MASTER'S THESIS

MEASURING ARCHITECTURAL COMPLEXITY

Quantifying objective and subjective complexity in enterprise architecture

Jeroen Monteban

Business Information Technology

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Enschede

May 24, 2018

Deloitte.

UNIVERSITY OF TWENTE.

Measuring Architectural Complexity

Quantifying objective and subjective complexity in enterprise architecture

Author

Jeroen Monteban

Programme	Business Information Technology
Track	IT Management & Innovation
Faculty	Electrical Engineering, Mathematics and Computer Science
Student number	1220837
E-mail	j.monteban@alumnus.utwente.nl

Graduation Committee

Maria Iacob

Department	Industrial Engineering and Business Information Systems
E-mail	m.e.iacob@utwente.nl

UNIVERSITY OF TWENTE.

Marten van Sinderen

Department	Computer Science
E-mail	m.j.vansinderen@utwente.nl

UNIVERSITY OF TWENTE.

Erik Hegeman

Department	Enterprise Architecture
E-mail	ehegeman@deloitte.nl

Deloitte.

Kiean Bitaraf

Department	Enterprise Architecture
E-mail	kbitaraf@deloitte.nl

Deloitte.

Preface

The writing of this Master's thesis marks the end of my days as a student at the University of Twente. These days started almost seven years ago, when a younger version of myself moved to Enschede to study Business Information Technology. After a relatively calm first year, I started a series of extracurricular adventures, during which I met countless of amazing people, and learned much about the world and myself. These years have been a pleasant mix of studying, board years, committees, foreign trips and of course the occasional drink. They have brought me many great experiences, and taught me a great deal, both personal and professional.

The final adventure I embarked upon as a student was this Master's thesis. In October of last year, I started as a graduate intern at the Enterprise Architecture department of Deloitte Consulting. The road towards a completed thesis is a tricky one, with many hurdles to take and crossings to navigate. Unavoidably, I took some side paths that lead nowhere, and miscalculated the distance to some milestones. However, I never lost sight of the destination, and am glad to have finally arrived at a completed thesis.

Of course, such a long road cannot be journeyed alone. First of all, I would like to thank my University supervisors, Maria and Marten, for their guidance and support. Their experience provided valuable insights in this project, and helped to improve its quality. Next, I would like to thank Erik en Kiean for their supervision, inspiring ideas and critical view; their detailed feedback and practical knowledge greatly contributed to this research. I have really enjoyed our collaboration over the past months. Additionally, I would like to thank my colleagues at the Enterprise Architecture department. Their insights, ideas, feedback and critical questions always inspired me to keep improving this research. And of course, I really enjoyed the coffee breaks, foosball matches and Friday afternoon drinks!

A special thanks goes out to the companies that helped me by providing interviews or a case study. Their involvement supplied the data for this research, and provided interesting insights in their organizations.

Last but not least, I would like to thank my family and friends for their support during this thesis, and during my days as a student.

This journey now comes to an end, and I hope you enjoy reading its results!

Jeroen Monteban
Amsterdam, May 23rd, 2018

Executive Summary

The fast pace of economic growth and technological advances have increased the dependence of business on IT and vice versa. Due to these developments, enterprise architecture (EA) - a discipline aiming to integrate business and IT - has witnessed a surge of complexity, often leading to an inefficient use of the architecture and a lack of control. Yet, an architecture that is too simple for its complex environment cannot support the functionalities required to operate in that environment. Complexity management has become an essential undertaking for enterprise architecture. It strives for an optimal level of complexity to efficiently and effectively deal with the complexity of the architecture's environment.

The basis for effective complexity management is measurement, yet no standardized or proven method for EA complexity measurement currently exists, nor is there consensus about the attributes contributing to complexity. Additionally, the many stakeholders involved in an enterprise architecture all have a different perception of complexity, which impedes their collaboration. Understanding this difference in complexity perception is essential to enable effective complexity management. To accomplish this, this research aims to incorporate objective and subjective complexity metrics in an EA complexity measurement model.

A literature review was conducted to gain insight into the state of the art of EA complexity research, and create an overview of the existing complexity metrics. Next, a series of twelve interviews were held at four organizations, during which participants were queried on the complexity of the enterprise architecture in their organization, and the attributes influencing their perception of this complexity. Using the data obtained through literature and interviews, a conceptual model of EA complexity was designed. This conceptual model contains constructs that influence EA complexity and stakeholders' perception thereof, and describes the relations between these constructs. Next, constructs have been operationalized through the design of metrics. Using these metrics, the constructs influencing complexity can be measured, thus creating a measurement model for EA complexity. Finally, the measurement model was validated through three expert interviews, and a case study applying the model in practice.

The identification of the constructs influencing objective and subjective complexity showed many aspects affecting EA complexity that are currently not considered in literature and practice. Whereas constructs such as size and heterogeneity are well-represented in literature, many other factors are ignored. These include enterprise- and environment-related constructs such as politics, technical debt or industry. Additionally, several constructs were found to specifically influence the perception of complexity, such as documentation, communication between stakeholders, the presence of an architectural vision, and several stakeholder qualities. A full list of these constructs and the metrics required to measure them can be found in this research.

The results from this research contribute to both the theory and practice of EA complexity. It consolidates the existing research concerning objective complexity measurement, and provides a first insight into the previously unexplored area of subjective complexity. In practice, this research be used to enable effective complexity management and overcome stakeholder differences.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction

Throughout history, humankind has busied itself with the creation of increasingly complex tools and systems. What started out with a stone spear, over the ages turned into water wheels, pyramids, steam engines and the International Space Station. The evolution of human tools has known periods of rapid innovation and acceleration, the most famous being the agricultural and industrial revolution, and the most recent being known as the digital or information revolution (Freeman & Louçã, 2001). With the invention of computers and the rise of Information Technology (IT), humans have created systems of unprecedented complexity, creating possibilities unimaginable before (Kandjani, Bernus & Wen, 2014). The use of IT in modern enterprises is widespread, and its value to business undeniable and well-established in literature (Cardona, Kretschmer & Strobel, 2013; Liang, You & Liu, 2010; Mithas, Ramasubbu & Sambamurthy, 2011). The size and complexity of these IT landscapes have been rising ever since; a trend that is expected to continue (Landthaer, Kleehaus & Matthes, 2016).

Due to their growing size and complexity, the need rose for a structured approach in the development of IT systems and landscapes. To this end, architecture has had an increasingly important role. In their Foundations for the Study of Software Architecture, Perry & Wolf (1992) already stated the benefits of architecture and argued to increase efforts on its development. A proper architectural basis has shown to benefit both the development and maintenance of a system (Shaw, DeLine, Klein, Ross, Young & Zelesnik, 1995) and can yield a competitive advantage (Bradley & Byrd, 2006). Either directly or indirectly, architecture impacts cost, maintainability and interoperability (Osvalds, 2001). Especially in an enterprise-wide context, architecture is an important basis for IT management (Rood, 1994). However, an often reported problem is that over time, architectures tend to become more complex. Subsequent expansions and modifications of an IT landscape can result in a lack of architectural structure, or having multiple design patterns intertwined. This increase in architectural complexity can, in turn, impact a system and its environment in many ways. Besides higher costs, increased architectural complexity can lead to lower adaptability and maintainability of the entire IT landscape (Wehling, Wille, Seidl & Schaefer, 2017), increase operational risk and reduce flexibility (Schmidt & Buxmann, 2011). Moreover, the increasing size and complexity of IT landscapes poses the need to ensure IT is properly aligned with, and enables, business goals. Hence, the field of enterprise architecture (EA) was born.

Complexity has been identified as one of the major challenges faced by the discipline of enterprise architecture (Lucke, Krell & Lechner, 2010), and it has been attributed as one of the causes of high failure rates in IT projects (Daniels & LaMarsh II, 2007). Complexity reduction is a popular remedy in large IT landscapes, but at the same time, a certain level of architectural complexity is necessary to properly support business goals and requirements, and to enable extensive functionality (Wehling et al., 2017; Schmidt, 2015). This poses a challenge: to maximize the performance of an architecture, it is important to find its optimal level of complexity (Heydari & Dalili, 2012; Collinson & Jay, 2012; Schmidt, 2015).

The evidence above suggests architectural complexity can have an important influence on the performance of enterprises and their IT landscapes, and should be managed properly. In fact, Lange & Mendling (2011) found that in a panel of experts, almost 90% considered complexity management as one of the primary goals of enterprise architecture. Yet, hardly any existing enterprise architecture methodologies and research directly addresses complexity management. At the same time, little research on complexity management in other areas is applicable to the field of enterprise architecture (Lee, Ramanathan, Hossain, Kumar, Weirwille & Ramnath, 2014). One of the problems in this regard is measurement. The lack of any recognized methodology of complexity measurement indicates a shortage of research in this field. Simultaneously, complexity measurement can be considered a prerequisite for proper complexity management. As Schütz, Widjaja & Kaiser (2013, p.1) note: "Measurability is the essential basis for management". However, quantifying architectural complexity is in itself a very complex process, indicated by that fact that no standardized or proven method currently exists.

This study aims to create a model of enterprise architecture complexity measurement, including the variables influencing architectural complexity and the appropriate metrics to measure these. Ultimately, this will enable the management of complexity in enterprise architectures.

1.2 Background

1.2.1 Architecture

Effective measurement requires a thorough understanding and proper definition of the target of measurement and its context. "Architecture" is quite a general term; it is defined in many different contexts and disciplines. The International Organization for Standardization (ISO) provides widely applicable and accepted standards and definitions, and this research will adopt their definition of architecture:

Definition 1. *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 (ISO/IEC/IEEE, 2011).

Though widely applicable, this definition is very dependent on the interpretation and scope of the terms "system" and "environment", since both are again very general terms.

The environment in which this research operates is that of organizations. However, not only organizations as a whole are considered; focus may lie on a subset or superset of an organization as well, such as departments or an ecosystem of organizations. Thus, the environment of focus is that of an enterprise, which can be defined as follows:

Definition 2. *Enterprise:* Any collection of organizations that has a common set of goals (The Open Group, 2011).

Defining the system under consideration requires an elaboration on the exact meaning of enterprise architecture. Though up for debate and different interpretations, a generally accepted definition for enterprise architecture is as follows:

Definition 3. *Enterprise architecture:* A coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure (Lankhorst, 2017).

This definition shows that the practice of enterprise architecture distinguishes two different viewpoints. First of all, one can discuss *an* enterprise architecture, which considers the organizational structure, business processes, information systems, and infrastructure of an enterprise, represented by a set of models designed by an architect. Secondly, one can consider the *practice* of enterprise architecture, which is the set of principles and methods used to design and realize the enterprise architecture. In practice, the process of enterprise architecture entails the definition of a current and future state architecture. In this research, "the complexity of an enterprise architecture" refers to the complexity of the architecture of the enterprise's elements, not the process of their design or realization.

The definition of enterprise architecture suggests that it regards to multiple aspects of an enterprise, and would thus consist of multiple "sub-architectures". Indeed, the popular enterprise architecture framework TOGAF identifies four architectures, called domains, commonly accepted as subsets of an overall enterprise architecture (The Open Group, 2011): business, data, application and technology architecture. This subdivision of enterprise architecture is supported by additional frameworks, such as the Integrated Architecture Framework by Capgemini (van 't Wout, Waage, Hartman, Stahlecker & Hofman, 2010), ArchiMate (The Open Group, 2017), and the research by Wagter, van den Berg, Luijpers & van Steenbergen (2005). The different domains focus on different aspects of the enterprise architecture, and are visualized in Figure 1.1. The *business architecture* concerns the strategy and objectives, products and services, governance, organizational structure and processes of an enterprise. The *data architecture* regards the data and information that an enterprise holds, and its structure, relationships and management. *Application architecture* refers to the applications deployed in the enterprise, their relationships and their support of the business processes. Finally, the *technology architecture*, also called infrastructure domain, concerns all the hardware, network and middleware components required to support the other domains, describing their types and structure. All domains are considered to be related and interdependent, and it is therefore considered to be a holistic approach to architecture.

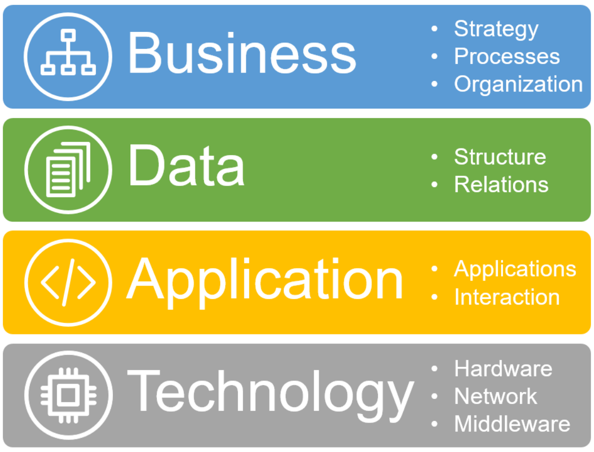


Figure 1.1: Enterprise architecture domains

1.2.2 Complexity

The concept of complexity seems hard to measure, but perhaps even harder to define. It is used throughout many research disciplines, and open to an array of different interpretations. Since this research focuses on the complexity of (enterprise) architectures, the definition of complexity should be focused likewise.

The Cambridge Dictionary defines complexity as "the state of having many parts and being difficult to understand or find an answer to", and a lot of existing architecture research endorses this view. They relate complexity to the number of components or elements, their relationship, and the variation or heterogeneity of these (Davis & LeBlanc, 1988; Flood & Carson, 1993; Kinsner, 2008). Schütz et al. (2013) share this view; they add that to consider the total complexity of an enterprise architecture, complexity within each domain as well as interrelations between its domains should be considered. Several studies look at patterns in these elements and relations: Kazman & Burth (1998) define complexity by considering the pattern coverage of an architecture, whereas Efatmaneshnik & Ryan (2016) calculate its distance from reference simplicity. Other studies define complexity in terms of their proposed metrics (Gao, Warnier, Splunter, Chen & Brazier, 2015; Lankford, 2003). Interestingly, all of these papers on complexity use measurable terms to define complexity, such as the number of elements and relations. Literature on business complexity, the non-technological domain of enterprise architecture, agrees on this as well (Collinson & Jay, 2012; Gharajedaghi, 2011). Therefore, this research adopts the view that complexity is best defined in measurable terms.

Although all of the previously mentioned researchers aim for the measurement of complexity, their exact interpretations of complexity differ. As a result, Schneider et al. (2014) take a more abstract approach to complexity in enterprise architecture. They note that different interpretations of complexity throughout research impedes the common acceptance of the field. Therefore, they propose a conceptual framework aimed at unifying these views on complexity. According to Schneider et al. (2014), the various aspects of complexity can be specified by four dimensions, as exhibited in Figure 1.2, which can be described as follows.

1.2.2.1 Objective versus subjective complexity

The first dimension is based on the role and influence of the observer. Objective complexity is independent of any observer, and therefore an inherent property of the object of study. Subjective complexity occurs when complexity is a part of the relationship between the object of study and its observer, and therefore dependent on this relationship. In enterprise architecture, different stakeholders may have a different perception of an architecture's complexity. In other words: subjective complexity exists in the eye of the beholder.

1.2.2.2 Structural versus dynamic complexity

This dimension relates to the internal structure of a system and the time frame considered. Structural, or static, complexity looks at system components and their cause-and-effect relationships in a static snapshot of the system. Dynamic complexity, on the other hand, refers to the interaction between components within the system, and the change of their relationship over a period of time. Beese, Aier, Haki & Aleatrati Khosroshahi (2016) argue that these concepts are closely connected; dynamic complexity is highly influenced by structural complexity and results from the interaction between a system's structural complexity and its dynamics.

1.2.2.3 Quantitative versus qualitative complexity

The next dimension refers to the way certain properties or attributes are evaluated. In the qualitative notion, complexity is evaluated through the qualitative assessment of elements within a system, and therefore not dependent on the quantity of these elements. In contrast, the quantitative notion concerns the quantification of elements within a system, relating its complexity to these numbers.

1.2.2.4 Ordered versus disordered complexity

The final dimension relates to the number of attributes considered when evaluating the system's complexity. In a system consisting of a substantial amount of attributes, with individually inconsistent attributes or an unknown amount of attributes, statistics become applicable. Although individual attributes are not predictable, the system as a whole may have analyzable attributes and behaviour. This statistical analysis of system behaviour applies a disordered notion of complexity. On the other hand, ordered complexity refers to a moderate and known number of attributes, with strong and clear internal relations. Although statistics are not applicable, these attributes and their relations may be studied to predict system behaviour.

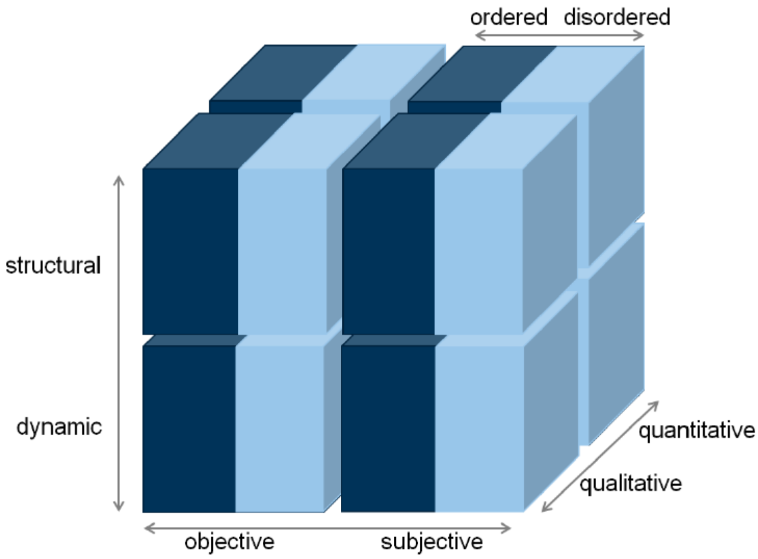


Figure 1.2: Complexity dimensions (Schneider et al., 2014)

As is illustrated by the complexity cube in Figure 1.2, these dimensions are independent of each other and can be combined in any way applicable in practice. Furthermore, they show that a system can combine both complexity notions along a single dimension. This is illustrated by the research of Beese et al. (2016), where structural complexity is considered to be an indispensable element of dynamic complexity.

This research adopts this four-dimensional view on complexity. Using this theory, definitions of complexity can be classified by identifying the appropriate dimensions. As a result, a complexity metric (CM) can be defined as a quadruple:

$$CM = (x_1, x_2, x_3, x_4)$$

This is based on the four dimension sets:

$$x_1 \subseteq \{objective, subjective\}$$
$$x_2 \subseteq \{structural, dynamic\}$$
$$x_3 \subseteq \{quantitative, qualitative\}$$
$$x_4 \subseteq \{ordered, disordered\}$$

Using these complexity dimensions “allows [researchers] to apply their own choice of complexity notions without having to argue for a specific definition of complexity” (Schneider et al., 2014, p.8). Therefore, this research will not adhere to a specific definition of complexity, but rather proposes the view that complexity is a property of a system that is defined by the relevant set of metrics used to measure it.

Definition 4. *Complexity:* A property of a system that is defined by the relevant set of metrics used to measure it.

1.2.3 Metrics

In order to properly measure the complexity of an architecture, it is important to clearly understand and define the process of measurement, as well as the object to be measured and its attributes.

Definition 5. *Attribute:* A quality or feature regarded as a characteristic or inherent part of someone or something (Oxford Dictionary, 2018a)

To define the measurement of these attributes, the International Organization of Standardization has adopted the International Vocabulary of Metrology, providing definitions for measurement and measurement standards (Joint Committee for Guides in Metrology, 2008). The measurement definitions formulated in this research are based on that document, though adopting its own terminology.

In order to quantify and measure the attributes of an architecture, one or more metrics have to be designed.

Definition 6. *Metric:* The property of an attribute, where the property has a magnitude that can be expressed as a number and a reference.

Definition 7. *Measurement:* The process of obtaining one or more metric values that can reasonably be assigned to an attribute.

From these definitions it can be inferred that measuring architectural complexity is the process of finding the value of architectural complexity metrics. In order to do this, the attributes contributing to complexity have to be found, and the appropriate metrics to measure these attributes designed.

Abran (2010) describes the process of designing a measurement method, consisting of three steps:

1. *Measurement principle:* gives the description of the attribute to be measured and its metrics;
2. *Measurement method:* operationalizes the principle, defining the steps to be followed in order to measure the attribute;
3. *Measurement procedure:* implements one or more measurement principles by implementing a measurement method.

The first step of measurement, defining the measurement principle, is an essential basis for proper measurement, and is the focus of this research. To develop a measurement principle for architectural complexity, the appropriate attributes and metrics should be identified first. Vasconcelos, Sousa & Tribolet (2007) introduce a template for architecture metrics, which include the following information:

1. *Name and description*
2. *Value*
3. *Computation:* describes the method of value calculation
4. *Architectural level:* the correlated architectural domain

Additionally, while developing a measurement principle for architectural complexity in software, Mccabe & Butler (1989, p.1423) present a list of properties that support the applicability of metrics when quantifying complexity:

1. "The metric intuitively correlates with the difficulty of comprehending a design." When using the metrics, a high level of complexity should also yield a high value of the metric. When quantifying a simple design, the metric value should be low.

2. “The metric should be related to the effort to integrate the design.” Since the integration of an architectural design is the most costly phase, the metrics should relate to this aspect. When measuring the complexity of an enterprise architecture, this means that the integration of the four architecture domains should be strongly considered when drafting metrics.

3. “The metric should help generate an integration test plan early in the life cycle.” Metrics should be applicable early in the life cycle of an architecture, so they can be used to positively influence its complexity in an early stage. In enterprise architecture, this means that metrics should be applicable on an architecture design.

4. “The metric and associated process should be automatable.”

1.2.4 Definition overview

Figure 1.3 provides an overview of the definitions introduced so far and the relations between them. This shows that the architecture of an enterprise has a certain amount of attributes, one of which is complexity. Complexity is, in turn, influenced by one or more of these attributes. Each of these attributes in measured by one or more metrics.

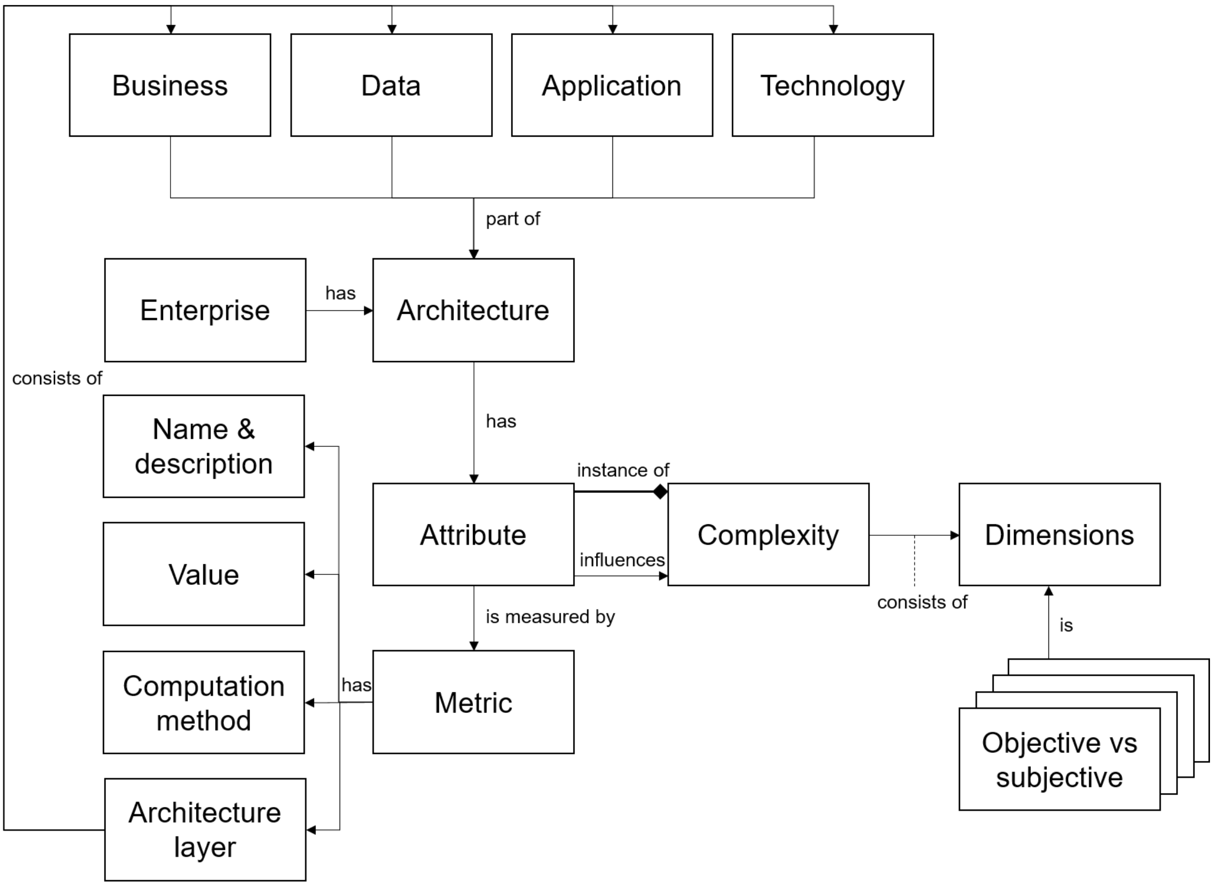


Figure 1.3: Definition overview and relations

1.3 Research objectives

1.3.1 Problem statement

Complexity has been identified as one of the key challenges of enterprise architecture (Lucke et al., 2010), and finding the optimal level of complexity is essential in order to maximize the performance of organizations and their IT landscapes (Collinson & Jay, 2012; Heydari & Dalili, 2012; Schmidt, 2015). The goal of complexity management in enterprise architecture is to achieve the appropriate level of complexity for the architecture's context and purpose. To determine the current and desirable level of complexity for an enterprise architecture, measurement is an important basis (Schütz et al., 2013). However, quantifying architectural complexity is in itself a very complex process, and no standardized or proven method exists (Schneider et al., 2014). Furthermore, there is no consensus on the attributes contributing to complexity. Without these attributes and the appropriate metrics to measure them, measurement of architectural complexity is not possible, making complexity management impracticable.

1.3.2 Objective

The measurement of architectural complexity is required to enable proper enterprise architecture complexity management. To accomplish this, the attributes contributing to this complexity should be identified, and their impact on complexity described. Additionally, metrics should be composed to measure these attributes. This research will propose a model which includes these attributes and their impact on complexity, and provides metrics for their measurement, enabling the quantification of complexity in enterprise architectures.

1.3.3 Scope

In this research, complexity has been defined according to the dimensions proposed by Schneider et al. (2014). To properly manage architectural complexity, all dimensions have to be considered. However, Schneider et al. (2014) indicate that current research is limited in its investigation of different dimensions. This was confirmed by a structured literature review carried out in this research, suggesting little to no research is focused on the subjective and dynamic notions of complexity (see section 2.3 "Objective complexity metrics", specifically Table 2.4). Whereas both dimensions are relevant to enterprise architecture complexity, the subjective notion seems particularly interesting in regards to complexity management, for several reasons. Firstly, an enterprise architecture is usually designed and maintained by multiple stakeholders. By definition, each of these stakeholders will perceive the complexity of the architecture differently. Understanding this difference in perception will help to consolidate their views and could greatly improve their collaboration. Additionally, there are many other stakeholders that are required to work with an enterprise architecture. Considering their perception of its complexity may help to increase their understanding of the architecture. This might in turn lead to a higher compliance with architecture standards or increase stakeholders' efficiency when working with the architecture. Finally, knowing which factors influence perceived complexity may help to simplify an inherently complex architecture, making the architecture more manageable (Jochemsen, Mehrizi, van den Hooff & Plomp, 2016). Therefore, this research will focus on the dimension of objective/subjective complexity, and will not differentiate between the other complexity dimensions.

The process of designing a measurement methodology has three main steps: measurement principle, method and procedure (Abran, 2010). This research will focus on the development of a measurement principle, meaning it will research the attributes of complexity and propose metrics, but will not define the exact steps to be followed for their measurement.

1.4 Research design

1.4.1 Research questions

This research will answer the following main research question to achieve the stated objective:

How can objective and subjective complexity metrics be incorporated in a unified enterprise architecture complexity model?

This research question reflects the goal of this study to measure architectural complexity, and incorporates its scope on the objective/subjective complexity dimension. To answer the main research questions, several sub-questions have to be answered regarding the context of the enterprise architecture and both types of complexity. The following sub-questions are studied:

1. Which existing metrics are most prevalent for measuring objective complexity in an enterprise architecture?

Existing literature already defines several metrics for the measurement of objective complexity in an enterprise architecture. This research aims to review this literature and identify the most prevalent and appropriate objective complexity metrics.

2. How can stakeholders be defined in a context-agnostic way?

According to the Oxford dictionary, subjectivity means to be "dependent on [...] an individual's perception" (Oxford Dictionary, 2018b). Therefore, in order to measure the subjective complexity of an architecture, it is important to define the different stakeholders that interact with it. This stakeholder definition has to be context-agnostic in order to be applicable in all different types of enterprises.

3. Which attributes influence a stakeholder's perception of enterprise architecture complexity?

The definition of subjectivity introduced above makes clear that subjectivity is influenced by the object of study, and a stakeholder's perception of this. Therefore, to measure subjective complexity, the attributes influencing stakeholder perception should be identified.

4. How does stakeholder perception lead to subjective complexity?

Subjective complexity is based on the object of study and the perception of stakeholders. This relation needs to be carefully specified and modelled.

5. What metrics are suitable for measuring subjective complexity in an enterprise architecture?

Based on the attributes found by previous research questions, appropriate metrics to measure these will be proposed.

6. How can both types of metrics be combined to create a unified model for enterprise architecture complexity measurement?

Combining the objective and subjective complexity metrics in a single model for enterprise architecture complexity leads to the complexity model defined in the main research question.

Figure 1.4 shows the relation between these research questions and enterprise architecture complexity.

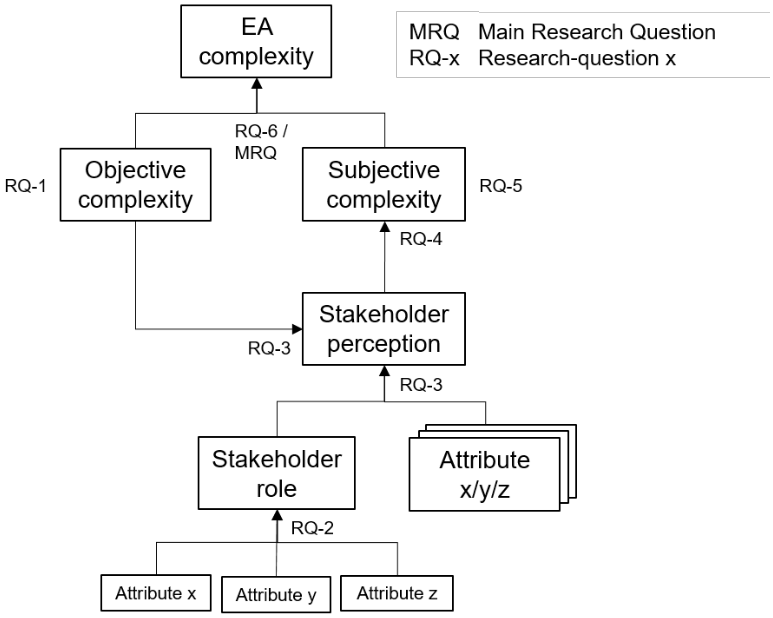


Figure 1.4: Research questions relations

1.4.2 Relevance

Consolidating the existing research on architectural complexity measurement and complementing this with subjective complexity has both an academic and practical relevance. Firstly, the current efforts on enterprise architecture complexity measurement are dispersed: many metrics are suggested by different studies. Consolidating the existing research will help to create an insight of the current state of the art. Furthermore, complexity research in enterprise architecture seems to be developing, but still incomplete. Schneider et al. (2014) observes an underrepresentation of the subjective complexity dimension in existing literature, which was verified in the structured literature review of this research. No more than 2% of existing enterprise architecture complexity metrics consider subjectivity (see section 2.3 “Objective complexity metrics”, specifically Table 2.4). Extending the state of the art on complexity research by studying a dimension that has been little-researched will contribute to the existing body of literature.

Additionally, this research has great potential relevance in practice. Excessive complexity in the architecture of an enterprise or its IT landscape has been found to have a series of negative effects. Figure 1.5 presents the negative effects of excessive complexity found by five empirical studies (Beese et al., 2016; Beese, Kazem Haki, Aier & Winter, 2017; Mocker, 2009; Schmidt & Buxmann, 2011; Wehling et al., 2017).

In an enterprise, many stakeholders are involved with an architecture and its development, ranging from C-level executives and lower management, to architects and developers. Each of these will have their own view on the architecture: business executives may focus on its value delivery, management on its functionalities and costs, architects on its maintainability and developers on its flexibility. Every stakeholder will therefore have a different perception of its complexity. Lack of understanding among stakeholders, which can be caused by a different perception of the architecture’s complexity, may be a cause of disagreement and resistance to change. This can lead to the mismanagement of the architecture: responsible stakeholders might take incorrect or ineffective decisions based on their perception of the complexity of the architecture. Exploring the subjective dimension of complexity will help to better understand how this complexity is enacting among the different stakeholders involved. In turn, this can help organizations to manage their enterprise architecture more effectively (Jochemsen et al., 2016).

Ultimately, this could have a positive influence on the effects influenced by enterprise architecture complexity, as visualized in Figure 1.5.

