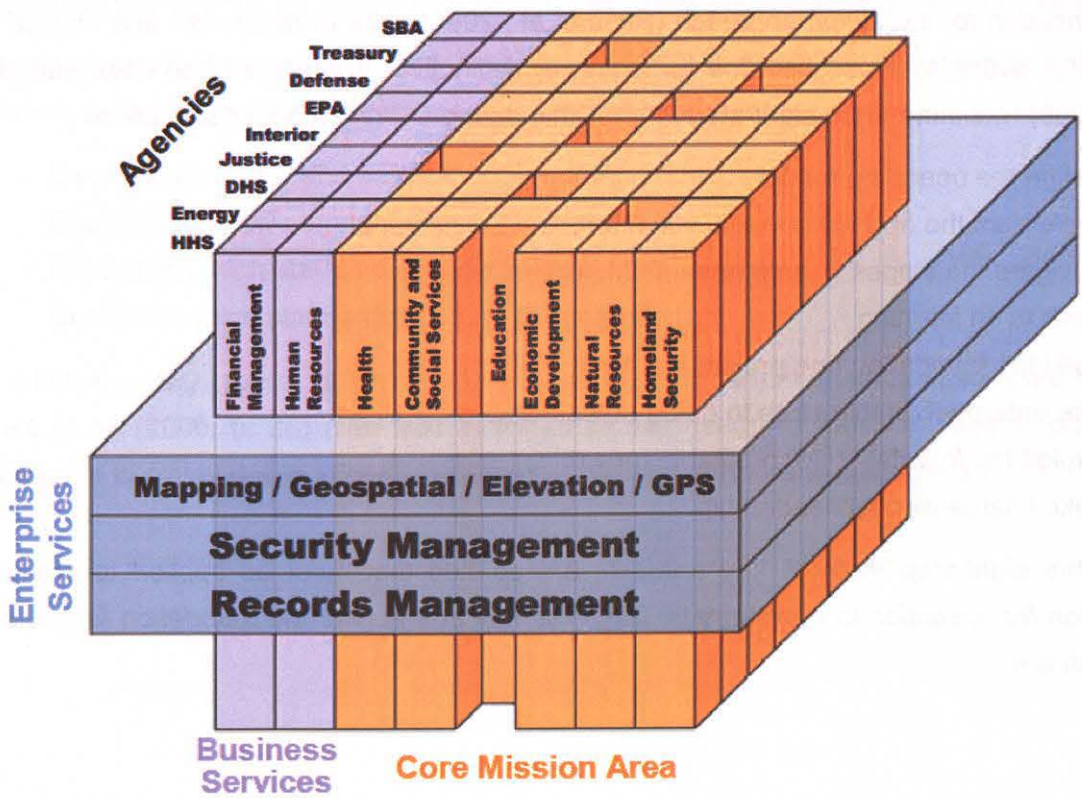


Figure 30: Segments and services, a direct copy (OMB, 2007b, p. 3)

The OMB approach is primarily concerned with *alignment/collaboration* between different federal agencies. Due to its restricted use by US government only (Sessions, 2007), this thesis does *not* provide an extensive business-IT alignment contextualisation of the OMB approach. However, the OMB approach contributed towards the development of the *alignment mechanisms and practices* of the BIAM, as discussed in section 4.3.

3.3.4 The Gartner approach

Gartner, an IT research and consulting enterprise, developed a Gartner Enterprise Architecture Method (GEAM) that consists of a Gartner EA process model and a Gartner EA framework. The Gartner EA process model represents key characteristics and a synthesis of best practices for developing and maintaining an EA, while the Gartner EA framework articulates the relationships between enterprise business architecture (EBA), enterprise information architecture (EIA), enterprise technical architecture (ETA), and their synthesis with enterprise solutions architecture (ESA) (Bittler & Kreizman, 2005).

The Gartner approach is primarily concerned with creating an *alignment methodology* for designing/changing the enterprise. Due to the restricted access to Gartner publications and copyright on Gartner materials, this thesis does *not* provide an extensive business-IT alignment contextualisation of the Gartner approach later in this thesis. Still, the Gartner approach contributed towards the development of the *alignment mechanisms and practices* of the BIAM, as discussed in section 4.3.

3.3.5 The foundation for execution approach

The *foundation for execution* approach (Ross et al., 2006) aims to rationalise and digitise both the routine, everyday processes and competitively distinctive capabilities of an enterprise. Ross et al. (2006) recommend an eight-step *method* in creating a *foundation for execution*:

1. Define the operating model
2. Implement the operating model via enterprise architecture
3. Navigate the stages of enterprise architecture maturity
4. Cash in on learning
5. Build the foundation one project at a time
6. Use enterprise architecture to guide outsourcing
7. Exploit the foundation for profitable growth
8. Take charge through leadership

During the eight-step *method*, key artefacts are defined that must be applied to create the *foundation for execution* in a systematic way. The key artefacts of the *foundation for execution* approach are:

1. The operating model
2. The core diagram
3. Four stages of architecture maturity
4. The IT-engagement model

The key artefacts are discussed subsequently.

3.3.5.1 The operating model

The *operating model* (OM) is used to establish the "necessary level of business process integration and standardisation for delivering goods and services to customers" (Ross et al., 2006, p. 44) and has two main dimensions: (1) business process standardisation, and (2) business process integration. The two dimensions require separate decisions, due to different *end results*.

Standardisation of business processes means defining how a process will be executed regardless of who or where it is executed. The *end result* of process standardisation, is a reduction in variability and therefore dramatic increases in throughput and efficiency. However, process standardisation has a cost, since standardisation limits local innovation and may require expensive rip-and-replace efforts to replace legacy systems with the new standard.

Integration of business processes, links business units via shared data. The *end result* of process integration is an increase in efficiency, coordination, transparency and agility. Integration speeds up the flow of information and transactions throughout an enterprise. Yet, integration may be difficult and time-consuming, since enterprises need to develop standard definitions and formats for data that will be shared across business units and functions.

Based on the two main dimensions, Ross et al. (2006) defined four general types of operating models, based on the levels of standardisation and integration:

1. Diversification (low standardisation, low integration)
2. Coordination (low standardisation, high integration)
3. Replication (high standardisation, low integration)
4. Unification (high standardisation, high integration)

In addition, every type of operating model also exhibits certain characteristics (see Figure 31). Ross et al. (2006, p. 28) aver that every enterprise needs to "position itself in one of these quadrants to clarify how it *intends* to deliver goods and services to customers".

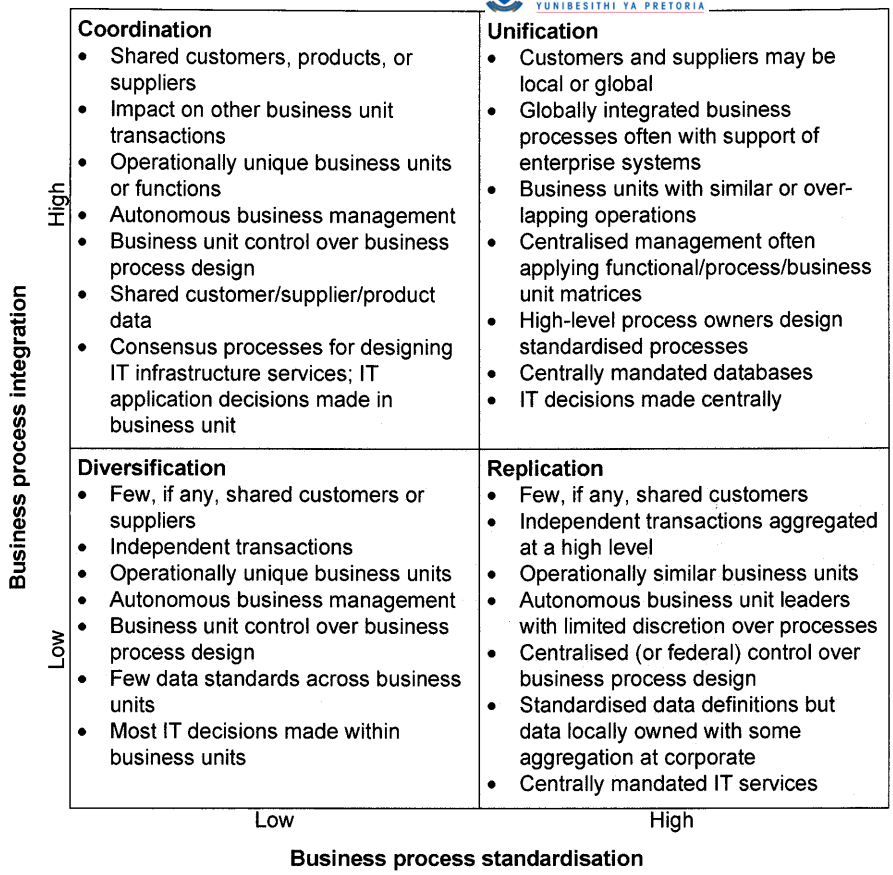


Figure 31: Characteristics of four operating models, based on Ross et al. (2006, p. 29)

