Drivers and Metrics for Quantifying IT Landscape Complexity

Eva Stoica, João Moreira[0000−0002−4547−7000], Sebastian Piest[0000−0002−0995−6813], and Faiza Bukhsh[0000−0001−5978−2754]

[1] University of Twente, Enschede, The Netherlands
[2] evastoica4@gmail.com, j.luizrebelomoreira@utwente.nl, j.p.s.piest@utwente.nl, f.a.bukhsh@utwente.nl

Abstract. Mastering complexity is an important topic within the field of Enterprise Architecture (EA), and many companies perceive this as a difficult endeavour due to the growth and perpetual evolution of the Information Technology (IT) landscape. Numerous EA frameworks and methods exist that describe what an IT landscape is. However, there is no consensus regarding the definition of IT landscape complexity, its drivers and how to measure it. To help create a better understanding of what IT landscape complexity entails, this paper reviews established EA literature regarding the common elements of such a landscape and investigates prevalent drivers and metrics to quantify complexity. This paper introduces a standard way of describing the IT landscape through a conceptual model, focusing on the elements in the application layer. Furthermore, the prevalent IT complexity drivers and metrics are illustrated. The aim of this research is to support organisations in communication and architecting information systems by considering different kinds of IT elements and their particular characteristics, such as connectivity, adaptation, and complexity. Then, a discussion on how the drivers and metrics of complexity can be used to quantify the complexity of such a landscape follows. Our approach is a first step towards better processes for managing complexity. This research supports mastering complexity by allowing organisations to quantify IT landscape complexity when aligning their architectures with the proposed conceptual model.

Keywords: IT Landscape · Complexity · Enterprise Architecture

1 Introduction

The information technology (IT) present in a contemporary enterprise is in a continuous transformation and evolution [1]. An overwhelming number of components and elements allow an organisation to perform its day-to-day operations and achieve certain goals and outcomes. The rapid growth and perpetual evolution of elements, and their relationships, which create the IT landscape, lead to complexity. From a systems thinking lens, complexity is concerned with a set of parts or components that have vague or difficult-to-understand relationships varying across time [2]. The complexity of an IT landscape, thus represents the number of components or elements of an architecture, their relationships, and the variation or heterogeneity of these [38].

Given the IT growth and dynamicity, managing the IT complexity of an organisation can be considered troublesome. Companies must use new technologies on top of their existing stack to match business needs or create temporal measures to address these. Usually, companies consider that more IT elements can support the delivery of more capabilities, however, these additions can increase the complexity of the landscape and negatively influence agility or increase risks and vulnerabilities [19]

This complexity of IT landscapes is an intangible factor and most organisations experience difficulties in understanding its size or its impact [16,17,43]. Measuring the complexity associated with a change in the IT landscape can be perceived as a very difficult endeavour. This problem is further amplified by the fact that no consensus or well-defined standards are present in either academia or practice to explain what an IT landscape is comprised of or what factors can lead to complexity in such a landscape. Without a common understanding of elements of an IT landscape, subjectivity is observed in both modelling the landscape and thinking about what complexity means (e.g., complex landscape as a larger number of IT components versus intricate relationships between components over time). Additionally, the existence of means to quantify complexity is limited and scattered. An acknowledged problem or a "major limitation" [26] is that few companies have suitable methods or tools to systematically assess and evaluate complexity.

To grasp the expected benefits, commonly associated with IT in an organisation, such as lower costs, flexibility, or efficiency [19], complexity needs to be managed and measured. EA methods, tools and techniques can be leveraged for this purpose. With the support of an exploratory and a systematic literature review (SLR), performed according to the extensions proposed by Wolfswinkel et al. [51] to the works of Webster and Watson [47] an answer is provided to the following question: *Which drivers and metrics can be used in the complexity analysis of an IT landscape?*

This paper introduces a conceptual model which supports a precise definition of the IT landscape, focusing on the elements in the application layer. The model fosters a shared understanding of IT elements which aid with IT modelling processes. Moreover, prevalent complexity drivers and metrics are summarised, enabling stakeholders to define and quantify IT landscape complexity. Based on the conceptual model and the list of drivers and metrics, a blueprint for quantifying IT landscape complexity is developed.

This paper is structured as follows. Section 2 covers the background. Section 3 describes the research methodology for the literature reviews. In Section 4 the results of the reviews are analysed. Here, the conceptual model for the IT landscape is introduced, alongside the overview of existing drivers and metrics of complexity. Section 5 discusses the implications of the results and Section 6 concludes this paper.