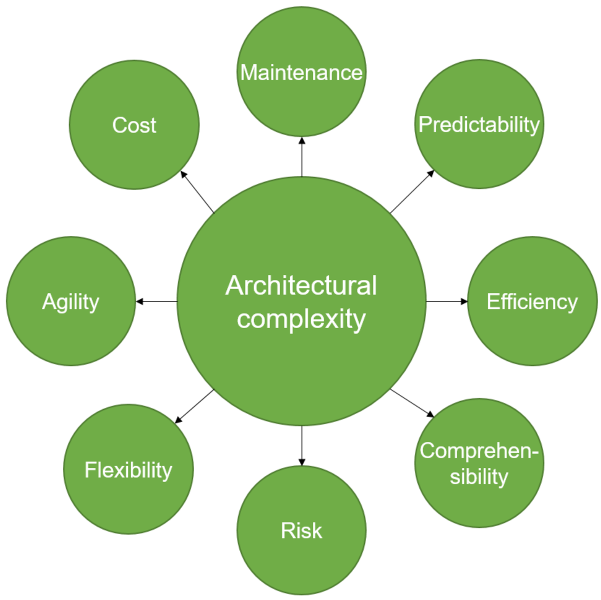


Figure 1.5: Negative effects of architectural complexity in EA

1.4.3 Research process

Answering the research questions posed earlier requires the design of a complexity measurement model. In Information Systems (IS) research, a widely-accepted methodology for the design of IS artifacts is Design Science. This methodology structures the process of investigation of the context, the design of an artifact and its validation. This research adopts the Design Science Methodology (DSM) introduced by Wieringa (2014) as overarching research process.

The DSM introduces the design cycle as a basis for the development of IS artifacts, which consists of three stages. The **problem investigation** prepares the design of an artifact by learning more about the problem it aims to solve and the context in which is operates. Next, the **treatment design** encompasses the actual design of the artifact. Finally, the **treatment validation** checks whether the artifact helps to achieve the desired goals. Note that the DSM is an overarching research process, and the steps described are accomplished through the use of several other methodologies. Figure 1.6 visualizes the entire research process and specifies the appropriate methodologies used for each step of the design cycle. These methodologies are introduced below, and described in more detail in chapter 3 “Methodology”.

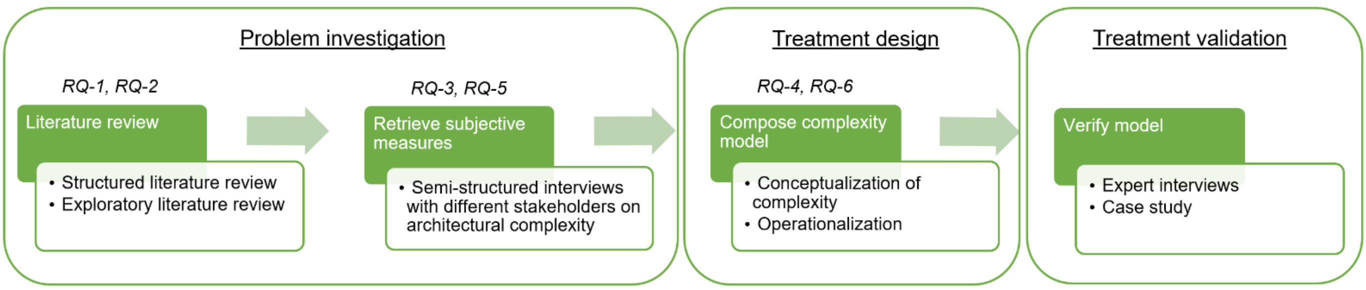


Figure 1.6: Research process flowchart

First, a structured literature review has been conducted to answer the first research question. Since objective complexity metrics have already been described in existing literature, they will be accumulated through a literature review. Through the structural approach by Kitchenham & Charters (2007), all metrics proposed by literature have been searched and analyzed. Data was extracted in a systematic way, using the concept matrix as suggested by Webster & Watson (2002). Based on these results, appropriate metrics have been selected for further use in the complexity model. By structurally searching all available literature and extracting the appropriate objective complexity metrics, broad applicability and academic support of these metrics can be ensured. An exploratory literature review has been conducted on the second research question. Stakeholders are an essential element to consider in subjective complexity, and their definition should be applicable in the different contexts of enterprise architecture. More details on the methodologies used for literature can be found in section 2.1 "Literature review methodology".

The third and fifth research questions have been answered empirically. Since very little literature considers the subjective notion of complexity in enterprise architecture, the attributes contributing to this are unknown. To identify these attributes, semi-structured interviews have been held with the different stakeholders found during the literature review. Details on the methodology used for these interviews can be found in section 3.2 "Interviews". The interview protocol used is added in Appendix C "Interview Guide". After identifying the attributes influencing subjective complexity, appropriate metrics to measure these have been designed according to the template introduced by Vasconcelos et al. (2007). These metrics have been based on the results of both the literature review and the interviews.

From this, a measurement model has been designed in order to answer research questions four and six. This was, in turn, done in two steps. The first and most important step is the conceptualization of complexity. Based on the data found in the previous parts of this research, descriptive inference was used to build a conceptual model of enterprise architecture complexity. This entails the identification of constructs influencing complexity and defining their relationships (Wieringa, 2014). Next, each of the constructs identified has been operationalized through the design of metrics. Using these metrics, the constructs influencing complexity can be measured, thus creating a measurement model for enterprise architecture complexity. A detailed explanation of this process can be found in section 3.3 "Model design".

Finally, the composed measurement model has been validated. This was done using expert interviews and a case study. During the expert validation, three experts were asked to judge the correctness of the designed model, and make suggestions for its improvement. Following its optimization using the expert feedback, the model was applied in practice during a case study. More details on the validation of the measurement model can be found in section 3.4 "Validation".

1.4.4 Research structure

This research is structured in several chapters. Chapter 2 presents the findings from the literature review that was conducted. Chapter 3 gives a detailed description of the methodologies that were used in the chapters that follow. Chapter 4 presents the results obtained from the interviews conducted during this research. In chapter 5, results from the literature review and interviews are combined to create a measurement model for architectural complexity. Chapter 6 describes the validation of this measurement model, using expert interviews and a case study. Chapter 7 interprets the results obtained, elaborates on their implications, discusses several limitations and sets forth possible directions for future research. Finally, chapter 8 concludes the research by answering the research questions posed in this chapter.

2. Literature Review

2.1 Literature review methodology

A literature review can serve multiple purposes in research, dependent on the question it attempts to answer. This research aims to answer multiple research questions, which require a different approach to literature review. According to Schneider et al. (2014), there is a substantial amount of literature to be found on the measurement of objective complexity in enterprise architecture, whereas subjective complexity has little to no research available. In this section, a structured literature is performed to review all available literature on complexity measurement, and identify and classify the available complexity metrics. The unexplored area of subjective complexity measurement requires an exploratory literature review to investigate the existing theories and information available on the topic (Adams, Khan, Raeside & White, 2007). Below, a methodology is provided for both types of literature review.

2.1.1 Structured literature review

Conducting a structured literature review should be a structured and rigorous process, and result in an exhaustive overview of all available literature on the topic. An essential step in this process is the creation of a protocol, explicating the steps to be taken in the review (Kitchenham & Charters, 2007). Such a protocol was drafted based on the works of Kitchenham & Charters (2007) and Webster & Watson (2002). It was used to guide the search and selection of, and data extraction from, relevant studies. The review protocol can be found in Appendix A “Structured Literature Review Protocol”. This protocol was used to extract the existing metrics for complexity measurement, and can be found in section 2.3 “Objective complexity metrics”.

2.1.2 Exploratory literature review

According to Adams et al. (2007), an exploratory literature review aims at finding the existing theories, empirical evidence and research methods related to a certain research topic. The goal of the review is not so much to provide a comprehensive meta-analysis of the existing literature, like a structured review aims to; rather it aims to provide a better insight in the existing theories, methods and data on the topic. This can be used as a theoretical foundation for further research.

In contrast to a structured literature review, an exploratory literature review does not aim to answer a research question, but rather focuses on a research topic. The second part of this review, based on the results of the structured literature review, revolves around the topic of subjective complexity measurement in the context of enterprise architecture. The search strategy used for this review is presented in Figure 2.1. After searching the databases listed in section A.3.1 “Database search” with a list of initial search terms, relevant literature that contributes to the research topic were selected. Based on the concept-centric approach of Webster & Watson (2002), relevant concepts were identified and described. Using this literature, search terms may be refined, or more search terms may be added. This iterative process was repeated until no more relevant literature was found, and eventually resulted in the identification of several concepts as described in this chapter. An overview of the search terms used can be found in Table 2.1.

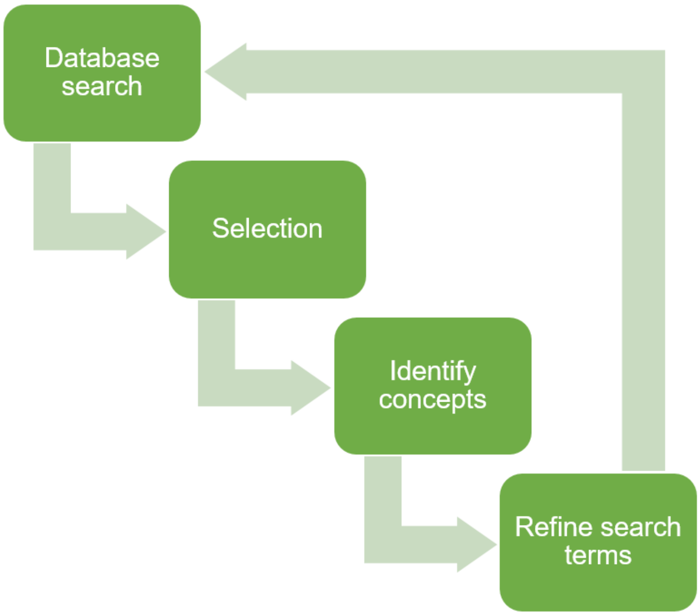


Figure 2.1: Exploratory search strategy

Search term		Search term		Search term
complex	AND	systems		
complex	AND	systems	AND	characteristics
”complex systems”	AND	”enterprise architecture”		
“enterprise architecture”	AND	(stakeholder*	OR	“stakeholder framework”)
“enterprise architecture”	AND	stakeholder	AND	perception
subjective	AND	complexity	AND	“enterprise architecture”
cognitive	AND	complexity	AND	“enterprise architecture”
complexity	AND	perception	AND	“enterprise architecture”
(subjective	OR	cognitive)	AND	complexity
cognitive	AND	complexity	AND	architectur*
requisite	AND	complexity		
optimal	AND	complexity	AND	“enterprise architecture”
optimal	AND	system	AND	complexity

Table 2.1: Search term overview

2.2 Complex systems

Introduced in the 1920s by Ludwig von Bertalanffy, General Systems Theory argues to view the world in terms of systems. A system can be defined as "a set of elements standing in interrelation among themselves and with the environment" (Von Bertalanffy, 1972, p.417). The theory provides a conceptual framework to approach science, namely by researching a system in relation with itself and its environment (Von Bertalanffy, 1975). This approach was necessitated by the increasing complexity of systems in modern technology (Von Bertalanffy, 1968). The research on these complex systems advanced the field of complexity science, looking for a theory to unify the research on complexity in different fields, and studying the characteristics and behaviour of complex systems (Kandjani et al., 2014).

Research on complex systems extends the General Systems Theory and thus focuses on (complex) systems in their environments. A substantial amount of research has been done on the measurement of these complex systems, and could therefore be applicable to enterprise architecture. That is, if an enterprise architecture were to be considered a complex system. To determine whether the research in complexity science is applicable to enterprise architecture, four studies have been considered that provide one or more characteristics that make a system complex, shown in Table 2.2.

Study	**Characteristics**
Ottino (2003)	1. Displays organization without any external organizing principle being applied.
Norman & Kuras (2006)	1. Large number of useful potential arrangements of its elements; 2. Elements can change when interacting with their neighboring elements; 3. Structure and behavior is not deductible from its components; 4. Its own complexity is increased, when given a steady inflow of resources; 5. Independent change agents are present.
Ladyman, Lambert & Wiesner (2013)	1. Nonlinearity; 2. Feedback; 3. Spontaneous order; 4. Robustness and lack of central control; 5. Emergence; 6. Hierarchical organisation; 7. Numerosity.
Cilliers (1998)	1. Large number of elements; 2. Elements interact dynamically; 3. Every element influences, and is influenced by, quite a few others; 4. Interactions are non-linear; 5. Interactions have a limited range; 6. There are loops in the interaction (recurrency); 7. Interacts with its environment; 8. Operates far from equilibrium; 9. Evolves through time, and its past is co-responsible for its present behavior; 10. Each element is ignorant of the behavior of the system as a whole.

Table 2.2: Complex system characteristics

Applying these characteristics to an enterprise architecture shows that it does not fulfill all characteristics of a complex system. For example, the structure and behavior of an enterprise architecture often is, or should be, deductible from its components. It does not show spontaneous order, nor does it increase it own complexity. Most importantly, an enterprise architecture does not display self-organization, which is often considered an important aspect of complex systems. Consequently, despite adhering to many of the characteristics listed in Table 2.2, an enterprise architecture cannot be considered a complex system as defined in literature.

However, these characteristics can also be applied to the environment of an enterprise architecture, the enterprise itself. Doing so shows that an enterprise is clearly a complex system, which is confirmed in literature (Kandjani, Bernus & Nielsen, 2013; Kandjani et al., 2014; Bar-Yam, 2003). Kandjani et al. (2014) argue for a classification of complexity in complex systems, providing three aspects of complexity:

1. Complexity of the system's function;
2. Complexity of the process creating the system;
3. Complexity of its architecture.

This third complexity aspect relates to enterprise architecture. Therefore, although not being a complex system itself, enterprise architecture does contribute greatly to the complexity of its environment, the enterprise. Consequently, results from this study can be used not only to identify the complexity of enterprise architectures, but has applications in research on the complexity of enterprises as well.

2.3 Objective complexity metrics

2.3.1 Search and selection

2.3.1.1 Database search and selection

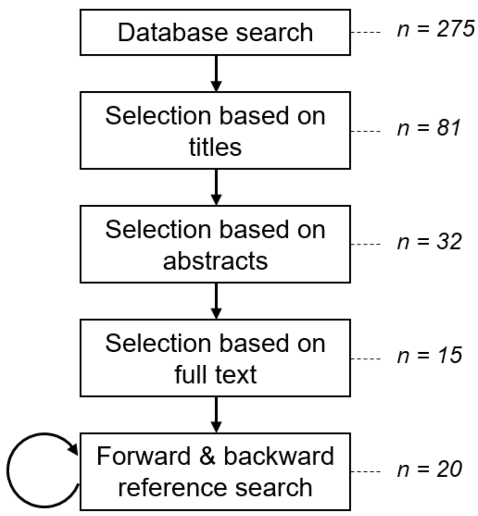


Figure 2.2: Structured literature review results

The results of the search and selection process, as described in section A.3 "Search strategy", are visualized in Figure 2.2. The first database search with the defined search terms resulted in a total of 275 unique results. This initial set of studies was refined based on the predefined selection criteria, as found in Table A.1. First, the selection criteria were applied based on study titles. This resulted in the exclusion of 194 studies, most of which obviously did not meet the first inclusion criteria and concerned different types of architecture. For example, many studies focused on the architectural complexity of (micro)processors. Of the remaining 81 studies, abstracts were read. Here, another 49 studies were excluded. Most of these were discarded because they did not meet the second inclusion criteria, whereas some met one of the exclusion criteria. Finally, for the remaining 32 studies, the full text was obtained and read. By carefully applying all selection criteria on these texts, another 17 studies were excluded. With this, a set of 15 studies remained that met all criteria. It is important to note that in the above process, the selection criteria were applied conservatively, in order to ensure no relevant studies were missed. This means that when in doubt, studies were included.

2.3.1.2 Reference search

After the first database search and the refinement of the selection, a forward and backward reference search was performed. First, all references in the remaining selection were scanned to search for additional relevant research, meaning those studies that meet the selection criteria. Subsequently, using the previously listed databases, all studies that referenced to the selected studies were evaluated. The reference search is an iterative process, meaning the search was repeated until no new studies were found. This step found another 5 relevant studies. Therefore, the final selection of relevant research contained a total of 20 studies. This selection can be found in Table B.1 in Appendix B.1 "Review paper selection".

2.3.2 Quality assessment

The final set of papers were assessed on their quality based on the the checklist provided by Kitchenham & Charters (2007). No papers were excluded in the process.

2.3.3 Data extraction

The process of data extraction from the resulting set is based on the concept matrix of Webster & Watson (2002), and can be summarized by Figure 2.3.

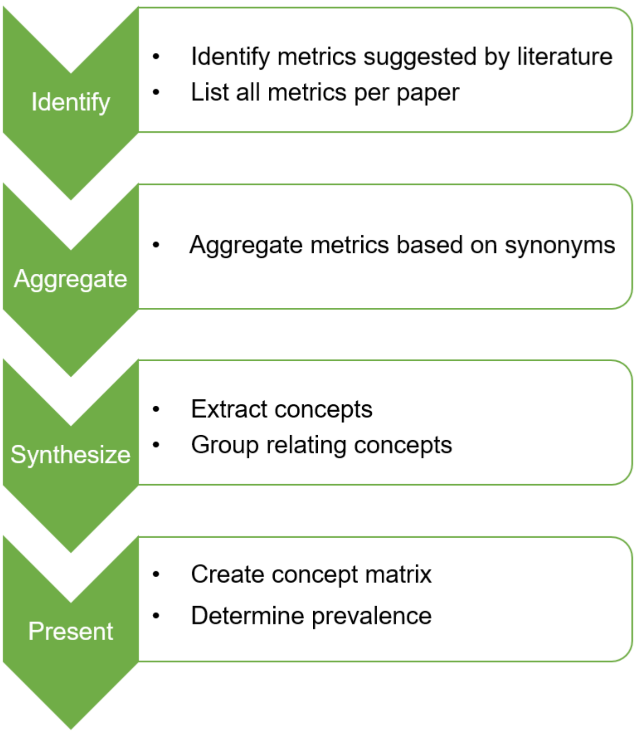


Figure 2.3: Data extraction process

First, all suggested metrics were identified and listed for each paper. Metrics that use synonyms and/or are obviously highly similar, were aggregated to form a list of forty-two unique metrics that are suggested by literature at least once. Next, different concepts were extracted from these metrics, based on the author's judgment. This resulted in the identification of twelve concepts, which can be combined into four concept-groups. Using this data, a concept matrix was created, and the prevalence of the concepts among literature was determined.

2.3.4 Results

2.3.4.1 Metric matrix

The aim of this structured literature review was to answer the following research question, as defined in Appendix A "Structured Literature Review Protocol":

1. Which existing metrics are being used to measure complexity in an enterprise architecture, and how can these be classified using the complexity dimensions by Schneider et al. (2014)?

In order to answer this question, all metrics suggested by the selected papers were identified and listed. This included a lot of similar results, which were aggregated in a resulting list of forty-two metrics. In Table B.2 in Appendix B.2 "Metric matrix", the metric matrix can be found. This matrix contains all identified complexity metrics, showing their relation to the included papers and their resulting prevalence. As the matrix shows, the metric *number of relations* is the most prevalent, being used in 55% of the papers, closely followed by *number of elements*, with a prevalence of 40%. Of all forty-two metrics identified, thirteen were used in more than one paper, together accounting for about two-thirds of the total number of mentioned metrics. Following the metric matrix, Table B.3 in Appendix B.3 "Metric description" offers a list providing a short description of each metric.

2.3.4.2 Concepts

Based on the list of metrics, twelve concepts were extracted, which can be divided into four groups, as shown in Table 2.3. Table B.4 in Appendix B.4 "Metric-concept overview" shows an overview of all metrics and their assigned concept(s). Below, a short description of each concept is provided.

Graph-based	**Functional**	**Non-functional**	**Other**
Elements and relations	Functions	Quality	Expert opinion
Patterns	Redundancy	Reliability	Cost
	Application-data	Conformity	
	Hardware-data	Service time	

Table 2.3: Identified concepts

Graph-based
The metrics related to these concepts propose to see an architecture as a graph, consisting of elements and their relations. Graphs can be made on multiple abstraction-levels, therefore the definition of an element and relation is dependent on context. The concept *elements and relations* uses data about the graph's elements to measure the complexity of the graph, and thus the architecture it represents.

The *patterns* concept identifies existing design-patterns within the graph by using pattern-matching algorithms. An architecture that is based on an existing design-pattern is easier to comprehend, and pattern-matching can therefore be used to measure complexity.

Functional

These concepts look at functional properties of the system and/or its elements. *Functions* relates to the functions a system can perform, or the processes to accomplish these. *Redundancy* occurs when the functionality of multiple elements overlap, meaning the architecture is larger than necessary. *Application-data* and *hardware-data* refers to data or statistics of the software or hardware deployed in the architecture. These concepts are related to a specific architectural domain: the application domain and infrastructure domain.

Non-functional

Non-functional concepts relate to the non-functional properties of an architecture. *Quality* and *reliability* look at the quality and reliability requirements set, since high requirements in these aspects could lead to higher complexity. *Conformity* evaluates to what extent elements conform to standardization. *Service time* is a concept that looks at the required time to perform a specific service or function.

Other

Finally, *expert opinion* uses the opinion of experts to provide weight factors for other metrics. *Cost* looks at the cost and effort required to make changes to the architecture.

2.3.4.3 Concept matrix

As a next step, a concept matrix was created, which can be found in Table B.5 in Appendix B.5 "Concept matrix". This matrix creates an overview of the twenty papers of the literature review and the concepts they use to measure architectural complexity. Included is again the total number of papers that mention each concept, and their prevalence. This shows that the elements and relations concept is by far the most popular, being used in 80% of the papers. After this, conformity is most prevalent (35%), followed by functions and application-data (both 30%). The concept-groups show that graph-based metrics are highly used (with a prevalence of 85%), followed by functional properties (55%). Half the papers use a single concept to measure complexity, whereas the other half combines multiple concepts, often from multiple groups.

2.3.4.4 Complexity dimensions

Schneider et al. (2014) researched the different notions of complexity in enterprise architecture literature, and designed a framework to classify existing metrics. In order to gain insight in the state of the art of complexity measurement in the context of enterprise architecture, all identified metrics, as can be found in Appendix B.2 "Metric matrix", have been classified according to their dimension(s) of complexity. The resulting classification matrix can be found in Appendix B.6 "Complexity classification", and is summarized by Table 2.4. This table shows the total count of identified metrics that can be categorized as that specific dimension. Next, its prevalence among the total of fourt-two metrics is shown. Please note that some metrics were assigned more than one value along the same dimension, for example having both quantitative and qualitative aspects, sometimes resulting in a total prevalence exceeding 100%.

Prevalence	Count	Dimension	Dimension	Count	Prevalence
98%	41	Objective	Subjective	1	2%
100%	42	Structural	Dynamic	1	2%
98%	41	Quantitative	Qualitative	4	10%
86%	36	Ordered	Disordered	7	17%

Table 2.4: Complexity dimension distribution

As is apparent from this data, the most common complexity metric (CM) can be described as follows:

$$CM = (objective, structural, quantitative, ordered)$$

This quadruple describes 79% of the enterprise architecture complexity metrics in the current literature.

2.3.5 Discussion

In this section, the current state of the art on the measurement of architectural complexity was reviewed. Existing metrics have been broken down into concepts, and classified into complexity dimensions. Most of these metrics propose to look at an architecture as a graph; 17 out of 20 studies found made use of graph-based metrics in some form. However, over half of the studies found combined multiple types of metrics to get a thorough view of an architecture's complexity. This indicates that, since complexity is such a comprehensive property, multiple metrics with different viewpoints should be combined. The division of existing metrics into the complexity dimensions shows a very low variation: 79% of the metrics can be described with the quadruple $(objective, structural, quantitative, ordered)$. Moreover, 98% of the metrics found focused on the objective dimension of complexity, confirming the observation by Schneider et al. (2014) that subjective complexity metrics are yet to be found in the context of enterprise architecture.

The list of metrics found can be employed to extract metrics regarding the objective complexity of enterprise architecture. Of course, employing the entire list of metrics in an enterprise architecture complexity model would neither be efficient nor feasible. Therefore, a selection of metrics has to be chosen to be included in the model. The selection of objective metrics is described and effectuated in section 3.3.2 "Operationalization".

2.4 Stakeholders

Subjective complexity, per definition, is dependent on the perception of a stakeholder, and therefore different for each stakeholder. In order to measure the subjective complexity of an architecture, the different stakeholders that interact with it should be defined. Making this definition context-agnostic is an important requirement to ensure applicability in different organizations and contexts. To make the concept of stakeholder applicable in a model intended for complexity measurement, attributes should be identified that collectively constitute any enterprise architecture stakeholder.

2.4.1 Stakeholder definition

Stakeholders have been an important part of management theory since the publication of *Strategic Management: A Stakeholder Approach* by Freeman (1984). In this influential work, Freeman introduces the stakeholder as a concept and important factor in enterprises. Ever since, stakeholders and their management have been considered an essential aspect of organizations.

Over the past decades, numerous definitions of the word “stakeholder” have been proposed, addressing different organizations and contexts. In enterprise architecture, the leading definition of stakeholder is adopted from the 42010 IEEE standard:

Definition 8. *Stakeholder:* Individual, team, organization, or classes thereof, having an interest in a system (ISO/IEC/IEEE, 2011).

In the context of enterprise architecture, this means that anyone who has an interest in the architecture and its results can be considered a stakeholder.

2.4.2 Stakeholders in enterprise architecture

In the study and practice of enterprise architecture, stakeholders play an important role. Insufficient involvement of business (or other organizational) stakeholders has been identified as one of the main threats to enterprise architecture (Gartner, 2009). Collaboration with stakeholders is considered challenging, but a requisite for successful implementation of enterprise architecture (van der Raadt et al., 2008; Niemi, 2007; Puspitasari, 2016). Since different stakeholders have different and often conflicting needs and perspectives (Niemi, 2007), identification of the various types of stakeholders involved with enterprise architecture is an important first step in analyzing their perception.

2.4.2.1 Views and viewpoints

In enterprise architecture, stakeholders are seldom interested in the full scope of the architecture, including all its details. In order to cater the needs of different stakeholders, views were introduced. A view is a representation of the architecture intended for a specific (group of) stakeholder(s) (Lankhorst, 2017). By constructing views for different stakeholders, only relevant information can be included, leaving out unnecessary details. To create these views, viewpoints are composed. A viewpoint contains all information needed to construct a specific view; it is a template from which an individual view can be developed, establishing the purpose and audience of a view, and dictating the techniques used for its creation (Lankhorst, 2017).

2.4.3 Stakeholder classification frameworks

Over the past years, numerous frameworks have been proposed for the classification of stakeholders. However, leading stakeholder classification models, such as Mitchell, Agle & Wood (1997), are neither intended for, nor validated in the context of, enterprise architecture. Furthermore, most stakeholder theories are mainly focused on management and decision making. Although this does play a part in enterprise architecture, modelling complexity perception requires a more comprehensive approach to stakeholder classification. Recently, enterprise architecture research has increased its focus on stakeholders, identifying collaboration as a requisite for success.

2.4.3.1 Role

Niemi (2007) aims to create a holistic view on the stakeholders of an enterprise architecture. By analyzing a large body of existing literature on the subject and supplementing this with interviews, he proposes three roles, which can be used to classify stakeholders:

- **Producers** are the stakeholders that carry out the planning and development of the enterprise architecture. They usually do not use the architecture other than for their primary work, and are little

involved with its management. Producers aim to satisfy the facilitators' and users' requirements.
Example: (enterprise/domain/solution) architect, project manager.

- **Facilitators** are the stakeholders involved with the governance, management, maintenance or sponsorship of the enterprise architecture. Contrary to producers, they do not directly conduct planning or development of the architecture. They differ from users in the sense that their work affects the architecture.
Example: board of directors, project management office.
- **Users** are the stakeholders that utilize enterprise architecture work and its products in their daily work. In contrast to the other roles, they do not carry out work that directly affects the architecture. However, they can be involved in architecture work by providing requirements or feedback.
Example: application developer, IT operator.

2.4.3.2 Domain-level matrix

In their paper *Stakeholder Perception of Enterprise Architecture*, van der Raadt et al. (2008) propose a matrix for the classification of stakeholders. For this, they identify two attributes:

- **Level** refers to the organizational level a stakeholder operates in. Four levels can be distinguished:
    - At *enterprise* level, management focuses on the strategy of the architecture, making decisions regarding the future state of the architecture. This level typically operates at C-level.
    - At *domain* level, domain owners and change managers coordinate and mange the programs running in their domain, and IT focuses on the domain architecture. This level operates at business unit-level.
    - At *project* level, stakeholders run project and implement changes, executing the decisions made by the first two levels.
    - At *operational* level, stakeholders are responsible for the stability and continuity of the operational environment, the day-to-day operations, and the maintenance and optimization of platforms.
- **Domain** refers to the architectural domain that constitute enterprise architecture. These domains are *business*, *data*, *application* and *technology*, and are described in detail in section 1.2.1 "Architecture".

	Business	**Data**	**Application**	**Technology**
Enterprise	CEO	CIO	Enterprise architect	CTO
Domain	Head of business unit	Domain architect	Domain architect	Platform manager
Project	Project manager	Information analyst	Solution architect	Infrastructure engineer
Operational	Business process engineer	Data administrator	Application administrator	Infrastructure administrator

Table 2.5: Example stakeholders and their classification in van der Raadt et al.'s framework

2.4.3.3 Zachman Framework

The Zachman Framework is one of the most widely-known enterprise architecture frameworks. In this framework, enterprise architecture is viewed in a matrix of perspectives and dimensions. The columns of the matrix focus on different dimensions: data, function, network, people, time and motivation. The rows represent the view of different stakeholders, taking into consideration those who are involved in the planning, conception, building, using and maintaining activities of an enterprise architecture (Pereira & Sousa, 2004):

- The **planner** focuses on positioning the architecture in its environment and specifies its scope.
- The **owner** focuses on the management and application of the architecture.
- The **designer** composes the requirements of the architecture, ensuring it will fulfill the owner's expectations.
- The **builder** manages the creation of the actual products.
- The **subcontractor** fabricates out-of-context components based on the specification of the builder.

These views can be identified as roles focused on the production and facilitation of the enterprise architecture. They can be mapped to the roles of producer (planner, designer, builder, subcontractor) and facilitator (owner) introduced earlier, and can be viewed as a specifications of the more generally applicable model of Niemi (2007).

2.4.3.4 Viewpoint classification framework

Views and viewpoint in enterprise architecture are used to create a representation of the architecture intended for a specific (group of) stakeholder(s). Steen, Akehurst, ter Doest & Lankhorst (2004) propose a classification framework for viewpoints, aimed at enterprise architecture. In this framework, they include a classification of stakeholders and their goals:

- **Design** is focused on stakeholders in the design process. These viewpoints focus on the details of a specific architectural domain, or the interdependencies between domain.
- **Decide** is focused on manager in the process of decision making, and offers insights into different architectural domains and their relations.
- **Inform** is focused on aiding other stakeholders to understand the architecture, and obtain their commitment.

This categorization of viewpoints can be mapped to the framework of roles by Niemi (2007):

$$\text{Design} \Rightarrow \text{Producer}$$
$$\text{Decide} \Rightarrow \text{Facilitator}$$
$$\text{Inform} \Rightarrow \text{User}$$

This supports the categorization of roles by Niemi (2007) and provides the possibility to create viewpoints focused on these roles.

2.4.4 Stakeholder attributes

Based on the classification frameworks introduced above, some important stakeholder attributes are identified that can be used to classify stakeholders. Combining the frameworks of Niemi (2007) and van der Raadt et al. (2008) produces a stakeholder classification based on three attributes, each with a set of possible values. This is visualized in Figure 2.4 "Stakeholder attributes and their possible values".

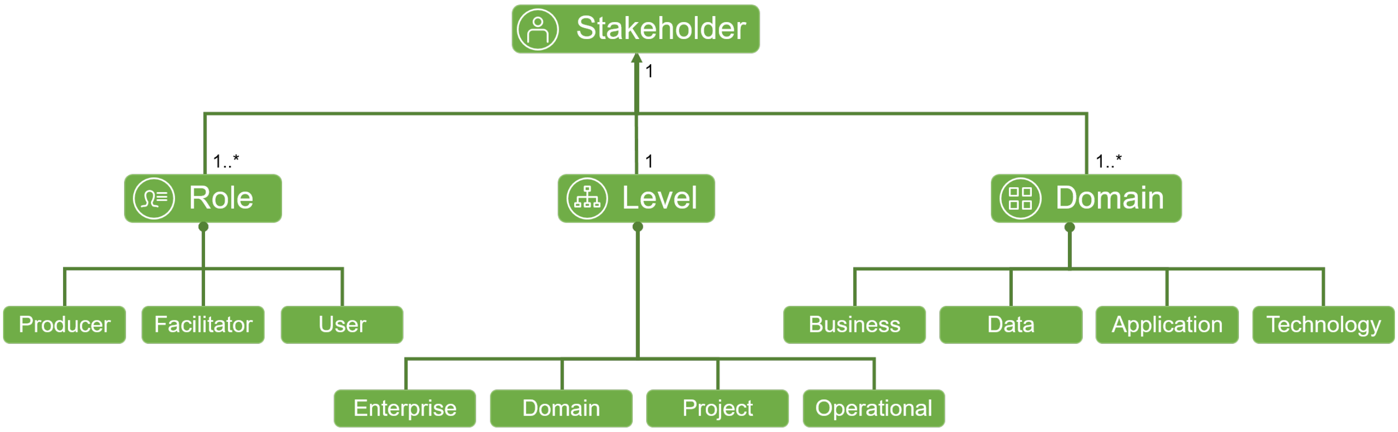


Figure 2.4: Stakeholder attributes and their possible values

2.5 Subjective complexity

Enterprise architures are socio-technical systems, meaning their functioning heavily depends on their interaction with stakeholders. This means that subjective aspects, stemming from stakeholders interacting with the architecture, are an inherent part of an enterprise architecture. Capturing the subjective aspects of complexity can therefore help organizations to manage their IT landscape more effectively (Jochemsen et al., 2016). Subjective complexity is dependent on the perception of stakeholders, meaning it is essential to first understand perception itself.

Cognitive informatics is the science studying human perception and the internal processing of information. It is an interdisciplinary research area focusing on cognition, problem understanding, information processing and artificial intelligence (Singh & Misra, 2006). Human cognition defines the property of comprehension or understandability, which in turn influences whether an entity is perceived to be complex or simple (Singh & Misra, 2006). Cognitive informatics considers the transfer of information essential in this process. It proposes “information” as the third essence making up the natural world, supplementing the traditional essences matter and energy (Wang, 2007), and defines information as follows:

Definition 9. *Information:* A generic abstract model of properties or attributes of the natural world that can be distinctly elicited, generally abstracted, quantitatively represented, and mentally processed (Wang, 2007, p.4).