Not only does an OM decision represent a general vision of a how an enterprise will enable strategies, but each operating model presents different *opportunities and challenges for growth*. For example, *process standardisation*, evident in the replication OM (see Figure 32, *Replication*), enables *organic* growth by expanding into new markets, replicating standard practices and innovations in new markets. However, growth via *acquisition* requires rip-and-replace of infrastructure to leverage the existing foundation.

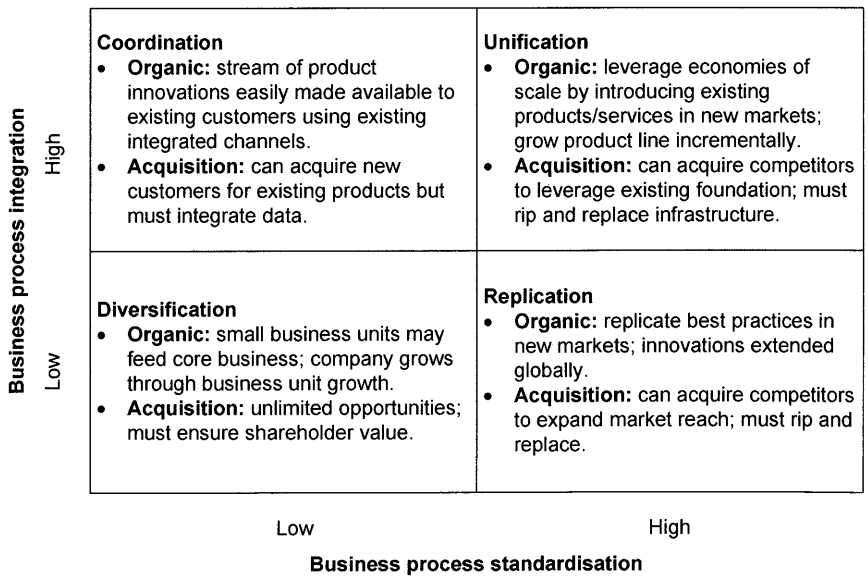


Figure 32: Different operating models position enterprises for different types of growth, based on Ross et al. (2006, p. 39)

Another key artefact that is derived from the OM, is called the core diagram, which will be discussed next.

3.3.5.2 The core diagram

The *core diagram* translates OM decisions into a visual representation of the processes, data and technologies that need to be shared across the enterprise. Ross et al. (2006) define four common elements in a core diagram:

- *Core business processes.* The stable set of enterprise processes required to execute its operating model and respond to market opportunities.
- *Shared data driving the core processes.* Customer data shared across product lines or business units of an enterprise.
- *Key linking and automation technologies.* Technologies that enable integration of applications (middleware) to shared data, major software packages such as ERP systems, portals providing standardised access to systems and data, and electronic interfaces to key stakeholder groups.
- *Key customers.* Major customer groups served by the *foundation for execution*.

The elements highlighted in a core diagram depend on the type of OM. Each OM consequently requires a different process and template for its design. As an example, the unification OM requires a *process* (see Figure 33, top half) to identify key customers to be served, key processes to be standardised and integrated, and shared data to integrate processes and serve customers. Finally, key technologies may also be added (optionally) to automate or link processes. The *template* for a unification OM (see Figure 33, bottom half) reflects the highly standardised and integrated processes and shared data that make products and services available to customers. Linking and automating technologies are only shown if they are signification in terms of management vision (Ross et al., 2006).

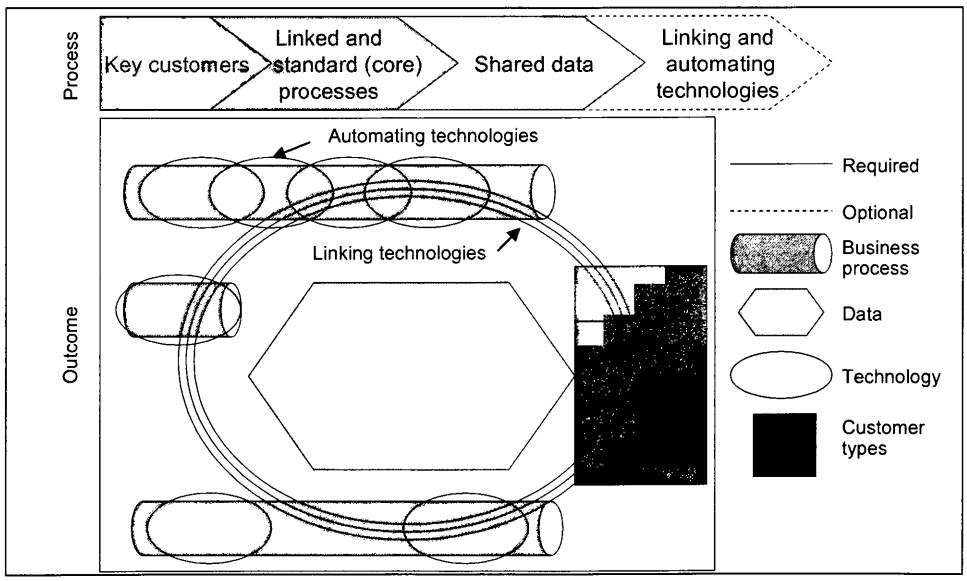


Figure 33: Core diagram process and template for a unification OM, based on Ross et al. (2006, p. 54)

The core diagram provides a graphical representation of enterprise vision in terms of standardisation requirements. In pursuit of this vision, enterprises gradually advance through four stages of architecture maturity. The four stages of architecture maturity are discussed next.

3.3.5.3 Four stages of architecture maturity

The *four stages of architecture maturity* refer to the consistent pattern used by enterprises for building their *foundation for execution*. When enterprises advance through the stages of architecture maturity, they realise benefits ranging from reduced IT operating costs to greater strategic agility (Ross et al., 2006).The four stages are:

1. *Business silos architecture*, where enterprises maximise individual business unit needs or functional needs.
2. *Standardised technology architecture*, i.e. gaining IT efficiencies through technology standardisation and increased centralisation of technology management.
3. *Optimised core architecture*, i.e. providing enterprise-wide data and process standardisation, appropriate for the OM.
4. *Business modularity architecture*, where enterprises manage and reuse loosely coupled IT-enabled business process components to preserve global standards while enabling local differences.

Since each stage requires enterprise changes, enterprises need to acquire learning in several areas (e.g. business objectives, funding priorities, and management responsibilities), whereas learning objectives within the areas differ from one stage to the next.

When an enterprise advances through different stages of architecture maturity, governance mechanisms assist with the process of transformation. The IT engagement model portrays a set of required governance mechanisms and will be discussed next.

3.3.5.4 The IT engagement model

An *IT engagement model* (see Figure 34) is used to portray the set of governance mechanisms that will be required by an enterprise to transform itself into a future design. The *IT engagement model* contains three main ingredients:

1. Company-wide IT governance, defined as the "decision rights and accountability framework to encourage desirable behaviour in using IT" (Ross et al., 2006, p. 119).
2. Project management, which requires a formalised project methodology with clear deliverables and checkpoints.
3. Linking mechanisms, which incorporates processes and decision-making bodies that need to align incentives and connect the project-level activities to the companywide IT governance.