2 Background

Section 2 offers an overview of the related works that serve as the basis for the literature study. First, it will be acknowledged what an IT landscape is. Then, the overview of what complexity represents in an IT context is introduced.

2.1 IT Landscape Architecture

Architecture Starting with the term architecture, this represents the philosophy underlying a system which describes its purpose, intent, and structure [23]. Architecture defines the fundamental organisation of a system embodied in its components, their relationships to each other and the environment, as well as the principles guiding its design and evolution [20],[24].

Enterprise Architecture Moving to EA, this represents “a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise’s organisational structure, business processes, information systems, and infrastructure” [24]. Even more, it is a means to support the business and IT alignment [35] and help bridge the gap between the current and future state of an enterprise [12]. A better alignment leads to “lower cost, higher quality, better time to market and greater customer satisfaction” [12]. For using EA, methodologies, viewpoints and modelling languages are needed. The most used combination that helps deliver enterprise architecture is using TOGAF [12], [18] as a methodology and ArchiMate [13] as a modelling language.

IT Landscape Next, in literature, the combination of the terms IT and landscape is used interchangeably with concepts such as application architecture, IT infrastructure, technical architecture, or information systems architecture. The IT landscape can represent “A set of hardware, software and facility elements, arranged in a specific configuration, which serves as a fabric to support the business operation of an enterprise” [27]. However, no standard definition of such a landscape or its composing elements exists [15,23,34]. Most current IT landscapes are the results of a history of projects that introduced their specialised software, hardware, and infrastructure [23]. Furthermore, over the years, organisations have combined older technologies, e.g., legacy systems, with newer ones [5], [50] leading to intricate relationships between components, including artificial intelligence, digital twins, and cloud computing [36]. This fast pace of developments can also influence the prioritisation of short-term or quick fixes which can result in architectural technical debt [22],[46].

2.2 IT Landscape Complexity

Definition IT landscape complexity refers to the number of components or elements of an architecture, their relationships, the variation or heterogeneity of these and the use of different levels of observation and granularity in modelling [32], [38].

Classification For the field of information systems or IT landscape complexity, Schneider [39] unified the existing research by creating a multi-dimensional framework. Four dimensions are included in this framework for classifying complexity: organised vs. disorganised, qualitative vs. quantitative, subjective vs. objective and static vs. dynamic [39]. These dimensions can guide the analysis of EA models.

3 Research Methodology

In Section 3, the overarching research question for this paper is introduced. Then, the approach selected to answer it is highlighted.

3.1 Research Questions

The main research question is stated as: *Which drivers and metrics can be used in the complexity analysis of an IT landscape?*. The answer to this question is presented with the help of an exploratory and systematic literature review (SLR). Two sub-research questions were formulated to further investigate the main knowledge question.
RQ1. What are the composing elements of an IT landscape?
RQ2. Which factors drive the complexity of an IT landscape?

As no accepted definition of an IT landscape components is present in the literature, the first sub-question investigates this by employing an exploratory review. The components and building blocks are pieced together to scope the IT landscape. The second question delves into an SLR to gain a better understanding of the drivers of complexity. Once the drivers are identified, existing metrics for their operationalisation have been distinguished via an exploratory literature review.

The motivation for the overarching research question is to have a unified view of what an IT landscape means and how complexity can be quantified based on measuring its drivers. The quantification can support organisations in their process of managing the complexity of their IT landscape.

3.2 Literature Review Approach

To structure the research, a mixed-method literature review was performed according to the extensions proposed by Wolfswinkel et al. [51] to the works of Webster and Watson [47]. By combining both an exploratory and a SLR, comprehensive answers can be shaped, alongside a foundation of existing knowledge and a holistic view of the investigated topics.

The scope of the review was to obtain answers to the questions introduced above. By performing a literature review, the questions can progressively become answerable. The databases used for this research are IEEE Xplore, ACM Digital, Scopus and ISI Web of Science.

Exploratory Literature Execution To answer RQ1, via an exploratory literature review, the following steps were carried out: the definition of the research question, the scoping of review and data sources, then the database search (based on the keywords from the question such as: "IT architecture elements") preceded by theme and concept identification and lastly the analysis and presentation of the findings. During the theme identification, Wolfswinkel et al.'s advice [51] of prioritising concepts rather than classifying individual articles and creating links between relevant ideas was used. This structure was created by the researcher, as an exploratory literature review does not require one. 6 papers were analysed for the RQ1 out of a pool of 20 selected works.