The latter part of this definition - elicitation, abstraction, representation and processing - can be seen as the perception of information. Perception can therefore be defined in terms of information processing:

Definition 10. *Perception:* A set of interpretive cognitive processes of the brain at the subconscious cognitive function layers that detects, relates, interprets, and searches internal cognitive information in the mind (Wang, 2007, p.7).

In cognitive informatics, complexity is considered to be related to the easy of comprehension. Bieri (1955) states that a system of constructs that has a highly differentiated interpretation among persons, i.e. one that is difficult in interpret and thus interpreted differently, is cognitively complex. The application of cognitive complexity in enterprise architectures has not yet been studied[1]. However, cognitive software complexity has been studied for quite some time, and has resulted in a series of cognitive complexity metrics (Cant et al., 1995; Singh & Misra, 2006; Wang, 2006; Wijendra & Hewagamage,

[1] Extensive literature searches on the application of cognitive complexity in enterprise architecture yielded no results.

2016). Application of cognitive informatics in the field of enterprise architecture might take lessons from its results in software design. In the field of software engineering, several cognitive complexity metrics have been formulated. Most studies leverage the insights of cognitive informatics by supplementing traditional complexity metrics with cognitive weights based on the understandability of basic software patterns, such as iteration or concurrency. This understandability is measured in terms of the relative time it takes a test group to understand an instantiation of a pattern. Cant et al. (1995) defines the mental processes a software engineer uses when interpreting code, which are searching ("tracing") and processing ("chunking"). The cognitive complexity of a software element is expressed by the time taken to trace and chunk the element and all its nested elements.

Before introducing these metrics, Cant et al. (1995) defines a theoretical classification of software complexity. Although the previously mentioned metrics are hard to transfer into architectural metrics, this theoretical basis introduced might prove useful for application in enterprise architecture. This classification, visualized in Figure 2.5, further specifies the cognitive complexity of software as consisting of problem complexity, stakeholder characteristics and structural (in this research: objective) complexity. These categories can be applied to enterprise architecture as well, hypothesizing that subjective complexity is influenced by (among others) problem complexity, stakeholder characteristics and objective complexity.

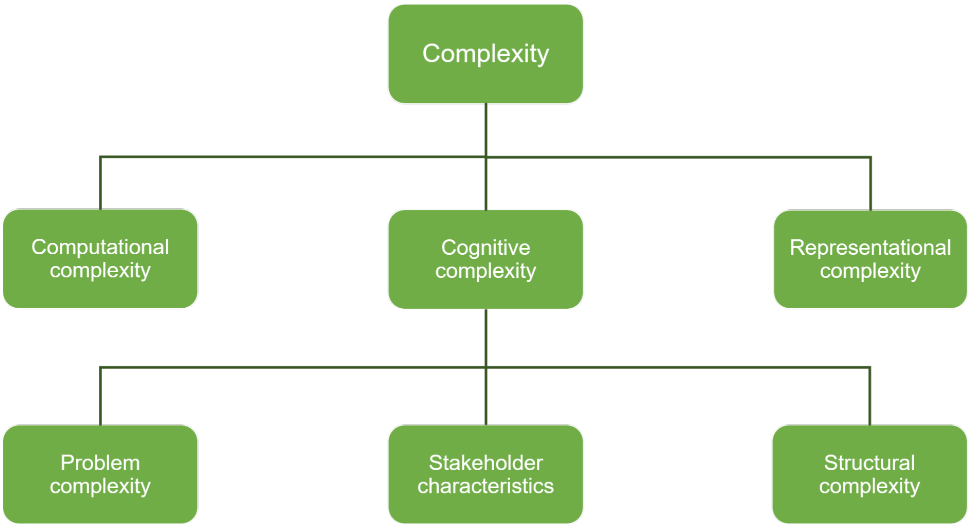


Figure 2.5: Classification of software complexity (Cant et al., 1995)

One of the stakeholder characteristics theorized by cognitive informatics to influence human perception is the way of information processing. Cognitive information is classified into four categories: knowledge, behavior, experience and skills (Wang, 2007). Since perception is stated to be the processing of cognitive information, it can be concluded that a stakeholder's ability of processing these types of information influences their perception of complexity. Although this is often thought to mean intelligence, Suedfeld & Coren (1992) found that there is no strong correlation between intelligence and complexity perception. Unfortunately, there are currently no concrete metrics to be found measuring the processing of these information types, other than reflective metrics such as time taken to understand a visual representation of the information.

Another aspect of complexity stated in the model proposed by Cant et al. (1995) focuses on representational complexity. Although this is not elaborated in their paper, several other studies focus on this phenomenon. Campbell, Halpin & Proper (1996) and Lantow (2014) find that the level of detail of documentation influences understandability. Moody (2009) states that the notation of visual form, in the

[1] Extensive literature searches on the application of cognitive complexity in enterprise architecture yielded no results.

case of enterprise architecture its documentation, greatly affects understanding. Moody defines several principles aimed at maximizing "cognitive effectiveness". These principles focus on four steps in processing visual notations:

- *Perceptual discrimination:* revolves around the detection of different features in images, such as color, shape or size. Based on these features, the brain parses the image into elements and their background.
- *Perceptual configuration:* based on visual characteristics, the structure and relations of the elements are inferred.
- *Working memory:* the information is temporarily stored for processing. Since the human working memory has very limited capacity, this is an important bottleneck in image processing.
- *Long-term memory:* for information to be transferred from working to long-term memory, it has to be linked to prior knowledge. Similarity to prior knowledge greatly influences speed and accuracy of processing.

Although the influence of documentation on enterprise architecture seems likely, there is no empirical data to support this yet. The existing research is focused on software models, and has contradictory findings. For example, Campbell et al. (1996) find that when model size increases, it becomes less comprehensible, which may be explained by the limited working memory of the brain. However, Lantow (2014) finds that a lower level of detail, and thus a smaller size of documentation, leads to lower understandability, and theorizes that this is due to the loss of important context when lowering the level of detail.

Concluding, the research on the perception of enterprise architecture complexity is limited. Existing research on cognitive informatics and cognitive software complexity does indicate some areas that may be of influence on subjective complexity, such as the information processing of the four types of cognitive information, and the visual notation of documentation. However, empirical research is needed to confirm these findings in the context of enterprise architecture, and to explore further variables influencing subjective complexity.

2.6 Requisite complexity

Complexity is in itself not necessarily a negative quality in a system. A large organization employing an enterprise architecture containing only a handful of applications, will most likely not achieve its best possible performance. This means that complexity management should not solely focus itself on the reduction of complexity; rather, it should aim to find the optimal level of complexity (Heydari & Dalili, 2012; Schmidt, 2015). Boisot & Mckelvey (2011) have captured this aim by recasting Ashby's Law of Requisite Variety into their *Law of Requisite Complexity*. Their law states that "to be efficaciously adaptive, the internal complexity of a system must match the external complexity it confronts" (Boisot & Mckelvey, 2011, p.279). The law can be captured in a simple graph defining internal and external complexity, as is visualized in Figure 2.6.

Every system, including enterprise architectures, should have a minimum level of complexity to deal with requirements derived from its environment (Schmidt, 2015). This required level of complexity in an enterprise architecture is essential to manage and achieve an organization's strategy and goals. When this minimal level of architectural complexity is not met - i.e. there is a complexity deficit - the flexibility and adaptability of the architecture is compromised (Heydari & Dalili, 2012). However, many enterprise architectures that do meet this minimum complexity also retain a level of unnecessary complexity, which

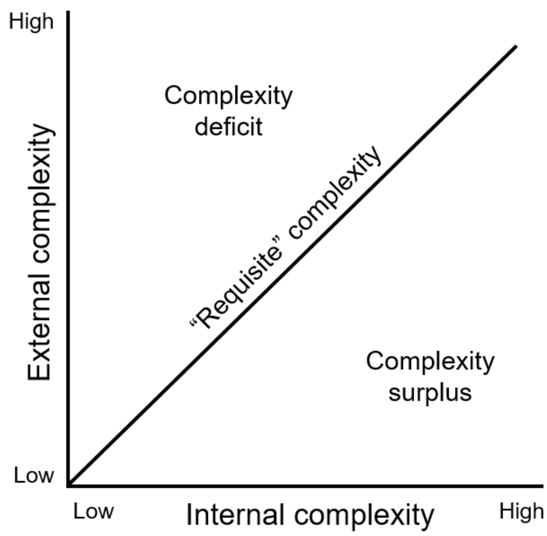


Figure 2.6: Requisite complexity: balancing between internal and external complexity

cannot be justified by the complexity of its environment (Wehling et al., 2017). This complexity surplus also harbours several threats to the organization, such as extra costs and lower stability (Heydari & Dalili, 2012).

Ultimately, the goal of enterprise architecture complexity management is to minimize complexity deficit or surplus, and to balance the architecture at "requisite", or optimal, complexity. Measuring complexity, as is this research's goal, is but a first step in this process. The question rises, however, how this optimal level of complexity can be determined. The environment of an enterprise architecture largely consists of the enterprise itself, which has a non-negligible level of complexity, as was found in section 2.2 "Complex systems". Being a complex system, the complexity of an organization - and thus the external complexity of its architecture - will be volatile. Although the exploration of the requisite complexity levels are not in scope of this research, it is an interesting next step in the optimization of enterprise architecture complexity.

3. Methodology

3.1 Conceptual model

An enterprise architecture is an artifact that operates in a certain context. One of its characteristics is complexity, which is influenced by both the architecture and its context. In order to measure architectural complexity, it is therefore essential to know what this context looks like, and what aspects could influence architectural complexity. This requires an abstract model of enterprise architectures and their context, defining the general concepts at play.

A conceptual model is an abstract model containing an interlinked set of concepts. Concepts are properties or characteristics associated with objects, events or people that can be generalized (Bhattacherjee, 2012). Combining the relevant concepts and defining their relations results in an abstract model applicable to all enterprise architectures, reflecting the existing knowledge on architectures and their environment. The goal of this conceptual model is to enable clear communication regarding its concepts, and to provide a basis for the design of other models (Wand, Monarchi, Parsons & Woo, 1995; Thomas, 2006). The conceptual model introduced below has been used as an overview of the current knowledge on architectural complexity, and as a basis for the design of a measurement model.

The International Organization of Standardization defines a conceptual model depicting system architectures in their context (ISO/IEC/IEEE, 2011). Lankhorst (2017) defines its own conceptual model, based on a different version of the same ISO standard. The model introduced in ISO/IEC/IEEE (2011) has been employed as a basis of the conceptual model to be designed, including minor adaptations inspired by Lankhorst (2017). This model features the concepts “enterprise” and “architecture”, along with several other concepts that collectively constitute their context. This conceptual basis is extended with several concepts introduced earlier in this research, the most important being “complexity”. The resulting conceptual model can be found in Figure 3.1.

As is clear from the conceptual model, complexity is a concept associated with architecture, perceived by stakeholders and measured by metrics. The goal of this research is to design a measurement model of the concept “complexity”, meaning it aims to find the relevant set of metrics for complexity. The measurement model is designed in several steps. First, interviews were conducted to determine the attributes that contribute to subjective complexity, thereby complementing the attributes found in the structured literature review; section 3.2 “Interviews” describes the methodology used. Jointly, these results form the input of the next step in the process: the design of the measurement model. The approach taken for this step is described in section 3.3 “Model design”. Finally, the designed model is validated, as described in chapter 6 “Validation”

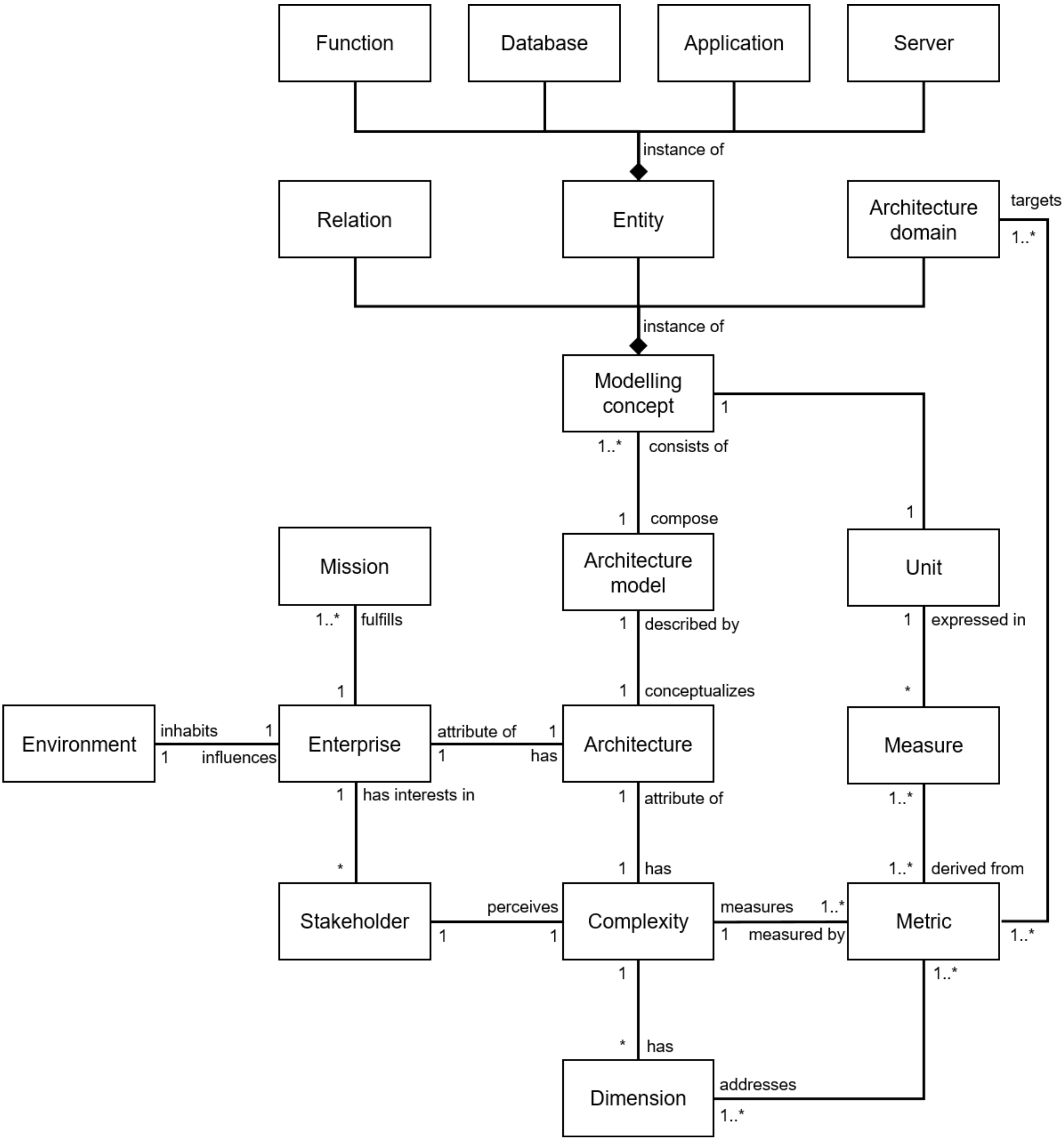


Figure 3.1: Conceptual model of enterprise architectures and their context

3.2 Interviews

3.2.1 Research design

Designing a model that describes subjective enterprise architecture complexity requires insight into the factors influencing complexity perception. This is reflected in the research question formulated in section 1.4.1: *Which attributes influence a stakeholder's perception of enterprise architecture complexity?* For the collection of data based on opinions, a qualitative research design is recommended (Denscombe, 2010). To answer the research question stated, twelve in-depth interviews were conducted. Though a wide range of interview strategies is available, semi-structured interviews are considered most appropriate when exploring the perception of participants (Dicicco-Bloom & Crabtree, 2006; Denscombe, 2010). Semi-structured interviews are organized around a set of open-ended questions formulated beforehand. However, the interviewer is free to seek further clarification from participants, with additional questions emerging from the dialogue of the interview (Doody & Noonan, 2013). To ensure data is collected from all participants in a similar way, an interview guide should be developed (Doody & Noonan, 2013). The interview guide contains a script for the opening and wrap-up of the interview, discusses a general flow of topics, and defines a set of open questions that should eventually support the answer to the main research question. The developed interview guide can be found in Appendix C "Interview Guide".

3.2.2 Context

3.2.2.1 Organizations

Niemi (2007) states that stakeholder concerns may differ per organization and per industry. This research aims to obtain interview results that are generalizable and applicable cross-industry. To this end, the twelve interviews have been conducted at four different organizations, all in different industries. The following industries have been identified: Consulting Services (CS), Financial Services (FS), Public Sector (PS) and Energy & Resources (ER). These industries have been chosen based on the contacts within Deloitte, and the feasibility of finding participants. Below, a short description of each organization is provided.

Organization A (CS)
The first organization to provide three interviews was Deloitte Consulting, department Enterprise Architecture. Three senior employees with extensive experience in enterprise architecture were interviewed. All participants have been active in different industries, providing insights in their specific situations, and further adding to the cross-industry applicability of the results.

Organization B (FS)
Organization B is a financial organization mostly active in the Dutch market, providing their service to several millions of customers. As with most large financial organizations, organization B has an extensive enterprise architecture to support their processes.

Organization C (PS)
Organization C is part of the public sector. As one of the largest Dutch governmental agencies, they are heavily influenced by (changing) legislature and national politics, causing a high environmental complexity. This unique dynamic can provide additional insight in comparison to the other (private) organizations.

Organization D (ER)
Organization D operates in the energy and resources market. Contrary to the other organizations, they are a relatively new player, having been founded less than five years ago. Due to their fast growth and "clean slate" in the digital era, they provide fresh insights into enterprise architecture and its complexity.

3.2.2.2 Roles

To gain insight into the complexity perception of different stakeholders within an organization, stakeholders should be classified, and their difference in complexity perception explored. When identifying stakeholders and their concerns, ISO/IEC/IEEE (2011) groups stakeholders based on their role. This research followed this approach, classifying stakeholders based on the roles identified in section 2.4.3.1 "Role", as shortly described below.

- *Producers:* carry out EA planning and development
- *Facilitators:* involved with governance, management, maintenance or sponsorship of EA
- *Users:* Utilize EA work and its products

At each of the four organizations, three participants were interviewed corresponding to the roles identified. The participants of organization A are consultants that have been involved in different organizations and experienced multiple roles, and are therefore categorized in all three roles. Table 3.1 poses an overview of the interviews conducted at the four organizations and the roles of the participants.

	A	**B**	**C**	**D**
Producers		1	1	1
Facilitators	3	1	1	1
Users		1	1	1

Table 3.1: Number of participants interviewed per role and industry

3.2.3 Data collection

All interviews were conducted face-to-face and audio recorded. On average, an interview took 55 minutes and 30 seconds. An overview of the participants and their job title can be found in Table C.1 in Appendix C "Interview Guide". The interview started with a few introductory questions regarding the participant's role and an explanation of some terminology, such as enterprise architecture and complexity. These questions were asked in the form of small-talk, to establish rapport between the interviewer and participant, which is an important aspect in one-on-one interviews (Dicicco-Bloom & Crabtree, 2006). After this, the interview was led by a series of key questions, additional questions and probes that were established before the interview. Additionally, questions were asked based on the responses of the participants. A full overview of the key questions can be found in section C.3.3 "Key questions". The key questions asked during the interview cover all concepts relating to complexity as defined in the conceptual model of Figure 3.1. Ensuring that all concepts are discussed is an important part of data collection (Babbie, 2010).

3.2.4 Data analysis

Interview data was analyzed using an inductive approach, based on the Grounded Theory Method introduced by Glaser & Strauss (1967). This approach argues for the development of theory based solely on the examination of data, establishing results that are fully grounded in data. In this specific case, the data to be analyzed is verbal, which is the process of "quantifying the subjective or qualitative coding of the contents of verbal utterances" (Chi, 1997, p.273). Several steps in the process of verbal data analysis were identified based on the work of Glaser & Strauss (1967), Chi (1997) and Burnard (1991), with the latter two specifically describing the process of the analysis of recorded data from semi-structured interviews. The steps followed in the data analysis process are visualized in Figure 3.2.

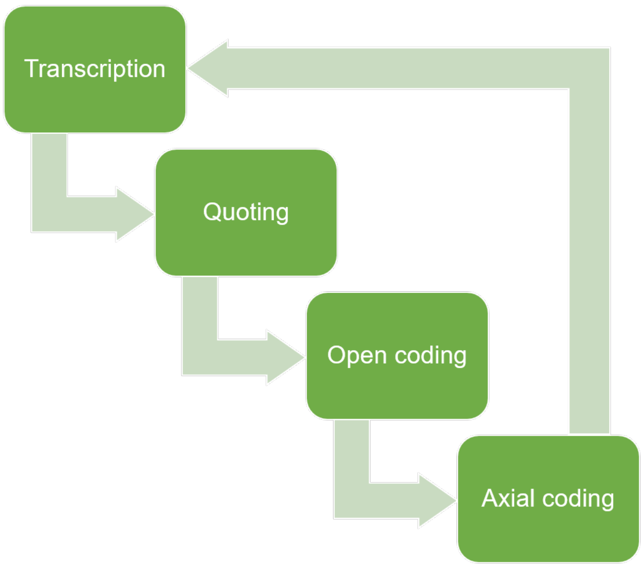


Figure 3.2: Data analysis process

First, interviews were transcribed using the audio recording. Interviews were transcribed semi-verbatim; since the aim of the data analysis is to extract factual information, albeit on the participant's opinion, full verbatim transcription including filler words was not necessary (Denscombe, 2010). Next, the transcripts were read and quotes and passages were highlighted that were related to one of the constructs introduced in section 3.1 "Conceptual model". This step precedes the actual coding, and is aimed at reducing the amount of text to be analyzed for coding and to smooth the process (Chi, 1997).

The most important aspect of the data analysis process was the coding of the transcripts. In qualitative data analysis, open coding is the process of initial classification and labeling of concepts (Babbie, 2010). This process is not an exact science, but based on the judgment of the author. Data has been read, reread, and the key concepts that describes it have been identified. These concepts were added to the list of codes, and linked to particular quotes. Quotes may contain multiple codes, and likewise codes may be found in multiple quotes. The final step in data analysis was axial coding. After fully coding an interview, the list of codes may be analyzed itself: by identifying overarching core concepts that emerge from the data, the existing list of codes can be grouped (Babbie, 2010). Coding is an iterative process, meaning the results of axial coding may influence the process of open coding, and vice versa. Furthermore, the entire data analysis is an iterative process: results from coding may be kept in mind during the interviews themselves and all following steps (Chi, 1997). For the coding process, *Atlas.ti* was used as a tool. *Atlas.ti* is designed for qualitative data analysis and is widely considered to be a suitable tool for academic purposes. It facilitates the open and axial coding process, and offers several features that enable in-depth analysis of the data.

After several iterations of the data analysis process, as described in Figure 3.2, a list of grouped codes remained as a final data set. This list of codes identifies several properties contributing to subjective complexity and served as an input for the model design.

3.2.5 Interview validity

The process of coding is often considered more of an art than an exact science, since the composition of a coding scheme is done by the researcher, based on his interpretation of the data. Therefore, the analysis process was validated based on suggestions made by Burnard (1991). Three of the participants interviewed were contacted and asked to produce a list of main themes in the transcripts of their own interviews. By comparing these to the coding scheme created by the researcher, it can be ensured that the interpretation of the participant's interview is correct. Overall, the coding scheme appeared correct, although some minor modifications were made based on this validation.

3.3 Model design

Designing a measurement model consists of two steps: conceptualization and operationalization (Babbie, 2010). Conceptualization is the process of defining relevant concepts and their relations, and results in a conceptual model. This conceptual model is a further specialization of the conceptual model introduced earlier in section 3.1 "Conceptual model", and identifies and models those concepts that influence complexity and a stakeholder's perception thereof. Subsequently, this conceptual model of subjective complexity has been operationalized by defining appropriate metrics that measure its concepts.

3.3.1 Conceptualization

Conceptual models consist of constructs and their relationships. A construct, also called a latent variable, is a concept, or a combined set of concepts, that is created to explain a certain phenomenon that is not directly observable (Bhattacherjee, 2012). By definition, constructs cannot be measured directly, and should therefore be measured using one or more metrics. Some constructs are easy to measure indirectly, such as a person's age. Others require an extensive set of metrics to describe them, such as a person's generosity, a society's norms and values, or an architecture's complexity.

The construct this research aims to measure is "complexity". To do so, all related constructs and their relations must first be specified. The conceptualization of complexity was based on the data acquired in chapter 2 "Literature Review" and chapter 4 "Interview Results". Construct identification through this data leads to a final set of constructs contributing to subjective architectural complexity. Using these constructs and the relations defined in the model in Figure 3.1, a conceptual model was designed, specifying all relevant constructs and their relations.

3.3.2 Operationalization

The next step in the design of a measurement model for architectural complexity is the operationalization of the conceptual model. Operationalization is the process of defining metrics that measure constructs (Bhattacherjee, 2012). A construct can be measured by defining one or more metrics that can, in contrast to the construct itself, be measured directly. Metrics were designed using the results of the literature review and interviews. Based on the metric template introduced in section 1.2.3 "Metrics", the following template was designed and used to describe metrics.

1. Name — Each metric will have a distinctive name
2. Construct — The underlying construct measured by this metric
3. Description — A short description of the metric
4. Type — Reflective or formative, further described below
5. Computation — Method of value calculation, further described below

Type

The relationship between a construct and a metric can be causal in either direction. Firstly, variation in a construct can lead to variation in its specified metric. In this case, the construct is reflected by its metric, and these are therefore termed reflective metrics (Edwards & Bagozzi, 2000). In contrast, variation in a construct can also be caused by variation in its metric. In this case, the construct is formed by its metric, and these are therefore termed formative metrics (Edwards & Bagozzi, 2000). Reflective and formative metrics are fundamentally different in their nature, and should be validated differently as well. The incorrect identification of causality between constructs and their metrics, called the misspecification of metrics, is a widespread problem in research, and can affect both current and future research (Freeze & Raschke, 2007). Therefore, when designing metrics measuring constructs, their causal direction has been carefully considered. The nature of reflective and formative metrics and their associated characteristics are summarized in Figure 3.3.

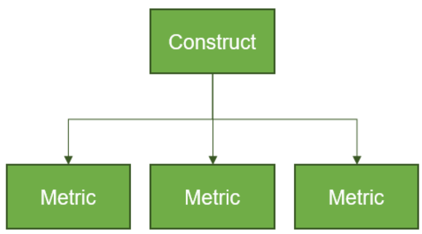
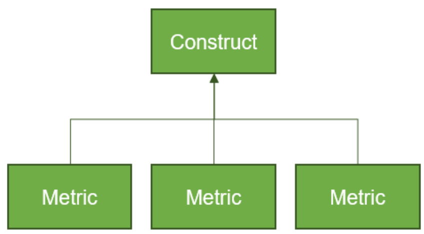
	Reflective	Formative
Causality direction	Construct to metric	Metric to construct
Metric exclusion	Does not alter the meaning of the construct	Fundamentally changes the construct
Correlation	High	Not expected
Interchangeability	High	Not expected
Visualization		

Figure 3.3: Reflective and formative metrics and their characteristics

Computation

One of the components making up a metric is its computation method, describing the way the value of the metric can be calculated. As is shown in the model of Figure 3.1, this is based on one or more measures. The difference between a metric and a measure is subtle: a metric is aimed at measuring a construct and can consider multiple variables, whereas a measure consists of a single variable. To illustrate the difference between a metric and a measure, consider one of the most widely-used metrics for software complexity, McCabe's metric of cyclomatic complexity. McCabe's metric is described by the following formula:

$$c = e - n + 2p$$

In this example, the calculated value of c (complexity) is a metric, whereas e (number of edges), n (number of nodes) and p (number of connected components) are all measures (Weyuker, 1988).

Mccabe & Butler (1989, p.1423) provide a list of properties that support the applicability of metrics when quantifying complexity. Based on this, the following list of properties has been defined and adopted in the metric design:

- "The metric intuitively correlates with the difficulty of comprehending a design." When using the metrics, a high level of complexity should also yield a high value of the metric. When quantifying a simple design, the metric value should be low.
- "The metric should be related to the effort to integrate the design." Since the integration of an architectural design is the most costly phase, the metrics should relate to this aspect. When measuring the complexity of an enterprise architecture, this means that the integration of the four architecture domains should be strongly considered when drafting metrics.
- "The metric should help generate an integration test plan early in the life cycle." Metrics should be applicable early in the life cycle of an architecture, so they can be used to positively influence its complexity in an early stage. In enterprise architecture, this means that metrics should be applicable on an architecture's design.

The first two stages of the research - the literature review and interviews - have both resulted in a list of (potential) metrics and measures. To operationalize the constructs defined in the conceptual model, a subset of these metrics have been selected and adapted if necessary, to be used in the measurement model. To select the appropriate and relevant metrics, the full list of metrics has been evaluated based on the selection criteria described in Table 3.2.

Selection criteria	**Description**
Usability	The metric must be usable in practice. The required data should be reasonably acquirable and the computation method should be easy.
Applicability	The metric must accurately describe the construct it is assigned to.
Prevalence	Metrics with a high prevalence among literature and interviews are likely more important. In other words, the metrics should be thoroughly grounded (Glaser & Strauss, 1967).
Overlap	If a metric has a high overlap with another metric, the most appropriate metric should be selected based on the selection criteria above.

Table 3.2: Metric selection criteria

After evaluating all metrics based on the selection criteria, a final selection of metrics describing all constructs has been made. For each construct, a final (set of) metric(s) has been specified, based on the metric template introduced earlier, resulting in the measurement model.

3.3.3 Reliability and validity

This research designs a model based on the definition of several constructs and their operationalization. These constructs are partly imaginary, and do not physically exist. Furthermore, the metrics are based on subjective research data. Therefore, when designing a measurement model, it is important to consider the psychometric properties of the metrics: reliability and validity (Bhattacherjee, 2012). These properties are visualized in Figure 3.4.

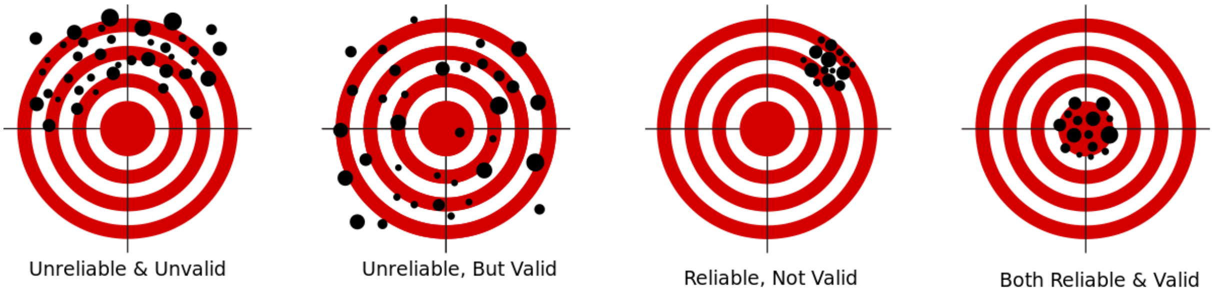


Figure 3.4: Comparison of reliability and validity[1]

3.3.3.1 Reliability

Reliability is the degree to which a metric is consistent and dependable (Bhattacherjee, 2012). If the same construct is measured multiple times using the same metrics, and the underlying phenomenon does not change, it should yield the same results. In the measurement model designed in this research, the main threat to reliability stems from its subjective nature. One of the main sources of unreliability in social science are the observer's subjectivity, asking ambiguous questions and asking irrelevant questions (Bhattacherjee, 2012). To remedy this threat to reliability, an extensive interview protocol was used during the interviews to standardize the process of acquiring data. Furthermore, data analysis was validated by the participants themselves, as described in section 3.2.5 "Interview validity". There are several ways to assess the reliability of metrics. However, all these require a minimum number of measurements to be performed. Due to time constraints, this research cannot perform enough empirical measurements and has therefore not assessed the reliability of its metrics.

3.3.3.2 Construct validity

Construct validity refers to "the extent to which a [metric] adequately represents the underlying construct that it is supposed to measure" (Bhattacherjee, 2012, p.59). Construct validity can be assessed both theoretical and empirical. Theoretical construct validity, also called **translational validity**, assesses how well a construct is captured by its metrics. Within translational validity, **face validity** looks at a metric "at face value", and evaluates whether it seems like a reasonable metric of the underlying construct. In other words, it assesses whether the metric logically seems like an appropriate measurement instrument. **Content validity** amplifies this view, and checks whether each measure within the metric is relevant for the construct being measured, and if the content of the measures covers the entire construct. For example, a metric "software quality" should contain measures that jointly cover all aspects of software quality that can be found in literature. Measuring face validity and content validity is a subjective process that is done using a panel of experts. For this research, the translational validation of metrics is performed using a panel of 3 experts, coming from the field of enterprise architecture. More details are given in section 3.4.1 "Expert validation".

Criterion validity is an empirical method of measuring construct validity, and employs statistical analysis to measure validity. As with reliability, the evaluation of criterion validity requires a minimum sample size of measurements to be performed. Due to time constraints, this research has not assessed the criterion validity of its metrics.

3.4 Validation

Validation of the designed measurement model is an important final step in the process of its design. In design science, an artifact can be validated by showing it satisfies the requirements that were set, and that it contributes to stakeholder goals (Wieringa, 2014). To validate the resulting measurement model, it is necessary to first validate the underlying conceptual model (Sargent, 2013). There is no formal procedure to validate a conceptual model before it is operationalized (Liu, Yu, Zhang & Nie, 2011). Therefore, it is important to ensure that the designed model has face validity. This is done by reviewing the model with experts in the enterprise architecture field. The experts can judge whether the model is correct and applicable in its intended context. Next, the design of the metrics and the resulting measurement model should be validated. The validation of a measurement model should be empirical, and requires a large data set (n>100 (Collins, Ogundimu & Altman, 2016)). Due to time constraints, this research has not performed an empirical validation of the measurement model. Instead, a case study was conducted in which a part of the measurement model was be applied in a practical context.

3.4.1 Expert validation

Expert validation is adopted as a first step of validation. During expert validation, one or more experts are presented with the artifact, its intended context, and its goals. Subsequently, they are asked to imagine the interaction between the artifact and its context, and predict how the artifact will behave and whether this will satisfy its goals. Based on recommendations by the experts, the artifact may be redesigned or optimized (Wieringa, 2014).