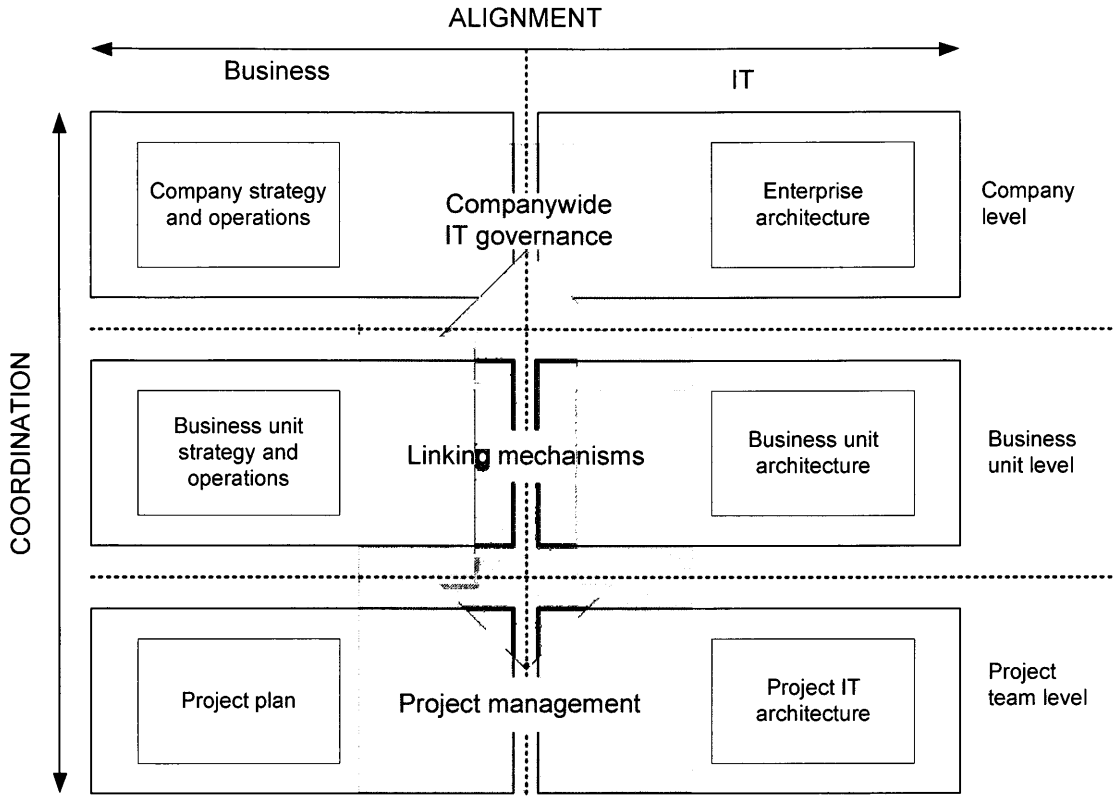


Figure 34: The IT engagement model, based on Ross et al. (2006, p. 120)

Figure 34 presents the three main ingredients of the IT engagement model, as well as the *coordination* and *alignment* between different stakeholder groups. Whereas *coordination* is required between different enterprise levels (company, business unit, and project team levels), *alignment* is required between two perspectives, i.e. business and IT.

The *foundation for execution* approach is primarily concerned with creating an *alignment* vision embedded in the required operating model. Although this section introduced the *foundation for execution* approach to the reader, one can only compare the *foundation for execution* approach to other alignment approaches if a common business-IT alignment model exists. Section 7.2 therefore re-visits the *foundation for execution* approach after defining a Business-IT Alignment Model (BIAM) in section 4.3.

3.3.6 The essence of operation approach

Similar to Zachman (see section 3.3.1), Dietz (2006) in his *essence of operation approach*, also applies the generic *system design process* to demonstrate alignment between requirements and implementations. Similar to Zachman (see section 3.3.1), his objective is to create an enterprise ontology, but ontology in this case defined as the "*essence of construction and operation*" of an enterprise (Dietz, 2006, p. 8). Since Dietz maintains that the organisation of an enterprise is a *social system*, and the active elements of a social system are human beings who operate on and communicate about things in the object world, the *essence of construction and operation* need to contain the communicative aspects of the enterprise. The *essence of operation approach* thus draws on the theory of communicative action of Habermas (1981) to provide an

explanation of how communication works, and how communication is used to perform coordination acts and production acts in an enterprise.

This section first differentiates between *coordination acts and production acts* and the distinct human abilities required to communicate. The distinct human abilities are then used to discuss the *organisation of the enterprise* as layered system. With reference to three different layers, the section concludes with an introduction to the *ontological aspect models* that are required to represent the ontological aspect system of the enterprise, and a *methodology* to develop the ontological aspect models.

3.3.6.1 Coordination acts vs. production acts

Humans perform two kinds of acts within their position of authority and responsibility*: production acts* and *coordination acts*. *Production acts* render goods and/or services that are delivered to the environment of the enterprise, and may be either material (e.g. manufacture product) or immaterial (e.g. decision to grant an insurance claim). *Coordination acts* however, ensure that humans enter into and comply with commitments towards each other regarding the performance of a production act. In performing coordination acts and production acts, humans apply three kinds of communicative acts that correspond with their human abilities (Figure 35):

- The *forma* ability (meaning 'form') concerns the form aspects of communication and information, and requires *coordination acts* (e.g. uttering information or perceiving information) to perform *production acts* (e.g. transmitting or storing data).
- The *informa* ability ('what is in the form') concerns the content aspects of communication and information, and requires *coordination acts* (e.g. expressing thought or educing thought) to perform *production acts* (e.g. deducing or reasoning).
- The *performa* ability ('through the form') concerns creation/design of new, original things linked to communication, and requires *coordination acts* (e.g. exposing or evoking commitment) to perform *production acts* (e.g. deciding or judging (Dietz, 2006)).

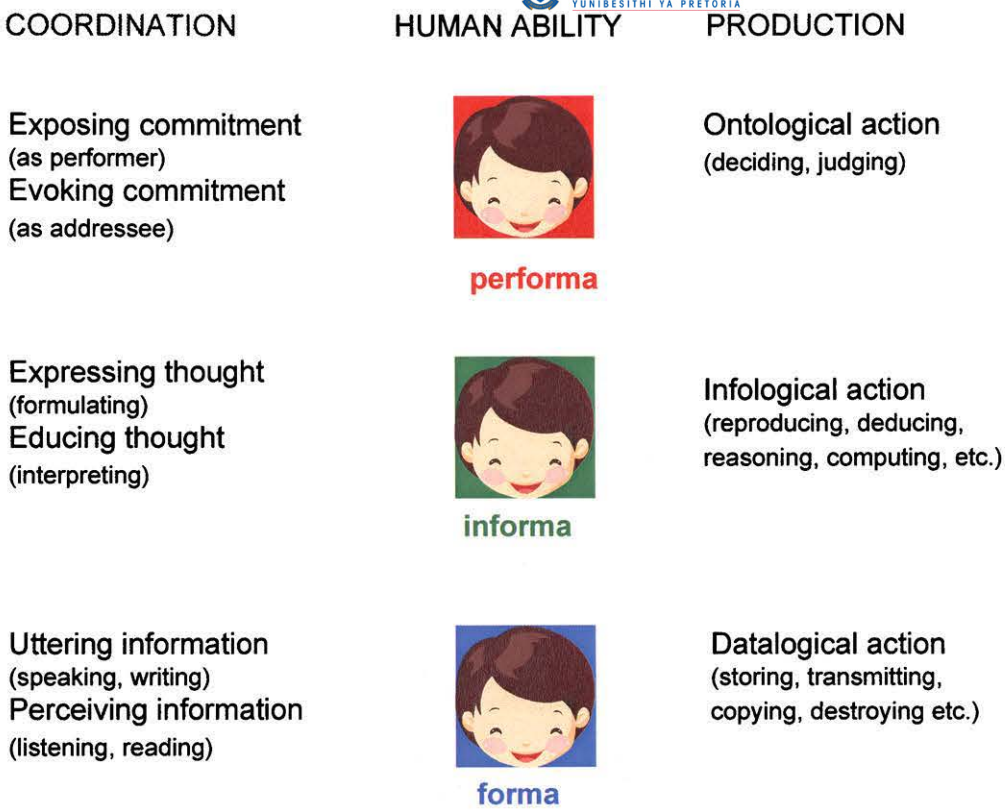


Figure 35: Three kinds of communicative acts, based on Dietz (2006)

The distinct human abilities (Figure 35, *Performa, Informa* and *Forma* abilities) provide the opportunity to create three abstraction layers in representing the *organisation* of the enterprise, which is discussed in the next section.

