2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop

Measuring Enterprise Architecture Complexity

Maria Eugenia Iacob
University of Twente
Enschede, The Netherlands
m.e.iacob@utwente.nl

Jeroen Monteban
University of Twente
Enschede, The Netherlands
j.monteban@student.utwente.nl

Marten van Sinderen
University of Twente
Enschede, The Netherlands line 5:
m.j.vansinderen@utwente.nl

Erik Hegeman
Deloitte Consulting
Amsterdam, The Netherlands
ehegeman@deloitte.nl

Kiean Bitaraf
Deloitte Consulting
Amsterdam, The Netherlands
kbitaraf@deloitte.nl

***Abstract*— Complexity management has become an essential undertaking for enterprise architecture (EA). It strives for an optimal level of complexity to efficiently and effectively use the EA for its intended purposes. The basis for 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 EA all have a different perception of complexity, leading to the notion of subjective complexity. This research aims to incorporate objective and subjective complexity metrics in a single EA complexity measurement model. A systematic literature review has been carried out to make an inventory of existing complexity metrics. Semi-structured interviews were used to gain insights in stakeholder perceptions and subjective complexity attributes. Based on these results, a conceptual model of EA complexity was designed. The constructs in this model have been operationalized with metrics to create a measurement instrument of EA complexity. The model and its operationalization was then validated through expert interviews, and tested during a case study, where it has been applied in practice.**

Keywords*— *enterprise architecture, objective complexity, subjective complexity, measurement instrument, metric

I. Introduction

A. *Problem statement*

Complexity has been identified as one of the major challenges faced by the discipline of enterprise architecture [22], and it has been attributed as one of the causes of high failure rates in IT projects [7]. Complexity reduction, for example through application consolidation, is a popular remedy in large IT landscapes. At the same time, a certain level of architectural complexity is necessary to properly support business goals and requirements, and to enable extensive functionality ([36], [27]), such as quantitative analysis [14]. In other words, this ensures the alignment of the enterprise goals and strategy with the EA, and, most importantly, it makes possible the analysis of EA quality issues and of the compliance with non-functional requirements. This poses a challenge: in order to maximize the effectiveness of an architecture, it is important to find the right balance between complexity excess and deficit, that is, to find an optimal level of complexity ([13], [6], [27]).

Architectural complexity can have an important influence on the performance of enterprises and on their IT landscapes, and should be managed properly. In fact, [19] 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 EA 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 [21]. One of the problems in this regard is measurement. The lack of any generally accepted methodology for complexity measurement indicates a shortage of research in this field. At the same time, measurement is a prerequisite for proper complexity management. As [31] (p.1) noted, “Measurability is the essential basis for management”. However, quantifying architectural complexity is difficult, as many different aspect of may play a role, which explains why no standardized or proven method currently exists.

B. *Contribution and relevance*

The concept of complexity seems hard to measure, but it is 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. We take as baseline the study by [29], which proposed a more abstract approach to complexity in EA. They note that different interpretations of complexity throughout research impedes a common acceptance and understanding in the field. Therefore, they propose a conceptual framework aimed at unifying these views on complexity. According to [29], the various aspects of complexity can be specified through four dimensions as shown in Fig. 1.

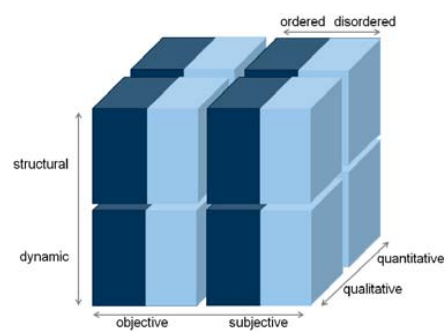


Fig. 1 Complexity dimensions [29]

Central to our research is the distinction between objective and subjective complexity. This 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 EA, different stakeholders may have very different perceptions of an architecture’s complexity. In other words: subjective complexity exists in the eye of the beholder, and depends on a combination of various factors,

2325-6605/18/$31.00 ©2018 IEEE
DOI 10.1109/EDOCW.2018.00026

which are related to both the object of study and the observer.

The main contribution of this paper is a conceptualization of EA complexity measurement, including the variables influencing architectural complexity, and a collection of appropriate metrics to measure them. We investigate how objective and subjective complexity metrics can be incorporated in this theoretical model, to ultimately facilitate EA complexity management.

Although for a proper architectural complexity management approach, all dimensions defined in Fig. 1 [29] have to be considered, we decided to limit the scope of this research to the dimension objective-subjective complexity. As also observed by [29] and confirmed by our own systematic literature review, the notions of subjective and dynamic complexity (see Section II.B) are very much underrepresented in the extant literature. Whereas both notions are relevant to EA complexity, we chose to first deal with the subjective notion. This seems particularly interesting in regards to complexity management, as a better understanding of the mechanisms underlying this type of complexity may lead to a significant improvement of architecture stakeholders' collaboration and to a shared understanding/simplification/compliance of architectural artifacts. Dynamic complexity (which constitutes a subject of future work), on the other hand, is primarily relevant for situations in which capturing complexity in architectural designs is most challenging. One should think of architectures that include for example cyber-physical systems-of-systems, in which autonomy and evolution, combined with emergent behavior are critical aspects of the design. Existing architectural approaches and modelling languages are to a large extent still not capable to capture this type of dynamic complexity aspects, and metrics to quantify them are still to be defined.