The designed measurement model has been validated on several aspects using the opinion of three experts. The following aspects were discussed during the expert interviews:

- **Constructs**: validate whether the proposed construct actually contribute to subjective complexity
- **Construct relations**: validate the causal relations between the constructs
- **Conceptual model**: validate the conceptual model as a whole
- **Translational validity:** validate whether each metric appropriately represents the underlying construct
- **Measurement model**: validate the measurement model as a whole

For the expert validation, three senior employees of the Enterprise Architecture department of Deloitte Consulting were asked to provide their knowledge and expertise. Due to the nature of their work they have considerable experience with enterprise architectures over different organizations and industries, as well as with the identified stakeholder roles. This makes them suitable experts for this step of validation. Table 3.3 provides an overview of the experts and their experience with enterprise architecture. Each expert was interviewed in a one-on-one interview that took approximately one hour. The measurement model has been adapted and optimized based on the feedback of the experts.

#	Role	Experience with EA
1	Director	25 years
2	Manager	6 year
3	Partner	20 years

Table 3.3: Overview of validation experts

3.4.2 Case study

In design science, artifacts can be validated by applying them to their problem context, and verifying whether they contribute to the solution of this problem. Therefore, the designed measurement model has been validated through its application in its intended context. Several research approaches are aimed at studying the interaction between an artifact and its environment, most prominently experiments, case studies and action research (Wieringa, 2014; Yin, 2009). Both Yin (2009) and Dresch, Lacerda & Miguel (2015) provide an overview of different characteristics that are found in these research approaches. Applying these characteristics to this research shows that the case study is the most appropriate approach to validate the designed measurement model. Case studies focus on phenomena in a real-life context in which the researcher has little or no possibility to control the situation. Case studies are especially suitable for studying complex situations with many variables of interest, making it a fitting research approach for this validation (Yin, 2009).

3.4.2.1 Research design

The goal of this case study was to apply the designed model in practice, observe its interaction with the intended context and verify that it contributes to its described goals. To this end, an explanatory case study was performed, aimed at studying the designed model in its context.

Due to the size of the designed measurement model, the scope of this case study was limited. Since the major contribution of this research lies in the addition of subjective constructs into complexity measurement, the case study focused on the constructs *stakeholder qualities, vision* and *understandability*. Of course, subjective complexity is also influenced by objective complexity. To eliminate objective complexity as a contributing factor, the case study only studied participants working at the same department of the same organization. Consequently, the objective complexity was equal for all participants, and can therefore be eliminated as a contributing factor. This lead to the following research question: “How do the constructs *stakeholder qualities, vision* and *understandability* influence subjective complexity?”.

To answer this research question, data on these constructs was acquired through a survey (more details on data collection can be found in section 3.4.2.3 “Data collection”). Using the data from the survey, the metrics associated with these constructs have been calculated. Finally, the participants were asked to judge the complexity of the architecture on a scale of 1-10. This was their subjective complexity. The analysis of this data, further described in section 3.4.2.4 “Data analysis”, was used as input for the one-on-one interviews with each participant. During this interview, the resulting scores served as a starting point to discuss the contribution of each construct to subjective complexity. The insights gained from the interviews have been used as input for the further optimization of the measurement model.

3.4.2.2 Case selection

The selection of a suitable case was done based on both the availability of contacts with organizations and several key characteristics. Based on possible selection criteria listed by Benbasat, Goldstein & Mead (1987), the following requirements were considered when selecting a case:

- To confirm cross-industry applicability, the organization should operate in a different industry than the interview organizations
- It should be a large company with a complex architecture
- At least three participants from the same department should be available for interviews

Based on these requirements, organization E was found willing to participate in the case. Organization E is a large company active in the logistics-sector, operating in the Netherlands, serving several million customers each year. Having a large infrastructure, they have an extensive enterprise architecture supporting their processes.

3.4.2.3 Data collection

Data was collected in two ways. First, a survey was sent to all participants to acquire data about the subjective constructs. The survey was based on the metrics designed for the constructs *stakeholder qualities, vision* and *understandability*, as defined in section 5.4 “Operationalization”, and can be found in Appendix E “Case Study Survey”.

Next, interviews were conducted with all participants. During these interviews, the results of the survey were discussed. The goal of the interviews was to gain a better understanding of the motivation of the answers given by each participant. These insights were used in the optimization of the conceptual model and the defined metrics.

3.4.2.4 Data analysis

Using the conceptual model from Figure 5.9, it can be inferred that subjective complexity is dependent on objective complexity, stakeholder qualities, vision and understandability. Given that objective complexity is equal for all participants in this case and that all constructs have a different relative weights, subjective complexity can be calculated as follows:

$$(x{*}\textit{stakeholder qualities}) + (y{*}\textit{vision}) + (z{*}\textit{understandability}) = \textit{subjective complexity}$$

For all of these constructs, questions were included in the questionnaire based on the metrics designed. From the survey results, values have been calculated for stakeholder qualities, vision and understandability. Subjective complexity itself was queried in the survey as well. Therefore, all variables from both sides of the equation are known through the survey data. Next, values for x, y and z were calculated so that the equation can be solved for all participants. These values represent the relative weights of the different constructs towards subjective complexity. The result is a weighed formula calculating subjective complexity.

This data was used as input for interviews with the participants. During the interviews, both the relations between the constructs (i.e. the conceptual model) and their relative importance (i.e. the construct weights) were discussed. Participants were asked to provide feedback on the construct relations and weights found during the research. Using both the data acquired from the survey as well as the participant feedback, the conceptual model and construct weights have been optimized.

3.4.2.5 Case validity

Case studies typically have a strong internal validity, but lack external validity (Yin, 2009). The internal validity of the case study was ensured by using multiple sources of evidence (surveys and interviews) and establishing a chain of evidence, as is suggested by Yin (2009).

The results of this case study are based on a single case with four participants. Therefore, the results are neither statistically significant nor a definitive validation of the measurement model. However, in-depth analysis of a single case does show the applicability of the model, and can serve as a starting point for further research. Suggestions for further validation of the measurement model can be found in section 7.7 “Future research”.

4. Interview Results

4.1 Analysis

4.1.1 Transcription & quoting

The first step in the analysis of the interviews was the semi-verbatim transcription of the audio recordings. This resulted in a set of twelve transcripts with a total of 92 pages, meaning an average of almost eight pages per transcript. The anonymized transcripts are not attached with this report, but are available upon request. All transcripts were loaded in the data analysis tool Atlas.ti for further analysis through open and axial coding.

After loading the transcript in Atlas.ti, relevant passages were marked as quotation. Here, "relevant" was defined as all quotes that were somehow related to one of the constructs introduced in section 3.1 "Conceptual model". Since these constructs are linked to the complexity of enterprise architecture as well as a stakeholder's perception of this, these quotes could potentially hold information on the attributes leading to this complexity or the perception thereof. The process of quoting aimed to reduce the amount of data that needed careful analysis for coding, in order to increase the quality of coding and smooth the process.

4.1.2 Open coding

The core of the data analysis is the process of open coding, which was done inductive and iterative. No predetermined coding scheme was used; rather, the analysis of the first interview led to an initial coding scheme which was evaluated, revised and extended with each subsequent interview. After twelve interviews, an additional final iteration of evaluating the coding scheme lead to a resulting set of 48 codes, listed in Table 4.1. Combined, these 48 codes were mentioned 165 times, meaning an average of almost fourteen (unique) codes per interviews.

4.1.3 Axial coding

Based on the list of codes, an overarching set of core concepts was identified, grouping the 48 codes in smaller sets. The core concepts identified during this process overlap with several of the concepts established in the conceptual model of Figure 3.1. This can be explained by the fact that the aim of the model is to identify all aspects relating to the enterprise architecture and its complexity, and is therefore inherently an appropriate categorization of all attributes relating to enterprise architecture complexity.

4.2 Results

4.2.1 Code list

The final list of codes resulting from the open and axial coding process can be found in Table 4.1. Note that the prevalence of a code was calculated based on its unique prevalence in each interview. The amount of times a certain code was mentioned during an interview did not influence the results. In Appendix D "Detailed Interview Results", a detailed analysis of the prevalence of codes for each role is given. Although some differences in prevalence among roles can be found this is relatively limited, meaning the different roles generally agree on the aspects influencing complexity.

Code group	Code	Count	Prevalence
Architecture	Application complexity	4	33%
	Business complexity	4	33%
	Conformity	2	17%
	Coupling	5	42%
	Data complexity	1	8%
	Dependency	6	50%
	Functional redundancy	2	17%
	Interoperability	1	8%
	Modularity	8	67%
	New, unknown technologies	1	8%
	Number of elements and relations	4	33%
	Presence of structure	1	8%
	Security	1	8%
	Standardization of landscape	6	50%
	Technical debt	1	8%
	Variation in technology	4	33%
Enterprise	Available resources	2	17%
	Business culture	2	17%
	Change effort	2	17%
	Control	2	17%
	Governance	3	25%
	History of architecture	6	50%

Code group	Code	Count	Prevalence
Enterprise	Internal politics	3	25%
	Legacy	5	42%
	Number of stakeholders involved	7	58%
	Priorities	1	8%
	Quality of the architect	2	17%
	Transparency in decision making	1	8%
Environment	Complexity of the environment	6	50%
	Complexity of the problem	4	33%
	Industry	5	42%
	Market developments	2	17%
Mission	Architecture performance	1	8%
	Goal of the architecture	1	8%
	Presence of vision or strategy	5	42%
Model	Abstraction level of documentation	5	42%
	Communication to stakeholders	3	25%
	Insight in the architecture structure	4	33%
	Maintainability of documentation	1	8%
	Notation of documentation	4	33%
	Presence of documentation	6	50%
	Terminologies and language	1	8%
Stakeholders	Biological: left- or right-brained	1	8%
	Stakeholder education and background	3	25%
	Stakeholder expectations	1	8%
	Stakeholder interest and affinity	6	50%
	Stakeholder knowledge and experience	12	100%
	Stakeholder role	7	58%

Table 4.1: Identified codes and their prevalence

4.2.2 Architecture

Most codes were classified in the “architecture”-group. This group consists of those attributes directly relating to the elements and relations in an enterprise architecture. As was defined in the conceptual model of Figure 3.1, an enterprise architecture consists of elements and their relations, categorized in different architectural domains (business, data, application and technology). Each architectural domain consists of elements and relations, although these elements differ for each domain. The codes in this group directly relate to characteristics of these elements and relations.

An often mentioned metaphor, especially by the producer stakeholders, was the “spaghetti versus lasagna” comparison. An important goal of enterprise architecture is to take the spaghetti that is the IT landscape - a complex and chaotic whole of intertwined elements - and create a lasagna: structured and organized in different layers.

4.2.3 Enterprise

The second-largest group of codes that was found in the interviews are enterprise-related elements. This group considers the immediate environment of the enterprise architecture, which is the enterprise itself. Within the enterprise, participants identified several aspects that can have an important influence on complexity or the perception thereof. Important factors here were the number of stakeholders involved - more people means more opinions and more requirements - and the history of the architecture. Ineffective or inefficient decisions taken in the past, often taken because of time and/or money constraints, tend to accumulate and be ignored until they cause a problem in the present.

4.2.4 Environment

The theory of requisite complexity, as introduced in section 2.6 “Requisite complexity”, could also be found in the interviews. Many stakeholders noted that the enterprise architecture was complex because its environment is complex. The ineffective or inefficient decisions mentioned earlier are often caused by a turbulent environment posing pressure on the enterprise. In many cases, the environment in which the enterprise operates is simply so complex, that the enterprise architecture needs to be complex to handle this. This was especially true in organization C, which, as a large governmental agency, has to handle a lot of complex legislation.

4.2.5 Mission

Several participants noted that when considering complexity, it is important to be aware of the vision of the architecture. They noted that if you do not know where you are going, things seem more complex than when there is a clear vision or strategy. An often quoted metaphor was the “dot on the horizon” that has to be set for and by the architecture.

4.2.6 Model

During the interviews, documentation arose as an important factor when considering subjective complexity. Most interview participant considered the documentation in their organization to be lacking in some way, and saw this as one of the contributors of complexity. Mentioned most was the presence of documentation - in some cases, there was no documentation at all. The resulting lack of insight in the architecture’s structure is seen as an important cause of perceived complexity: if you don’t know how an architecture is built, it quickly seems very complex. Several characteristics of documentation were mentioned as well, such as its abstraction level (i.e. level of detail) and its notation. But also the communication of this documentation towards other stakeholders is considered to be an important aspect. Most mentioned here was the match between documentation details and its audience. Participants emphasized that the level of detail in documentation should be dependent on the audience viewing it. Having an enormously detailed documentation is not necessarily a bad thing, as long as you show it to your colleague architects, and not the board of directors. Therefore, understandability follows from the correct matching of documentation detail and notation and the audience viewing it.

4.2.7 Stakeholders

Finally, the characteristics of different stakeholders appear to have an important influence on subjective complexity as well. The knowledge and experience of a stakeholder was the only code to be mentioned in all twelve interviews. But also role and technological affinity of a stakeholder are mentioned to be of influence for perceived complexity.

5. Model Design

5.1 Construct identification

The first step in the design of the measurement model is the conceptualization of subjective complexity. All constructs relating to complexity have been identified based on the literature review and interview results. Combining these resulted in a final list of constructs that is supported by literature and empirical data. However, including all results from both literature and interviews would result in a list of constructs that is simply too large to process into a model. Therefore, only constructs with a prevalence of at least 25% were included.

5.1.1 Literature

The structured literature review into existing complexity metrics resulted in the identification of several metrics. These were then categorized into concepts, which are listed in Table 2.3 and can be mapped to constructs. In Appendix B.5 “Concept matrix”, the prevalence of each concept was calculated. By eliminating the concepts with a prevalence lower than 25%, a set of four concepts remained:

- Elements & relations
- Functions
- Application data
- Conformity

These concepts have either been used directly as a construct, or been mapped on other constructs identified in the interviews.

5.1.2 Interviews

The coding of twelve interviews, held among four companies, yielded a total of 48 unique codes. However, many of these were only mentioned once or twice, and do therefore not meet the requirement of a prevalence of 25% or higher. After elimination of these codes, 26 remained. Each of these 26 remaining codes was assigned a construct. This process is similar to that used in the analysis of the structured literature review, and done based on the judgment of the author. One or more codes were assigned to these constructs, resulting in a list of 21 constructs. Table 5.1 shows a list of the codes and their assigned constructs. Note that some of the codes mentioned during the interviews were directly usable as constructs, whereas others had to be interpreted and converted into a construct that best fits it.

Code group	Code	Construct
Architecture	Application complexity	Application complexity
	Business complexity	Business complexity
	Coupling	Coupling
	Dependency	Coupling
	Modularity	Modularity
	Number of elements and relations	Size
	Standardization of landscape	Standardization
	Variation in technology	Heterogeneity
Enterprise	Governance	Governance
	History of architecture	Technical debt
	Internal politics	Politics
	Legacy	Legacy
	Number of stakeholders involved	Business complexity
Environment	Complexity of the environment	Environmental complexity
	Complexity of the problem	Environmental complexity
	Industry	Industry
Mission	Presence of vision or strategy	Vision
Model	Abstraction level of documentation	Documentation quality
	Communication to stakeholders	Communication
	Insight in the architecture structure	Understandability
	Notation of documentation	Documentation quality
	Presence of documentation	Documentation quality
Stakeholders	Stakeholder education and background	Education
	Stakeholder interest and affinity	Affinity
	Stakeholder knowledge and experience	Experience
	Stakeholder role	Role

Table 5.1: Identified codes and their assigned constructs

5.1.3 List of constructs

Combining the input from both literature and the interviews resulted in a final set of constructs that influence subjective complexity. Some of the constructs identified through literature were mapped onto the constructs identified in the interviews:

- *Elements & relations* was mapped onto the constructs *size* and *heterogeneity*. This was done because almost all of the prevalence of this concept stems from the use of size and/or heterogeneity metrics.
- *Functions* was mapped onto *business complexity*. This was done because most of the metrics underlying this concept related to functionality or processes, which are the elements of the business domain and therefore part of business complexity.
- *Application data* was mapped onto *application complexity*.

An overview of the resulting constructs and their grouping can be found in Figure 5.1.

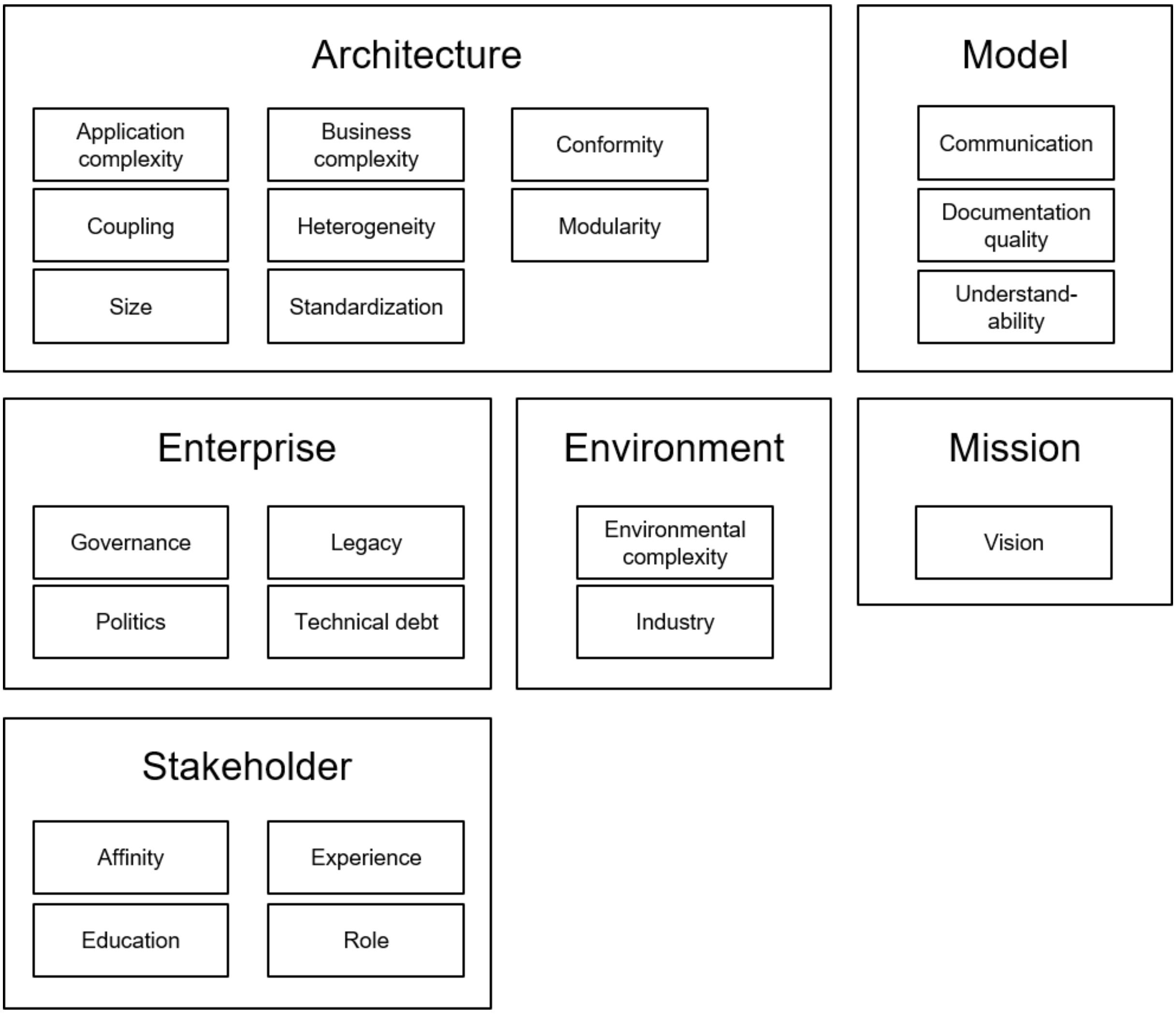


Figure 5.1: Overview of constructs and their grouping

5.2 Construct relations

For each group, relations between the constructs were identified and visualized. These relations were based on a combination of literature and empirical evidence from the literature review and interviews. The identification of relations within these groups is the first step towards creating a full conceptual model. Below, the relations between constructs are visualized for each identified group, including an explanation and substantiation of each relation.

5.2.1 Architecture

The first group is related on the architectural elements of the enterprise architecture and therefore called the "architecture" group. As was introduced earlier in this research, an enterprise architecture consists of four domains: business, data, application and technology. According to the model introduced in Figure 3.1, each domain consists of a series of entities (elements) and relations between them. What these elements are, depends on the domain of focus: in the business domain, an element may be a function or business process, whereas in the application domain elements are applications, and the relations their interfaces. The technology domain often consists of a wide array of different entities, such as servers (physical and virtual), Database Management Systems or Operation Systems. In current literature, the combined complexity of these four domains are often considered to be the objective complexity. However, the inter-domain complexity also plays a role in this (Schütz et al., 2013). Inter-domain complexity is dependent on the same aspects as intra-domain complexity. Together, the complexity of the different domains, combined with the inter-domain complexity, contributes to the objective complexity of an architecture (as will be shown in the further design of the conceptual model).

There are several properties of the elements and relations within each domain and between the domains that cause complexity. The most evident of these are size and heterogeneity. Both of these aspects are frequently mentioned in literature are the prime source of complexity within enterprise architecture, as was shown in the structured literature review. Therefore, both of these are modelled as causes for complexity. Conformity also emerged as a construct during the literature review as an important influence on complexity. Conformity can also be seen as a cause for heterogeneity: an architecture that is highly conform its standards will have a less heterogeneous landscape than one that has a low conformity. This is of course also dependent on how clear and heterogeneous the set standards are. Therefore, both standardization, which was introduced as a construct through the interviews, and conformity lead to heterogeneity in the model.

Finally, participants of the interviews often contributed complexity to the ability to the separation of concerns of different architectural components. One participants quoted: "For me, [complexity] is how modular it is built. So how clean is the architecture, how well are functionalities separated. If this is badly done, the landscape is very complex". This is an indication that the level of separation between different functionalities within domains, in other words, modularity, is an important cause for complexity. Related to this is coupling, which was also indicated as a cause for complexity and influences the interdependence between different modules. One participant noted that: "In my daily work, I could point complexity out to coupling. So if I have a tight coupling between systems, that's typically an issue for complexity". Whether an architecture is tightly coupled is therefore also a clear cause for its modularity. Figure 5.2 visualizes all of the identified relations between these constructs.

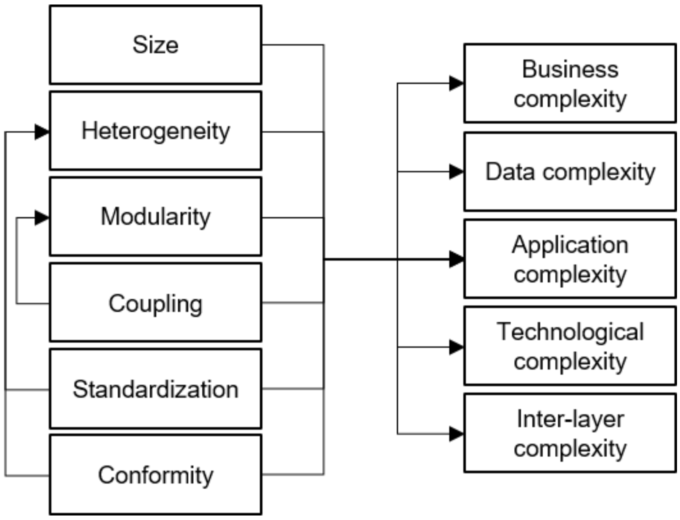


Figure 5.2: Architecture-related constructs and their relations

5.2.2 Enterprise

The "enterprise" group contains constructs that relate to the immediate environment of the architecture, which is the enterprise itself. When discussing the influence of the enterprise on the complexity of the architecture, an aspect that was mentioned frequently was that it is important to consider the historical decisions made regarding the architecture. Imperfect design decisions made in the past irrevocably have an influence on the current architecture. Although the structural/dynamic dimension of complexity, which considers complexity as a function over time, was deliberately not considered in this research, this does have a structural manifestation in the architecture: technical debt. Technical debt is a concept that reflects the additional cost caused by imperfect design or development decisions made in the past.

Another construct that the participants considered an important cause of complexity are legacy systems. Proper life cycle management of systems was often considered to be a prerequisite for limited legacy-induced complexity. Although legacy is not necessarily caused by imperfect design decision and can simply be a product of time, it can be seen as a cause for technical debt in the broad sense. Governance was also mentioned as a cause for technical debt. When asked about complexity, one of the participants mentioned that an important aspect was "whether we actually have a vision on [product], or are a department that just delivers components for other domains, makes a lot of difference. That whole governance-aspect: who's responsible for what". Additionally, he mentioned that lacking governance lead to several departments, not wanting to wait on a large redesign of the enterprise architecture, creating parallel sub-architectures. This inherently lead to a lot of additional technical debt and complexity.

Finally, many of these factors can be influenced by the internal politics of the company. One example was mentioned of the IT leadership that did not challenge the business on the feasibility of their requests. Caused by pressure from the business, several components had suffered change on change for years, ignoring restructuring or maintenance. Eventually, this lead to an architectural deadlock. According to the participant, this could have been prevented by a stronger IT leadership, convincing the leadership of the organization to invest in maintenance over new functionality. Therefore, politics can be seen as an important factor in the enterprise group, directly influencing governance and therefore indirectly influencing technical debt as well.

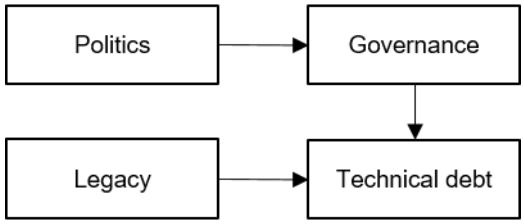


Figure 5.3: Enterprise-related constructs and their relations

5.2.3 Environment

The law of requisite complexity, introduced in section 2.6 “Requisite complexity”, emphasizes the importance of the environment an enterprise and its architecture operates in. This was confirmed during the interviews, when many participants acknowledge environmental factors to be a contribution to the complexity of the architecture. This especially came forward at organizations having an inherently complex environment with a lot of regulations and legislation. This is of course very dependent on the industry the enterprise operates in; an organization in the public or financial sector will usually have to handle more regulations than an organization in retail. One of the participants noted that ”complexity arises due to the fact that, since it’s a new organization operating in a very dynamic market, growth requires speed”, confirming that the market in which an organization operates is an important aspect influencing environmental complexity.

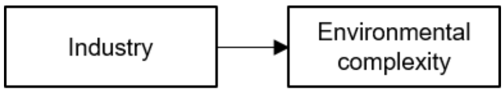


Figure 5.4: Environment-related constructs and their relations

5.2.4 Mission

The presence of a vision can be of great importance when considering complexity. One of the participants explained: “The goal of your organization influences how people look at the complexity of your architecture. Subjectively, if a large change is taking place and the architecture is impeding this, people consider it to be very complex. Whereas if there is no large change happening and the organization is more stable, even though the architecture might be in worse shape, subjectively people will find it to be less complex. It influences how people look at the architecture”.

Figure 5.5: Mission-related constructs and their relations

5.2.5 Model

Subjective complexity of an enterprise architecture is largely influence by the understandability of said architecture. This is established by literature, and understandability was therefore included in the key questions of the interviews, as can be found in Appendix C “Interview Guide”. In the context of understandability of the architecture, most interview participants mentioned documentation. Several aspects of documentation were mentioned to influence understandability, mostly focusing on its level of detail and notation. Another aspect that was considered to influence understandability is the communication towards stakeholders. When talking about understandability and the perception of complexity, one participant noted: “If you manage to explain it clearly, it will be considered less complex than when you’re stammering an incoherent story that nobody understands”.

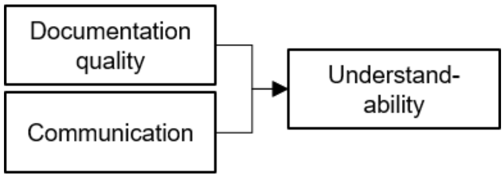


Figure 5.6: Model-related constructs and their relations

5.2.6 Stakeholder

As was already established in the introduction of this research, subjectivity is dependent on the object of study, but also of the stakeholder’s perception of this. Several stakeholder qualities can influence their perception of complexity. Having a prevalence of 100%, all participants of the interviews agreed that the experience of a stakeholder, both in the organization itself as well as in the field of enterprise architecture, influences their perception of complexity. But also the role of a stakeholder, their general affinity with architecture and technology, and their education can play a role in their perception of complexity, according to the interview results. These constructs have been combined in the general construct of “stakeholder qualities”, influencing subjective complexity.

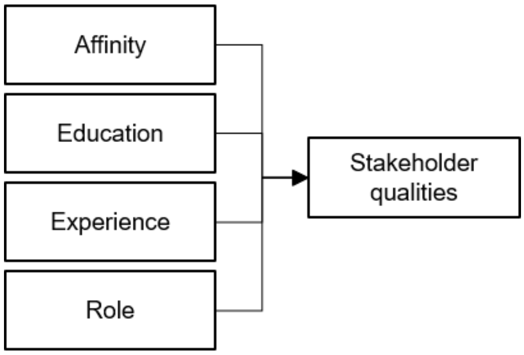


Figure 5.7: Stakeholder-related constructs and their relations

5.3 Conceptual model

After establishing the relations within each construct group, these groups should be unified to form the final conceptual model of complexity. This requires the formulation of two additional constructs representing complexity itself. In the introduction of this research, it was argued that subjective complexity is influenced by stakeholder perception, but also by the objective complexity of the architecture. Therefore, both “objective complexity” and “subjective complexity” are introduced as constructs.

Objective complexity is reflected by the complexity in the individual domains and the inter-domain complexity of the architecture. This was shown to be influenced by several aspects shown in Figure 5.2. Technical debt, caused by several other constructs, was found in the interviews to influence objective complexity as well. Finally, environmental complexity is also reflected in the objective complexity of an architecture. The law of requisite complexity states that “the internal complexity of a system must match the external complexity it confronts” (Boisot & Mckelvey, 2011, p.279). This means that a higher environmental complexity will inherently lead to a higher internal, i.e. objective, complexity.

Combining these constructs leads to the formulation of a conceptual model for objective complexity, as is visualized in Figure 5.8.

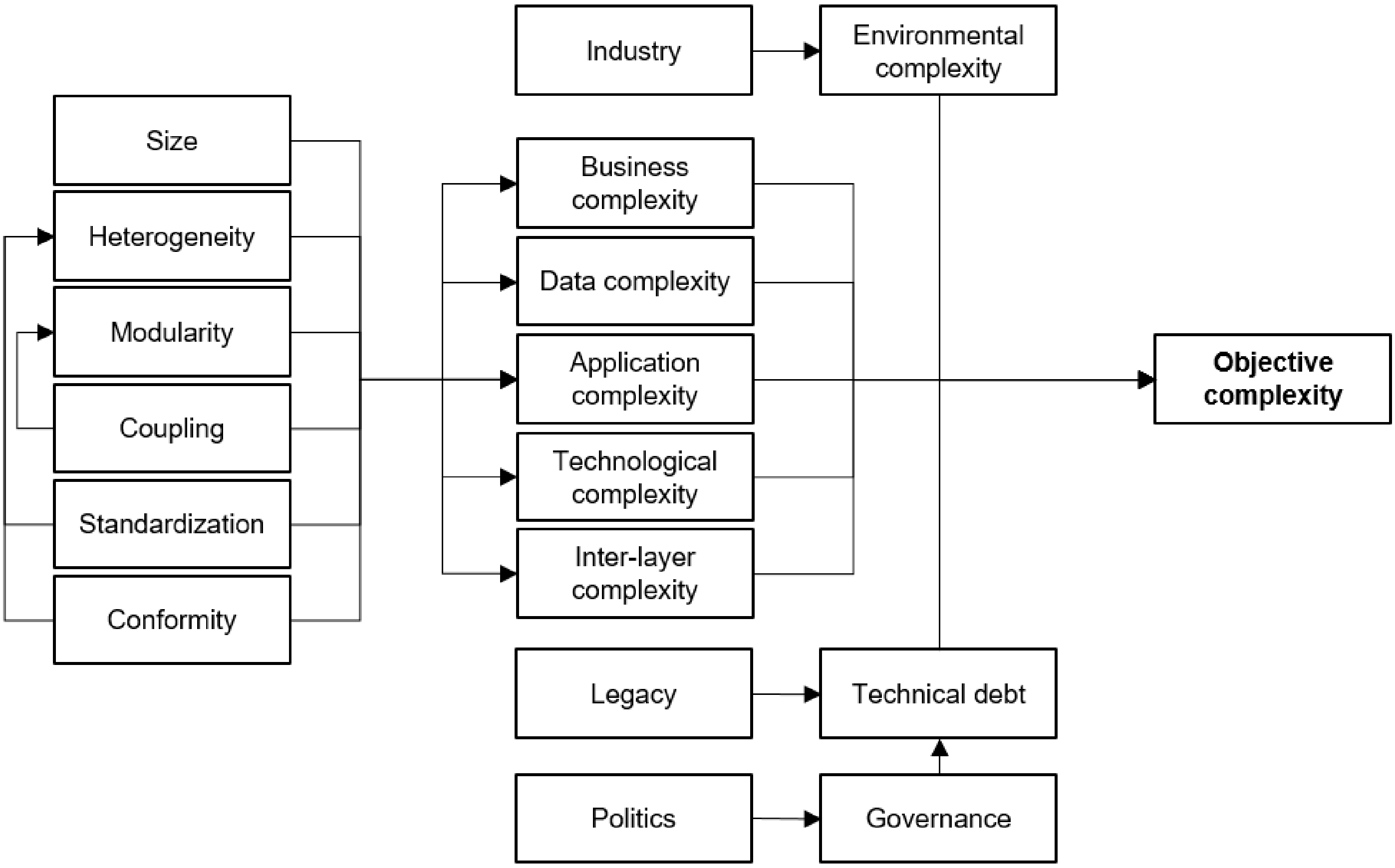


Figure 5.8: Conceptualization of objective complexity

The final construct to be included in subjective complexity. This was established to be influenced by both the objective complexity, as well as stakeholder perception. A stakeholder's perception is first of all influenced by the qualities of the stakeholder itself. The constructs defined in Figure 5.7 show which characteristics were found to influence their perception of complexity. In literature, it has also been established that the understandability of the architecture influences its subjective complexity, as was explained in section 5.2.5 "Model". Finally, interviews show that the presence or lack of a vision or strategy also influences the perception of stakeholders, as was illustrated in section 5.2.4 "Mission". Combining these relations with the conceptualization of objective complexity made above, leads to the formulation of a conceptual model. This model is visualized in Figure 5.9.