SLR Execution Moving to the SLR executed for answering RQ2, the process behind this type of review can be explained according to the five stages[1] proposed by Webster and Watson [47] .

In the define phase, the scope of the review is to gather answers to RQ2 by performing searches on the four academic repositories mentioned previously. In this stage, inclusion and exclusion criteria are defined. These aid in increasing the number of adequate results for valuable insights. Furthermore, the number of citations and the peer-review status of articles guided the search. Table 1 showcases the criteria for RQ2.

Table 1. Search Queries

Inclusion	Based on relevance
	Language - English
	Title, abstract and keywords – Include the term complexity or its derivates
	Thematic relevance – Fall under research areas concerned with information systems, enterprise architecture or computer science
Exclusion	Articles from medicine, social sciences, business management, mathematics natural sciences, education
	Articles discussing IT projects, government, management, supply chain, industry-specific concerns or software design

Next, the search phase is concerned with performing the search on the four above-mentioned repositories. Different queries, as seen in Table 2 were initially created and through episodic searches, they were altered such that sufficient and relevant outcomes could emerge. Wolfswinkel et al. mention that for the sake of transparency, it is important that the search terms, queries, and sources are documented [51].

In the selection phase, the doubles were filtered, the samples were refined based on the abstract and the title, and the full text was considered. Furthermore, forward and backward citations were performed. For the RQ2, 417 papers

[1] For more information the researchers can share additional documentation of the SLR process.

Table 2. Search Queries

Main Search Queries	Scopus	General	(TITLE-ABS-KEY (("IT" OR "IT architecture" OR "IT landscape" OR "Application architecture" OR "Application landscape" OR "Enterprise architecture" OR "Enterprise landscape" OR "Information systems architecture") AND ("complexity" OR "Complexity driver" OR "driver* of complexity" OR "Complexity origin" OR "origin* of complexity" OR "Complexity factor" OR "factor* of complexity" OR "Complexity source" OR "source* of complexity" OR "Complexity cause" OR "cause* of complexity" OR "Complexity root cause" OR "Complexity influence" OR "influence* of complexity")) AND TITLE (complex*) AND KEY (it OR "information system architecture"))
	IEEE Xplore	Title	Complexit*
		Abstract	IT OR IT architecture OR IT landscape OR Application architecture OR Application landscape OR Enterprise architecture OR Enterprise landscape OR Information systems architecture
			Complexity driver OR driver* of complexity OR Complexity origin OR origin* of complexity OR Complexity factor OR factor* of complexity OR Complexity source OR source* of complexity OR Complexity cause OR cause* of complexity OR Complexity root cause OR Complexity influence OR influence* of complexity
	ACM Digital	General	("IT" OR "IT architecture" OR "IT landscape" OR "Application architecture" OR "Application landscape" OR "Enterprise architecture" OR "Enterprise landscape" OR "Information systems architecture") AND ("complexity" OR "Complexity driver" OR "driver* of complexity" OR "Complexity origin" OR "origin* of complexity" OR "Complexity factor" OR "factor* of complexity" OR "Complexity source" OR "source* of complexity" OR "Complexity cause" OR "cause* of complexity" OR "Complexity root cause" OR "Complexity influence" OR "influence* of complexity") AND TITLE (complex*)
	ISI Web of Science	Title	IT OR IT architecture OR IT landscape OR Application architecture OR Application landscape OR Enterprise architecture OR Enterprise landscape OR Information systems architecture
			Complexity driver OR driver* of complexity OR Complexity origin OR origin* of complexity OR Complexity factor OR factor* of complexity OR Complexity source OR source* of complexity OR Complexity cause OR cause* of complexity OR Complexity root cause OR Complexity influence OR influence* of complexity

were the result of the aggregated searches from the four repositories, however only 17 of them were chosen for further analysis. The PRISMA methodology [33] helped in structuring this selection (as seen in Figure 1). All the academic articles underwent this process of open coding. Axial coding followed next, as relationships between the concepts were discovered and finally, selective coding to integrate and refine the categories. The key concepts were categorised using concept matrices[3], as explained in [51].

The last step is the presentation of the findings from the prevalent works alongside the created relationships between concepts and insights. These are described in Section 4.