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 EA 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 EA seems to be developing, but still incomplete. [29] observes an underrepresentation of the subjective complexity dimension in existing literature. No more than 2% of existing EA complexity metrics consider subjectivity. Hence, advancing the state of the art on subjective complexity research would fill a gap in the existing body of knowledge. It should be also noted that all other complexity types (i.e., structural, quantitative, qualitative, ordered, disordered) are represented in our approach through the selection of metrics included in the proposed instrument.

Additionally, this research has great potential relevance in practice. 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 architecture complexity. Lack of a shared understanding among stakeholders, which can be caused by a different perception of the architectures complexity, may be causing disagreement and to mismanagement of the architecture, as responsible stakeholders might take incorrect or ineffective decisions. 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 [15]. This is important goal, as eexcessive complexity in the architecture of an enterprise or its IT landscape has been found to have a series of important negative organizational consequences. Fig. 2 presents the implications of excessive complexity found by five empirical studies ([1], [2], [28], [23], [36]). This is why a profound understanding of complexity mechanisms can have far reaching effects organizations.

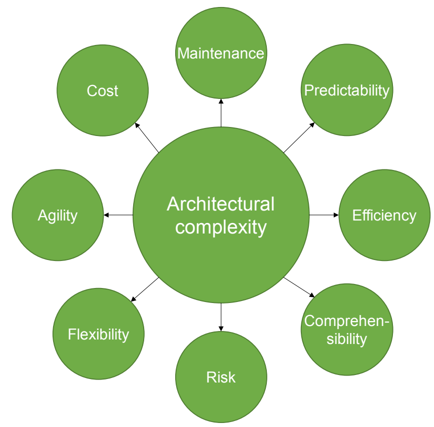


Fig. 2 Implications of architectural complexity

Methodology

In this study we make use of several (combined) research methodologies and approaches, as explained below:

- Systematic literature review to identify the attributes and currently used complexity metrics [18];
- Exploratory literature review to find stakeholders and how they interact with EA;
- Semi-structured interviews with stakeholders to identify attributes of subjective complexity;
- Design science research and descriptive inference to design a conceptual model of EA complexity and a measurement instrument to measure subjective complexity [37];
- Experts interviews and a case study to validate the proposed model.

The remainder of this paper is organized as follows. In Section II we analyze the relevant related work concerning architecture complexity. At the same time we introduce this area's underlying terminology, concepts and their taxonomy. In Section III we present the proposed theoretical model of architecture complexity, and its operationalization in the form of a measurement instrument. The model, and its' operationalization is then further tested and refined by means of a case study in a large organization (from the logistics sector), and expert interviews presented in Section IV. A discussion of the results of the present study and some pointers to future work conclude the paper in Section V.

II. Background on Enterprise Architecture Complexity

In this section we discuss the existing literature on EA complexity. The analysis is organized along the following topics. First we discuss the concept of architecture complexity, and its various characterizations. Then we go in depth on the topic of complexity measurement. The Sub-sections C and D are devoted to subjective complexity. Since subjective complexity is defined in relation with a specific "subject", we first review the different types of architecture stakeholders and their concerns regarding complexity. Finally, we identify several attributes of subjective complexity.

A. *Enterprise architecture complexity: definition and classification*

The Cambridge Dictionary defines complexity as "the state of having many parts and being difficult to understand or find an answer to". Much of the existing architecture research endorses this view, by relating complexity to the number of components or elements, their relationship, and to their variation/variety, and heterogeneity ([8], [10], [17]). [31] adds that the total complexity of an EA must take into account complexity within each domain, as well as the complexity of the interrelations between domains. Several studies look at patterns to be found in architecture descriptions, in the ways architecture concepts and relations are used: [16] define complexity by considering the pattern coverage of an architecture, whereas [9] calculate its distance from reference simplicity. Other studies define complexity in terms of their proposed metrics [11]. Interestingly, all of these studies use measurable terms to define complexity, such as the number of elements and relations. Literature on business complexity (the non-technological domain of EA) agrees on this as well ([6], [12]). 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 mentioned earlier, the taxonomy proposed by [29] is the most comprehensive and characterizes complexity according to four dimensions (Fig. 1):

- *Objective versus subjective complexity* is a dimension that distinguishes between complexity aspects that are independent of any observer, and perceived complexity which is the result of an observer's perceptions.
- *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.
- *Quantitative versus qualitative complexity*. This dimension refers to the way certain properties or attributes are evaluated/quantified.
- *Ordered versus disordered complexity*. The final dimension relates to the number of attributes considered when evaluating the system's complexity (high or moderate).

These four dimensions are independent of each other and can be combined in any way applicable in practice. Furthermore, [29] argues that a system can combine both complexity notions along a single dimension. This is also confirmed by the research in [1], where structural complexity is considered to be an indispensable element of dynamic complexity.