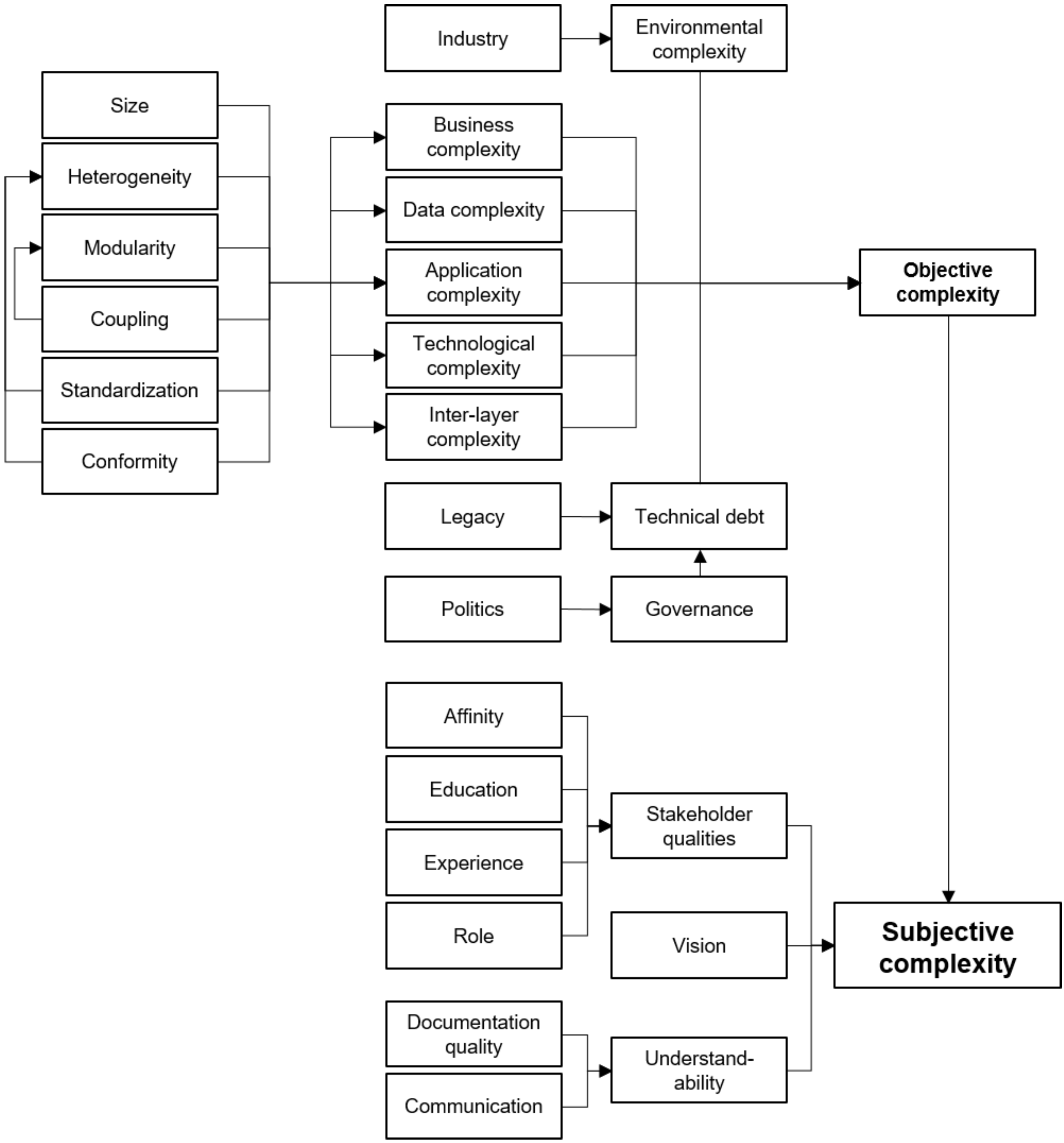


Figure 5.9: Conceptual model of architectural complexity

5.4 Operationalization

To create a measurement model, the identified constructs have been operationalized by defining one or more metrics. These metrics can be used to directly measure the constructs. Each metric was designed based on the metric template introduced in section 3.3.2 “Operationalization”, looking as follows:

1. Name
2. Construct
3. Description
4. Type
5. Computation

Due to time constraints, not all constructs have been operationalized. To still be able to fully measure complexity, those construct at the end of the causal chain have been operationalized, given that they can be measured without further specification. For example, for the enterprise group, only the construct “technical debt” has been operationalized, since all other constructs lead to technical debt. It is therefore the furthest construct along the causal chain that can be operationalized without further specialization. Below, the metrics required to measure complexity are introduced per construct group.

5.4.1 Architecture

An architecture is a system that can be mathematically described as follows:

$$S = (T, R)$$

Here, S is the system, i.e. the entire architecture or one of its domains, T is the set of components in that system, i.e. its elements, and R are the relations between these components (Schütz et al., 2013). Several of the metrics introduced below depend on this system notation of an enterprise architecture.

Please note that “conformity”, “coupling” and “standardization” have not been operationalized. This is due to the fact that these constructs were moved up the causal chain in a later iteration of the conceptual model, as is described in section 6.1 “Expert validation”. Since only those constructs at the end of the causal chain were operationalized, metrics for these constructs were excluded.

5.4.1.1 Size

The size of an architecture has been operationalized in many of the papers reviewed in the structured literature review, and has been an accepted construct contributing to complexity. Given the set notation of an architecture (or one of its domains) stated above, the size of an architecture can be measured using the metrics below.

Metric 1

1. Number of elements
2. Size
3. Measures the size of the set of elements in the respective architectural domain
4. Formative
5. $|T|$: count the total number of elements in the domain

Metric 2

1. Number of relations
2. Size
3. Measures the size of the set of relations in the respective architectural domain
4. Formative
5. $|R|$: count the total number of relations in the domain

5.4.1.2 Heterogeneity

As with size, heterogeneity has been operationalized in several papers. This research adopts the metrics introduced by Schütz et al. (2013). Given the set notation of an architecture (or one of its domains) introduced before, the heterogeneity of an architecture can be measured using the metrics below.

Metric 1

1. Element entropy
2. Heterogeneity
3. Measures the heterogeneity of the set of elements in the respective architectural domain
4. Reflective
5. $-\sum_{i=1}^{n} p_i ln(p_i)$ where

   $i \in T$

   $p_i =$ relative frequency of element i

Metric 2

1. Relation entropy
2. Heterogeneity
3. Measures the heterogeneity of the set of relations in the respective architectural domain
4. Reflective
5. $-\sum_{i=1}^{n} p_i ln(p_i)$ where

   $i \in R$

   $p_i =$ relative frequency of relation i

5.4.1.3 Modularity

Modularity refers to the degree to which a system can be divided in separate components. The measurement of modularity in architecture has not yet been specifically studied. However, Newman (2006) defined a metric for modularity in networks that has been generally accepted. This metric focuses on networks consisting of nodes and edges - which is exactly how an architecture was defined earlier. The metric is based on statistics: modularity exists when the number of edges between groups is significantly less than can be expected by chance.

Metric 1

1. Element modularity
2. Modularity
3. Measures the modularity of a network of elements
4. Reflective
5. $\frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})s_i s_j$ where

$$m = |R|$$
$$A_{ij} = \text{number of edges between element } i \text{ and } j$$
$$k_i = \text{degree of element } i$$
$$s_i s_j = 1 \text{ if } i \text{ and } j \text{ are in the same group and } 0 \text{ otherwise}$$

5.4.2 Enterprise

All enterprise-related constructs are theorized to lead to technical debt. Although some other minor influences may exist, the accumulation of technical debt through legacy, politics and the governance of information systems is the main reason for adding complexity, according to the interviews. Since most of the constructs are hard to measure directly, especially politics and governance, these will be measured through technical debt.

There are currently no metrics available that focus on technical debt on an architectural level. However, the metrics introduced by Nord, Ozkaya, Kruchten & Gonzalez-rojas (2012), initially aimed at software architecture, can also be translated to enterprise-level. They define the cost of an architecture to be the sum of the cost of implementation and the cost of rework. Here, the cost of rework represents technical debt: it quantifies the future work that needs to be done on the architectural elements.

Metric 1

1. Cost of rework
2. Technical debt
3. Measures the cost of rework (Cr) that needs to be done on elements in the architecture
4. Reflective
5. $\sum_k Cr(E_k)$ for all new elements E_k where

   $Cr(E_k) = \sum_j Cr(E_j)$ for all pre-existing elements E_j where

   $Cr(E_j) = D(E_j, E_k) * Ci(E_j) * Pc(n-1)$ where

   $D(a,b)$ is the number of dependencies between a and b, Ci is the implementation cost and $Pc(n-1)$ is the propagation cost of release $n-1$. The calculation of propagation cost is described by Baldwin, Maccormack & Rusnak (2014)

5.4.3 Environment

Environmental complexity is a construct that is affected by many factors and is therefore hard to measure. Cannon & John (2007) summarize and consolidate several earlier studies on environmental complexity, and propose a model for its measurement. They tested these metrics on an existing set of data, and suggest several correlations. This research adopts the use of their model for measuring environmental complexity, but emphasizes that their correlations and relative contribution to environmental complexity should be further investigated.

Metric 1

1. Herfindahl-Hirschman index
2. Environmental complexity
3. Measures the market concentration by squaring the market share of each competing organization in a market, and summing the resulting number
4. Reflective
5. $\sum_{i=1}^{N} s_i^2$ where

   $N =$ number of firms

   $s_i =$ market share of firm i

Metric 2

1. Size diversity
2. Environmental complexity
3. Measures the distribution of organizations in different size categories
4. Reflective
5. $\frac{\sum_{j=1}^{m} O^2}{(\sum_{j=1}^{m} O)^2}$ where

   $O =$ number of organizations in the industry

   $m =$ number of organizational size classes

Metric 3

1. Heterogeneity of output
2. Environmental complexity
3. Measures the number and distribution of industries to which a given industry sells its output
4. Reflective
5. $\frac{\sum_{j=1}^{m} D^2}{(\sum_{j=1}^{m} D)^2}$ where

   $D =$ dollar volume of outputs in the industry

   $m =$ number of organizations buying outputs

Metric 4

1. Specialization rate
2. Environmental complexity
3. Measures the proportion of an industry's shipments accounted for by primary products. This reflects the diversity of products offered by an organization in the industry
4. Reflective

5. $\frac{PP}{TS}$ where

   $PP =$ number of primary product shipments

   $TS =$ total shipments of all products

Metric 5

1. Labor diversity
2. Environmental complexity
3. Measures the diversity of different labor types present in the industry. Labor types are defined based on the Standard Occupational Classification
4. Reflective
5. $\frac{\sum_{j=1}^{m} L^2}{(\sum_{j=1}^{m} L)^2}$ where

   $L =$ number of employees

   $m =$ number of occupational codes represented in the industry

Metric 6

1. Asset size
2. Environmental complexity
3. Measures the average asset size of an organization in the industry
4. Reflective
5. $\frac{AS}{O}$ where

   $AS =$ total asset size of the industry

   $O =$ number of organizations in the industry

Metric 7

1. Capital intensity
2. Environmental complexity
3. Measures the ration of the value of assets to the value of outputs for an average organization in the industry
4. Reflective
5. $\frac{AS}{D}$ where

   $AS =$ total asset size of the industry

   $D =$ dollar volume of outputs in the industry

Metric 8

1. Technical level of workforce
2. Environmental complexity
3. Measures the percentage of the workforce classified in scientific, engineering or other technical occupations, as defined by the Standard Occupational Classification
4. Reflective

5. $\frac{TO}{W}$ where

$TO =$ number of employees working in a technical occupation

$TS =$ total workforce

5.4.4 Mission

The mission of the company is an important aspect in perceived complexity. During the interviews, multiple stakeholders indicated that the presence of a clear vision for the company and the architecture was of importance for complexity. When it is clear for a stakeholder what the goal of an architecture is, this makes is easier for them to understand its purpose and design.

No metric measuring this construct currently exists. Therefore, this research proposes new metrics, as introduced below.

Metric 1

1. Company vision
2. Vision
3. Measures to what extent the stakeholder is aware of the goal or strategy of the organization
4. Reflective
5. Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree):
   I know and understand the vision and strategy of [organization] as an organization

Metric 2

1. Architecture vision
2. Vision
3. Measures to what extent the stakeholder is aware of the goal or strategy of the enterprise architecture
4. Reflective
5. Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree):
   I know and understand the architecture vision of [organization]

5.4.5 Model

During the interviews, understandability of the architecture was posed by stakeholders to be an important aspect of perceived complexity. This understandability was linked to "insight in the structure of the architecture". From the interviews, it can be traced that this understandability is dependent on two construct: documentation, and the communication of this from the producers towards the other stakeholders.

Three aspects influencing complexity can be inferred from the cognitive effectiveness principles stated by Moody (2009), as introduced in section 2.5 "Subjective complexity":

- **Perceptual discrimination**: Element shape, size, colour, etc.
- **Working memory**: Size of the documentation
- **Long-term memory**: Consistency in documentation notation

These aspects were confirmed during the interviews. The notation of elements was specifically mentioned by several participants: "Do people know the documentation language (...) or is it just blocks and arrows to people? We as architects (...) assign meaning to a block and an arrow. However, for an average reader, these are just blocks and arrows. They don't assign the same meaning to that". Regarding the size of the documentation, participants emphasized that this should be dependent on the target audience: "with some you should share the details, and with others you shouldn't". When discussing consistency, one participant noted that "presentation is very important, especially when it concerns the perception of complexity. It's all about repitition, using standardized structures and patterns that people recognize".

Literature showed that the level of detail of documentation does not necessarily influence understandability positively or negatively. However, the interviews showed that the main source of increased perceived complexity lies in the mismatch between the audience and the documentation detail. Having extremely detailed documentation is in itself not a bad thing, as long as it is shown to a fellow architect, and not the board of directors. Another aspect that was mentioned, in both literature and interviews, is the notation of documentation. UML may be logical to architects, but may not be to those unfamiliar with its syntax. Here, notational elements should be based on the audience as well.

Therefore, understandability follows from the correct matching of documentation detail and notation and the audience, and consistency in applying this. This is dependent on four aspects:

1. The availability of documentation to stakeholders
2. The availability of different levels of documentation details and notational elements
3. Consistency in their use
4. Proper communication of the appropriate level of detail and notational elements to different stakeholders

5.4.5.1 Documentation quality

Metric 1

1. Available levels of detail
2. Documentation quality
3. Measures the availability of different levels of detail in the documentation
4. Formative
5. Count how many levels of abstraction are available in the documentation

Metric 2

1. Available notational elements
2. Documentation quality
3. Measures the availability of different notational elements for an element in the architecture. Ideally, this number should be 1
4. Formative
5. Divide the total number of notational elements by the number of element types in the architecture

5.4.5.2 Communication

Metric 1

1. Documentation availability
2. Communication
3. Measures the availability of the documentation
4. Reflective
5. Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree):
   I know where to find relevant documentation on the enterprise architecture

Metric 2

1. Documentation detail suitability
2. Communication
3. Measures the suitability of the documentation's abstraction level for individual stakeholders
4. Reflective
5. Answer the following question on an ordinal scale of 1 (way too little/much detail) to 5 (right amount of detail):
   To what extent does the documentation of the enterprise architecture that you work with have the correct level of details?

Metric 3

1. Documentation notation suitability
2. Communication
3. Measures the suitability of the documentation's notational elements for individual stakeholders
4. Reflective
5. Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree):
   The notational elements of the enterprise architecture documentation are understandable and clear

Metric 4

1. Documentation consistency
2. Communication
3. Measures the consistency of the documentation
4. Reflective
5. Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree):
   The documentation of the enterprise architecture is consistent

5.4.6 Stakeholder qualities

During the interviews, different stakeholder qualities were among the constructs with the highest prevalence. According to interview participants, several stakeholder qualities can influence the complexity perception of a stakeholders. One construct (stakeholder experience) was even mentioned at all twelve interviews. Below, each stakeholder quality is quantified using several scales.

5.4.6.1 Affinity

A stakeholder's affinity with technology was often mentioned as one of the indicators of whether a stakeholders would perceive the architecture as complex. In general, the interviews agreed that a higher affinity with technology leads to a lower perception of complexity. This metric is a scale that was adapted from Edison & Geissler (2003). The paper measures technology affinity by defining ten statements that need to be answered using a Likert scale of 1 (strongly disagree) to 5 (strongly agree). The technology affinity of a person can be calculated using the answers of these ten questions. The questions have been slightly adapted to serve the context of enterprise architecture better.

Metric 1

1. Technology affinity
2. Affinity
3. Measures a stakeholder's affinity for technology
4. Reflective
5. Score the ten questions below on a Likert scale of 1 to 5 and calculate their average

Questions

1. Technology is an important part of my life
2. I enjoy learning about new technologies
3. My job requires me to know about different technologies
4. I usually have no trouble learning new technologies
5. I relate well to the technology used in my job
6. I am comfortable with new technologies required for my job
7. In my job, I know how to deal with technological malfunctions or problems
8. Solving a technological problem is a fun challenge
9. I find most technologies easy to learn
10. I feel as up-to-date on technology as my peers

5.4.6.2 Education

During the interviews, the education or background of a stakeholder was mentioned as one of the factors influencing their perception of complexity. Specifically, interviewees mentioned that those with a technical education will usually have a lower perception of complexity. Based on this, this research proposes a new metric to measure this.

Metric 1

1. Education area
2. Education
3. Measures a stakeholder's area of education
4. Formative
5. Ask participant what area they studied. See options below. Each area will be assigned a score. The final education score is calculated by summing the education area and level

Education area — Score	
Alpha	2
Beta	5
Gamma	2

Metric 2

1. Education level
2. Education
3. Measures a stakeholder's education level
4. Formative
5. Ask participant what level they studied. See options below. Each level will be assigned a score. The final education score is calculated by summing the education area and level

Education level[1] — Score	
University	5
Hbo	4
Mbo	2
High school	1

5.4.6.3 Experience

Experience was mentioned in 100% of the interviews conducted as a factor influencing the perception of complexity. During the interviews, participants mentioned both experience with the organization (e.g. knowledge of the organizational structure) and with enterprise architecture as a discipline (e.g. experience with reading documentation or with the different aspects of EA). The metrics below quantify both.

Metric 1

1. Experience in organization
2. Experience
3. Measures the experience of a stakeholder in the organization
4. Formative
5. Years active in the organization: [0-2], [2-5], [5-10], [10-20], [20+]

Metric 2

1. Experience with enterprise architecture
2. Experience
3. Measures the experience of a stakeholder with enterprise architecture
4. Formative
5. Years of experience with enterprise architecture: [0-2], [2-5], [5-10], [10-20], [20+]

[1] Currently only suited for the Netherlands

5.4.6.4 Role

Metric 1

1. Role
2. Role
3. Identifies the role of the stakeholder: producer, facilitator or user
4. Formative
5. Ask the participant for their job description and classify them in one of the three roles

5.5 Measurement model

The final measurement model and the result of this research is created by combining the conceptual model in Figure 5.9 with the metrics described. Combined, these provide a method for measuring subjective complexity. It should however be noted that, of course, all these constructs and metrics may have different weights. This measurement model is but a first design, and the definitive calculation of subjective complexity requires comprehensive empirical data on the use of the model. Additionally, the model should be validated through empirical research. The next chapter, Validation, describes the first steps of validation that were taken by this research. Finally, section 7.7 “Future research” describes several suggestions for the further validation of the measurement model.

6. Validation

6.1 Expert validation

To validate the results obtained in the previous steps of the research, experts have been consulted. Three expert interviews of one hour each were conducted to validate several steps in the model development: construct identification, conceptualization and operationalization. Due to the size of the model, it was not feasible to discuss the entire model with each expert within the time frame of one hour. Therefore, the validation was split, discussing different aspects with each expert. Below, a short overview of each expert interview is given. The interview number correspond to the expert numbers given in Table 3.3.

6.1.1 Interview 1

One of the experts was interviewed regarding objective complexity. During the interview, all constructs influencing objective complexity were discussed. First, the expert was presented with the model and given an explanation of the modelling choices. Next, he was asked to validate the identified constructs and their relations. Each construct was discussed independently, evaluating its influence on complexity as well as its relations to other constructs. Due to time constraints, the translational validity of the metrics measuring these constructs was not discussed during the interview.

6.1.2 Interview 2

This expert was interviewed regarding subjective complexity, discussing all construct influencing this. A similar approach was used, presenting the model and explaining the modelling choices first. Additionally, the expert was asked to validate the metrics designed to measure each construct, ensuring their translational validity. This lead to several minor modifications of some metrics, that are already included in section 5.4 “Operationalization”.

6.1.3 Interview 3

Finally, the third expert was asked to validate the measurement model as a whole, and to provide feedback on its potential use in practice. The discussion mainly focused around the validity of the model as an artifact to satisfy stakeholder goals. This lead to several suggestions of its practical use, which have been included in section 7.5 “Recommendations”.

6.1.4 Model changes

Based on the expert interviews, several modifications of the conceptual model were made. The first modification concerns the architecture group. One of the experts noted that in the model introduced in Figure 5.2, the constructs conformity, coupling and standardization contribute to domain complexity in two ways: directly, and through the constructs of heterogeneity and modularity. The expert suggested to remove the direct link between the constructs and domain complexity, including the constructs only as causes for heterogeneity and modularity, to prevent hem from being "counted twice". This suggestion was adopted, leading to the respecification of the architecture group as shown in Figure 6.1.

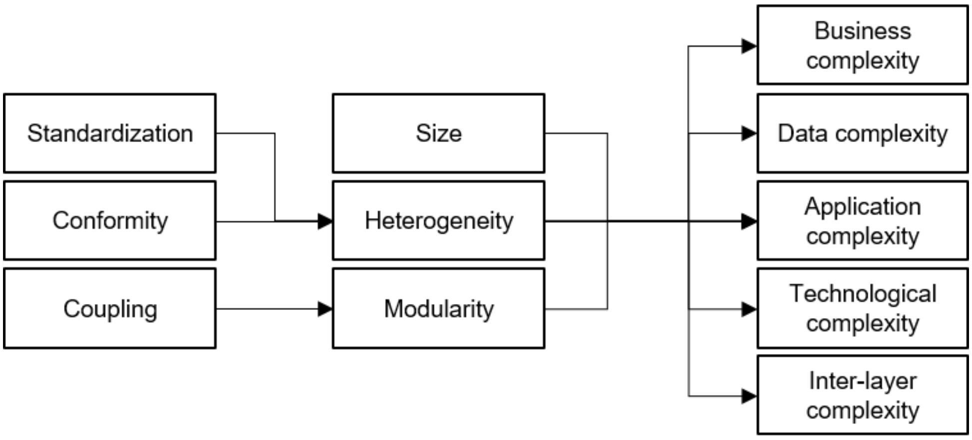


Figure 6.1: Respecification of architecture-related constructs and their relations

Secondly, a respecification of the enterprise group was suggested by adding an additional relation. The expert suggested that politics, currently only influencing governance, can lead to legacy as well. An example that was mentioned was the IT manager that poured his heart and soul into a system when he was still a developer. As a manager, he might exert political pressure to keep this system running, even if it turns into legacy, due to his personal history. The suggested relation was adopted, leading to the respecification of the enterprise group as shown in Figure 6.2.

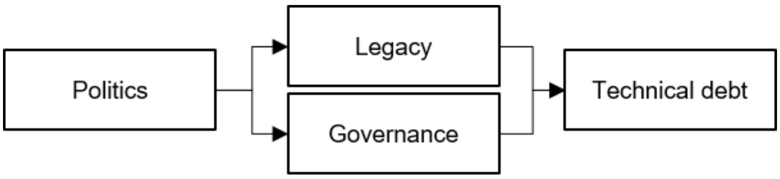


Figure 6.2: Respecification of enterprise-related constructs and their relations

Finally, a suggestion to the design of the model was made. One of the experts argued that constructs from both the enterprise and the environment group should directly relate to domain complexity. In Figure 5.9, environmental complexity and technical debt directly influence objective complexity. However, these constructs can directly influence the elements and relations of the different domains. Technical debt influences not only the application and technological complexity through the existence of legacy, but also heavily influences the structure of all domains. The same can be argued for environmental complexity. Therefore, the relations of these construct will be moved up in the causal chain, leading to objective complexity through domain complexity. Combining these suggestions leads to the formulation of an adapted conceptual model. Figure 6.3 visualizes this improved conceptual model.

Several additional suggestions to the model were made by the experts that were not adopted. This was either due to the scope of the research, or because the suggestions were contradicting with the interview data. In this latter case, the empirical data found in the study was leading. These expert suggestions have been included in section 7.2.3 "Alternative models".

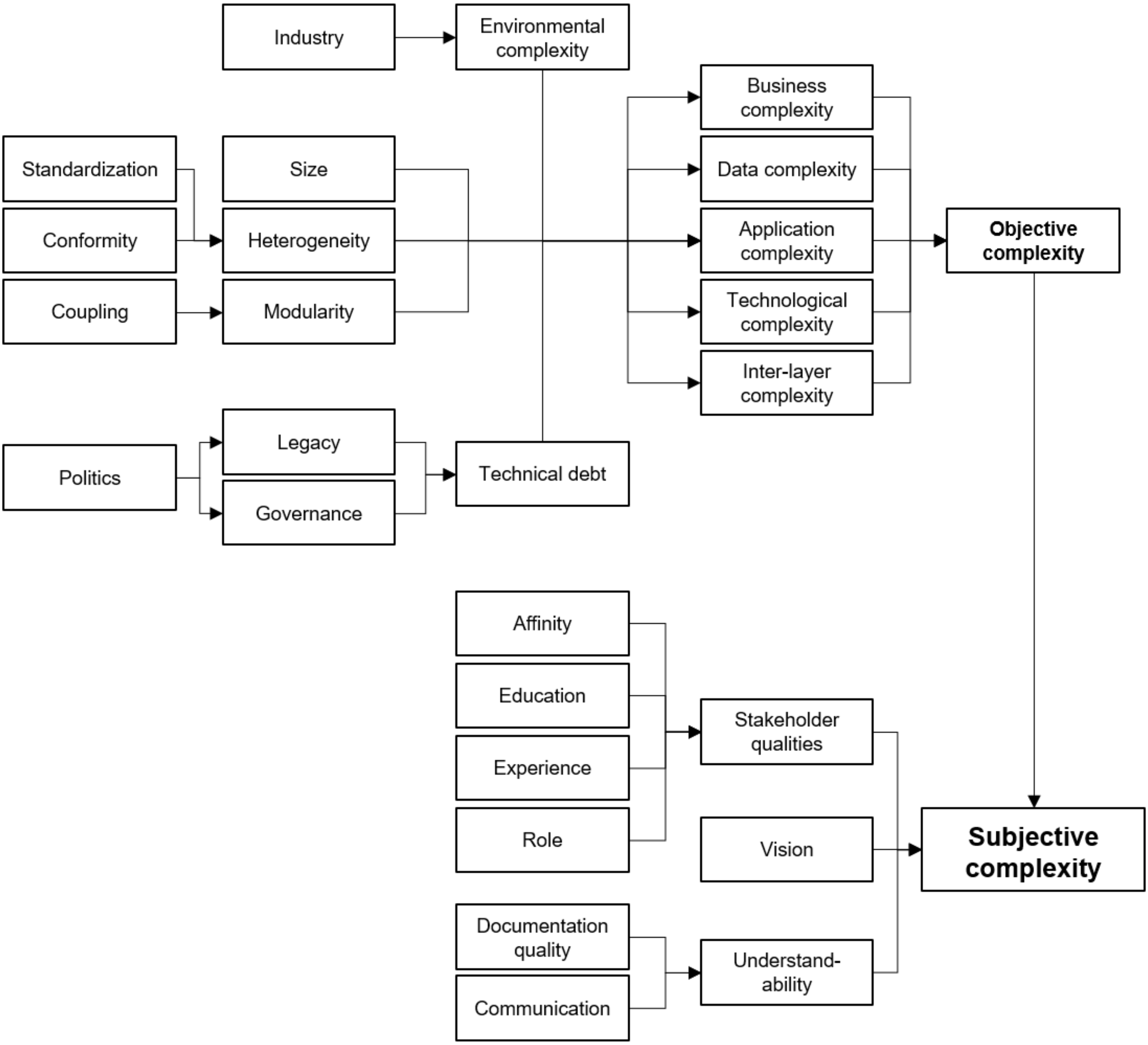


Figure 6.3: Improved conceptual model of architectural complexity

6.2 Case study

To further validate the designed measurement model, it was applied in practice during a case study. At a large organization in the logistics-sector, data was collected on the subjective complexity of three participants, one from each of the roles identified in this research. Data was collected through a survey - in which the metrics introduced in section 5.4 “Operationalization” were queried - and interviews. The goal of the case study was to validate the relations between the constructs *stakeholder qualities*, *vision*, *understandability* and *subjective complexity*. Additionally, the case study aimed to determine the relative weight of each of these constructs.

6.2.1 Survey results

Data was acquired from three participants using an online survey, which can be found in Appendix E “Case Study Survey”. Table 6.1 displays the results of each participant for the corresponding constructs. The construct “stakeholder qualities” (SQ) was divided into several constructs in accordance with the conceptual model of Figure 6.3.

	Participant 1	**Participant 2**	**Participant 3**
SQ: Role	User	Facilitator	Producer
SQ: Education score	10	7	7
SQ: Affinity	4.2	4.0	3.9
SQ: Experience within organization	[2-5]	[0-2]	[2-5]
SQ: Experience with EA	[0-2]	[5-10]	[2-5]
Understandability	3.25	3.25	3
Vision	3.5	4.5	4.5
Subjective complexity	8.0	8.0	8.0

Table 6.1: Validation survey results

6.2.1.1 Stakeholder qualities

The constructs education, affinity and experience were measured directly using the metrics designed in section 5.4 “Operationalization”. Each participant was assigned a role based on the job description given in their survey. To correctly represent diversity in roles, one participant was found for each of the roles identified in section 2.4 “Stakeholders”:

- Participant 1 is a project architect and therefore a user of the architecture. Although also a producer on some level, this validation was focused on the enterprise architecture at a domain level. On this abstraction level, a project architect can be considered a user of the domain architecture.
- Participant 2 is a business information manager. His role is closely related to that of the domain architect, but is more focused on the translation of business requirements towards IT and the facilitation of projects. Therefore, he is considered a facilitator.

- Participant 3 is an enterprise architect and producer, specifically responsible for one domain and together with his fellow architects responsible for the entire enterprise architecture.

6.2.1.2 Understandability

The metrics used for measuring documentation and communication were combined into a single score for understandability. This was done by taking a participant's average over all questions. The results indicate that all participants agree to a certain extent on the quality of the documentation and communication thereof.

6.2.1.3 Vision

Similarly, the score for vision was calculated by averaging the two metrics introduced for this construct. It shows that the vision at organization E is fairly well-communicated towards its stakeholders.

6.2.2 Interview results

Using the survey data as input, each participant was subsequently interviewed. Taking about an hour each, the one-on-one interviews provided a better insight into the subjective complexity of each participant and the influence of the different constructs on this.

The interviews confirmed the existing list of constructs and the relations between them. Each participant endorsed the constructs in the current model as influencing their perception of complexity. Although several suggestions were made for additional constructs, none were mentioned more than once. Combining these suggestions with the existing list of codes provided in Table 4.1 did not result in any additional codes obtaining a prevalence larger than 25%. Therefore, the existing list of constructs can be considered complete when employing a minimum prevalence of 25%. Additionally, the relations between the constructs were validated for those constructs influencing subjective complexity. This means that, all else being equal, stakeholder qualities, vision and understandability can be used to measure and predict subjective complexity. Therefore, the subset of the conceptual model visualized in Figure 6.4 can be considered valid in this case.

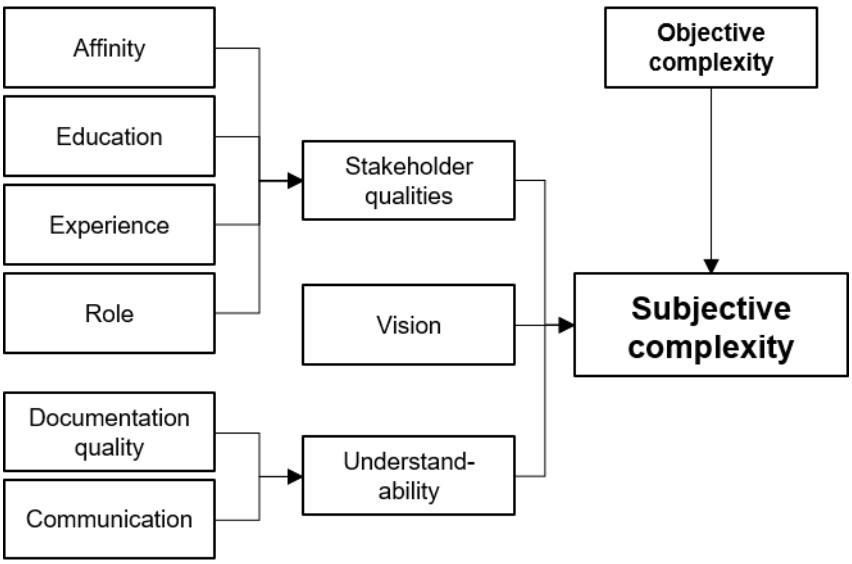


Figure 6.4: Validated conceptual model of subjective complexity

When asked about the respective weight of the different constructs, participant estimated role, experience, documentation and communication as the most important constructs, contributing most to subjective complexity. This is in line with earlier results, where role, experience, presence of documentation and affinity had the highest prevalence (all 50% or higher).

6.2.3 Construct weights

Based on the results from the initial interviews as well as the survey and validation interviews, the different weights of the constructs can be determined. This is done by the formula introduced in section 3.4.2 "Case study":

$$(x * \textit{stakeholder qualities}) + (y * \textit{vision}) + (z * \textit{understandability}) = \textit{subjective complexity}$$

Using the data from the survey, those values of x, y and z can be calculated that complete the formula for each participant.