4 Analysis of Research Results

Section 4 details the insights gathered from the mixed-method literature review. First, the concept of IT landscape is addressed, thereby facilitating an understanding of its constituent elements. Subsequently, the drivers of complexity in such a landscape are showcased.

4.1 IT Landscape

Components To be able to manage a complex environment, it is crucial to observe which elements or components are part of this high-level design space,

[3] These matrices can be shared upon request

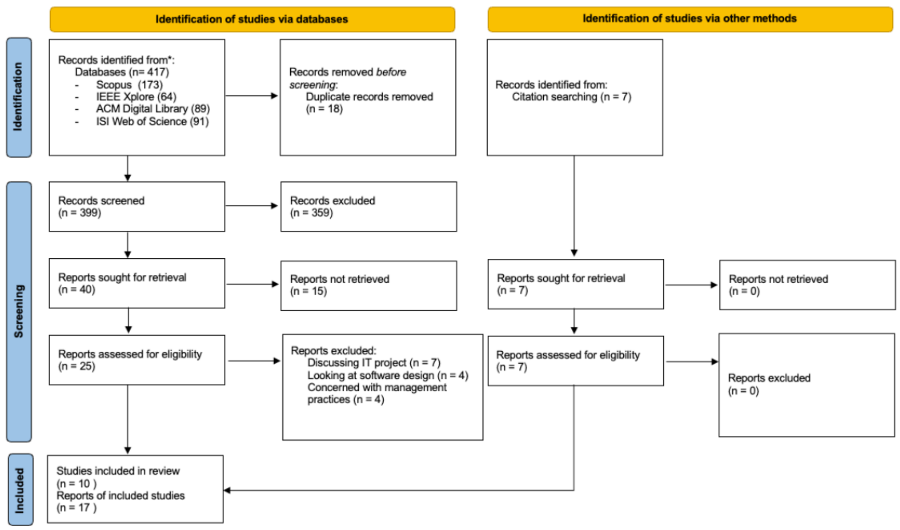

Fig. 1. PRISMA - Complexity Drivers

or the IT landscape. Firstly, a landscape can be represented with a three-dimensional coordinate system showing architectural relationships between chosen domains such as business functions, application components and products [37], [45].

The IT landscape is commonly associated with "A set of hardware, software and facility elements, arranged in a specific configuration, which serves as a fabric to support the business operation of an enterprise" [27]. Throughout the years, the term evolved and started being used interchangeably with others such as information systems or technical architecture and nowadays a standard definition of its composing elements is absent [15,23,34].

In 1987, Zachman [52], proposed the first framework for IS architecture. Nowadays, it represents one of the best-known architectural frameworks. Zachman started by looking at the traditional architecture components to build the basis of the IS architecture. Initially, the scope of the framework focused on data (what something is made of), functions/processes (how something is working) and networks (where something is located) and what those meant to different stakeholders. As the years passed, it expanded to include people, time, and motivation.

Rohloff [35] offers a differentiation between the IT landscape and the IS architecture. As opposed to IS architecture which focuses on a single system, the IT landscape is concerned with the design on a larger scale. To support this larger-scale perspective, one can consider the work of Hammer [15]. He describes IT architecture as a common high-level model or design that supports various disciplines and stakeholders in "taking design decisions in a multi-dimensional design space", "defining designs in a multi-disciplinary environment" and "communicating solutions"[15].

Another well-acknowledged book, written by Laan [23], discusses the building blocks of IT for describing what an IT infrastructure is. The model explains how from business processes, required for fulfilling an enterprise vision, information is stored and managed using applications, which in turn need platforms and infrastructure to run. There are three types of applications: client (e.g., web browsers, word processors), office (e.g., email servers, collaboration tools), and business specific (custom-built or highly customised applications, e.g., Enterprise Resource Planning) and various components such as databases or front-end servers that enable them to work. Each of the four layers (business, application, platforms and infrastructure) influences the management function, and their functionality "is supported by non-functional attributes" [23]. These non-functional attributes describe the qualitative behaviour of a system and can range from availability, scalability, and security to testability and recoverability.

Rohloff [35] also explains that the main IT blueprints, the ones creating a basis for a landscape, are the application landscape (showing how each business process is supported by applications), the data repository (deployment of databases and how they support information clusters), and the service landscape (infrastructure services). These blueprints follow the same division as what Laan [23] described: processes, applications, services, and infrastructure.