3.3.6.2 The organisation of the enterprise

The previous section indicated that the *performa* abilities are associated with ontological production acts (Figure 35, *Ontological action*), whereas *informa* abilities are associated with infological production acts (Figure 35, *Infological action*), and *forma* abilities are associated with datalogical production acts (Figure 35, *Datalogical action*).

Using the three distinct human abilities, Dietz (2006) thus represents the *organisation* of the enterprise as a heterogeneous *social* system that consists of a layered integration of three homogeneous social systems: the ontological, infological and datalogical aspect systems (see Figure 36). The three aspect systems are of the same category, i.e. *social* systems, but differ in terms of their *kind of production*: the ontological aspect system produces ontological acts, such as decisions and judgements; the infological aspect system produces infological acts, such as reproducing, deducing, reasoning and computing; whereas the datalogical aspect system produces datalogical acts, such as storing, transmitting, copying and destroying.

The distinction between different aspect systems enables one to focus on the essential/ontological aspect system in describing the essential operation of an *organisation*, irrespective of its realisation (i.e. integration with the other two aspect systems) or implementations (using technology to make the organisation operational). The three aspect

systems thus only represent the *organisation* of the enterprise system and *exclude* the implementation (incorporating technology) of the enterprise system.

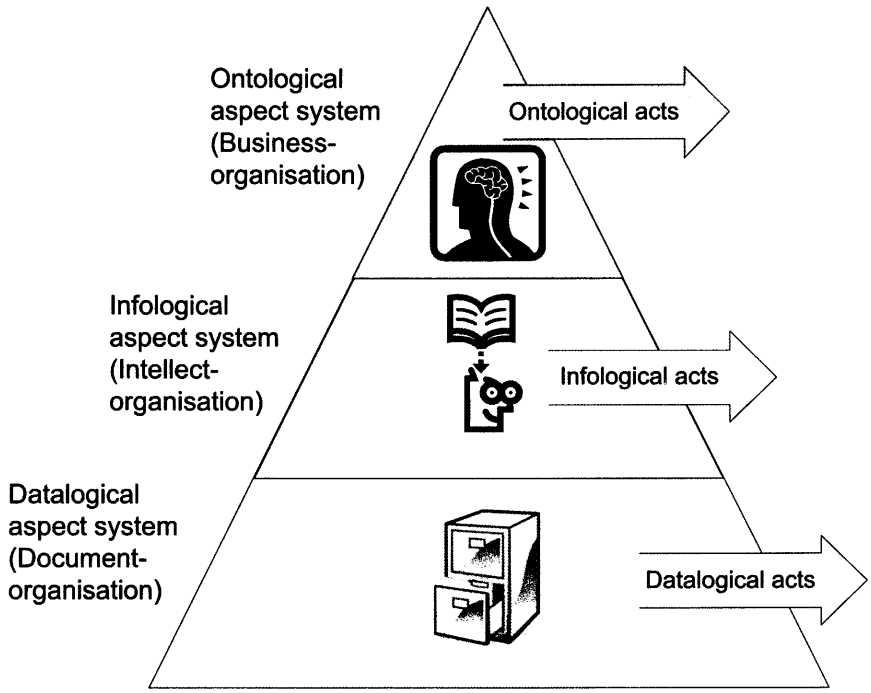


Figure 36: The three aspect systems, based on Dietz (2006)

Dietz (2006) focuses on the essential/ontological aspect system (Figure 36, *Ontological aspect system*) using ontological aspect models (OAMs) to represent the ontological knowledge of an enterprise. The next section introduces the OAMs.

3.3.6.3 The ontological aspect models

The main contribution of the *essence of operation approach* is the ontological aspect models (OAMs) that convey the ontological knowledge of *enterprise construction*. Figure 37 illustrates the three aspect systems and the set of OAMs to represent the ontological knowledge of an enterprise. The OAMs are *white box* models that provide a *constructional notion* of the *ontological aspect* system (see section 3.2.1.1), rather than *black box* models that convey the function or behaviour of a system. Dietz (2006, p. 82) equates the set of OAMs to the skeleton of the enterprise, which provides the "rigorous basis for effective and elegant movements but does not determine the external beauty of the 'body'". Many other human abilities are thus required to achieve an optimal-performing enterprise.

3.3.6.4 A methodology for developing the ontological aspect models

In assisting the practitioner to develop the OAMs (Figure 37) in the right way, Dietz developed a *methodology,* called DEMO (Design and Engineering Methodology for Organisations) and suggests that the OAMs are developed in the following sequence:

1. Develop the interaction model (re-visited in section 8.2.3) to represent the actors and transaction types that are involved during an enterprise operation.

2. Derive the process model from the interaction model to demonstrate the transaction patterns for each transaction type.
3. Detail the action model based on the individual steps of the process model to serve as guidelines for actors in dealing with their agenda.
4. Derive the state model from the action model to specify the state space of the production world.
5. Convert the interaction model to an interstriction model by adding the passive influences (facts that were created) as detailed in the state model (Dietz, 2006).

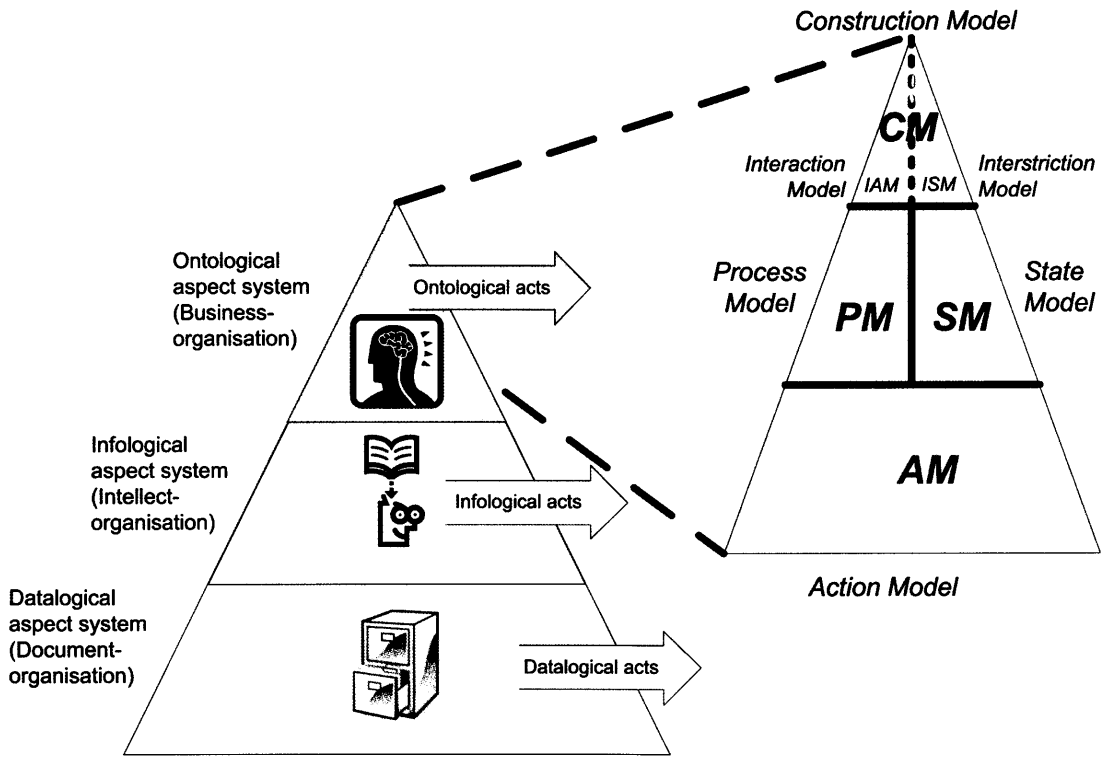


Figure 37: The ontological aspect models, based on Dietz (2006, p. 140)

The *essence of operation* approach is primarily concerned with creating the essential constructional view of the *organisation* of the enterprise system (called the enterprise ontology), as a starting point for *alignment* with the ICT system. Hoogervorst (2009) already acknowledged the value of enterprise ontology as defined by Dietz (2006) by using *enterprise ontology* and *architecture guidance* as two pillars in his *enterprise engineering* approach (see section 3.4.7). Although this section provided an introduction of the *essence of operation* approach, section 8.2 re-visits the *essence of operation* approach using the Business-IT Alignment Model (BIAM) as discussed in section 4.3.