6.2.3.1 Score calculation

First, all metrics should provide results on a similar scale to enable their comparison. To this end, several metrics will be recalculated to fit on a scale of 1-10. Below, the recalculation of several constructs is elaborated.

Role
This construct is measured through a nominal scale, rather than an ordinal scale, and assigning a value of 1-10 to each role would be arbitrary. However, interviews did suggest a difference in subjective complexity between roles. Generally, facilitators can be expected to have the highest subjective complexity, followed by users, with producers having the lowest complexity. This is due to the level of involvement of the different roles with the architecture.

To take this difference between roles into account, the final score of the other stakeholder qualities will be corrected for their role. Producer scores will be multiplied by 1.1, whereas facilitator scores will be multiplied by 0.9. User scores will remain unchanged.

Affinity, documentation & vision
Affinity, documentation & communication and vision are all calculated on a Likert-scale of 1 to 5. This will be recalculated by doubling the scores[1].

Experience score
Experience metrics are based on an ordinal scale containing five brackets. Brackets can be therefore be assigned a score of 1 (0-2 years of experience) to 5 (20+ years of experience). Similar to the other constructs rated on an ordinal scale, this will be recalculated by doubling its value.

Stakeholder qualities
The constructs role, education, affinity and experience can be combined in a single construct of stakeholder qualities. This can be done using the following calculation:

$$S = \frac{\sum(X * W_x)}{\sum W_x} * R \text{ where} \tag{6.1}$$

[1] Although this technically results in a scale of 2-10 instead of 1-10, this minor difference will be ignored

$$
\begin{aligned}
S &= \text{Stakeholder qualities score} \\
X &= \text{Score construct } X \\
W_x &= \text{Weight construct } X \\
X &\in \{\text{Education, Affinity, Experience}\} \\
R &= \text{Role score } (0.9, 1.0 \text{ or } 1.1)
\end{aligned}
$$

6.2.3.2 Complexity calculation

In order to calculate subjective complexity, the constructs of stakeholder qualities, vision and understandability will be combined. This can be done similar to calculating the stakeholder qualities:

$$
C = \frac{\sum(Y * W_y)}{\sum W_y} \text{ where} \tag{6.2}
$$

$$
\begin{aligned}
C &= \text{Complexity score} \\
Y &= \text{Score construct } Y \\
W_y &= \text{Weight construct } Y \\
Y &\in \{\text{Stakeholder qualities } (S)\text{, Vision, Understandability}\}
\end{aligned}
$$

However, it is important to note that this final complexity score does not yet equal subjective complexity. The metrics use a scale of 1-10, with 10 being highest. This means that a higher value of C means a better understanding of the architecture and its vision, a relevant education and role, and more affinity and experience. All of these characteristics have been validated to lead to lower subjective complexity. Therefore, subjective complexity is the inverse of C. Since $C = 1$ means no understanding, experience, etc., this should lead to the higher complexity, i.e. 10. Likewise, $C = 10$ means a perfect understanding, a lot of experience, etc., and should lead to the lowest complexity, i.e. 1. Assuming this relation is directly proportional, subjective complexity (SC) can be calculated by:

$$
SC = 11 - C \tag{6.3}
$$

6.2.3.3 Weight calculation

Since the values for all constructs are known, as well as the value for subjective complexity, the remaining unknown variables are W_x and W_y. Finding the correct weights should have an empirical basis. To determine the importance and weight of each construct, this research will look at the prevalence of the constructs in the initial interviews, as can be found in Table 4.1. Summing the prevalence of all codes related to the constructs (provided their prevalence is 25% or higher), gives the following results:

Construct	Related code(s)	Prevalence
Education	Stakeholder education and background	3
Affinity	Stakeholder interest and affinity	6
Experience	Stakeholder knowledge and experience	12
Stakeholder qualities	• Stakeholder education and background • Stakeholder interest and affinity • Stakeholder knowledge and experience • Stakeholder role	28
Understandability	• Abstraction level of documentation • Communication to stakeholders • Insight in the architecture structure • Notation of documentation • Presence of documentation	22
Vision	Presence of vision or strategy	5

Table 6.2: Total prevalence per construct

Dividing the prevalence by the smallest number (3 for education, affinity, and experience and 5 for stakeholder qualities, understandability, and vision) gives the relative weight of each construct. Using these weights and formulas (6.1), (6.2) and (6.3), the subjective complexity can be calculated for each participant:

Construct	Weight	Participant 1	Participant 2	Participant 3
Role		1.0	0.9	1.1
Education	1	10	7	7
Affinity	2	8.4	8	7.8
Experience	4	3	4	4
Stakeholder qualities	5.6	5.5	5.0	6.1
Understandability	4.4	6.5	6.5	6.0
Vision	1	7.0	9.0	9.0
Complexity score		6.1	6.0	6.3
Subjective complexity		**4.9**	**5.0**	**4.7**

Table 6.3: Calculation of subjective complexity based on construct values

Obviously, the calculated score does not match the actual subjective complexity, which was 8 for all participants. However, this calculation does not consider objective complexity. Although objective complexity is equal for all participants and can therefore be ignored as a factor, it does still influence subjective complexity, as is shown in Figure 6.4. Therefore, it is likely that the difference between the calculated an actual subjective complexity stems from a high objective complexity in organization E.

Nevertheless, the calculation in Table 6.3 does show an approximately equal score for each of the participants, which is consistent with the actual subjective complexity. Although the actual numbers do not match (5 versus 8), the numbers are in both cases equal for each participant. Taking into account the possible influence of objective complexity, which is also equal for all participants, indicates that the weights are valid for this particular case. Although it is hard to generalize from a single case with $n = 3$, this does suggest the results are in the right direction.

7. Discussion

This section discusses the results of this research for each of the steps taken in the design of the measurement model. It aims to find an explanation of the results found and interpret their meaning, as well as elaborating on their validity and generalizability. Subsequently, recommendations are made towards enterprise architecture practitioners on the implications of this research for practice. Several limitations in the approach taken and the results found are discussed as well. Finally, possible future research directions are reviewed, suggesting the next steps to be taken in the study of subjective complexity.

7.1 Constructs

7.1.1 Construct identification

The design of a measurement model was started by the extraction of several constructs from the literature and interview data. The structured literature review and interviews resulted in a long list of 12 concepts and 48 codes potentially influencing subjective complexity, which can be found in Table 2.3 “Identified concepts” and Table 4.1 “Identified codes and their prevalence”. These were then divided into six groups, indicating the different areas impacting complexity. The code groups found in the interview results show a similarity to several construct defined in Figure 3.1 “Conceptual model of enterprise architectures and their context”. This is in line with expectations, since Figure 3.1 is a conceptual model that defines the context of an enterprise architecture. The fact that the construct groups can be directly mapped onto the conceptual model attests to their completeness, since all influential construct are represented in the interview results. The metrics found in the literature review all measure the objective complexity of an architecture. Since these results are a subset of subjective complexity, the concept groups found in literature have been mapped onto the results from the interviews.

In the interview results, minor differences could be found between the three stakeholder groups interviewed. A breakdown of the codes per stakeholder group can be found in Appendix D “Detailed Interview Results”. For most codes, the difference between stakeholder groups is negligible and can be attributed to chance. There are however some interesting differences visible in some codes. An example is business complexity, which is mentioned by producers three times, once by users and never by facilitators. A possible explanation for this might be that facilitators, often originating from the business-side of the organization, have a better grasp of the business processes and therefore consider these to be less complex. However, the exact same distribution can be found in the coupling code. Providing the same explanation for the distribution is unlikely for this code. Here, the difference in distribution might be explained due to the fact that coupling as a factor could be invisible or unknown to facilitators, and therefore not considered. Looking at the total number of codes identified by stakeholder groups, it shows that producers and users mention more possible causes for complexity (94 and 89 respectively) than facilitators (70). This might be due to the fact that some levels of complexity are hidden from facilitators and therefore not considered, as was mentioned with coupling. However, this is merely an assumption, and a definite explanation to this discrepancy would require further research.

The size of the list of literature concepts and interview codes was too large to be processed into a model. To ensure the results are thoroughly grounded in practice, a prevalence margin of 25% was upheld. Due to this margin, several suggestions made by literature and practice were discarded for use in the conceptual model. However, this does not mean they do not influence subjective complexity to some extent. Both complexity and perception are very comprehensive and intricate constructs, that are likely influenced by a near endless number of other constructs. However, the case study performed in this research used the prevalence of the interviews as an indicator of the relative weight for each construct. This turned out to be a correct indication of the relative weights, since the subjective complexity given by the participants could be calculated from their input using these weights. This suggests that the relative prevalence of each construct in the interviews is an accurate representation of its importance towards subjective complexity. A higher prevalence means a larger contribution of that construct to subjective complexity. From this, it can be concluded that the constructs disregarded through the prevalence margin have a very limited influence on subjective complexity. Including all constructs in a conceptual model would make it unpractical and defeat its purpose of measuring architectural complexity. Therefore, the prevalence margin of 25% seems justifiable.

Several of the constructs present in the final model have been adopted from existing literature. Since previous research was focused on the measurement of objective complexity, these constructs focus around the construct group of "architecture". Additionally, metrics suggested by this literature have been adopted to measure these constructs. Since many of these papers carried out some form of empirical validation of these constructs and metrics, a large part of the constructs from the architecture group, including their metrics, can be considered validated to some extent.

7.1.2 External validity

An important factor to consider when drawing conclusions from empirical data is the external validity or generalizability of the results. Since this research is aimed at measuring subjective complexity in the context of enterprise architecture, the results should be valid across different organizations and for different stakeholder. To ensure cross-industry applicability, the interviews have been held at different organizations operating in different industries. To ensure the applicability of the results on different stakeholders, employees in different roles were interviewed at each organization. This makes the results generalizable over stakeholders from different roles. The resulting division of data collection over different industries and roles can be found in Table 3.1 "Number of participants interviewed per role and industry". Of course, a stakeholder was defined using more attributes than just role, as can be found in Figure 2.4 "Stakeholder attributes and their possible values". Ideally, a similar cross-validating approach should have been used for the other attributes defining stakeholders, i.e. level and domain. Unfortunately, the number of interviews was limited, and these attributes have not been considered.

7.2 Conceptual model

7.2.1 Model design

Using the constructs identified, a conceptual model was designed by defining relations between these constructs. The relations were defined based on data from literature and interviews. During this process the interview transcripts were analyzed for indicators of relations mentioned by participants. An example of this is one of the participants mentioning: "Our processes being complex, is actually simply because the legislation is complex", which is a clear indicator of environmental complexity directly influencing objective complexity. Another source of these relations was the participant being asked directly regarding the cause of a construct. During the introduction, understandability was identified as inherently influencing subjective complexity. Therefore, participants were directly asked about the understandability of the architecture and the constructs influencing this (see Appendix C.3.3 “Key questions”). Their answers to these questions can therefore be considered to have a direct relation to that construct, e.g. understandability. Finally, relations were defined using the information acquired through the literature review.

Of course, the translation of this data into construct relations leaves room for interpretation, creating multiple possible conceptual models. To ensure a valid conceptual model, its final design has been realized through several iterations. The first draft of the model created by the author was discussed with the supervisors of this research, and refined using their feedback. Subsequently, three expert interviews were performed, which lead to several adjustments of the conceptual model. After additional validation of the model through a case study, the final and definitive version of the conceptual model was produced, which can be found in Figure 6.3 “Improved conceptual model of architectural complexity”.

7.2.2 Model relations

An important note to make concerning the conceptual model is the fact that it is, of course, not a 100% accurate representation of reality. Although many important relations have been defined based on the available data, most of the constructs found are likely to be interrelated to some extent. For example, documentation quality can be expected to influence objective complexity as well, since incomplete documentation can lead to incorrect governance and the development of technical debt. Likewise, politics is likely to have a direct effect on subjective complexity, since the involvement of a large number of (high-ranking) stakeholders could lead stakeholders to believe the situation must be very complex. However, while recognizing the existence of these relations, this research believes them to be relatively insignificant and have a minor impact on the measurement model. Since it is not feasible to define and model all existing relations, the proposed model is a suitable basis for the measurement of subjective complexity.

7.2.3 Alternative models

During the expert interviews, several suggestions were made for the improvement or redesign of the conceptual model. Although part of these suggestions were adopted, others were not. One of the suggestions made by an expert entails the entire redesign of the conceptualization of objective complexity. The expert suggested that the constructs found in the enterprise group are viewpoints that can be used to scope the complexity of an enterprise to a specific set of elements and relations. This is similar to the current modelling of business, data, application and technology complexity. Indeed, the expert argued that the enterprise-related constructs should be modelled alongside these constructs. Objective complexity is then defined by the size, heterogeneity and modularity of all these viewpoints. These are in turn influenced by standardization, conformity, coupling and environmental complexity. A visualization of this alternative model of objective complexity is shown in Figure 7.1.

This model implicates the incompleteness of the TOGAF model of four architecture domains, as introduced in section 1.2.1 “Architecture”, suggesting that technical debt, governance and politics should also be considered as domains. In fact, the expert argued there are many more domains to consider, and one of the pitfalls of enterprise architecture is ignoring these additional aspects, focusing only on the four defined domains. Although very interesting, adopting this view has quite far-going implications on the conceptual model and, in fact, on enterprise architecture as a whole. Since this discussion is out of scope for this research and enterprise architecture was defined in this research using TOGAF, the change was not adopted in the final model.

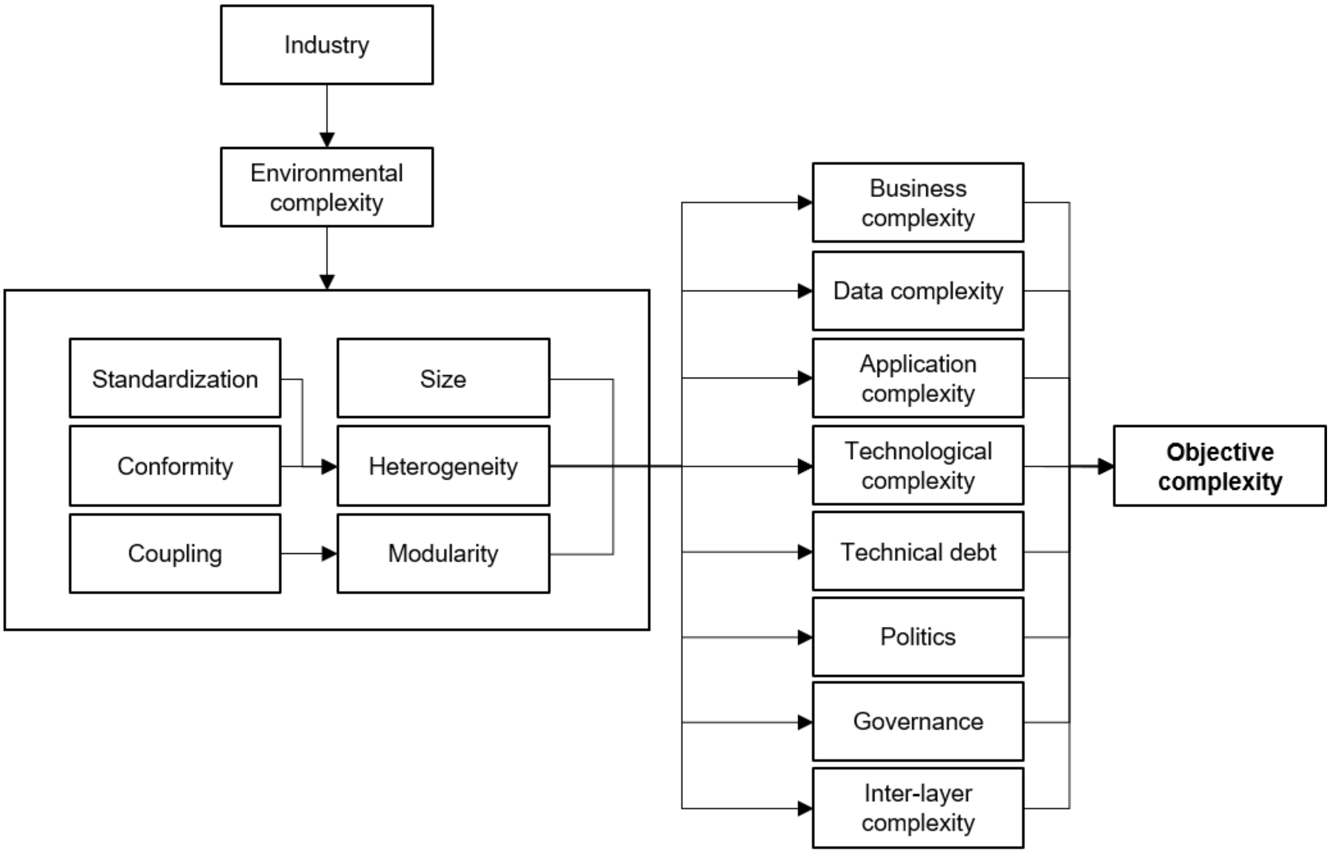


Figure 7.1: Alternative conceptualization of objective complexity

7.3 Measurement model

The designed measurement model operationalizes the conceptual model. The formulation of a series of metrics enables the measurement of the constructs and therefore of subjective complexity. An important aspect of the operationalization is the construct validity of the suggested metrics.

Existing research on complexity measurement is extended by adopting several metrics widely used for the measurement of size and heterogeneity. Additionally, environmental complexity and technical debt are operationalized through the adoption of existing metrics for these constructs. These metrics are suggested by literature, such as the papers identified in the structure review, which can be found in Appendix B.1 “Review paper selection”. Several of these papers include some form of empirical validation of the metrics they propose. Consequently, the construct validity of these metrics can be assumed to some extent. However, it should be noted that it is inadequate to assume full construct validity based on research performed in a different context, and the construct validity of these metrics should be assessed in future research.

Contrarily, no existing metrics have been found for the measurement of the constructs influencing subjective complexity. New metrics have therefore been proposed by this research. Their translational validity has been tested through a series of expert interviews, during which the metrics were presented to experts. Based on the feedback received during these interviews, several metrics were adjusted.

As a result of this process, both the objective and subjective metrics have their construct validity tested to some extent. However, the ensure construct validity for all metrics, a large-scale empirical study is required. Using a large survey to obtain a data set, statistical analysis could be performed in order to test the construct validity of the metrics. This is described in more detail in section 7.7 “Future research”.

7.4 Construct weights

During the case study, weights were determined for each subjective construct based on survey data. These weights were derived from the prevalence of the constructs in the interviews and literature, and applied on the survey data in section 6.2.3.3 “Weight calculation”. This calculation resulted in a partially corresponding result: the calculated complexity was nearly equal for each participant, which was also the case for the actual subjective complexity stated by the participants. However, there was a discrepancy, because whereas the calculated values suggested a complexity of 5 (out of 10) for each participant, the actual complexity indicated by participants was an 8 (out of 10). A potential explanation lies in the fact that objective complexity is an important influence on subjective complexity which in turn was not considered in this calculation. The objective complexity was ignored as a factor during the case study since it was equal for all participants. However, on the final value of subjective complexity, it is still a significant aspect. If the objective complexity in the organization is high, this can explain the discrepancy between the calculated (5) and actual (8) subjective complexity. This effect is demonstrated by a hypothetical calculation in Table 7.1. This shows that with an objective complexity of 9 (out of 10) and the respective weight of 3 to 1 for objective and subjective complexity, the calculated complexity would be correspond with the indicated complexity. Here, the subjective scores are the values that resulted from the case study in Table 6.3 “Calculation of subjective complexity based on construct values”.

	Weight	**Participant 1**	**Participant 2**	**Participant 3**
Subjective score	1	4.9	5.0	4.7
Objective complexity	3	9.0	9.0	9.0
Subjective complexity		8.0	8.0	7.9

Table 7.1: Hypothetical calculation of a new subjective complexity by including objective complexity

Of course, this case study was performed with a sample size of $n = 3$, and no statistically significant conclusions can be drawn based on the results. The case study serves as an example, and an indication that the model could be applied in practice. To further test the measurement model, and determine reliable weights for each construct, additional empirical data is required. If the relative weights of the constructs are known, subjective complexity could be calculated using the metrics provided.

The possibilities of the application of this model extends beyond the measurement of complexity. Theoretically, when the relative weights of all constructs are known, the model could be used to predict certain constructs. An example of this was demonstrated in Table 7.1, where the objective complexity of an architecture was derived from its subjective complexity. If the weights derived from the case study are correct, the objective complexity has been predicted to be 9 (out of 10), given the data from the survey. Since subjective complexity is easier to measure, this could be a reliable method to conduct a quick assessment of an architecture’s complexity. Similarly, the measurement model could be used to predict any construct, given that the values of the other constructs and their relative weight are known. This method of calculating different constructs has many potential use cases in enterprise architecture complexity management, some of which are described in section 7.5 “Recommendations”. Of course, more research is needed to validate the weights found, and to test whether these hold true for different organizations and contexts. This requires a large-scale empirical study, which is described further in section 7.7 “Future research”.

7.5 Recommendations

The results found in this research have several implications for enterprise architecture in practice. As was formulated in the introduction of this research, complexity is one of the major challenges for enterprise architecture. Suboptimal levels of complexity within an architecture can lead to a series of negative impacts for an organization, as was visualized in Figure 1.5. Therefore it was established that the management of complexity is, or should be, an essential occupation for enterprise architecture as a discipline. In the sections below, recommendations are made on how to leverage this research's results to accomplish this.

7.5.1 Recommendations for practitioners

Literature has shown that enterprise architecture strives for an optimal level of complexity called requisite complexity. Where the drawbacks of too much complexity are well-documented, too little complexity is argued to cause problems as well. An architecture that is too simple for its complex environment cannot support the functionalities required to operate in that environment. From this, it can be deduced that complexity reduction should never be a goal in itself. The goal of complexity management is to strive for an optimal level of enterprise architecture complexity, to efficiently and effectively deal with the complexity of the architecture's environment.

For an architecture's complexity to be managed properly, it is important to consider the stakeholders involved. Ultimately, the goal of the enterprise architecture is to enable the organization's strategic goals through the integration and alignment of business and IT. This requires the cooperation of many different types of stakeholders, all of which are involved and required to work with the architecture. The perception of complexity can cause these stakeholders to make inefficient or incorrect decisions, simply because they do not understand the architecture they are working with. Additionally, it can cause resistance to change, hindering the realization of strategic goals through the enterprise architecture. Furthermore, if the perception of complexity differs among stakeholders involved, this can impede their communication and collaboration. An often witnessed phenomenon is a large difference in complexity perception between the business (e.g. managers) and IT (e.g. the architect). This difference in perception causes friction and disagreement, and hinders the proper utilization and development of the architecture.

The statements above make clear that complexity management should consider both objective and subjective complexity. The measurement model designed in this research considers both of these complexity dimensions. Therefore, it is recommended to apply the model when examining architectural complexity. To that end, the model can be employed in two ways. First of all, the underlying conceptual model can be used as an indicator of which attributes to consider. Often, enterprise architecture practitioners do not employ a unified view of complexity; everyone has their own idea of its definitions and underlying attributes. By adopting the conceptual model designed in this research, insight is created in the aspects influencing complexity. From this shared understanding, decisive attributes can be identified and complexity management efforts can be efficiently directed. Secondly, the proposed metrics can be used to quantify the values for each construct, and derive which constructs require attention. An assessment of an organization's complexity can lead to insights in the source of objective and subjective complexity, providing a more precise direction for complexity management efforts. For example, a quantitative assessment could show that the source of complaints about complexity do not stem from a large number of applications, but rather from a lack of documentation and communication. Instead of focusing on reducing the number of applications, architects should focus their efforts on the documentation of the existing architecture, and clearly communicate this towards other stakeholders. This avoids unnecessary (and expensive) technical projects, will help to improve communication and collaboration between business and IT, and ultimately makes the architecture more effective and efficient.

Concluding, the designed measurement model should be leveraged to gain more insight into the sources of complexity, identify critical attributes and facilitate complexity management. Ultimately, this will help enterprise architectures to be used effectively and efficiently, enabling the strategic goals of the organization.

7.5.2 Recommendations for Deloitte

This research was conducted in cooperation with the Enterprise Architecture department of Deloitte Consulting in the Netherlands. As a consultant, Deloitte encounters many organizations struggling with the complexity of their enterprise architecture. This research can benefit Deloitte in several ways.

Firstly, the measurement model can be further operationalized and even automated. A tool could be developed calculating objective complexity based on input from architecture models and/or a configuration management database (CMDB). This could enable a quick insight into the objective complexity of an architecture, and help to make substantiated claims about the state of the architecture. The complexity level could, for example, be included in a present-state architecture document. A note should be made here that this methodology relies on the correctness of architecture models and the CMDB, which is hardly ever the case. However, a decent approximation of the objective complexity serves this purpose well enough.

Additionally, the measurement of subjective complexity could be employed to gain more insight in the situation of the organization. Measuring the subjective complexity of different stakeholders could provide insights that help to better understand the situation of a client. Therefore, the measurement model introduced in this research can be used to quickly gain an insight in the situation when engaging with a new client.

The goal of a consultant often lies in accomplishing change in an organization; Deloitte aims to make an “impact that matters” at their clients, as is their motto. The perception of complexity often causes stakeholders to resist to change. Since they do not understand the architecture they are working with, they are hesitant to initiate projects that have a large impact, since they do not know what this impact will be. This resistance to change is widespread in organizations, and consultants often spend much of their time convincing and negotiating with many stakeholders. Better understanding the complexity perception of these stakeholders means a consultant can discern the underlying causes of this resistance to change and anticipate on this.

To accomplish change at an organization, it is not enough to consider individual stakeholders. The communication and collaboration between stakeholder groups is an important part of most processes. Having a different perception of enterprise architecture complexity impedes this communication and collaboration between stakeholders. Ultimately, these different views on the architecture and its complexity can cause friction between the business and IT of an organization, hindering the proper utilization and development of the architecture. Overcoming this requires an insight in the different perceptions of complexity. The model introduced in this research can help Deloitte to better understand the attributes underlying this discrepancy in perception.

Concluding: complexity perception and its difference between stakeholders is an important cause for ineffective decisions, resistance to change and lacking collaboration. Deloitte can leverage the insights gained through this research to better understand the dynamics of complexity and its perception, enabling them to make an impact that matters.

7.6 Limitations

Like any other, this research has certain limitations that need to be addressed. The first limitation is one that is applicable to many qualitative studies. Both the coding of the interviews as well as the identification of constructs based on its data was established through the insights of the author. A best effort was made to ensure the validity of this process: coding schemes were validated by the interview participants, and constructs were validated by experts. However, since this process is considered more of an art than an exact science and is mostly dependent on the judgment of one person, it provides a limitation to the research results. On a similar note is the limitation that all interviews were performed by the same interviewer (the author), which could introduce a bias in the interview results. Although an attempt was made to prevent this bias by introducing a comprehensive interview guide, results might be slightly biased.

Another limitation lies with the sample sizes of the empirical sections of this research. Twelve interviews have provided a decent insight into the constructs influencing complexity and have most likely uncovered the most important constructs. However, it is very well possible that one or more important constructs have been missed due to the choice of organizations or participants. Therefore, a larger research interviewing more participants in different industries is necessary to ensure the completeness of the measurement model.

Similarly, the case study performed had a limited sample size with $n = 3$. In order to ensure construct validity, data should be acquired on a large scale to enable the statistical analysis of the proposed model. Additionally, the relative weights calculated in this study bear no statistical significance and are mainly used as an illustration of the model's application. Finally, explanatory case studies are best served using multiple cases (Yin, 2009), which was not possible in this research due to time constraints. Performing multiple similar case studies with larger sample sizes will allow for more reliable conclusions on construct validity and the relative weight of constructs.

A last limitation that should be noted is the fact that the final measurement model is probably not fully complete. Many of the constructs in the model are likely to be interrelated to some extent. However, modelling all of these relations would likely make the model unpractical and would not be feasible. Therefore, this model can be considered a close approximation of reality.

7.7 Future research

This research provides several opportunities for the further exploration of enterprise architecture complexity. First of all, a large-scale empirical validation of the designed measurement model should be performed. This research should focus on the following aspects:

- Validating the relations between constructs and establishing their relative weight;
- Validating the construct validity of the proposed metrics.

All these aspects can be accomplished through the use of Structured Equation Modelling (SEM). SEM is a technique that combines factor analysis and multiple regression analysis (Hox & Bechger, 1998). It can be used to measure both the relation between a metric and its underlying construct (construct validity) as well as the causal relations between constructs (Hox & Bechger, 1998). SEM requires the precise definition of a conceptual model defining the relations between constructs, as well as the design of metrics for their measurement. Both of these steps have been taken in this research. Therefore, a recommended next step in the research on subjective complexity measurement would be a study applying SEM to the measurement model designed in this research.

Broadening the scope on complexity measurement also provides opportunities for future research. As was remarked by one of the experts during the expert validation, an important aspect missing in this research is time. According to the expert, complexity, both objective and subjective, arises when the architecture is unable to support architectural changes that need to be made. He quoted: “The inability of an architecture to support future change is an important cause for complexity”. This view takes a new type of construct into consideration: the ability to facilitate future change. Obviously, a vital aspect in this is the change of the architecture over time, which is not represented in this research. The importance of changes over time are also noted by literature, as dynamic complexity is suggested by Schütz et al. (2013) as one of the complexity dimensions. The structured literature review in this study confirmed the lack of research on dynamic complexity: similar to subjective complexity, only 2% of all existing complexity metrics address dynamic complexity (see Table 2.4 “Complexity dimension distribution”). Like this research has addressed subjective complexity, future research should explore the dynamic aspects of enterprise architecture complexity as well. Only by studying all complexity dimensions can enterprise architecture complexity be properly managed.

8. Conclusion

8.1 Conclusions

The goal of this research is to answer the following main research question:

> **How can objective and subjective complexity metrics be incorporated in a unified enterprise architecture complexity model?**

To answer this question, several sub-questions have been formulated. The sections below provide a recapitulation of the chapters in this research, and answer the sub-questions posed.

8.1.1 Objective complexity

1. Which existing metrics are most prevalent for measuring objective complexity in an enterprise architecture?

Objective complexity has already been studied quite extensively in previous research (Schütz et al., 2013). To design a measurement model including both objective and subjective metrics, an extensive review on the state of the art is necessary. To this end, a structured literature review has been performed in order to answer this sub-question. Appendix A "Structured Literature Review Protocol" provides an overview of the methodology used to rigorously scan the current body of literature on this topic. This resulted in a search and selection process scoping the initial 275 results down to 20 papers that define architectural complexity in measurable terms. Each of these 20 papers provides a (series of) metric(s) aimed at measuring objective complexity. Extensive analysis on these papers resulted in a list of 42 metrics used for architectural complexity. Appendix B "Structured Literature Review Results" provides an overview of these metrics, their description and their prevalence among the 20 papers.

Graph-based	**Functional**	**Non-functional**	**Other**
Elements and relations	Functions	Quality	Expert opinion
Patterns	Redundancy	Reliability	Cost
	Application-data	Conformity	
	Hardware-data	Service time	

Table 8.1: Identified concepts in literature

Additionally, concepts were designed that categorize these metrics; each metric was assigned a concept. These concepts were in turn categorized into several concept groups. This resulted in the identification of the concepts shown in Table 8.1. Finally, the prevalence of each of these concepts among the 20 papers was determined, which can be found in Appendix B.5 "Concept matrix".

8.1.2 Stakeholders

2. How can stakeholders be defined in a context-agnostic way?

Subjective complexity, per definition, is dependent on the perception of a stakeholder, and therefore different for each stakeholder. In order to measure subjective complexity it is important to have a context-agnostic definition of this stakeholder, ensuring applicability of the designed model to all stakeholders. This requires the identification of a set of attributes that collectively constitute any enterprise architecture stakeholder.

Several stakeholder classification frameworks have been developed in the context of enterprise architecture. Niemi (2007) aims to create a holistic view on the stakeholders of an enterprise architecture and proposes a classification based on three roles:

- **Producers** are the stakeholders that carry out the planning and development of the enterprise architecture. They usually do not use the architecture other than for their primary work, and are little involved with its management. Producers aim to satisfy the facilitators' and users' requirements.
- **Facilitators** are the stakeholders involved with the governance, management, maintenance or sponsorship of the enterprise architecture. Contrary to producers, they do not directly conduct planning or development of the architecture. They differ from users in the sense that their work affects the architecture.
- **Users** are the stakeholders that utilize enterprise architecture work and its products in their daily work. In contrast to the other roles, they do not carry out work that directly affects the architecture. However, they can be involved in architecture work by providing requirements or feedback.

In another study on enterprise architecture stakeholders, van der Raadt et al. (2008) propose a matrix using two attributes to classify enterprise architecture stakeholders:

- **Level** refers to the organizational level a stakeholder operates in. Four levels can be distinguished:
  - At *enterprise* level, management focuses on the strategy of the architecture, making decisions regarding the future state of the architecture. This level typically operates at C-level.
  - At *domain* level, domain owners and change managers coordinate and manage the programs running in their domain, and IT focuses on the domain architecture. This level operates at business unit-level.
  - At *project* level, stakeholders run projects and implement changes, executing the decisions made by the first two levels.
  - At *operational* level, stakeholders are responsible for the stability and continuity of the operational environment, the day-to-day operations, and the maintenance and optimization of platforms.
- **Domain** refers to the architectural domains that constitute enterprise architecture. These domains are *business*, *data*, *application* and *technology*, and are described in detail in section 1.2.1 "Architecture".

Several other stakeholder classification frameworks in the context of enterprise architecture are available. Many of these, however, can be mapped to one of the frameworks introduced above. Therefore, the framework by Niemi (2007) and van der Raadt et al. (2008) are adopted by this research. Since these frameworks are not mutually exclusive, they can be combined to define a context-agnostic definition of stakeholders in the context of enterprise architecture. This definition is visualized in Figure 8.1.