From the broad perspective introduced by Zachman [52] and the building blocks of IT discussed by Laan [23], a shift towards more tangible elements of the IT landscape must be in place. Bruckmann [6] mentions that an IT landscape is composed of various IT objects, or artifacts, which can be "hardware platforms, database products, operating systems, application servers, development tools, programming languages [etc.]". In this case, an application uses IT objects. Furthermore, he proposes a conceptual model to illustrate the breakdown of an IT object.

Building upon the proposed conceptual model introduced in [6], the attention can move to Hammer [15] which describes the dimensions of IT architecture. For complex landscapes, alongside components, such as an IT artifact, it is important to consider the behaviour of the system and its interactions. The design space of an IT architecture is composed of "functional and non-functional dimensions" [15]. The functional dimension is concerned with the high-level structure of an architecture, thus, the components, their specifications, interfaces, and restrictions. Moreover, for the functional perspective, behaviour, or dynamic structure, must be analysed. Behavior is touching upon, concurrent activities, interaction sequences or protocols between components. The non-functional dimension must investigate performance (e.g., response time, throughput), dependability (e.g., reliability, availability, security, and robustness) and general aspects (such as desirable future). This dimension is composed of requirements which were also perceived as important by Laan [23], as previously discussed.

Key Takeaways To combine the perspectives discussed and understand what an IT landscape is comprised of, Figure 2 is proposed by the researchers. According to the model, an IT landscape refers to the overarching structure, combination and configuration of applications (which can use several IT objects) and their relationships to support the business processes for delivering value to the customer.

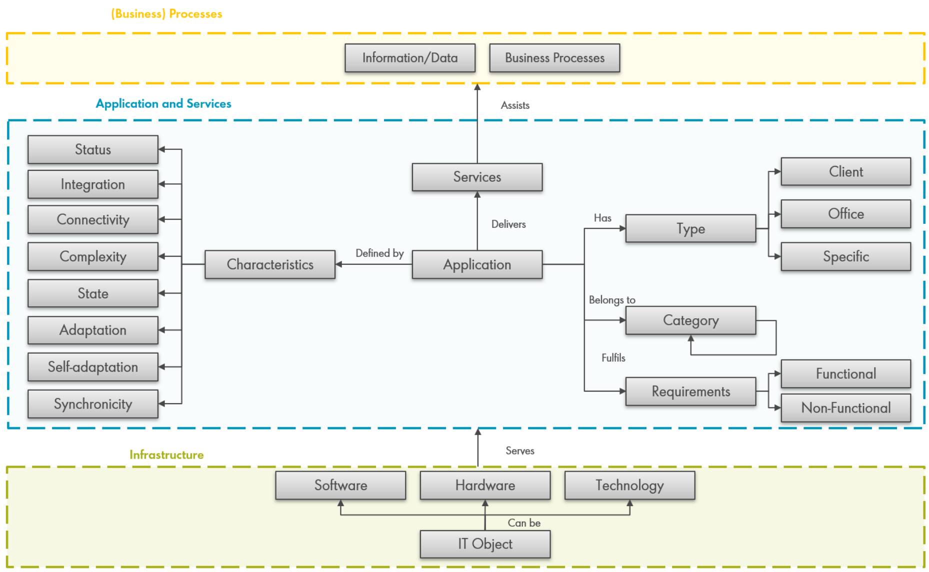

Fig. 2. Conceptual Model of IT Landscape Components

The model is comprised of three layers, similar to the core model of ArchiMate [13]. These are the business processes, applications and services, and the infrastructure layer.

The IT landscape is composed of a myriad of applications that support the business operations of an enterprise. The business has to complete certain processes, created based on information or data entities, that need to be supported by IT. Hence, from the business requirements, applications are selected and used by an organisation. These realise services that assist the business in delivering value to customers.

The applications can be combined and configured in various ways, and each of them has certain characteristics. According to Matook [30], there are seven different characteristics of an application: integration, connectivity, complexity, state, adaptation, self-adaptation, and synchronicity. Furthermore, each application has a status or a lifecycle phase, as mentioned by Bruckmann et al. [6]. For example, an application can be in one of the following phases "proposed, test, productive, standard, and retired" [6].

Next, applications can have a type. For example, Laan [23] distinguished three types of applications: client (e.g., web browsers, word processors), office (e.g., email servers, collaboration tools), and business-specific (custom-built or highly customised applications, e.g., Enterprise Resource Planning). These applications can also belong to a category.