B. *Existing metrics for measuring complexity*

The current state-of-the-art on the measurement of architectural complexity was reviewed following a rigorous systematic literature review process. An essential step in this process is the creation of a protocol, explicating the steps to be taken in the review ([18]). Such a protocol was drafted based on the methodologies proposed in [18] and [35]. The protocol, depicted in Fig. 3, was used to guide the search and selection of relevant studies, and the subsequent data extraction and analysis needed for the identification of existing metrics for objective complexity measurement.

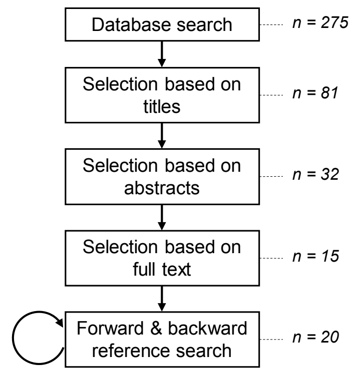


Fig. 3 Search & selection protocol

We found 42 metrics (see Appendix A), which have categorized according to their type, and their projection onto the complexity dimensions of [29]. Most of these metrics take as basis the underlying graph representation of the architecture, as 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 the architecture's complexity. This indicates that, since complexity is such a comprehensive property, multiple metrics with different viewpoints should be combined. The clustering 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 of [29] that subjective complexity metrics are yet to be found in the context of EA. Thus, the list of metrics found can primarily be employed to extract metrics regarding the objective complexity of EA. Of course, using the whole list of metrics in an EA complexity model would neither be efficient nor feasible. Therefore, a selection of metrics has to be chosen to be included in the model.

C. *Stakeholder attributes*

Niemi [26] aims to create a holistic view on the stakeholders of an EA. 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 who carry out the planning and development of the EA, facilitators who are involved with the governance, management, maintenance or sponsorship of the EA, and users who utilize enterprise architecture work and its products in their daily work.

[32] proposes a classification of stakeholders based on levels and domains. Four levels are distinguished: enterprise, domain, project, operational; as well as four dimensions: business, data, application, and technology.

When combining the frameworks of [25] and [32] we end-up with a stakeholder classification based on three attributes, each having a set of possible values. This is visualized in Fig. 4.

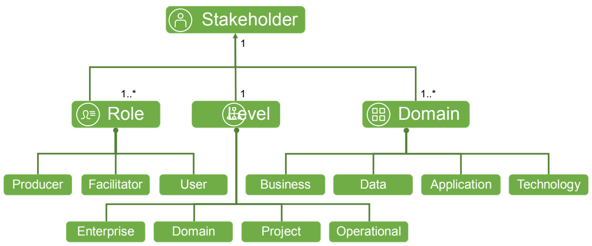


Fig. 4 Stakeholder classification

D. *Attributes of subjective complexity*

Enterprise architectures 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 EA. Capturing the subjective aspects of complexity can therefore help organizations to manage their IT landscape more effectively [15]. 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 [26]. Human cognition defines the property of comprehension or understandability, which in turn influences whether an entity is perceived to be complex or simple [26]. Cognitive informatics considers the transfer of information essential in this process. In cognitive informatics, complexity is considered to be related to the ease of comprehension. [3] states that a system of constructs that has a highly differentiated interpretation among persons (i.e., one that is difficult to understand, and, thus, it is interpreted differently), is cognitively complex. The application of cognitive complexity in enterprise architectures has not yet been studied. However, cognitive software complexity has been studied for quite some time, and has resulted in a series of cognitive complexity metrics ([4], [26], [33], [38]). 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. [4] defines the mental processes a software engineer uses when interpreting code, which consists of 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, [4] defines a theoretical classification of software complexity. Although the previously mentioned metrics are hard to transfer into architecture metrics, this theoretical basis might prove useful for application in enterprise architecture. This classification [4], shown in Fig. 5, further specifies the cognitive complexity of software as consisting of problem complexity, stakeholder characteristics, and structural (which in the context of this study can be interpreted as structural and objective complexity). These categories can be applied to EA as well, hypothesizing that subjective complexity is influenced by (among others) problem complexity, stakeholder characteristics and objective complexity.

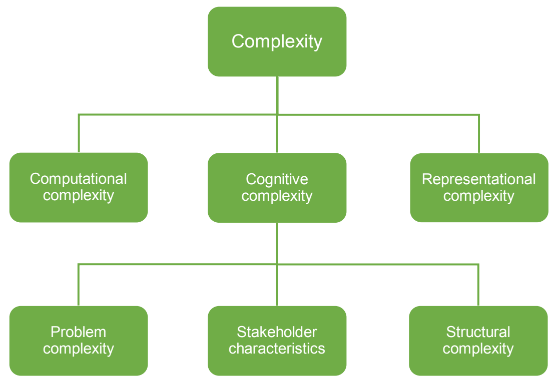


Fig. 5 Cognitive complexity taxonomy [4]

One of the stakeholder characteristics theorized by cognitive informatics to influence human perception is the way of processing information. Cognitive information is classified into four categories: knowledge, behavior, experience and skills [34]. 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, [30] 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 [4] focuses on representational complexity. Although this is not elaborated in their paper, several other studies focus on this phenomenon. [20] finds that the level of detail of documentation influences understandability. [24] states that the visual notation (in the case of EA the diagramming technique used to specify architecture descriptions and to create its documentation), greatly affects understanding. [24] 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.