This section introduced the six alignment approaches that were primarily used and referenced for the purpose of this thesis. The next section introduces eight *other alignment approaches*, applied as a secondary data source to provide additional motivation and explanation for the BIAM components (as discussed in Chapter 4).

3.4 OTHER ALIGNMENT APPROACHES

The previous section (section 3.3), introduced six alignment approaches, which were used as the primary data source for analysing alignment approaches to construct the BIAM. This section introduces eight *other alignment approaches* that were also referenced in this thesis as secondary data sources. Table 10 presents the eight *other alignment approaches*, their key mechanisms and referenced publications.

Table 10: Other alignment approaches

Alignment approach	Key mechanism(s)	Referenced publications
GERAM approach	Generalised Enterprise Reference Architecture and Methodology (GERAM) framework	(GERAM, 1999)
Schekkerman approach	E2AF (Extended Enterprise Architecture Framework)	(Schekkerman, 2004)
The dynamic architecture approach	DYA (Dynamic Architecture)	(Wagter, van den Berg, Luijpers, & van Steenbergen, 2005)
Bernard approach	EA3 Cube Framework	(Bernard, 2005)
Gharajedaghi approach	Interactive Management Model	(Gharajedaghi, 2006)
Capgemini approach	IAF (Integrated Architecture Framework)	(Capgemini, 2007)
Hoogervorst approach	Enterprise governance and design concepts	(Hoogervorst, 2009)
The Giachetti approach	EDM (Enterprise Design Methodology)	(Giachetti, 2010)

The following sections introduces each of the eight *other alignment approaches* in terms of their *main benefits* and *key mechanisms*.

3.4.1 The GERAM approach

The *GERAM approach* addresses the challenges that enterprises face in a rapidly changing environment and the need to adapt dynamically (GERAM, 1999). Acknowledging the value of existing reference architectures, the IFIP/IFAC task force evaluated existing enterprise

integration reference architectures (CIMOSA, GIM and PERA) and consolidated the existing reference architectures into a consolidated, generalised architecture. The proposed reference architecture was entitled GERAM (Generalised Enterprise Reference Architecture and Methodology). GERAM incorporates a set of method, models and tools, which are needed to *build and maintain* the integrated enterprise (GERAM, 1999).

The *key mechanism* of the *GERAM approach* is the GERAM framework, which consists of several components (see Figure 38). The most important component, is the GERA (Generalised Enterprise Reference Architecture), which includes basic concepts for enterprise engineering and integration (e.g. specifying enterprise entities, life cycles and life histories of enterprise entities). Other components include methodologies for enterprise engineering (EEMs), enterprise modelling languages (EMLs), which are used to produce enterprise models (EMs). The models guide the implementation of the operational system of the enterprise (EOS), which may also be supported by specific enterprise modules (EMOs). The methodology and languages are supported by enterprise engineering tools (EETs) (GERAM, 1999).

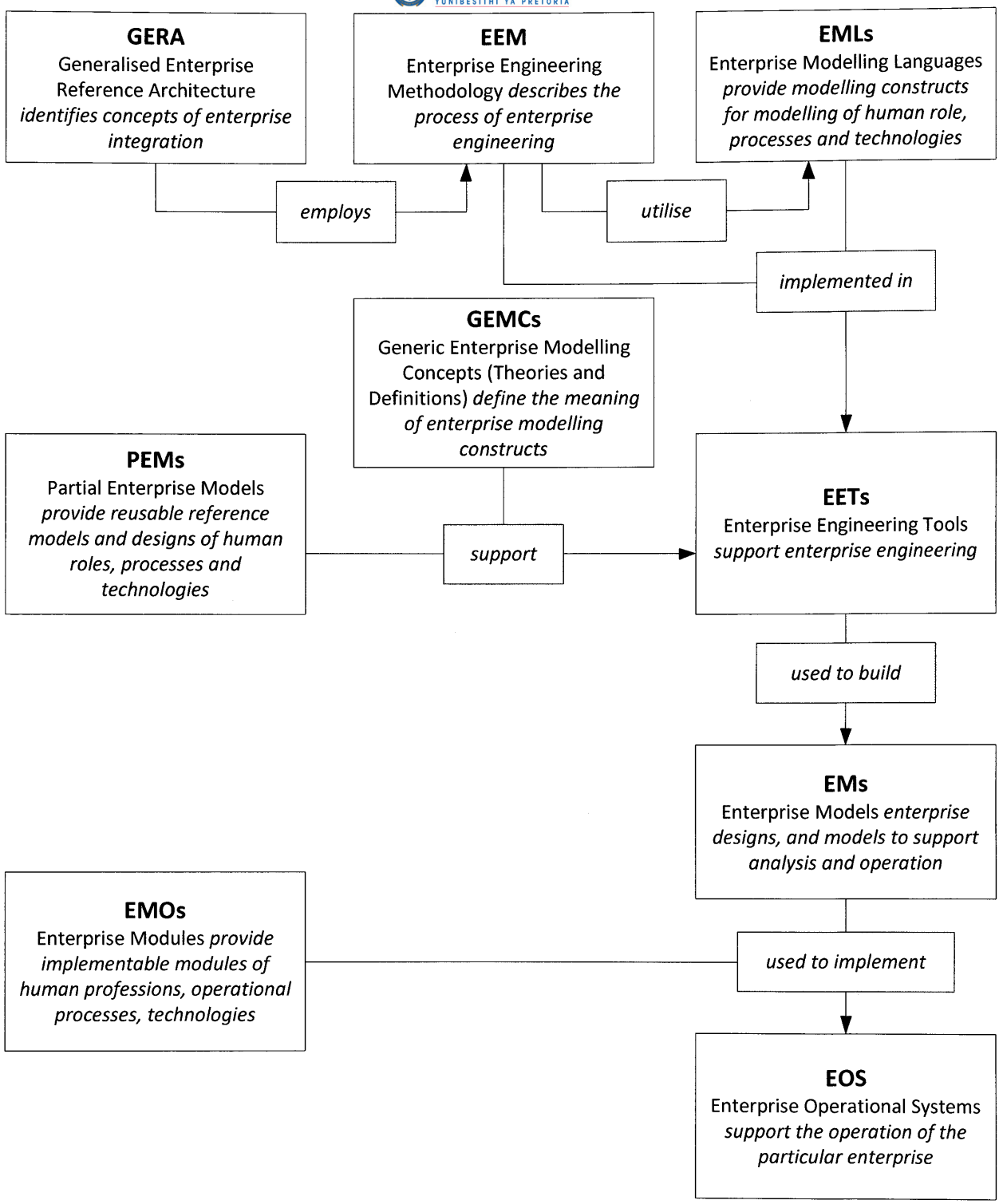


Figure 38: GERAM (Generalised Enterprise Reference Architecture and Methodology) framework components, based on GERAM (1999)

3.4.2 The Schekkerman approach

The *Schekkerman approach* addresses the need of enterprises to collaborate and communicate with all the extended stakeholders of the enterprise (Schekkerman, 2004).

The *key mechanism* of the *Schekkerman approach* is the Extended Enterprise Architecture Framework (E2AF). The E2AF was developed by the Institute for Enterprise Architecture

Developments in 2002, primarily influenced by the Zachman Framework, EAP (Enterprise Architecture Planning) and IAF (Integrated Architecture Framework). The E2AF resembles a matrix of six columns and four rows to distinguish between different levels of concerns and different aspects of the enterprise (Schekkerman, 2004).