Moreover, the applications in an IT landscape fulfil certain functional or non-functional requirements [15]. The functional dimension is concerned with the high-level structure of architecture, thus, the applications, their specifications, interfaces or interaction sequences, protocols, data exchange formats and restrictions [15]. The non-functional dimension must investigate performance (e.g., response time, throughput), dependability (e.g., reliability, availability, security, and robustness) and general aspects (such as desirable future). This dimension is composed of requirements which were also perceived as important by Laan [23], as previously discussed.

The model illustrates that applications are using "IT objects" (as defined by Bruckmann), such as software (e.g., operating systems, databases, compilers, en-/decryption software), hardware (e.g., servers, workstations, storage devices) and technology. The last IT object, technology, is seen as "software architecture patterns" or "application protocols, digital preservation formats, programming languages" [6]. Matook [30] also supports the view that the "IT artifact" is based on technology, software, and hardware. These three elements are integrated, and they help with the execution of information-processing tasks with a specific usage purpose.

4.2 Complexity Drivers and Metrics

Drivers The SLR on complexity drivers and metrics leads to numerous insights. The drivers are manifold for an IT landscape and range from well-acknowledged and debated ones such as the number and heterogeneity of applications and their relationships to less researched drivers, such as unnecessary variability [49], lack of knowledge or enterprise architecture debt [14], [29]. A high-level overview of these is introduced next.

Beese et al. [4] suggested five drivers for IT architectural complexity: size, diversity, integration, planning, and dynamics. Schüetz [41] classifies the origin of IT complexity into the following categories: the number and heterogeneity of component and their relationships, the rate of change, and the application to different IT architecture domains (function, data, technology, and interfaces). Similarly to Besse et al. and Schüetz, papers [50] and [40], find the number of applications, heterogeneity, standards or functional scope as causes for IT landscape complexity.

Another aspect that seems to cause complexity is the systems integration, as described by Jain et al. [21] in an eighteen-criteria framework. This includes discussions on interface openness or abstraction of system architecture, that impact the integration complexity. For a more specific origin of complexity that negatively impacts costs, architecture dependability, maintenance, and evolving systems, Wehling's work on unnecessary variability (components supporting the same business process) [48], [49] can be analysed.

For large-scale complex IT systems, Sommerville et al. [42] introduce two origins of complexity: trust relationships of components in a system and lack of knowledge about the system (from a human perspective). Mocker [31] also investigates the origins of complexity for application architecture for large-scale IT systems. He discovers that interdependence (interconnectedness), redundancy (supporting the same process), the number of different applications, as well as their age impact cost (e.g., maintenance and operation), agility, and the overall IT complexity. Furthermore, the complexity of the business requirement that an application is helping with can generate overall complexity.

Lentz and Bleizeffer [25] support the above-mentioned points, but they also discuss the impact of unintelligent design as a cause of IT complexity. Frequent application updates and fixes for a faster turnaround are high priorities in larger enterprises, such that everything can operate accordingly. These considerations can lead to architectural technical debt. Quick fixes in one area of the enterprise can mean accepting compromises in other areas, generating a higher complexity, or cost to restore the state of the system in the future [14].

Technical debt describes the delayed technical development activities for getting short-term payoffs or timely release of specific software [53]. Atchison [3] describes how technical debt and complexity go hand in hand and concludes that "Technical debt is the core of IT complexity".

Architectural debt resides under the umbrella of technical debt [29]. Martini [28],[29] illustrates possible causes leading to architectural technical debt, for example, business factors, the use of third-party/open-source systems that were not initially part of the architecture, parallel development, non-completed refactoring (e.g., using a new application programming interface, however, the previous one cannot be removed because of backwards compatibility) or the human factor (e.g., differences in knowledge, ignorance, error-prone situations).

As technical and architectural technical debt are both specific to the software domain, Hacks [14], [44] coined the term Enterprise Architecture Debt to create the link between business and IT. It is defined as "the deviation of the current present state of an enterprise from a hypothetical ideal state"[14]. This can mean elements not optimally implemented, the use of bad interfaces, interoperability issues or different priorities of stakeholders. To understand the taxonomy of EA debt the work of Daoudi et al. [8] serves as a reference point.