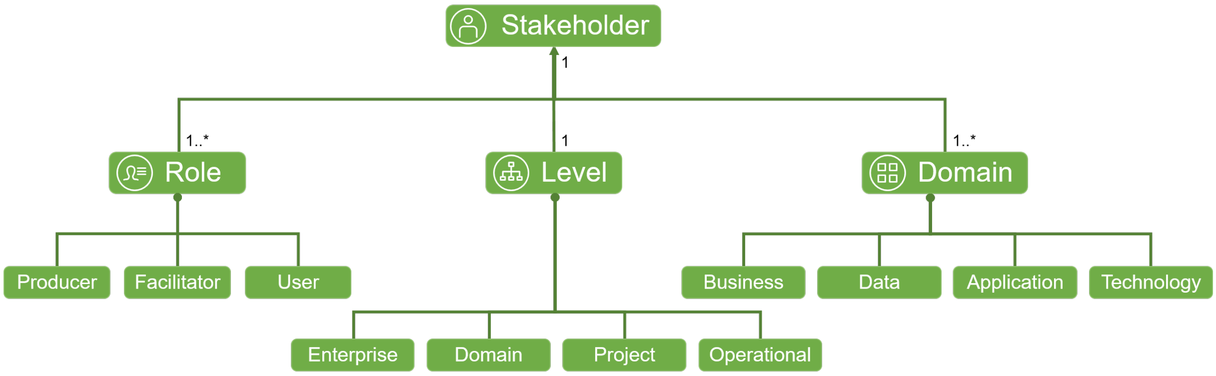


Figure 8.1: Stakeholder attributes and their possible values

8.1.3 Subjective complexity

3. Which attributes influence a stakeholder's perception of enterprise architecture complexity?

The answer to this research question requires the conceptualization of subjective complexity. In this process, empirical data was used to identify the constructs influencing subjective complexity. A construct is a concept, or combined set of concepts, that is created to explain a certain phenomenon that is not directly observable. A stakeholder's perception of complexity can be influenced by many different constructs. To be able to measure the complexity of a specific individual, it is necessary to identify these constructs first.

To this end, twelve interviews have been conducted over four organizations. During these interviews, participants were queried on the complexity of the enterprise architecture in their organization, and the attributes influencing their perception of this complexity. By interviewing one participant from each role (producer, facilitator and user) across different organizations, cross-industry applicability of the results can be ensured for each role. Details on the design of these interviews can be found in Appendix C "Interview Guide". After their transcription, these interviews were quantified by means of open coding and axial coding. This process led to the identification of a long list of codes influencing complexity, which can be found in section 4.2.1 "Code list". Similar to the concepts identified in the structured literature review, each code has a prevalence indicating in how many interviews it was mentioned. To ensure the wide applicability of the results, a prevalence margin was used: only concepts and codes with a prevalence of 25% or higher were included in the next step of the research.

Constructs were identified based on the results of the literature review and interviews. These constructs were then categorized based on a model introduced by the International Organization of Standardization (ISO/IEC/IEEE, 2011) that is visualized in Figure 3.1. The 22 identified constructs and their groups are shown in Figure 8.2.

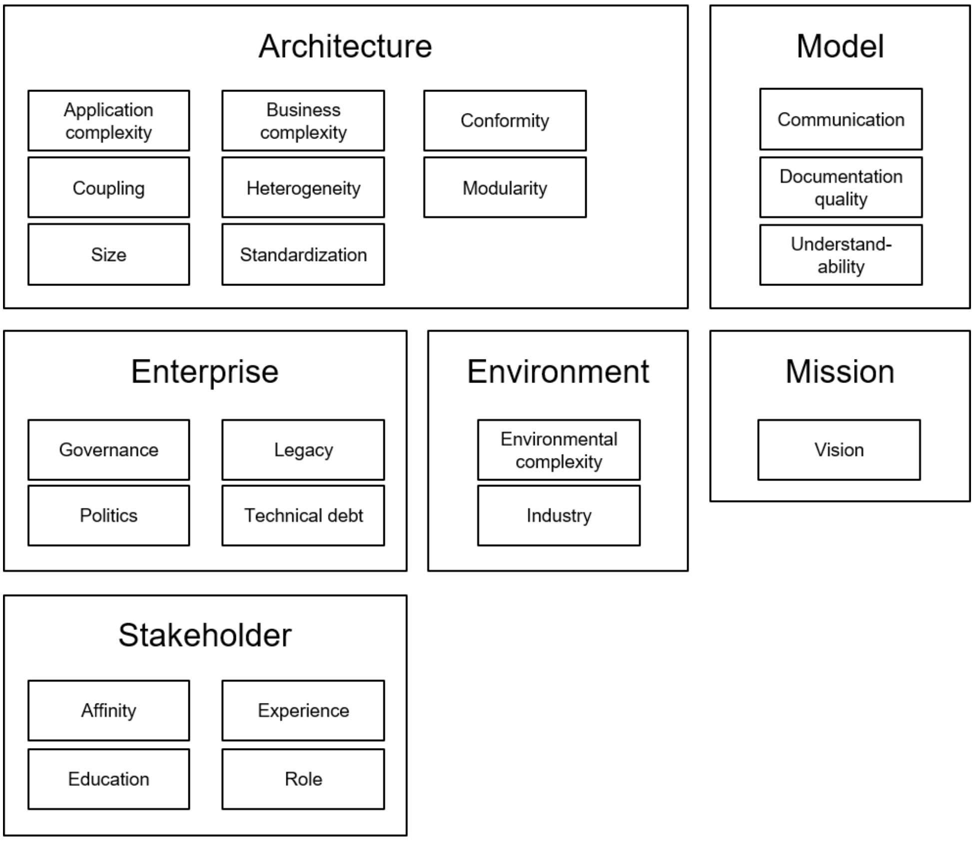


Figure 8.2: Overview of constructs and their grouping

8.1.4 Conceptual model

4. How does stakeholder perception lead to subjective complexity?

The next step in the conceptualization of subjective complexity is the design of a conceptual model. For each group of constructs, relations between the constructs were defined. This was done based on a combination of literature and empirical evidence from the literature review and interviews. For each group, an explanation and visualization of the defined relations can be found in section 5.2 “Construct relations”. Next, the relations between the groups were defined, creating a single model containing all constructs. Thus, based on the literature and empirical evidence, all constructs have been combined into a conceptual model. This model can be found in Figure 8.3. It has been validated through a series of expert interviews and a case study, and found to be correctly conceptualizing subjective complexity.

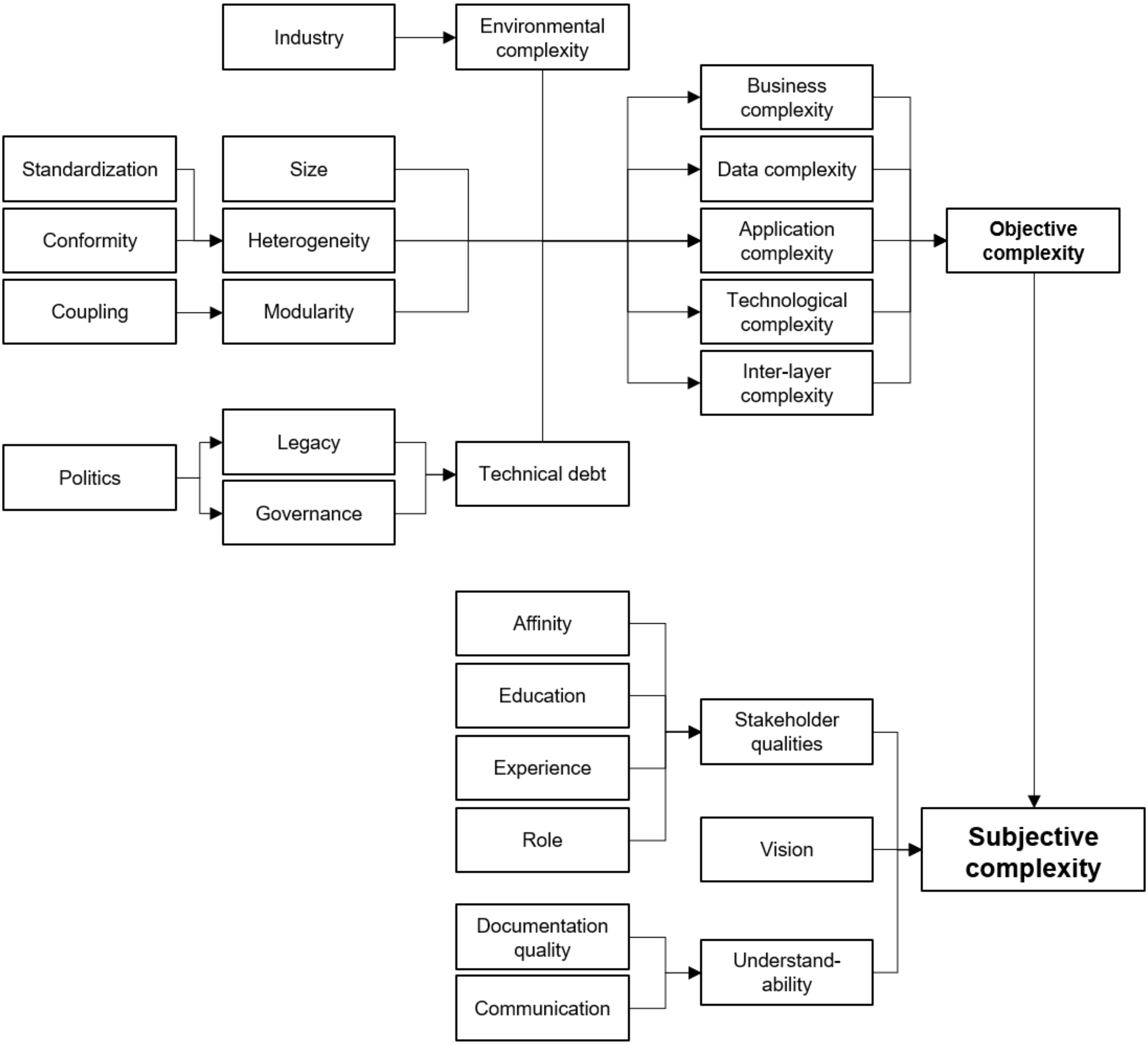


Figure 8.3: Final conceptual model of architectural complexity

8.1.5 Metric design

5. What metrics are suitable for measuring subjective complexity in an enterprise architecture?

After establishing the conceptual model of both objective and subjective complexity, metrics have been designed to measure this complexity. Of course, for some constructs, validated metrics are provided in existing literature. An example for this are several constructs measuring objective complexity: these metrics have been identified in the structured literature review. Therefore, several of the suggested metrics are based on existing literature and are validated metrics for that particular construct. However, since no studies have previously been performed into subjective complexity measurement, several constructs could not be operationalized using existing metrics. For these constructs, new metrics have been proposed. To test the validity of these metrics, expert interviews have been conducted. During these interviews, experts were presented with the metrics and the underlying constructs they should measure. Based on their experience, the experts provided feedback on the validity of these metrics. Several adjustments have been made based on this validation. More details on the expert validation performed in this research can be found in section 6.1 “Expert validation”.

The detailed list of metrics proposed by this research can be found in section 5.4 “Operationalization”. An overview of the metrics designed for each construct is provided in Table 8.2. All metrics have been designed conform an existing template for architectural metrics, containing the following information:

1.	Name	Each metric will have a distinctive name
2.	Construct	The underlying construct measured by this metric
3.	Description	A short description of the metric
4.	Type	A metric can be reflective or formative
5.	Computation	Method of value calculation

Construct	**Metrics**
Size	• Number of elements • Number of relations
Heterogeneity	• Element entropy • Relation entropy
Modularity	• Element modularity
Technical debt	• Cost of rework

Construct	**Metrics**
Environmental complexity	• Herfindahl-Hirschman index • Size diversity • Heterogeneity of output • Specialization rate • Labor diversity • Asset size • Capital intensity • Technical level of workforce
Mission	• Company vision • Architecture vision
Document quality	• Available levels of detail • Available notational elements
Communication	• Documentation availability • Documentation detail suitability • Documentation notation suitability • Documentation consistency
Affinity	• Technology affinity
Education	• Education area • Education level
Experience	• Experience in organization • Experience with enterprise architecture
Role	• Role

Table 8.2: List of metrics for each construct

8.1.6 Measurement model

6. How can both types of metrics be combined to create a unified model for enterprise architecture complexity measurement?

The final measurement model can be created by combining the conceptual model from Figure 8.3 with the metrics from Table 8.2. Combined, these provide a method for measuring both objective and subjective complexity.

Of course, it should be noted that simply combining the metrics for each construct does not give a final score for complexity; constructs almost certainly have different levels of influence on complexity. Therefore, the relative weight of several constructs have been determined during a case study. Three participants were sent a survey to acquire data concerning several constructs. All participants of this case study were, at the time of the case study, working for the same organization, in the same domain. Consequently, all participants engage with the same architecture. From this, it can be concluded that the objective complexity is equal for all participants, and can therefore be ignored as a variable. Using the survey data, values could be assigned to all constructs influencing subjective complexity (excluding objective complexity, which was ignored). Since the participants were also asked to rate their subjective complexity itself, the respective weights of the constructs could be calculated. The details of this process and calculation of the weights can be found in section 6.2.3 “Construct weights”. The resulting weights can be found in Table 8.3.

Construct	**Weight**
Role	N/A
Education	1
Affinity	2
Experience	4
Stakeholder qualities	5.6
Understandability	4.4
Vision	1

Table 8.3: Relative weight of constructs influencing subjective complexity

Combining the answers to the sub-questions as given in this chapter answers the main research question. The designed measurement model incorporates the constructs influencing objective and subjective complexity, and provides the metrics to measure these.

8.2 Contributions

8.2.1 Contributions to theory

This research provides a contribution to the existing theory on enterprise architecture complexity measurement in several ways. Firstly, the research summarizes the existing literature on objective complexity measurement. 20 papers were found that define complexity in measurable terms. However, these papers produced a wide variety of different metrics used for complexity measurement. By performing a structured literature review an overview was created of the state of the art on complexity measurement. Additionally, these metrics were categorized, providing an insight in the types of metrics used.

The main contribution to theory of this research lies in the focus on subjective complexity. As was indicated by Schütz et al. (2013) and found in the structured literature review of this research, subjective complexity had so far not been studied in the context of enterprise architecture. This research provides a first insight in a previously unexplored area of subjective enterprise architecture complexity, and offers several interesting possibilities and suggestions for subsequent studies.

By performing a series of cross-industry interviews, constructs were identified that contribute to subjective complexity. Using the interview data to discover relations between these constructs allowed for the conceptualization of subjective complexity. The conceptual model designed in this study can serve as a starting point for further research into subjective complexity measurement. Additionally, a measurement model is created by integrating existing complexity metrics and introducing new metrics. The introduction of these new metrics, which have been validated by experts, is another important contribution of this research to theory.

Finally, the integration of these metrics with the conceptual model creates a measurement model that can be used to measure both objective and subjective complexity. Using a case study, relative weights were determined for several constructs, providing a validated method of measuring subjective complexity.

8.2.2 Contributions to practice

This research provides a contribution to the enterprise architecture practice in two ways. First, the conceptualization of subjective complexity gives architects an insight into the factors influencing people's perception of complexity. So far, focus has been on the objective aspects of complexity, and complexity management mostly revolved around reducing the number of, and coupling between, elements. However, the interviews performed in this study showed that subjective complexity can play an important role in the management of an enterprise architecture. The complexity perception of different stakeholders can have a huge impact on the performance of the architecture. If managers or developers find the architecture too complex, this can limit the proper functioning of the organization. The conceptual model can be a starting point of factors to consider when making complexity management efforts.

Secondly, the formulation of a measurement model allows enterprise architecture practitioners to quantify the complexity of their architecture. By measuring the complexity for different stakeholders, an architect can determine where to focus his efforts. If the perceived complexity of the architecture throughout the organization is higher than the objective complexity, this means the architecture is most likely not functioning properly. By analyzing the metrics introduced by this research, an architect can correctly determine where to focus his efforts in order to reduce this perceived complexity. Finally, by establishing the relative weights of constructs, the measurement model can be used to determine the value of constructs. If the values of all other constructs are known, as well as their relative weights, the value of a remaining construct can be deduced.

9. Bibliography

Abran, A. (2010). The design of software measurement methods. In *Software Metrics and Software Metrology* (pp. 99–128). IEEE Computer Society.

Adams, J., Khan, H. T. A., Raeside, R., & White, D. (2007). Literature review and critical reading. In *Research Methods for Graduate Business and Social Science Students* (pp. 48–80). Thousand Oaks, CA: Response Books.

Babbie, E. (2010). Qualitative data analysis. In *The practice of social research* (pp. 393–420). Belmont, CA: Wadsworth.

Baldwin, C., Maccormack, A., & Rusnak, J. (2014). Hidden structure: Using network methods to map system architecture. *Research Policy*, *43*(8), 1381–1397.

Bar-Yam, Y. (2003). The dynamics of complex systems examples, questions, methods and concepts. In *Dynamics of Complex Systems* (pp. 1–15). CRC Press.

Beese, J., Aier, S., Haki, M. K., & Aleatrati Khosroshahi, P. (2016). Drivers and effects of information system architecture complexity: A mixed-methods study. In *24th European Conference on Information Systems (ECIS)*, Istanbul.

Beese, J., Kazem Haki, M., Aier, S., & Winter, R. (2017). Revisiting the impact of information systems architecture complexity: A complex adaptive systems perspective. In *38th International Conference on Information Systems*, Seoul.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly*, *11*(3), 368–386.

Bhattacherjee, A. P. (2012). *Social science research: Principles, methods, and practices*. Tampa, FL: Creative Commons Attribution License.

Bieri, J. (1955). Cognitive complexity-simplicity and predictive behavior. *The Journal of Abnormal and Social Psychology*, *51*(2), 263–268.

Boisot, M. & Mckelvey, B. (2011). Complexity and organization-environment relations: Revisiting Ashby's Law of Requisite Variety. In P. Allen, S. Maguire, & B. McKelvey (Eds.), *Complexity and Management* (pp. 279–298). SAGE.

Booch, G. (2008). Measuring architectural complexity. *IEEE Software*, *25*(4), 14–15.

Bradley, R. & Byrd, T. (2006). Information technology architecture as a competitive advantage-yielding resource: A theoretical perspective advantage-yielding resource : a theoretical perspective. In *AMCIS*, (pp. 2415–2425).

Burnard, P. (1991). A method of analysing interview transcripts in qualitative research. *Nurse Education Today*, *11*(6), 461–466.

Campbell, L. J., Halpin, T. A., & Proper, H. A. (1996). Conceptual schemas with abstractions - Making flat conceptual schemas more comprehensible. *Data and Knowledge Engineering*, *20*(1), 39–85.

Cannon, A. R. & John, C. H. S. (2007). Measuring environmental complexity: A theoretical and empirical assessment. *Organizational Research Methods*, *10*(2), 296–321.

Cant, S. N., Jeffery, D., & Henderson-Sellers, B. (1995). A conceptual model of cognitive complexity of elements of the programming process. *Information and Software Technology*, *37*(7), 351–362.

Cardona, M., Kretschmer, T., & Strobel, T. (2013). ICT and productivity: Conclusions from the empirical literature. *Information Economics and Policy*, *25*(3), 109–125.

Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The Journal of the Learning Sciences*, *6*(3), 271–315.

Cilliers, P. (1998). Approaching complexity. In *Complexity and postmodernism: Understanding complex systems* (pp. 1–24). London.

Collins, G. S., Ogundimu, E. O., & Altman, D. G. (2016). Sample size considerations for the external validation of a multivariable prognostic model: A resampling study. *Statistics in Medicine*, *24*, 214–226.

Collinson, S. & Jay, M. (2012). *From Complexity to Simplicity*. London: Palgrave Macmillan.

Daniels, C. B. & LaMarsh II, W. J. (2007). Complexity as a cause of failure in information technology project management. In *IEEE International Conference on System of Systems Engineering*, (pp. 1–7)., San Antonio, TX.

Davis, J. S. & LeBlanc, R. J. (1988). A study of the applicability of complexity measures. *IEEE Transaction on Software Engineering*, *14*(9), 1366–1372.

Denscombe, M. (2010). *The good research guide for small-scale social research projects*. London: Open University Press.

Dicicco-Bloom, B. & Crabtree, B. F. (2006). The qualitative research interview. *Medical Education*, *40*(4), 314–321.

Domerçant, J. C. & Mavris, D. N. (2011). Measuring the architectural complexity of military systems-of-systems.

Doody, O. & Noonan, M. (2013). Preparing and conducting interviews to collect data. *Nurse Researcher*, *20*(5), 28–32.

Dresch, A., Lacerda, D. P., & Miguel, P. A. C. (2015). A distinctive analysis of case study, action research and design science research. *Review of Business Management*, *17*(56), 1116–1133.

Edison, S. W. & Geissler, G. L. (2003). Measuring attitudes towards general technology: Antecedents, hypotheses and scale development. *Journal of Targeting, Measurement and Analysis for Marketing*, *12*(2), 137–156.

Edwards, J. R. & Bagozzi, R. P. (2000). On the nature and direction of relationships between constructs and measures. *Psychological Methods*, *5*(2), 155–174.

Efatmaneshnik, M. & Ryan, M. J. (2016). A general framework for measuring system complexity. *Complexity*, *21*(1), 533–546.

Flood, R. L. & Carson, E. R. (1993). Dealing with complexity: An introduction to the theory and application of systems science. In *Dealing with Complexity: An Introduction to the Theory and Application of System Science* chapter Systems an, (pp. 23–38). New York: Springer.

Freeman, C. & Louçã, F. (2001). *As time goes by: From the industrial revolutions to the information revolution*. New York: Oxford University Press.

Freeman, R. E. (1984). *Strategic management: A stakeholder approach*. Boston: Pitman.

Freeze, R. & Raschke, R. L. (2007). An assessment of formative and reflective constructs in IS research. In *European Conference on Information System (ECIS)*, (pp. 1481–1492).

Gao, L., Warnier, M., Splunter, S. V., Chen, L., & Brazier, F. M. T. (2015). Architectural complexity analysis for large-scale emergency rescue management systems: A preliminary study. In *International Conference on Complex Systems Engineering*, (pp. 1–6)., Storrs, CT.

Gartner (2009). Gartner identifies ten enterprise architecture pitfalls. https://www.gartner.com/newsroom/id/1159617. Last accessed on 2018-02-15.

Gharajedaghi, J. (2011). Operational thinking. In *Systems Thinking: Managing Chaos and Complexity, A Platform for Designing Business Architecture* (pp. 109–131). Elsevier Science.

Glaser, B. G. & Strauss, A. L. (1967). *The discovery of Grounded Theory: Strategies for qualitative research*. Piscataway, NJ, USA: Transaction Publishers.

González-rojas, O., López, A., & Correal, D. (2017). Multilevel complexity measurement in enterprise architecture models. *International Journal of Computer Integrated Manufacturing*, *30*(12), 1280–1300.

Heydari, B. & Dalili, K. (2012). Optimal system's complexity, an architecture perspective. *Procedia Computer Science*, *12*, 63–68.

Hox, J. & Bechger, T. (1998). An introduction to structural equation modeling. *Family Science Review*, *11*, 354–373.

ISO/IEC/IEEE (2011). ISO/IEC/IEEE 42010: Systems and software engineering - Architecture description. Technical report, International Organization for Standardization.

Jochemsen, E., Mehrizi, M. R., van den Hooff, B., & Plomp, M. G. A. (2016). Understandig IS complexity and the heterogeneity of frame: The illusion of agreement. In *European Conference on Information Sys tems 2016 Proceedings*.

Joint Committee for Guides in Metrology (2008). International vocabulary of metrology Basic and general concepts and associated terms (VIM). Technical report, BIPM/IEC/IFCC/ILAC/ISO/IUPAC/IUPAP/OIML.

Kandjani, H., Bernus, P., & Nielsen, S. (2013). Enterprise architecture cybernetics and the edge of chaos: Sustaining enterprises as complex systems in complex business environments. In *46th Hawaii International Conference on System Sciences*, (pp. 3858–3867).

Kandjani, H., Bernus, P., & Wen, L. (2014). Enterprises as complex systems. In P. Saha, L. Johnston, K. Wolfe, & J. Mull (Eds.), *A Systemic Perspective to Managing Complexity with Enterprise Architecture* (pp. 72–98). Hershey PA: IGI Global.

Kazman, R. & Burth, M. (1998). Assessing architectural complexity. In *Proceeding of the 2nd Euromicro Conference on Software Maintainance and Reengineering*, (pp. 104–112)., Florence.

Kinnunen, M. J. (2006). *Complexity measures for system architecture models*. PhD thesis, Massachusetts Institute of Technology.

Kinsner, W. (2008). Complexity and its measures in cognitive and other complex systems. In *7th IEEE International Conference on Cognitive Informatics*, (pp. 13–29)., Stanford.

Kitchenham, B. & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, Evidence-Based Software Engineering Project, Durham.

Ladyman, J., Lambert, J., & Wiesner, K. (2013). What is a complex system? *Europen Journal for Philosophy of Science*, *3*(1), 33–67.

Lagerstrom, R., Baldwin, C., MacCormack, A., & Aier, S. (2014). Visualizing and measuring enterprise application architecture: An exploratory telecom case. In *2014 47th Hawaii International Conference on System Sciences*, (pp. 3847–3856). IEEE.

Lakhrouit, J. & Baina, K. (2015). Evaluating complexity of enterprise architecture components landscapes. In *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, (pp. 1–5). IEEE.

Lakhrouit, J. & Baïna, K. (2016). A pattern based methodology for analyzing enterprise architecture landscape. *International Journal of Computer Science Issues*, *13*(6), 15–24.

Landthaer, J., Kleehaus, M., & Matthes, F. (2016). Organizational modeling. In Pergl, R., Molhanec, M., Babkin, E., & Fosso Wamba, S. (Eds.), *Enterprise and Organizational Modeling and Simulation*, (pp. 52–66)., Ljubljana. Springer.

Lange, M. & Mendling, J. (2011). An experts' perspective on enterprise architecture goals, framework adoption and benefit assessment. In *15th IEEE International Enterprise Distributed Object Computer Conference*, (pp. 304–313)., Helsinki.

Lankford, J. (2003). Measuring system and software architecture complexity. In *IEEE Aerospace Conference Proceedings*, volume 8, (pp. 3849–3857).

Lankhorst, M. (2017). *Enterprise architecture at work*. Berlin: Springer.

Lantow, B. (2014). Level of detail and understandability of enterprise models - Better understandability through higher complexity? In *Perspectives in Business Information Research*, (pp. 45–56).

Lee, H., Ramanathan, J., Hossain, Z., Kumar, P., Weirwille, B., & Ramnath, R. (2014). Enterprise architecture content model applied to complexity management while delivering IT services. In *2014 IEEE International Conference on Services Computing*, (pp. 408–415)., Anchorage, AL.

Liang, T.-P., You, J.-J., & Liu, C.-C. (2010). A resource-based perspective on information technology and firm performance: A meta analysis. *Industrial Management & Data Systems*, *110*(8), 1138–1158.

Liu, J., Yu, Y., Zhang, L., & Nie, C. (2011). An overview of conceptual model for simulation and its validation. *Procedia Engineering*, *24*, 152–158.

Lucke, C., Krell, S., & Lechner, U. (2010). Critical issues in enterprise architecting: A literature review. In *Americas Conference on Information Systems*, Lima.

Mccabe, T. J. & Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the Association for Computer Machinery*, *32*(12), 1415–1425.

Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *The Academy of Management Review1*, *22*(4), 853–886.

Mithas, S., Ramasubbu, N., & Sambamurthy, V. (2011). How information management capability influences firm performance. *MIS Quarterly*, *35*(1), 237–256.

Mocker, M. (2009). What is complex about 273 applications? Untangling application architecture complexity in a case of european investment banking. In *2009 42nd Hawaii International Conference on System Sciences*, (pp. 1–14). IEEE.

Moody, D. L. (2009). The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, *35*(6), 756–779.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*(23), 8577–8582.

Niemi, E. (2007). Enterprise architecture stakeholders - A holistic view. In *Americas Conference on Information Systems*, Keystone, CO.

Nissen, V., von Rennenkampff, A., & Termer, F. (2011). IS architecture characteristics as a measure of IT agility. In *17th Americas Conference on Information Systems 2011, AMCIS 2011*, volume 1, (pp. 770).

Nord, R. L., Ozkaya, I., Kruchten, P., & Gonzalez-rojas, M. (2012). In search of a metric for managing architectural technical debt. In *6th European Conference on Software Architecture*, (pp. 91–100).

Norman, D. O. & Kuras, M. L. (2006). Engineering complex systems. In D. Braha, A. A. Minai, & Y. Bar-Yam (Eds.), *Complex Engineered Systems* (pp. 206–245). New York: Springer.

Osvalds, G. (2001). Definition of enterprise architecture-centric models for the systems engineer. In *International Council on Systems Engineering*, number 11, (pp. 93–99)., Melbourne.

Ottino, J. M. (2003). Complex systems. *AIChE Journal*, *49*(2), 292–299.

Oxford Dictionary (2018a). Attribute. https://en.oxforddictionaries.com/definition/attribute. Last accessed on 2018-01-02.

Oxford Dictionary (2018b). Subjective. https://en.oxforddictionaries.com/definition/subjective. Last accessed on 2018-01-02.

Pereira, C. M. & Sousa, P. (2004). A method to define an enterprise architecture using the Zachman framework. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, (pp. 1366–1371)., Nicosia.

Perry, D. E. & Wolf, A. L. (1992). Foundations for the study of software architecture. *Software Engineering Notes*, *17*(4), 40–52.

Puspitasari, I. (2016). Stakeholder's expected value of enterprise architecture. In *14th IEEE International Conference on Software Engineering Research*, (pp. 243–248)., Towson, MD.

Richard Beetz, K. & Kolbe, L. M. (2011). Towards managing IT complexity: An IT governance framework to measure business-IT responsibility sharing and structural IT organization. volume 1, (pp. 832–843).

Rood, M. A. (1994). Enterprise architecture: Definition, content, and utility. In *Proceedings of 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, (pp. 106–111)., Morgantown, WV.

Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation*, *7*(1), 12–24.

Schmidt, C. (2015). Business architecture quantified: How to measures business complexity. In D. Simon & C. Schmidt (Eds.), *Business Architecture Management* (pp. 243–268). Springer.

Schmidt, C. & Buxmann, P. (2011). Outcomes and success factors of enterprise it architecture management: Empirical insight from the international financial services industry. *European Journal of Information Systems*, *20*(2), 168–185.

Schmidt, C., Widjaja, T., & Schütz, A. (2013). Messung der komplexität von IT-Landschaften auf der basis von Architektur-Metamodellen: Ein generischer ansatz und dessen anwendung im rahmen der Architektur-Transformation. In *GI Jahrestagung*, (pp. 1261–1275)., Koblenz.

Schneider, A., Zec, M., & Matthes, F. (2014). Adopting notions of complexity for enterprise architecture management. In *20th Americas Conference on Information Systems*, (pp. 1–10)., Savannah.

Schneider, A. W., Reschenhofer, T., Schutz, A., & Matthes, F. (2015). Empirical results for application landscape complexity. volume 2015-March, (pp. 4079–4088).

Schütz, A., Widjaja, T., & Kaiser, J. (2013). Complexity in enterprise architectures - Conceptualization and introduction of a measure from a system theoretic perspective. In *Proceedings of the 21st European Conference on Information Systems*, Utrecht, Netherlands.

Shaw, M., DeLine, R., Klein, D. V., Ross, T. L., Young, D. M., & Zelesnik, G. (1995). Abstractions for software architecture and tools to support them. *IEEE Transaction on Software Engineering*, *21*(4), 315–335.

Singh, D. & Misra, A. (2006). Cognitive software development process and associated metrics - A framework. In *5th IEEE International Conference on Cognitive Informatics*, (pp. 255–260)., Beijing.

Steen, M. W. A., Akehurst, D. H., ter Doest, H. W. L., & Lankhorst, M. M. (2004). Supporting viewpoint-oriented enterprise architecture. In *8th IEEE Enterprise Distributed Object Computing Conference*, (pp. 201–211).

Stroud, B. & Ertas, A. (2016). Enterprise cyclomatic complexity.

Suedfeld, P. & Coren, S. (1992). Cognitive correlates of conceptual complexity. *Personality and Individual Differences*, *13*(11), 1193–1199.

The Open Group (2011). *TOGAF Version 9.1*. U.S.: The Open Group.

The Open Group (2017). Language structure. http://pubs.opengroup.org/architecture/archimate3-doc/chap03.html{#}{_}Toc489945966. Last accessed on 2017-10-11.

Thomas, O. (2006). Understanding the term reference model in information systems research: History, literature analysis and explanation. In Bussler, C. & Haller, A. (Eds.), *Business Process Management Workshops*, (pp. 484–496). Springer.

van der Raadt, B., Schouten, S., & van Hans, V. (2008). Stakeholder perception of enterprise architecture. In *European Conference on Software Architecture*, (pp. 19–34)., Paphos.

van 't Wout, J., Waage, M., Hartman, H., Stahlecker, M., & Hofman, A. (2010). *The integrated architecture framework explained*. Berlin: Springer.

Vasconcelos, A., Sousa, P., & Tribolet, J. (2007). Information system architecture metrics: An enterprise engineering evaluation approach. *The Electronic Journal Information Systems Evaluation*, *10*(1), 91–122.

Von Bertalanffy, L. (1968). *General System Theory: Foundations, development, applications*. New York, NY: George Braziller.

Von Bertalanffy, L. (1972). The history and status of General Systems Theory. *Academy of Management Journal*, *15*(4), 407–426.

Von Bertalanffy, L. (1975). *Perspectives on General Systems Theory: scientific-philosophical studies*. New York: George Braziller.

Wagter, R., van den Berg, M., Luijpers, J., & van Steenbergen, M. (2005). *Dynamic enterprise architecture: How to make it work*. Hoboken, NJ: John Wiley & Sons.

Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems*, *15*, 285–304.

Wang, Y. (2006). Cognitive complexity of software and its measurement. In *5th IEEE International Conference on Cognitive Informatics*, (pp. 226–235)., Beijing.

Wang, Y. (2007). The theoretical framework of cognitive informatics. *International Journal of Cognitive Informatics and Natural Intelligence*, *1*(1), 1–27.

Webster, J. & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii–xxiii.

Wehling, K., Wille, D., Pluchator, M., & Schaefer, I. (2016). Towards reducing the complexity of enterprise architectures by identifying standard variants using variability mining. In *Automobil Symposium*.

Wehling, K., Wille, D., Seidl, C., & Schaefer, I. (2017). Decision support for reducing unnecessary IT complexity of application architectures. In *2017 IEEE International Conference on Software Architecture Workshops*, (pp. 161–168).