The six levels of *concern* include:

1. *Contextual level:* Describing the motivations of the enterprise.
2. *Environmental level*: Representing the business and technology relationships within the extended enterprise.
3. *Conceptual level:* Referring to the requirements of enterprise entities involved in various aspect areas of the enterprise.
4. *Logical level:* Representing the ideal logical solutions for each aspect area.
5. *Physical level:* Describing the physical solutions of products and techniques in each aspect area.
6. *Transformational level:* Describing the impact for the enterprise in terms of the proposed solutions (Schekkerman, 2004).

The four *aspect areas* include:

1. *Business or organisation:* Expressing the business elements and structures.
2. *Information:* Representing the information needs, flows and relations.
3. *Information – systems:* Referring to the automated support of specific functions.
4. *Technology – infrastructure:* Representing the supporting technology environment for the information systems (Schekkerman, 2004).

3.4.3 The dynamic architecture approach

The *dynamic architecture* approach addresses the challenge that enterprises face in finding the correct balance between *coherence* and *agility. Coherence* is required to ensure that the enterprise functions as a uniform entity, whereas *agility* requires dynamic enterprise changes to keep up with changes in products and markets (Wagter et al., 2005).

The *key mechanism* of the *dynamic architecture* approach is the Dynamic Architecture (DYA) model. The DYA model suggests information system development conforming to architecture standards, but also provide for information system development *without conformance* to architecture standards. Most of the development projects should be anticipative in nature, conforming to architecture standards. Development *without conformance* needs to be the exception, requires motivation and still happens in a controlled way. Thus an enterprise may need to develop an ad hoc, short-term solution (*without conformance*) when the enterprise is taken by surprise or if the enterprise needs to seize a once-off competitive advantage (Wagter et al., 2005).

3.4.4 The Bernard approach

The *Bernard approach* intends to *improve enterprise performance* by perceiving the enterprise in a holistic and integrated way, developing both current and future representations/artefacts of the enterprise (Bernard, 2005).

The *key mechanism* of the *Bernard approach* is the EA3 Cube Framework (Figure 39). The EA3 Cube (Figure 39) contains:

1. *Horizontal slices*: Sub-architectures for distinct functional areas.
2. *Vertical segments*: Segments of distinct activity, called *lines of business.*
3. *Common threads*: Threads of common activity that are present in all levels of the framework, e.g. security, standards and workforce.

Artefacts (Figure 39, *Artefacts*) are documentation about the horizontal slices and vertical segments, describing the current or future architecture of the enterprise.

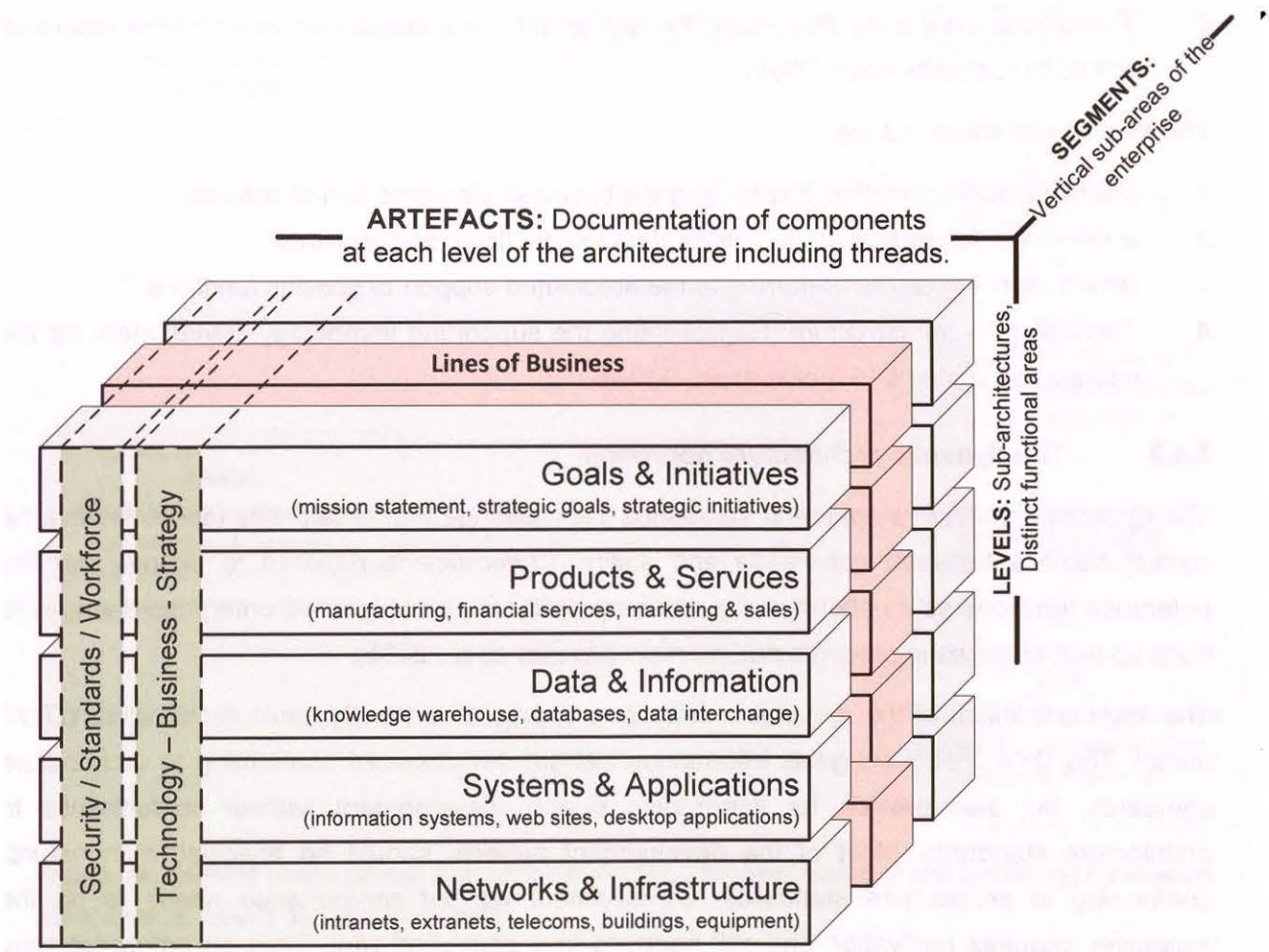


Figure 39: The EA3 Cube Framework, based on Bernard (2005, p. 38)

3.4.5 The Gharajedaghi approach

The *Gharajedaghi approach* addresses the challenges that enterprises face due to continuous "change of the game". Garajedaghi (2006) states that a dual paradigm shift is necessary to understand the enterprise, which would contribute towards effective enterprise redesign and

management. The dual paradigm shift requires (1) a shift in the *method of inquiry* (shifting from an analytical approach towards a systems approach), and (2) a shift in the *conception of the enterprise* (shifting from a mindless system towards a multiminded sociocultural system).

The *key mechanism* of the *Gharajedaghi approach* is the Interactive Management Model (see Figure 40). The Interactive Management Model suggests contextual knowledge within three areas, prior to the problem-definition and design within an enterprise:

- *Basic assumptions*: Making assumptions about the evolving game in which enterprises participate, drivers for change and basis for competition.
- *Systems principles*: Understanding systems principles, such as openness, purposefulness, emergent properties, multi-dimensionality and counter-intuitiveness.
- *System dimensions*: Understanding and describing an enterprise in terms of five dimensions, i.e. power, beauty, wealth, knowledge and values.

Knowledge of the environmental context, systems principles and systems dimensions is necessary to define problems and opportunities, using various techniques (system analysis, obstruction analysis and system dynamics). Based on the analyses, enterprise designers design a solution/idealised design. The idealised design could lead to several levels of output (redesigning the enterprise, its operations or products).