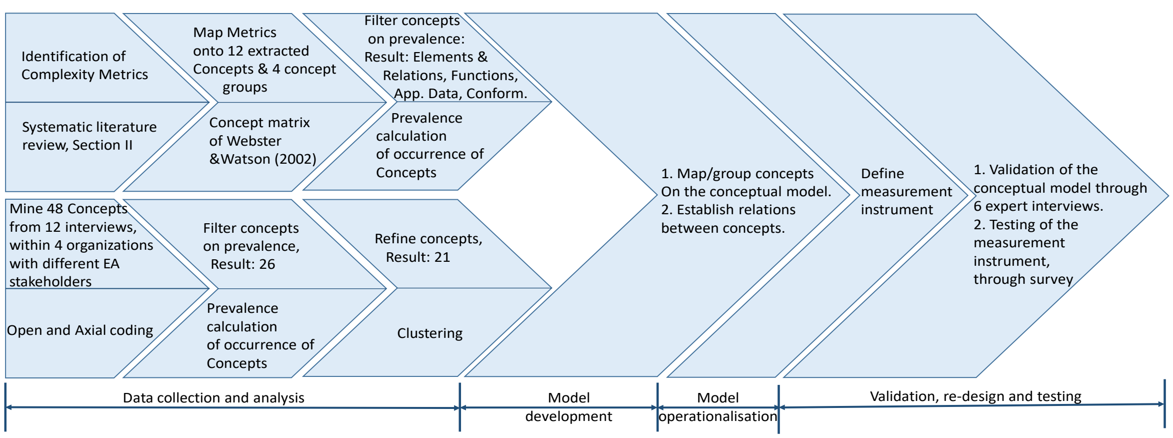
- *Perceptual configuration* refers to visual characteristics, based on which the structure and relations between elements in a diagram are inferred.
- *Working memory* makes possible that information is temporarily stored for processing. Since the human working memory has very limited capacity, this is an important bottleneck in image processing.
- *Long-term memory*: for information to be transferred from the working to the long-term memory, it has to be linked to prior knowledge. Similarity to prior knowledge greatly influences speed and accuracy of the processing.

Although the influence of documentation on EA perception seems very plausible, there is no empirical data to support this yet. The existing research is focused on software models for the production of such architecture descriptions (i.e., models), and has contradictory findings. For example, [5] finds that when model size increases, it becomes less comprehensible, which may be explained by the limited working memory of the brain. However, [20] finds that a less detailed/superficial documentation, leads to lower understandability, and theorizes that this is due to the loss of important context information.

Concluding, the research on the perception of EA complexity is still limited. Existing research on cognitive informatics, and cognitive software complexity does indicate some areas that may be of influence on subjective complexity. However, empirical research is needed to confirm these findings in the context of EA, and to explore further variables influencing subjective complexity.

III. A Theoretical Model for EA Complexity

The theoretical model we propose and its operationalization is the result of multi-step process shown in Fig. 6. In this section we will explain briefly the different phases of this process and their deliverables.



A. Data collection and analysis

The information sources for the development of the model and its operationalization were the systematic literature review, and of a series of 12 interviews with experts from four organizations, from different industries.

As mentioned earlier, through a *systematic literature review* a list of metrics have been identified (see Appendix A). These form the basis of the later operationalization of the constructs in the theoretical model. Furthermore, based on this list of metrics, twelve concepts were extracted. The metric and concept extraction from literature has been executed following the "concept matrix" method of [35], and is shown in Fig. 7. Metrics that use synonyms and/or are obviously highly similar, were aggregated to form a list of forty-two unique metrics that are mentioned in the 20 selected studies at least once. Next, different concepts were extracted from these metrics, based on the authors' judgment and semantic interpretation of the literature. This resulted in twelve concepts. Using this data, a concept matrix was created, and the concepts' prevalence (calculated as frequency of occurrence) in the literature was determined.

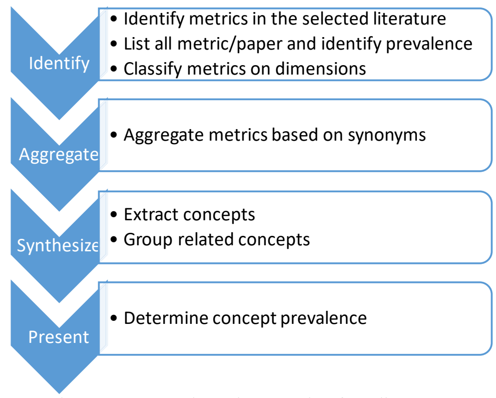


Fig. 7 Concept and metric extraction from literature

Finally, by eliminating the concepts with a prevalence lower than 25%, a set of four concepts were retained: Elements & relations, Functions, Application data, and Conformity. As it will be explained later, these concepts have either been used directly as a construct (i.e., conformity), or have been mapped on other constructs identified during the interviews.