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, *14*(9), 1357–1365.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Wijendra, D. R. & Hewagamage, K. P. (2016). Automated tool for the calculation of cognitive complexity of a software. In *2nd International Conference on Science in Information Technology*, (pp. 163–168).

Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). Los Angeles: Sage.

Appendix A. Structured Literature Review Protocol

Conducting a structured literature review should be a structured and rigorous process, and result in an exhaustive overview of the available literature. To that end this review protocol, based on the works of Kitchenham & Charters (2007) and Webster & Watson (2002), was drafted. The protocol was used to guide the search and selection of, and data extraction from, relevant studies.

A.1 Review overview

According to Kitchenham & Charters (2007), a structured literature review involves several successive activities. First, the review is carefully planned. After formulating a research question to be answered by the review, a search strategy and selection criteria should be chosen, focused on answering the research question formulated. Next, the search can be conducted, which consists of several steps further detailed in section A.3 "Search strategy". In order to assure a quality review, the studies found in the search process should be assessed on their quality. Finally, the data can be analyzed and the results synthesized, for which the concept matrix of Webster & Watson (2002) was used. The structured review process executed in this thesis is presented in Figure A.1.

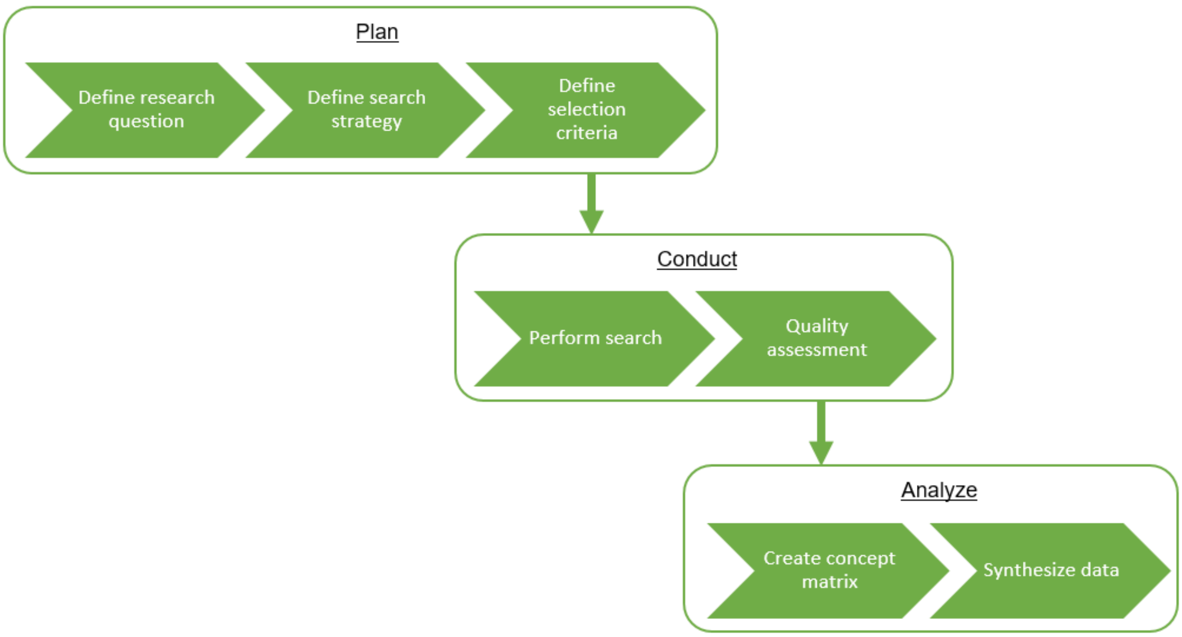


Figure A.1: Structured literature review process

A.2 Research question

For a structured literature review, a clear research question should be defined. The aim of this review is to answer the following research question, as defined in section 1.4.1 “Research questions”:

1. Which existing metrics are being used to measure complexity in an enterprise architecture, and how can these be classified using the complexity dimensions by Schneider et al. (2014)?

To answer this research question, an overview will be created of the metrics used for enterprise architecture complexity measurement in existing literature.

A.3 Search strategy

To ensure a complete overview of the existing literature, a thorough search strategy was carried out, as illustrated by Figure A.2.

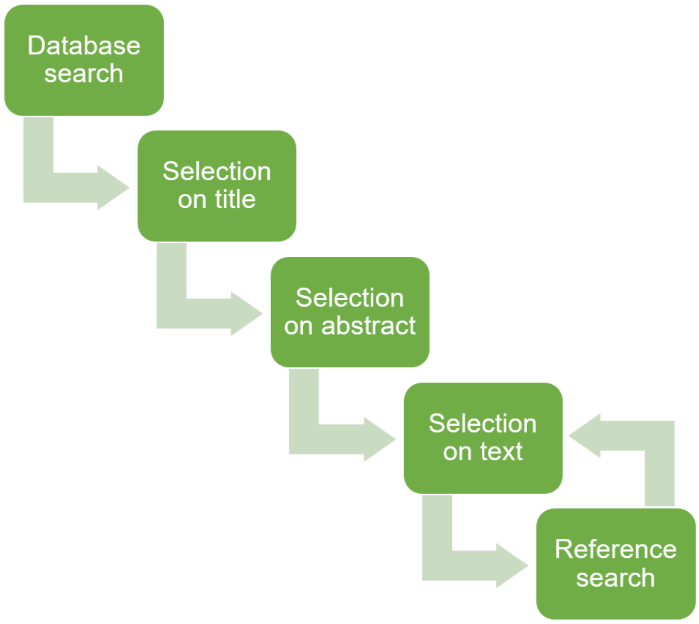


Figure A.2: Search strategy

A.3.1 Database search

The search will be conducted on some of the largest scientific databases of both generic and technical literature. The following databases will be searched:

1. Scopus
2. Web of Science
3. JSTOR
4. IEEE Xplore
5. ACM Digital library
6. AIS Electronic library

Essential to finding an exhaustive list of the current literature is the use of correct search terms. Multiple test searches and repeated refinement lead to the identification of the search terms below. A short justification for each term is included.

(A) (”enterprise architecture” AND complexity)

A literal search for complexity in the context of enterprise architecture is done. The keyword ’complex’ is not included, since this is used in a lot of situations that are outside the scope of this review.

(B) ”architectur* complexity”

Since the complexity of an EA is partly dependent on the complexity of its layers, a general search for architectural complexity is included as well.

(C) measur*

(D) metric*

The review is focused on the quantification of complexity, which is reflected by these keywords.

These terms are combined using Boolean expressions:

(A OR B) AND (C OR D)

A.3.2 Selection criteria

The initial set of studies yielded by the database will be refined based on the selection criteria found in Table A.1. To be included, studies will have to meet the inclusion criteria. Studies that meet the exclusion criteria are considered not relevant and will be excluded. In line with the search strategy in Figure A.2, the selection criteria will first be applied to the study titles. Next the selection criteria will be applied to the abstracts of the remaining set. Finally, the full text of the remaining studies will be read, again applying the selection criteria.

Inclusion criteria	**Exclusion criteria**
Focused on the enterprise architecture or one of its layers	Measures software complexity
Discusses complexity in measurable terms	Focused on the architecture of a single application

Table A.1: Study selection criteria

A.3.3 Reference search

After the first database search and the refinement of the selection, a forward and backward reference search will be performed. First, all references in the remaining selection are scanned to search for additional relevant research, meaning those studies that meet the selection criteria. Subsequently, using the databases listed in section A.3.1 “Database search”, all studies that referenced to the study selection were evaluated. This forward and backward reference search is an iterative process, which will be repeated until no new studies are found.

A.3.4 Quality assessment

In order to ensure the quality of the study selection, an assessment will be done based on the checklist provided by Kitchenham & Charters (2007).

A.4 Data extraction

According to Webster & Watson (2002), a literature review should be concept-centric, as opposed to the often-used author-centric approach. This means that in order to properly analyze and synthesize the available literature, it should be grouped on the concepts it describes. As a method of analysis, Webster & Watson (2002) suggests a concept matrix. Using this concept matrix, this review will identify metrics used in the existing literature, and the concepts these adhere to.

Appendix B. Structured Literature Review Results

B.1 Review paper selection

Number	Title	Author(s), year
1	A pattern based methodology for analyzing enterprise architecture landscape	Lakhrouit & Baïna (2016)
2	Architectural complexity analysis for large-scale emergency rescue management systems: A preliminary study	Gao et al. (2015)
3	Assessing architectural complexity	Kazman & Burth (1998)
4	Complexity in enterprise architectures: Conceptualization and introduction of a measure from a system theoretic perspective	Schütz et al. (2013)
5	Complexity Measures for System Architecture Models	Kinnunen (2006)
6	Decision support for reducing unnecessary IT complexity of application architectures	Wehling et al. (2017)
7	Empirical results for application landscape complexity	Schneider, Reschenhofer, Schutz & Matthes (2015)
8	Enterprise architecture content model applied to complexity management while delivering IT services	Lee et al. (2014)
9	Enterprise cyclomatic complexity	Stroud & Ertas (2016)
10	Evaluating complexity of enterprise architecture components landscapes	Lakhrouit & Baina (2015)
11	IS architecture characteristics as a measure of IT agility	Nissen, von Rennenkampff & Termer (2011)
12	Measuring architectural complexity	Booch (2008)
13	Measuring system and software architecture complexity	Lankford (2003)
14	Measuring the architectural complexity of military systems-of-systems	Domerçant & Mavris (2011)
15	Messung der Komplexitt von IT - Landschaften auf der basis von Architektur - Metamodellen : Ein generischer ansatz und dessen anwendung im Rahmen der Architektur - Transformation	Schmidt, Widjaja & Schütz (2013)
16	Multilevel complexity measurement in enterprise architecture models	González-rojas, López & Correal (2017)
17	Towards managing IT complexity: An IT governance framework to measure business-IT responsibility sharing and structural IT organization	Richard Beetz & Kolbe (2011)
18	Towards reducing the complexity of enterprise architectures by identifying standard variants using variability mining	Wehling, Wille, Pluchator & Schaefer (2016)
19	Visualizing and measuring enterprise application architecture: An exploratory telecom case	Lagerstrom, Baldwin, MacCormack & Aier (2014)
20	What is complex about 273 applications? Untangling application architecture complexity in a case of European investment banking	Mocker (2009)

Table B.1: Identified papers

B.2 Metric matrix

Metric Paper #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Count	Prevalence
# relations	x			x	x	x	x			x		x				x	x	x		x	11	55%
# elements	x			x		x	x			x		x					x	x			8	40%
# cardinal elements		x		x	x									x	x						5	25%
# cardinal relations		x		x										x	x						4	20%
Cyclomatic complexity		x							x				x	x							4	20%
Element entropy				x		x									x			x			4	20%
Relation entropy				x		x									x			x			4	20%
Conformity							x				x						x			x	4	20%
Interface Complexity Multiplier		x			x									x							3	15%
Redundancy							x										x			x	3	15%
# OS & middleware											x						x			x	3	15%
Functions/system		x												x							2	10%
# patterns			x									x									2	10%
Application age																	x			x	2	10%
# hardware platforms											x						x				2	10%
Betweenness centrality	x																				1	5%
Quantified expert opinion		x																			1	5%
Pattern coverage			x																		1	5%
Elements/type					x																1	5%
Relations/element					x																1	5%
Processes/element					x																1	5%

Metric Paper #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Count	Prevalence
Elements/process					x																1	5%
Service-time Actual								x													1	5%
Domains/application											x										1	5%
Software categories/app											x										1	5%
SLOC												x									1	5%
Halstead difficulty													x								1	5%
# functions																x					1	5%
Apps/user																	x				1	5%
Customization																	x				1	5%
# instances																	x				1	5%
# software platforms																	x				1	5%
Application type																	x				1	5%
# software frameworks																	x				1	5%
# new applications																	x				1	5%
# retired applications																	x				1	5%
# physical servers																	x				1	5%
# virtual servers																	x				1	5%
Visibility Fan-In																			x		1	5%
Visibility Fan-out																			x		1	5%
Requirements/app																				x	1	5%
Propagation cost																			x		1	5%
Count	3	5	2	6	7	4	4	1	1	2	5	4	2	5	4	2	17	4	3	6		

Table B.2: Identified metrics and their prevalence

B.3 Metric description

Metric	Explanation
# relations	Number of relations
# elements	Number of elements
# cardinal elements	Cardinal set of element types
# cardinal relations	Cardinal set of relation types
Cyclomatic complexity	Cyclomatic complexity by McCabe, adjusted for use on architectures
Element entropy	Entropy measure of elements, based on Entropy Measure by Shannon
Relation entropy	Entropy measure of relations, based on Entropy Measure by Shannon
Conformity	Conformity to enterprise standards
Interface Complexity Multiplier	Multiplier based on several interface characteristics, such as distance, volume of interchange, quality and reliability requirements, connection reversibility, tolerances, physical conditions, level of reuse, concurrency, standardization
Redundancy	Functional redundancy in systems
# OS & middleware	Number of Operating Systems and middleware in use
Functions/system	Number of functions performed per system
# patterns	Number of design patterns found in graph representation
Application age	Application age
# hardware platforms	Number of hardware platform types
Betweenness centrality	Number of shortest paths that pass through a given element
Quantified expert opinion	Weight factor based on expert opinion
Pattern coverage	Percentage of graph covered by patterns
Elements/type	Number of elements per type
Relations/element	Number of relations per element
Processes/element	Number of processes per element
Elements/process	Number of elements per process
Service-time Actual	Cumulative service-time of a graph element and all its child elements
Domains/application	Number of domains per application
Software categories/app	Number of software categories per application
SLOC	Source lines of code
Halstead difficulty	Halstead difficulty, adjusted for use on architectures
# functions	Number of functions
Apps/user	Applications per user
Customization	Total person-years used for customization of applications
# instances	Number of instances of major applications
# software platforms	Number of software development platforms in use
Application type	Percentage of service-, object- and non-object-oriented applications
# software frameworks	Number of software frameworks in use
# new applications	Number of new applications per year
# retired applications	Number of retired applications per year
# physical servers	Number of physical servers
# virtual servers	Number of virtual servers
Visibility Fan-In	Number of elements that directly or indirectly depend on a given element
Visibility Fan-out	Number of element that a given element directly or indirectly depends on
Requirements/app	Number of business requirements per application
Propagation cost	Costs incurred when a randomly chosen application is changed

Table B.3: Description of the identified metrics

B.4 Metric-concept overview

Metric	Concept	Concept-group
# relations	Elements and relations	Graph-based
# elements	Elements and relations	Graph-based
# cardinal elements	Elements and relations	Graph-based
# cardinal relations	Elements and relations	Graph-based
Cyclomatic complexity	Elements and relations	Graph-based
Element entropy	Elements and relations	Graph-based
Relation entropy	Elements and relations	Graph-based
Conformity	Conformity	Non-functional
Interface Complexity Multiplier	Elements and relations, quality, reliability, conformity	Graph-based, non-functional
Redundancy	Redundancy	Functional
# OS & middleware	Application-data	Functional
Functions/system	Functions	Functional
# patterns	Patterns	Graph-based
Application age	Application-data	Functional
# hardware platforms	Hardware-data	Functional
Betweenness centrality	Elements and relations	Graph-based
Quantified expert opinion	Expert opinion	Other
Pattern coverage	Patterns	Graph-based
Elements/type	Elements and relations	Graph-based
Relations/element	Elements and relations	Graph-based
Processes/element	Elements and relations, functions	Graph-based, functional
Elements/process	Elements and relations, functions	Graph-based, functional
Service-time Actual	Service time	Non-functional
Domains/application	Application-data	Functional
Software categories/app	Application-data	Functional
SLOC	Application-data	Functional
Halstead difficulty	Application-data	Functional
# functions	Functions	Functional
Apps/user	Application-data	Functional
Customization	Cost	Other
# instances	Application-data	Functional
# software platforms	Application-data	Functional
Application type	Application-data	Functional
# software frameworks	Application-data	Functional
# new applications	Application-data	Functional
# retired applications	Application-data	Functional
# physical servers	Hardware-data	Functional
# virtual servers	Hardware-data	Functional
Visibility Fan-In	Application-data	Functional
Visibility Fan-out	Application-data	Functional
Requirements/app	Functions	Functional
Propagation cost	Functions	Functional

Table B.4: Identified metrics and assigned concepts

B.5 Concept matrix

	Graph-based		**Functional**				**Non-functional**				**Other**		**Count**
	Elements & relations	Patterns	Functions	Redun-dancy	Application data	Hardware data	Quality	Reliability	Conformity	Service time	Expert opinion	Cost	
1	X												1
2	X		X				X	X	X		X		6
3		X											1
4	X												1
5	X		X				X	X	X				5
6	X												1
7	X			X					X				3
8										X			1
9	X												1
10	X												1
11					X	X			X				3
12	X	X			X								3
13	X				X								2
14	X		X				X	X	X				5
15	X												1
16	X		X										2
17	X			X	X	X			X			X	6
18	X												1
19			X		X								2
20	X		X	X	X				X				5
Count	16	2	6	3	6	2	3	3	7	1	1	1	
Prevalence	80%	10%	30%	15%	30%	10%	15%	15%	35%	5%	5%	5%	
Group	17 (85%)		11 (55%)				8 (40%)				2 (10%)		

Table B.5: Identified concepts and their prevalence

B.6 Complexity classification

Metric	Objective/subjective	Structural/dynamic	Quantitative/qualitative	Ordered/disordered
# relations	Objective	Structural	Quantitative	Ordered
# elements	Objective	Structural	Quantitative	Ordered
# cardinal elements	Objective	Structural	Quantitative	Ordered
# cardinal relations	Objective	Structural	Quantitative	Ordered
Cyclomatic complexity	Objective	Structural	Quantitative	Ordered
Element entropy	Objective	Structural	Quantitative	Disordered
Relation entropy	Objective	Structural	Quantitative	Disordered
Conformity	Objective	Structural	Quantitative, qualitative	Disordered
Interface Complexity Multiplier	Objective	Structural, dynamic	Quantitative, qualitative	Ordered, disordered
Redundancy	Objective	Structural	Quantitative, qualitative	Disordered
# OS & middleware	Objective	Structural	Quantitative	Ordered
Functions/system	Objective	Structural	Quantitative	Ordered
# patterns	Objective	Structural	Quantitative	Ordered
Application age	Objective	Structural	Quantitative	Ordered
# hardware platforms	Objective	Structural	Quantitative	Ordered
Betweenness centrality	Objective	Structural	Quantitative	Disordered
Quantified expert opinion	Subjective	Structural	Quantitative	Ordered
Pattern coverage	Objective	Structural	Quantitative	Ordered
Elements/type	Objective	Structural	Quantitative	Ordered
Relations/element	Objective	Structural	Quantitative	Ordered
Processes/element	Objective	Structural	Quantitative	Ordered
Elements/process	Objective	Structural	Quantitative	Ordered
Service-time Actual	Objective	Structural	Quantitative	Ordered
Domains/application	Objective	Structural	Quantitative	Ordered
Software categories/app	Objective	Structural	Quantitative	Ordered
SLOC	Objective	Structural	Quantitative	Ordered
Halstead difficulty	Objective	Structural	Quantitative	Ordered
# functions	Objective	Structural	Quantitative	Ordered
Apps/user	Objective	Structural	Quantitative	Ordered
Customization	Objective	Structural	Quantitative	Ordered
# instances	Objective	Structural	Quantitative	Ordered
# software platforms	Objective	Structural	Quantitative	Ordered
Application type	Objective	Structural	Qualitative	Ordered
# software frameworks	Objective	Structural	Quantitative	Ordered
# new applications	Objective	Structural	Quantitative	Ordered
# retired applications	Objective	Structural	Quantitative	Ordered
# physical servers	Objective	Structural	Quantitative	Ordered
# virtual servers	Objective	Structural	Quantitative	Ordered
Visibility Fan-In	Objective	Structural	Quantitative	Ordered
Visibility Fan-out	Objective	Structural	Quantitative	Ordered
Requirements/app	Objective	Structural	Quantitative	Ordered
Propagation cost	Objective	Structural	Quantitative	Disordered

Table B.6: Identified metrics and their classification

Appendix C. Interview Guide

C.1 Introduction

This interview study is done as a part of a Master's thesis on architectural complexity. The goal of this study is to measure both the objective and subjective complexity of an enterprise architecture. Whereas objective complexity metrics in the context of EA already exist in literature, subjective complexity has been an unexplored area so far. In order to manage complexity, it is essential to quantify it first, which requires finding the attributes leading to subjective complexity. After finding these attributes, we can draft metrics to measure them. The aim of these interviews is to find the attributes leading to subjective EA complexity.

This protocol serves as a guideline for the interviews, and structures its process. These interviews are semi-structured. Therefore, this protocol defines an introduction and ending to the interviews, and describes a general flow of topics. Furthermore, it defines the key questions that need to be asked in order to answer the research question of these interviews. Additionally, suggestions for additional in-depth questions are listed, as well as probes aimed to further engage the participant.

C.2 Interview blueprint

C.2.1 Main research question

The main research question that needs to be answered during the interviews is:

Which factors influence a stakeholders' perception of enterprise architecture complexity?

C.2.2 Introduction

The introduction of an interview will consist of the following aspects:

- Introduce the interviewer and the details of this research.
- State the goal of this research.
- Inform about the practical aspects of the interview:
    - The interview will take approximately one hour.
    - Ask for permission to record the interview, and ensure it is anonymous.
    - Transcripts are available on request.
- Ask some practical questions, to be found in section C.3.1 "Practical questions". This will be done in the form of small-talk, to establish report with the participant.
- Agree on the scope of terminology for the interview, to be found in section C.3.2 "Scoping questions".

C.2.3 Flow of topics

After the introduction, the content-focused part of the interview starts. This is done on the basis of the key questions to be found in section C.3.3 “Key questions”. The general flow of the topics is to first discuss the architecture of the IT landscape in the participants’ organization, and their understanding of it. Secondly, the interview will move on to its complexity, which is a naturally following topic if they feel their understanding is lacking. After asking about their opinion on complexity and the factors influencing it, we will focus on their role in the organization, and whether this affects their position towards IT architecture complexity.

C.2.4 Ending

After discussing all key questions and relevant other questions, the interview will be ended. Participants will be asked if they have anything important to add that was not yet discussed during the interview, and that could be valuable to the research. Finally, participants will be thanked for their participation and valuable input for this research, and asked if they are interested in its results, which will be shared if they are interested.

C.3 Interview questions

C.3.1 Practical questions

Some practical questions will be asked first, in order to get some more information about the participant and to properly classify them as a stakeholder.

1. What is your current role in your organization?
2. How long have you been active in your current role?
3. Can you briefly elaborate your daily tasks?
4. What role does Enterprise Architecture play in your job?

C.3.2 Scoping questions

Setting definitions and a scope for the interview make sure the interviewer and interviewee understand each other. Therefore, the scope is first set on the three important definitions of the interview *organization*, *(enterprise) architecture* and *complexity*.

1. In what organizational context do you usually work? With what part of the organization do you interact at least on a weekly basis?
2. What aspects do you consider to be part of an enterprise architecture? Do you work with actual elements of the architecture, or do you usually consider a model of the architecture?
3. What is your idea of complexity? When talking about the complexity of an architecture, what aspects of the architecture do you consider?

C.3.3 Key questions

In order to answer the main research questions, six key questions have been identified based on existing literature. These questions will be asked during the interview in order to answer the main research questions. With each key question, additional in-depth questions (listed) and probes (bullets) have been listed in order to acquire more information or further engage the participant. If deemed necessary or helpful, additional questions can be asked during the interview. For each question, a substantiation is provided in italics.

According to Cant et al. (1995), complexity consists, partly, of psychological complexity. Psychological, or cognitive, complexity is based on the ease of understanding or the property of comprehension (Singh & Misra, 2006). Therefore, it is important to first ask the interviewee about his understanding of the architecture.

1. Do you feel like you understand the architecture of your organizations IT landscape?
   - If not: why? What factors influence this? E.g. is it not explained well or just too complex?

   (a) Tell me about the architecture of your organizations IT landscape.

   (b) How can the architecture be altered to improve understandability?

   (c) How do you think the understanding of the architecture, or lack thereof, influences your organization?
      - Do you know of any problems arising from a lack of understanding?

   (d) What would help you to better understand the architecture?

According to Abran (2010), it is important to identify whether we measure the measurand itself, or a model of it. In the case of EA, this can vary based on the role of the stakeholder. Whereas users probably mostly interact with the actual architecture, facilitators are likely to interact more with a model of the architecture. Therefore, when questioning the understandability of the IT landscape, it is important for question its documentation as well. An important aspect in notational clarity is correct representation mapping, meaning all notational elements correspond to a single architectural element, and vice versa (Moody, 2009). This aspect of subjective complexity is called representational complexity (Cant et al., 1995).

2. How is the architecture of your organization documented?

   (a) Is het documentation easy to read and understand?

   (b) Does this documentation make the architecture simpler, or more complex?

   (c) Is there a model of the architecture? Is it easy to interprete?

   (d) Are all notational elements clearly corresponding to an architectural element?

After asking about the understandability of the architecture and its model, we move on to the complexity of the IT architecture. According to Figure 1.4, the subjective complexity of an architecture is dependent on the objective complexity and stakeholder perception. By asking their opinion of the complexity, we can assess how the understandability asked previously relates to their opinion of, i.e. the subjective, complexity.

3. What is your opinion on the complexity of the IT architecture in your organization?

   (a) Does the complexity of your IT landscape impact you in your job?

(b) What are some positive/negative influences of the complexity? How much do you know about the architecture of the IT landscape?
    - Please elaborate on your knowledge
(c) Do you feel the architecture is too complex or too simple?
    - Why?
    - How does this influence you in your role in the oragnization?
(d) In you opinion, what would help to reduce the complexity of you IT architecture?
    - What would help to reduce your perception of complexity?
(e) What is the influence of the organization's environment on its complexity?

Previously, we questioned understandability as one of the big influences on stakeholder perception. Next, we try to identify additional variables influencing subjective complexity by directly asking interviewees for their input on this.

4. In your opinion, what variables influence the complexity of an Enterprise Architecture?
    (a) What objective variables influence complexity?
    (b) What other variables influence complexity?
    (c) What aspects of the architecture make you feel like it is complex?
    (d) How do you feel about the architects of the IT landscape?
        - Do they have a positive influence on its complexity?
        - What could they do better?

In order to work with an architecture, information processing is an important aspect influencing stakeholders, which influences understandability as well. According to Wang (2007), information acquisition has four categories: knowledge, behavior, experience and skill. In order to assess the influence of stakeholder characteristics on subjective complexity, we will ask for these variables. Behavior and skill will be associated with different roles.

5. How does your role in the organization influence your view on EA complexity?
    (a) How would you feel about the complexity if you had a different role?
    (b) How does your experience influence your view on complexity?
    (c) How does your knowledge on architectures influence your view on complexity?

The goal of complexity management is to achieve the appropriate level of complexity for the architectures context and purpose. Therefore, subjective complexity is most likely influenced by the personal purpose of the architecture. This question verifies required architectural complexity as a variable.

6. Do you need the current level of architectural complexity to carry out your job? Or should it be more simple or complex?

C.4 Participants

#	Length	Organization	Job description	Stakeholder group
1	1:01:55	A	Enterprise architecture consultant	Producer, facilitator, user
2	0:57:17	B	Senior project manager	Facilitator
3	1:03:09	B	Business architect	Producer
4	0:51:05	A	Enterprise architecture consultant	Producer, facilitator, user
5	1:10:36	B	Delivery manager	User
6	0:55:54	A	Enterprise architecture consultant	Producer, facilitator, user
7	1:54:40	C	Service delivery lead	User
8	0:46:17	C	Opdrachtmanager	Facilitator
9	0:50:36	C	IT architect	Producer
10	0:21:45	D	Tech lead & Architect	Producer
11	0:34:20	D	Head of IT	Facilitator
12	0:38:20	D	Developer	User

Table C.1: Interview participant overview

Appendix D. Detailed Interview Results

Code group	Code	Facilitator	Producer	User	Total
Architecture	Application complexity	2	2	2	**6**
	Business complexity	0	3	1	**4**
	Cohesion and dependence of elements	4	4	5	**13**
	Conformity	2	2	2	**6**
	Coupling	0	3	1	**4**
	Data complexity	0	0	1	**1**
	Functional redundancy	1	2	1	**4**
	Interoperability	0	0	1	**1**
	Modularity	5	4	5	**14**
	New, unknown technologies	0	0	1	**1**
	Number of elements and relations	1	2	3	**6**
	Presence of structure	1	1	1	**3**
	Security	1	1	1	**3**
	Standardization of landscape	1	2	3	**6**
	Technical debt	0	1	0	**1**
	Variation in technology	1	1	2	**4**
Documentation	Abstraction level of documentation	3	4	2	**9**
	Communication to stakeholders	1	3	1	**5**
	Insight in the architecture structure	3	3	2	**8**
	Maintainability of documentation	0	1	0	**1**
	Notation of documentation	1	3	0	**4**
	Presence of documentation	3	4	3	**10**
	Terminologies and language	0	1	0	**1**
Enterprise	Available resources	1	0	1	**2**
	Business culture	0	0	2	**2**
	Change effort	1	1	2	**4**
	Control	2	2	2	**6**

Code group	Code	Facilitator	Producer	User	Total
Enterprise	Governance	1	1	3	**5**
	History of architecture	3	3	4	**10**
	Internal politics	1	2	2	**5**
	Legacy	2	2	3	**7**
	Number of stakeholders imvolved	2	3	4	**9**
	Priorities	0	1	0	**1**
	Quality of the architect	0	1	1	**2**
	Transparancy in decision making	0	1	0	**1**
Environment	Complexity of the environment	2	2	4	**8**
	Complexity of the problem	2	1	1	**4**
	Industry	2	3	2	**7**
	Market developments	1	2	1	**4**
Mission	Architecture performance	1	1	1	**3**
	Goal of the architecture	0	1	0	**1**
	Presence of vision or strategy	3	2	2	**7**
Stakeholders	Biological: left- or right-brained	1	1	1	**3**
	Stakeholder education and background	3	1	1	**5**
	Stakeholder expectations	0	0	1	**1**
	Stakeholder interest and affinity	2	5	3	**10**
	Stakeholder knowledge and experience	6	6	6	**18**
	Stakeholder role	4	5	4	**13**
	Total	**70**	**94**	**89**	**253**

Table D.1: Code prevalence among different stakeholder groups[1]

[1]Note that the numbers in the Total-column may exceed the total number of interviews (12), because some participants were classified in multiple stakeholder groups.

Appendix E. Case Study Survey

Subjective complexity

Thank you for participating in my research into the subjective complexity of IT architecture! In preparation of the interview, I would like to collect some data. I will use this data to make a predication of subjective complexity, based on the model developed in my research. The survey will take approximately 10 minutes, and its results will be processed anonymously.

Background

Firstly, I would like to ask you about some background information about yourself.

1. **Name**
   This will only be used to identify your results for the interview, this will not be included in any way in my research.

   ____

Role

2. **Please provide a short description of your job**
   Please describe your tasks and responsibilities in a few sentences.

   ____
   ____
   ____
   ____
   ____

Education

3. **In what education area did you study?**
   Mark only one oval.

   - ◯ Alpha
   - ◯ Bèta
   - ◯ Gamma

4. **What is you education level?**
   Mark only one oval.

   - ◯ University
   - ◯ Hbo
   - ◯ Mbo
   - ◯ Other: ____________________

Experience

5. **How many years have you worked at Schiphol?**
   Mark only one oval.

   - 0-2
   - 2-5
   - 5-10
   - 10-20
   - 20+

6. **How many years have you actively worked with enterprise architecture?**
   Mark only one oval.

   - 0-2
   - 2-5
   - 5-10
   - 10-20
   - 20+

Technology affinity

The questions below aim to measure your affinity with technology in general. Here, technology should be interpreted as technologies relating to IT (hardware, software, mobile devices, digital services, etc.)

Please indicate to what extent you agree with the statements below on a scale of 1 (strongly disagree) to 5 (strongly agree).

7. **Technology is an important part of my life**
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

8. **I enjoy learning about new technologies**
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

9. **My job requires me to know about different technologies**
   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

10. **I usually have no trouble learning new technologies**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

11. **I relate well to the technology used in my job**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

12. **I am comfortable with new technologies required for my job**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

13. **In my job, I know how to deal with technological malfunctions or problems**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

14. **Solving a technological problem is a fun challenge**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

15. **I find most technology easy to learn**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

16. **I feel as up-to-date on technology as my peers**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

Documentation

The questions below regard the documentation of the enterprise architecture. This documentation consists of all models or other description of the architecture's elements and their structure.

17. **I know where to find relevant documentation on the enterprise architecture**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

18. **To what extent does the documentation of the enterprise architecture have the correct level of details?**
Documentation can have different levels of detail; not all details may be relevant for your personal use. In general, does the documentation available to you have the right level of detail?
Mark only one oval.

	1	2	3	4	5	
Too little/much detail	◯	◯	◯	◯	◯	Correct level of details

19. **The notational elements of the enterprise architecture documentation are understandable and clear**
Documentation in the form of models consists of different notational elements (such as blocks and arrows). Is it understandable and clear what the semantic meaning of these elements is, i.e. what these elements represent?
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

20. **The documentation of the enterprise architecture is consistent**
Does the documentation always use the same notation, structure and terminology?
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

Vision

The questions below are regarding the vision of the company on their own strategic goals, and the goal of the enterprise architecture.

21. **I know and understand the vision and strategy of Schiphol as an organization**
To what extent are you familiar with the strategic goals and vision of Schiphol as an organization?
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

22. **I know and understand the architecture vision of Schiphol**

To what extent are you familiar with the strategic goals and vision of the enterprise architecture?

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	⬭	⬭	⬭	⬭	⬭	Strongly agree

Complexity

The final question is regarding the overall complexity of the enterprise architecture of Schiphol.

The complexity of an enterprise architecture is captured by its elements and their relations. These elements can be business processes, applications or even hardware. The entirety of these elements, their relations and structures are often considered to be very complex due to their size and interconnection.

Please note that the question below regards subjective complexity, and thus asks for your opinion on the complexity of the enterprise architecture.

23. **In your opinion, how complex is the entire enterprise architecture of Schiphol?**

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely simple	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Extremely complex