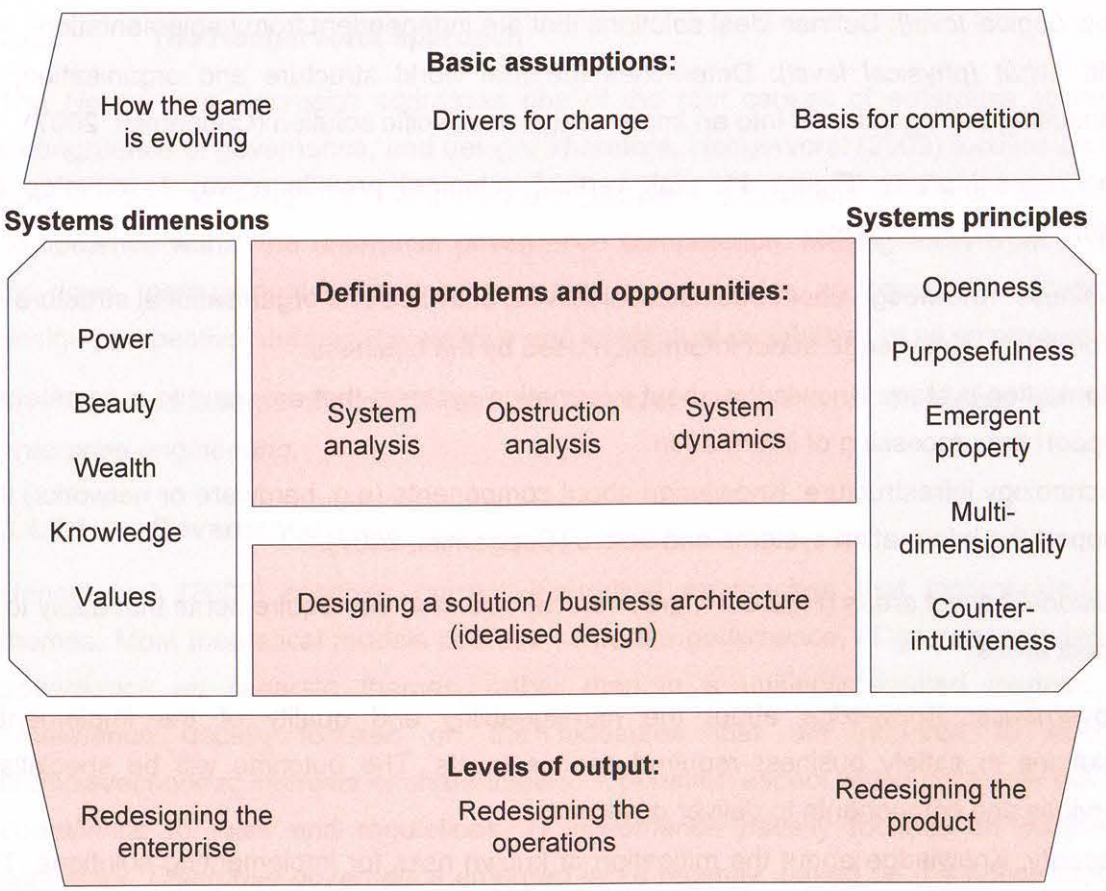


Figure 40: Interactive Management Model, based on Gharajedaghi (2006, p. 23)

3.4.6 The Capgemini approach

The *Capgemini approach* addresses the challenges of uncontrolled growth of information systems and technology in the late 1990s. Uncontrolled growth resulted in complex and costly information system landscapes. The *Capgemini approach* provides a solution to create better alignment between business and IT, deliver more flexibility for business and IT, and manage complexity better (Capgemini, 2007).

The *key mechanism* of the *Capgemini approach* is the Integrated Architecture Framework (IAF) (see Figure 41). Capgemini (2007, p. 4) views architecture as "providing a comprehensive and coherent view across business, information , systems and technology". The IAF is used to structure and define architecture content in terms of two dimensions: (1) abstraction levels, and (2) aspect areas.

The abstraction levels (Figure 41, horizontal bars) allows for a consistent definition within each aspect area:

- *Why (contextual level)*: Provides the scope and objectives for the new architecture and its content.
- *What (conceptual level)*: Elaborates and analyses the objectives, re-stated as requirements.
- *How (logical level)*: Defines ideal solutions that are independent from implementation.
- *With what (physical level):* Determines the real world structure and organisation, by translating the *logical level* into an implementation-specific solution (Capgemini, 2007).

Four *core aspect areas* (Figure 41, pink vertical columns) provide a way to develop the architecture of the enterprise:

- Business: Knowledge about business objectives, activities and organisational structure.
- Information: Knowledge about information used by the business.
- Information system: Knowledge about information systems that are used to automate and support the processing of information.
- Technology infrastructure: Knowledge about components (e.g. hardware or networks) that support the information systems and actors (Capgemini, 2007).

Two *additional aspect areas* (Figure 41, grey vertical columns) set requirements that apply to all core aspect areas:

- *Governance:* Knowledge about the manageability and quality of the implemented solutions to satisfy business-required service levels. The outcome will be specialised services and components to deliver governance.
- *Security:* Knowledge about the mitigation of known risks for implementing solutions. The outcome will be specialised services and components to deliver the required security (Capgemini, 2007).

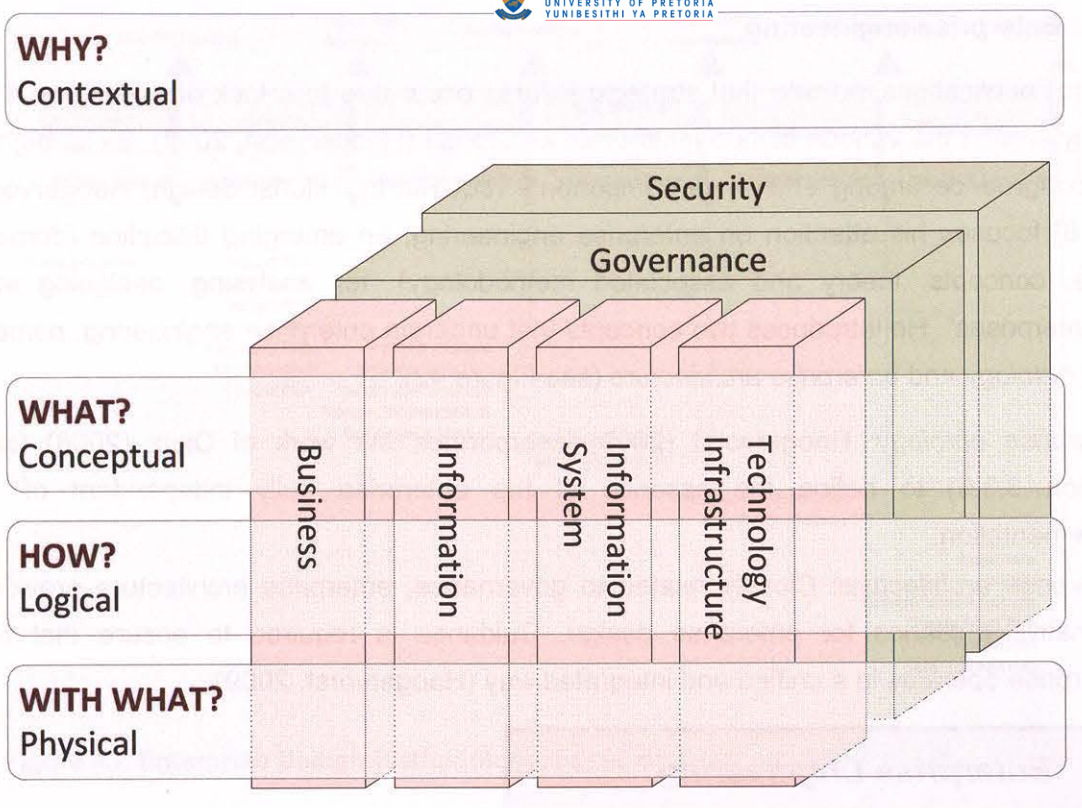


Figure 41: The Integrated Architecture Framework, based on Capgemini (2007, p. 13)

3.4.7 The Hoogervorst approach

The *Hoogervorst approach* addresses one of the root causes of enterprise strategic failure, incongruence of governance, and design. Therefore, Hoogervorst (2009) focuses on addressing governance and design from a unified perspective. He positions enterprise design as a core competence within the enterprise governance competence. Moving away from a mechanistic top-down management-focused perspective, he advocates an organismic governance and design perspective, utilising the creative and intellectual capabilities of all employees.

The *key mechanisms* of the *Hoogervorst approach* are concepts on unified *governance* and *enterprise engineering*.