The other source of data, namely the 12 *interview* transcripts with EA stakeholders have been coded (using open and axial coding) and yielded a total of 48 unique codes. However, many of these were only mentioned once or twice, and, therefore, do 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 some construct.

This process was similar to that used in the analysis of the structured literature review, and done based on the judgment of the authors. One or more codes were assigned to these constructs, resulting in a list of 21 constructs. Table 1 shows a list of the codes, their assigned constructs, and their clustering in a number of concept groups (e.g., Architecture, Enterprise, etc.) that were useful for eventually structuring the theoretical model into larger factors (i.e., groups of 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 the code's semantics.

Table 1. Constructs mined from interview transcripts

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

B. Development of the theoretical model

The first step of this phase is to come up with a consolidated list of constructs that influence objective and subjective EA complexity. To this end we combine the input from both literature and the interviews. In an effort to integrate the two sources, we have tried to find natural mappings of constructs identified through literature onto constructs identified through interviews, as shown in Table 2. One concept (conformity) for which no mapping could be found has been added to those mentioned in the last column of Table 1, as individual concept of the model, and part of the Architecture concept group.

Table 2. Construct mappings

Code group	Concepts extracted from literature	Mapping on constructs identified from intervies	Motivation
Architecture	Elements & relations	Size, Heterogeneity	this concept stems from the use of size and/or heterogeneity metrics
Enterprise	Functions	Business complexity	most of this concept's underlying metrics are related to functionality or processes, which are part of the business domain
Enterprise	Application data	Apllication complexity	clearly related to the application domain

The next step of the model construction was to define relationships between the identified constructs. Due to space limitations we will only briefly explain how this has been achieved. The source for the definition of relationships, was, as in the case of constructs, the surveyed literature and the interviews. As starting point, we took the following natural assumptions: i) objective complexity must be an important predictor of subjective complexity, as it sensible to expect that something which is inherently complex will also be perceived as complex, and ii) each of the concept groups has a direct influence on either objective, or subjective complexity. Thus, based on the nature of the metrics associated with these construct groups (i.e., objective or subjective), and on the semantic interpretation of the definitions given to them we established the construct groups Architecture, Enterprise, and Environment consist of predictors of objective complexity, while Mission, Model, and Stakeholders consist of predictors of subjective complexity. The only thing left was to identify whether also other relationships occur among the constructs within one group.

The complete list of intra-group relationships is depicted in Fig. 8. These have been derived through the analysis and interpretation of the interview transcripts and the literature. For example, in the case of the "Model" group, the underlying arguments for the depicted relationships are as follows. With respect to understandability of architectures, most interview participants mentioned documentation.

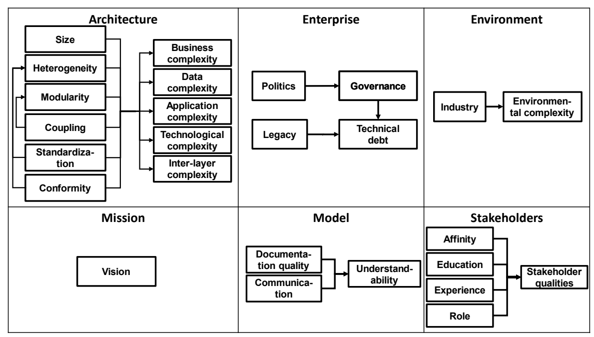


Fig. 8 Intra-group relationships

Several aspects of documentation were recognized to influence understandability, mostly related to 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".

The final result of the relationship definition exercise is shown in Fig. 9, and represents the first version of the model. This was validated through expert interviews (Section IV.A) which led to a slight re-specification of the model (Fig. 10).

C. Operationalisation of the theoretical model

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. Due to time constraints, not all constructs have been operationalized. To still be able to fully measure complexity, we focused on those independent variables in the causal graph from Fig. 9 that could be measured without further specification using available metrics. For example, for the Enterprise group, only the construct "technical debt" has been operationalized. Please refer to Appendix B, for the precise definition of all metrics included in the measurement instrument, and to Fig. 9 for an overview of the operationalized model. It should also be noted that metrics included in the instrument come from the surveyed literature (see Appendix A).

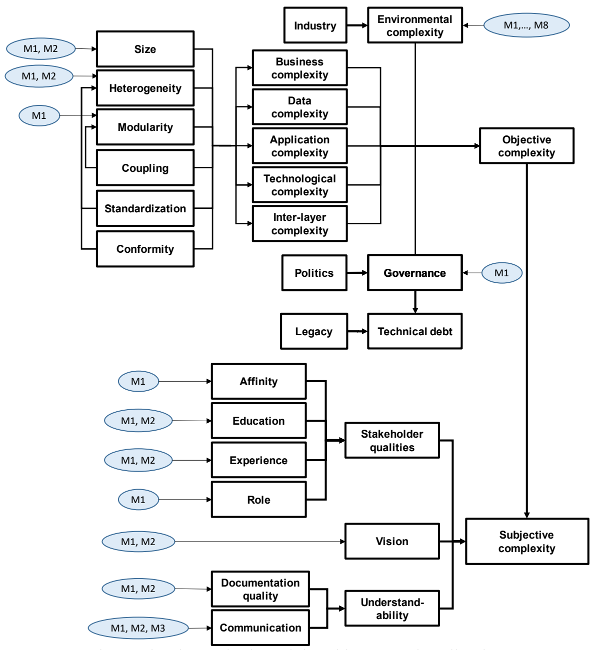


Fig. 9 The theoretical model and its operationalization

IV. Model Validation

A. Expert interviews

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, model development, 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.