Metrics Upon understanding the potential drivers influencing the IT landscape, it is also important to identify which metrics can quantify them. The most important work addressing this was performed by Iacob et al. [19]. Through a SLR an inventory of complexity metrics was created. This SLR was enhanced with the help of 12 semi-structured interviews with experts from different organisations. Based on the combined results, 42 metrics for quantifying EA complexity were identified. Some of these are counting the number of relations or the number of elements in an architecture, using cyclomatic complexity, conformity, or redundancy.

Beyond these 42 metrics, one recent development in the literature, focusing more on dynamic complexity is introduced by Daoudi [9]. This is called the Adaptive Enterprise Architecture, an approach for proactive sensing and responding to change and for managing trade-offs. In moving from as-is to to-be scenarios, the EA Degree of Dynamic Complexity (DDC) is proposed. Furthermore, paper [9] lists existing types of calculation for some complexity metrics, such as an interdependence matrix (for measuring layer interdependency Business-Application-Technology), entropy (for heterogeneity) or a context awareness capability matrix (for subjective complexity). Another important complexity measurement that can be used to choose between EA alternatives is proposed by Rojas [11]. He mentions that complexity metrics at the implementation stage identify elements that can either reduce or increase complexity, for example, a high cohesion between elements, respectively a high coupling between elements. For the design phase, to re-estimate project effort and understand the complexity of EAs, Rojas proposes the Structural Complexity Measure (SCM).

Key Takeaways To summarise the discussed research streams, Figure 3 has been created by the researchers. The complexity drivers and potential metrics for their quantification are proposed. The left-hand side column represents the categories of drivers, followed by potential examples of drivers for IT landscape complexity. The right-hand side column depicts possible metrics to quantify these.

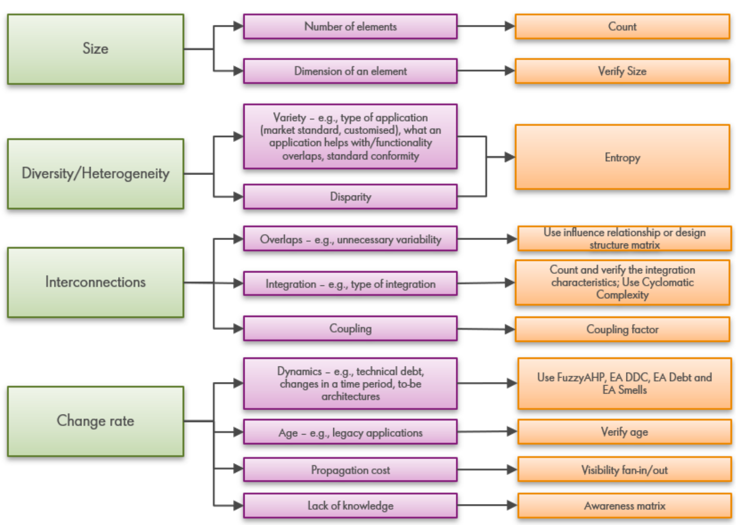

Fig. 3. Drivers and Metrics of IT Landscape Complexity

5 Discussion

This research introduces a conceptual model which unifies different views on what IT landscape components are. This model has been designed to support architects in aligning their views when modeling and communicating about IT landscapes. An important aspect brought forward with the help of the model is the emphasis on the characteristics of applications. Hence, this model can serve as a reference for the elements part of an IT landscape.

Organisations face challenges in managing and measuring the IT landscape complexity. With a defined and aligned view of what constitutes this landscape, as presented in Figure 2, an understanding of the root causes of complexity can start to form. The research also provides a model of drivers and metrics for IT landscape complexity, in Figure 3. IT architects can apply these to quantify landscape complexity for architectures aligned with the conceptual model.

By analysing EA designs of IT landscapes, architects can quantify complexity drivers using metrics. For example, in a model depicting applications, services, data, and interfaces, architects can count the number of services realised by an application, check the types of interfaces between the applications or observe the data usage and exchange. It is envisioned that this approach allows the aggregation of these metrics into an overall complexity metric for the IT landscape.

One use case for the approach can be the following: an architect comparing various future options for the current landscape, i.e., different target architectures, and making a choice based on the score of complexity. A certain viewpoint depicted by an architect will be assigned a complexity score. Here, one can observe if a higher level of complexity is linked with certain architectural patterns or changes. A possible way to calculate complexity for this specific use case (comparing as-is and to-be architectures) is shown in Equation 1, as introduced by Daoudi [10]. Complexity should be seen here as the state changes of the landscape over time.