3.4.7.1 Governance

Hoogervorst (2009) criticizes current theoretical approaches that incorporate governance themes. Most theoretical models address *corporate governance*, *IT governance* and *enterprise governance* as separate themes, rather than in a unified/integrated manner. *Corporate governance* usually focuses on the measures that are required to safeguard the financial/economic interests of shareholders. A pertinent aspect within corporate governance, is *compliance* to rules and regulations. *IT governance* usually focuses on *business and IT alignment*. *Enterprise governance* emerged more recently, based on the notion that enterprise *performance* (rather than compliance) safeguards shareholder interests. Hoogervorst (2009) unites corporate governance and IT governance under the umbrella-term, *enterprise governance*.

3.4.7.2 Enterprise engineering

A number of publications indicate that strategic failures occur due to a lack of coherence and consistency among the various components of an enterprise (Hoogervorst, 2009). Since higher levels of congruence among enterprise components requires intentional design, Hoogervorst (2009, p. 8) focuses his attention on enterprise engineering, an emerging discipline (domain knowledge, concepts, theory and associated methodology) "for analysing, designing and creating enterprises". He introduces two concepts that underpin enterprise engineering, namely *enterprise ontology* and *enterprise architecture* (see Figure 42):

- *Enterprise ontology*: Hoogervorst (2009) incorporates the work of Dietz (2006) (see section 3.3.6) to define the essence of the enterprise, fully independent of its implementation.
- *Enterprise architecture*: Closely related to governance, enterprise architecture provides normative guidance for enterprise design. Guidance is required to ensure that the enterprise operates in a unified and integrated way (Hoogervorst, 2009)

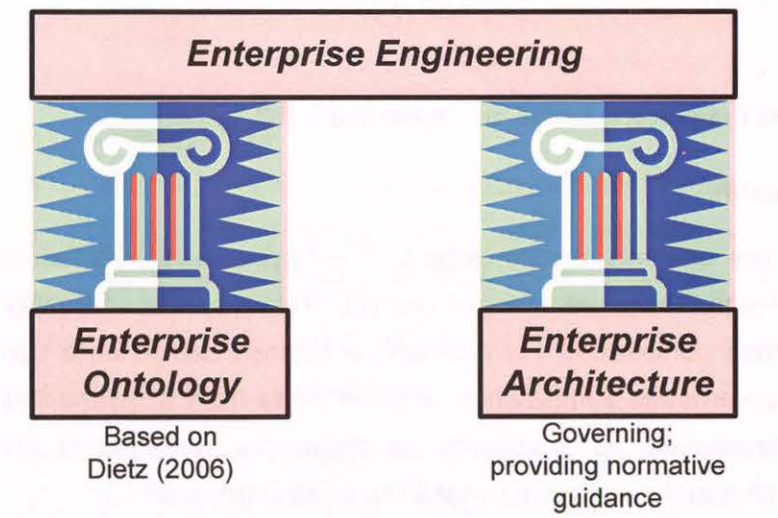


Figure 42: Pillars of enterprise engineering, based on Hoogervorst (2009)

3.4.8 The Giachetti approach

The *Giachetti approach* addresses the integration challenges of enterprise design. Knowledge for enterprise design is often fragmented and contained within different disciplines, preventing enterprises from achieving optimally (Giachetti, 2010). Giachetti (2010) states that enterprises require a system-wide perspective on the enterprise to integrate the specialised knowledge of separate enterprise aspects. As a solution, he provides an enterprise engineering methodology.

The *key mechanism* of the *Giachetti approach* is the Enterprise Design Methodology (EDM), which consists seven life-cycle phases (see Figure 43). Each phase contains several activities. Certain milestones mark the end of one phase and the beginning of the next, e.g. Kick-off meeting marks the end of *project initiation* and the start of *project planning.* The EDM forms the backbone to present several principles, models, methods and tools needed to design the enterprise (Giachetti, 2010).

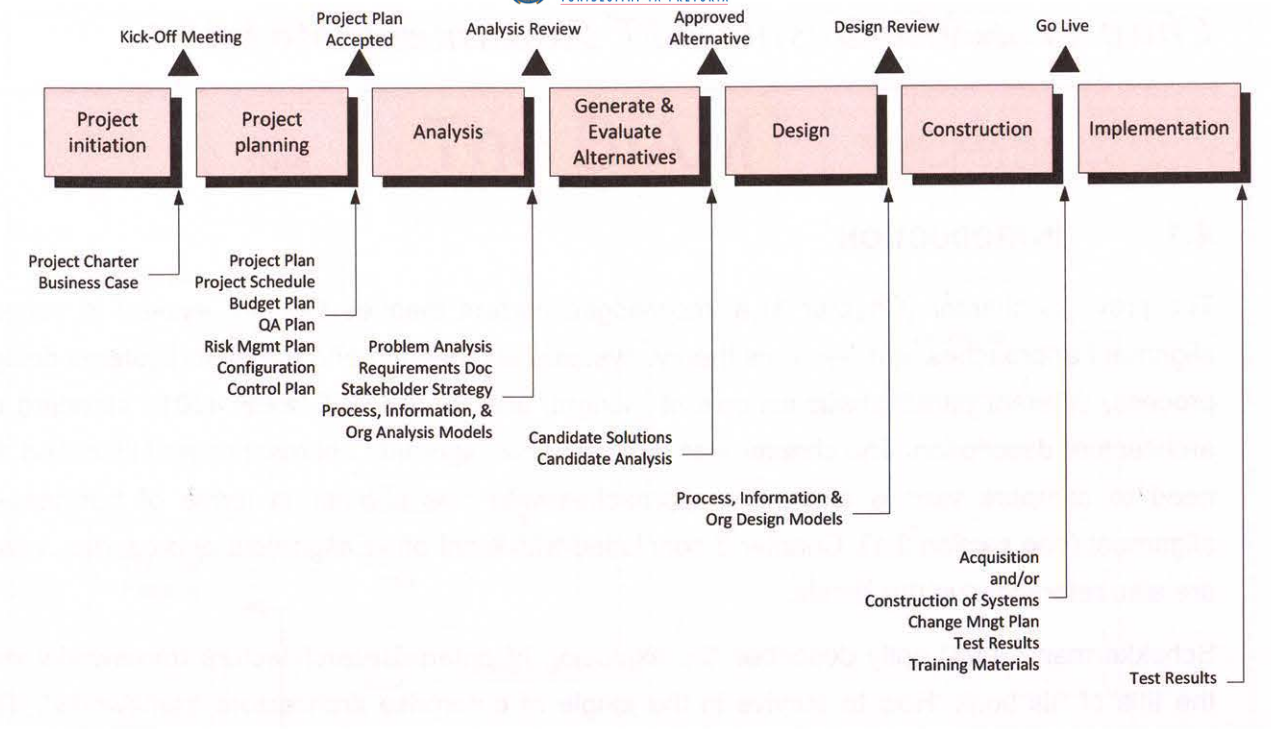


Figure 43: Enterprise Design Methodology, based on Giachetti (2010, p. 120)

3.5 CONCLUSION

Chapter 3 introduced the concept of business-IT alignment within a broader enterprise alignment context. Acknowledging the different approaches towards alignment, common theoretical foundations do exist. The theoretical foundations for business-IT alignment were delineated in section 3.2, and include systems theory, systems engineering and the basic systems design process, different paradigmatic schools of thought, and the ISO/IEC/IEEE 42010 standard on architecture description. Without providing a critical analysis in section 3.3, six alignment approaches are discussed, of which four approaches are prominent in literature (Zachman approach, Open Group approach, OMB approach and Gartner approach) and two less popular alignment approaches (the *foundation for execution* approach and the *essence of operation* approach). These six alignment approaches were used as the *main data source* during an inductive development of the BIAM (as discussed in Chapter 4).

Four of the six alignment approaches (the Zachman approach, Open Group approach, the *foundation for execution* approach and the *essence of operation* approach) will be used later in this thesis to verify the use of BIAM in providing a business-IT contextualisation.

Finally, this thesis referred to eight *other alignment approaches* as a *secondary data source*. The *other alignment approaches* (referenced in Chapter 4), provide additional motivation and explanation for the BIAM components.