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 conceptual model proposed in Section III, 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 them from being “counted twice”.

Secondly, a change of the enterprise group was suggested by adding an additional relation. The expert suggested that politics (only influencing governance in the original model) can lead to legacy as well.

Finally, one of the experts argued that constructs from both the enterprise and the environment group should directly relate to domain complexity. In the conceptual model proposed in Section III, 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 re-specification of the conceptual model shown in Fig. 10.

B. Case study

The second validation goal was to test the designed measurement model, through its application in practice in the context of a small case study. The scope of case study was intentionally limited to the predictors of subjective complexity (highlighted in blue in Fig. 10), as objective complexity has been extensively researched in the literature. The case study took place at a large organization from the logistics-sector. Data on subjective complexity was collected from three participants, one from each of the roles identified in this research (i.e., user, facilitator and producer). First we administered a questionnaire that has been developed according to the metrics included in the measurement instrument, followed by semi-structured 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 weights of each of these constructs.

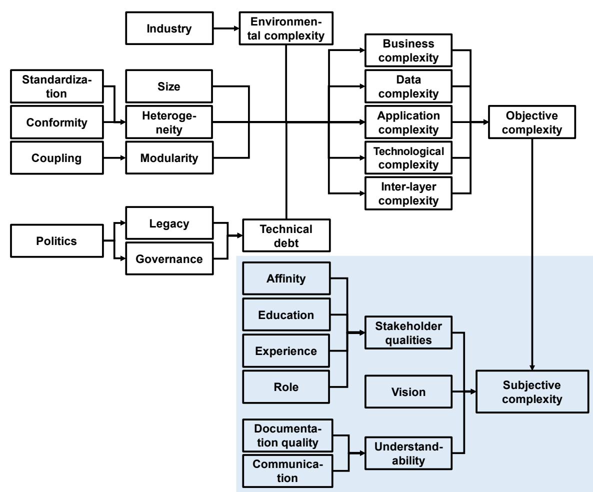


Fig. 10 Complexity conceptual model re-specified

Survey results: As mentioned earlier, a questionnaire has been developed based on the operationalization discussed in Section III, and using the metrics defined Appendix B. The goal was to perform a small scale test of the measurement instrument in a company that handles and maintains complex and large architecture descriptions, and see whether it leads to contradictory results, i.e., whether the calculation of the subjective complexity score based on the predictors' measurements significantly differs from the self-reported perception of architecture complexity. This small scale experiment showed that the tested part of the model (related to stakeholder qualities, vision and understandability) could account in for approximatively 62% from the self-reported subjective complexity. The difference could be explained by the fact that one predictor, namely the objective complexity was not included in this experiment and therefore its impact was not measured/taken into account. Although it is hard to generalize from a single case with n = 3, these early results are encouraging.

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 recognized 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, which did not lead to a re-specification of the model this time.

V. Conclusions

This research provides a contribution to the existing theory on EA complexity measurement in several ways. Firstly, the research summarizes the existing literature on EA complexity measurement. 20 papers were selected (following a systematic literature survey) that define complexity in measurable terms. They reflect the current state-of-the-art concerning EA complexity.

The main contribution to theory resides in its subjective complexity focus. 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 us to create a conceptualization of subjective complexity. Additionally, a measurement model has been devised by integrating existing complexity metrics and introducing new metrics.

By proposing a conceptualization and a measurement instrument for subjective (and objective) complexity this research fills a gap in the extant literature and opens several interesting possibilities for subsequent research.

Future work should at least consider the following avenues:

- Although the results of our case study were promising we are aware that further validation (in larger cases) of the model and its operationalization is desirable.
- Empirical validation of the relationship between objective and subjective complexity has been omitted from our case study due to time constraints. However, this is part of our short term follow-up research, as the participating organization is willing to give us access to their architecture descriptions for a complete assessment of the objective complexity of their EA.
- Simply combining the metrics for each construct does not give a final complexity "score". Model constructs almost certainly have different levels of influence on complexity. Therefore, during the case study we determined the relative weights of the several constructs involved in the model. However a fine-tuning of these weights, through larger experiments, is still needed.
- One other rather un-explored area of complexity is that of dynamic complexity. It is of major interest to understand how to measure, and, most importantly how to capture this type of complexity in architecture designs, in particular in relation with the advent of the new Industry 4.0 paradigm. The design of complex, autonomous systems-of-systems requires such capabilities, while the current EA approaches are only to a very limited extent capable to deal with dynamic aspects, such as emergent behavior, and evolution.

References

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

[2] J. Beese, M.K. Kazem Haki, S. Aier, and R. Winter, "Revisiting the impact of information systems architecture complexity: A complex adaptive systems perspective," In *38th International Conference on Information Systems*, Seoul, 2017.

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

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

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

[6] S. Collinson, and M. Jay, "From Complexity to Simplicity. London: Palgrave Macmillan," 2012.

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

[8] J. S. Davis, and R. J. LeBlanc, "A study of the applicability of complexity measures," *IEEE Transac- tion on Software Engineering*, *14*(9), 1366–1372, 1988.

[9] M. Efatmaneshnik, and M.J. Ryan, "A general framework for measuring system complexity," *Complexity*, *21*(1), 533–546, 2016.

[10] R. L. Flood, and E. R. Carson, "Dealing with complexity: An introduction to the theory and applica- tion of systems science," In *Dealing with Complexity: An Introduction to the Theory and Application of System Science*, 23–38, Springer, 1993.

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

[12] J. Gharajedaghi, "Operational thinking. In Systems Thinking: Managing Chaos and Complexity, A Platform for Designing Business Architecture", 109–131, Elsevier Science, 2011.

[13] B. Heydari, and K. Dalili, "Optimal system's complexity, an architecture perspective," In *Proc. Comp. Science*, 12, 63–68, 2012.

[14] M.-E. Iacob, H. Jonkers, "Quantitative analysis of service-oriented architectures", *International Journal of Enterprise Information Systems*, vol. 3, no. 1, Jan.-Mar. 2007, 42-60.

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

[16] R. Kazman, and M. Burth, M., "Assessing architectural complexity," In *Proceeding of the 2nd Euromicro Conference on Software Maintainance and Reengineering*, 104–112, Florence, 1998.

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

[18] B. Kitchenham, and S.Charters, "Guidelines for performing Systematic Literature Reviews in Soft- ware Engineering," *Technical report*, Evidence-Based Software Engineering Project, Durham, 2007.

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

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

[21] H. Lee, J. Ramanathan, Z. Hossain, P. Kumar, B. Weirwille, and R. Ramnath, "Enterprise architecture content model applied to

complexity management while delivering IT services," In *2014 IEEE International Conference on Services Computing*, 408–415, Anchorage, AL, 2014.

[22] C. Lucke, S. Krell, and U. Lechner, "Critical issues in enterprise architecting: A literature review," In *Americas Conference on Information Systems*, Lima, 2010.

[23] M. Mocker, "What is complex about 273 applications? Untangling application architecture com- plexity in a case of european investment banking," In *Proc. of the 42nd Hawaii International Conference on System Sciences*, 1–14, IEEE, 2009.

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

[25] E. Niemi, E., "Enterprise architecture stakeholders - A holistic view. In *Americas Conference on Information Systems*, Keystone, 2007.

[26] D. Singh, and A. Misra, "Cognitive software development process and associated metrics - A framework," In *5th IEEE International Conference on Cognitive Informatics*, 255–260, 2006, Beijing.

[27] C. Schmidt, "Business architecture quantified: How to measures business complexity," In D. Simon and C. Schmidt (Eds.), *Business Architecture Management*. Springer, 243–268, 2015.

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

[29] A. Schneider, M. Zec, and F. Matthes, "Adopting notions of complexity for enterprise architecture management," In *20th Americas Conference on Information Systems*, 1–10, Savannah, 2014.

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

[31] A. Schütz, T. Widjaja, and J. Kaiser, "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, 2013.

[32] B. van der Raadt, S. Schouten, and V. van Hans, "Stakeholder perception of enterprise architecture," In *European Conference on Software Architecture*, 19–34, 2008, Paphos.

[33] Y. Wang, "Cognitive complexity of software and its measurement," In *5th IEEE International Conference on Cognitive Informatics*, 226–235, 2006, Beijing.

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

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

[36] K. Wehling, D. Wille, C. Seidl, and I. Schaefer, "Decision support for reducing unnecessary IT complexity of application architectures," In *2017 IEEE International Conference on Software Architecture Workshops*, 161–168, 2017.

[37] R.J. Wieringa, "Design science methodology for information systems and software engineering," Springer, 2014.

[38] D. R. Wijendra, and K. P. Hewagamage, "Automated tool for the calculation of cognitive complexity of a software," In *2nd International Conference on Science in Information Technology*, 163–168, 2016.

APPENDIX A: CLASSIFICATION OF COMPLEXITY METRICS

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

APPENDIX B: OPERATIONALISATION OF THE THEORETICAL MODEL

Code group	Construct	Metric name	Metric definition	Type	Calculation
Architecture	Size	M1: Number of	Measures the size of the sets of elements in	formative	$\lvert T \rvert$: count the total number of elements in the domain
		M2: Number of	Measures the size of the set of relationships in the respective architectural domain	formative	$\lvert R \rvert$: count the total number of elements in the domain
	Heterogeneity	M1: Element Entropy	Measures the heterogeneity of the set of elements in the respective architectural	reflective	$-\sum_{i=1}^{n} p_i \ln p_i$, where i ϵ T, pi = relative frequency of element i
		M2: Relation entropy	Measures the heterogeneity of the set of relationships in the respective architectural	reflective	$-\sum_{i=1}^{n} p_i \ln p_i$, where i ϵ R, pi = relative frequency of element i
	Modularity	M1: Element modularity	Measures the modularity of a network of elements	reflective	$\frac{1}{4m}\sum_{ij}\left(A_{ij}-\frac{k_i k_j}{2m}\right)s_i s_j$, where $m = \lvert R \rvert$, A_{ij} = number of edges between *i* and *j* k_i= degree of element *i* $s_i s_j = 1$ if i and j are in the same group and 0 otherwise
Enterprise	Technical debt	M1: Cost of rework	Measures the cost of rework (*Cr*) that needs to be done on elements in the architecture	reflective	$\sum_k C_r(E_k)$, for all new elements E_k, where $C_r(E_k)$=$\sum_j C_r(E_j)$, for all pre-existing elements E_j, where $C_r(E_j) = D(E_j, E_k) * C_i(E_j) * Pc(n-1)$, where $D(a,b)$ is the number of dependencies between a and b, C_i 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)
Environment	Environmental complexity	M1:Herfindahl-Hirschman index	Measures the market concentration by squaring the market share of each competing organization in a market, and summing the	reflective	$\sum_{i=1}^{N} s_i^2$, where *N* = number of firms s_i= market share of firm *i*
		M2: Size diversity	Measures the distribution of organizations in different size categories	reflective	$\frac{\sum_{j=1}^{m} O^2}{(\sum_{j=1}^{m} O)^2}$, where *O* = number of organizations in the industry *m* = number of organizational size classes
		M3: Heterogeneity of output	Measures the number and distribution of industries to which a given industry sells its output	reflective	$\frac{\sum_{j=1}^{m} D^2}{(\sum_{j=1}^{m} D)^2}$, where *D* = dollar volume of outputs in the industry *m* = number of organizations buying outputs
		M4: Specialization rate	Measures the proportion of an industry's shipments accounted for by primary products. This re- flects the diversity of products offered by an organization in the industry	reflective	*PP/TS*, where *PP* = number of primary product shipments *TS* = total shipments of all products
		M5: Labor diversity	Measures the diversity of different labor types present in the industry. Labor types are defined based on the Standard Occupational Classification	reflective	$\frac{\sum_{j=1}^{m} L^2}{(\sum_{j=1}^{m} L)^2}$, where L= number of employees, m = number of occupational codes represented in the industry
		M6: Asset size	Measures the average asset size of an organization in the industry	reflective	*AS/O* where *AS* = total asset size of the industry, *O* = number of organizations in the industry
		M7: Capital intensity	Measures the ration of the value of assets to the value of outputs for an average organization in the industry	reflective	*AS/D* where, *AS* = total asset size of the industry, *D* = dollar volume of outputs in the industry
		M8: Technical level of workforce	Measures the percentage of the workforce classified in scientific, engineering or other technical occupations, as defined by the Standard Occupational Classification	reflective	*TO/TS* where *TO* = number of employees working in a technical occupation, *TS* = total workforce
Mission	Vision	M1: Company vision	Measures to what extent the stakeholder is aware of the goal or strategy of the organization	formative	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
		M2: Architecture vision	Measures to what extent the stakeholder is aware of the goal or strategy of the enterprise architecture	formative	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]
Model	Documentation quality	M1: Available levels of detail	Measures the availability of different levels of detail in the documentation	reflective	Count how many levels of abstraction are available in the documentation
		M2: Available notational elements	3. Measures the availability of different notational elements for an element in the architecture. Ideally, this number should be 1	reflective	Divide the total number of notational elements by the number of element types in the architecture
	Communication	M1: 1. Documentation availability	3. Measures the availability of the documentation	reflective	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
		M2: Documentation detail suitability	Measures the suitability of the documentation's abstraction level for individual stakeholders	reflective	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?
		M3:	Measures the suitability of the documentation's notational elements for individual stakeholders	reflective	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
		M4: Documentation consistency	Measures the consistency of the documentation	reflective	Answer the following question on an ordinal scale of 1 (strongly disagree) to 5 (strongly agree): The documentation of the enterprise architecture is consistent
Stakeholders	Education	M1: Eductation Area	Measures a stakeholder's area of education	formative	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 scoring: Alpha 2, Beta 5, Gamma 2.
		M2: Education level	Measures a stakeholder's education level	formative	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 Scoring: University 5, Hbo 4, Mbo 2, High school 1
	Affinity	M1: Technology affinity	Measures a stakeholder's affinity for technology	reflective	Score the ten questions below on a Likert scale of 1 to 5 and calculate their average. 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
	Experience	M1: Experience in organisation	Measures the experience of a stakeholder in the organization	formative	Years active in the organization: [0-2], [2-5], [5-10], [10-20], [20+]
		M2: Experience with enterprise architecture	Measures the experience of a stakeholder with enterprise architecture	formative	Years of experience with enterprise architecture: [0-2], [2-5], [5-10], [10-20], [20+]
	Role	M1: Role	Identifies the role of the stakeholder: producer, facilitator or user	formative	Ask the participant for their job description and classify them in one of the three roles