Considering that adaptation is essential in highly competitive and dynamic business environments, the equation proposed by Daoudi et al. targets proactive sensing and responding to change. In moving from the baseline to the target scenarios, the EA Degree of Dynamic Complexity (DDC) computes the complexity degree of the changes, where i is the indicator of the EA version (i is the current architecture, $i+1$ is a possible target architecture), the first element of the sum is concerned with the values of dynamic factors (changing in time), whereas the second one is related to static factors. N represents the number of factors selected for the analysis.

$$\text{DDC i, i+1(t)}=\sum_{j=1}^{n} \frac{fj(t)+fj}{n}, \text{ where } n \in N \tag{1}$$

Based on the work performed for this research, some further research should be considered as this will bring benefits to both academia and practice. First, the models proposed have to be fully validated in organisations with a high complexity level, e.g., more than 1.000 (preferably integrated) applications in the IT landscape. In such a landscape, one can observe if there are any elements of an IT landscape which are not included in the model. Moreover, drivers of complexity can emerge which might not be discussed in detail in the literature.

One of the researchers is working on creating an aggregated metric for the complexity of an IT landscape, piecing together insights from academia and from a case study. As the metric is being developed, the connection between the IT landscape components and the drivers and metrics influencing complexity needs to be exploited.

6 Conclusions

Starting by creating a conceptual model of the most commonly discussed elements which form the core of the IT landscape and going into more detail on which drivers and metrics could be used for the quantification of complexity, this paper aims at summarising prevalent works for supporting architects in managing complexity.

A conceptual model of IT landscape components is proposed based on a mixed-method literature review. The main contribution of this paper is therefore the provision of a standardised description of the IT landscape. Furthermore, the research identifies complexity drivers and metrics and proposes a method for measuring IT landscape complexity. This method's overarching goal is to quantify IT landscape complexity.

This approach establishes a foundation for the management of IT landscape complexity, for both academics and practitioners. A standard terminology for the IT landscape, or a common language, is essential for mastering complexity. Furthermore, by using identified complexity drivers and metrics, enterprises can prioritise and combine these to indicate their respective complexity levels. With such an aggregation, the lack of measurements for complexity can be tackled, as an organisation can find ways in which they can quantify the metrics based on their existing resources. Hence, the model and metrics overview assist enterprises in mastering complexity.

The model and an initial measurement method are under a formal validation with IT experts and the preliminary results show a satisfactory level to address the goal of this research. However, further validation is needed through a set of scenarios within organisations characterised by IT landscape complexity. Each scenario should include a baseline architecture along with potential target architectures that can be inputted into the measurement method, and, preferably, the result of the complexity calculations should be well explained to the IT experts.

As one of the core future avenues of research, improving the semantic expressiveness of the conceptual model can bring additional value to the overall framework. Therefore, making an ontological analysis of this conceptual model based on a foundational ontology might result in such expressiveness gain. In particular, a proposal is made to use existing well-founded ontologies that target system modelling, such as [7], to address this open issue.

References

1. Adomavicius, G., Bockstedt, J.C., Gupta, A., Kauffman, R.J.: Making Sense of Technology Trends in the Information Technology Landscape: A Design Science Approach. MIS Q. **32**, 779–809 (2008), https://api.semanticscholar.org/CorpusID:10592597
2. Arnold, R.D., Wade, J.P.: A Definition of Systems Thinking: A Systems Approach. Procedia Computer Science **44**, 669–678 (2015). https://doi.org/10.1016/j.procs.2015.03.050, https://linkinghub.elsevier.com/retrieve/pii/S1877050915002860
3. Atchison, L.: Overcoming IT complexity: simplify operations, enable innovation, and cultivate successful cloud outcomes. O'Reilly, Cambridge (2023), oCLC: on1351696495
4. Beese, J., Aier, S., Haki, K., Khosroshahi, P.A.: DRIVERS AND EFFECTS OF INFORMATION SYSTEMS ARCHITECTURE COMPLEXITY: A MIXED-METHODS STUDY. Research Papers (Jun 2016), https://aisel.aisnet.org/ecis2016_rp/74
5. Beetz, K., Kolbe, L.: Towards Managing IT Complexity: An IT Governance Framework to Measure Business-IT Responsibility Sharing and Structural IT Organization. In: 17th Americas Conference on Information Systems 2011, AMCIS 2011. vol. 1 (Jan 2011)
6. Brückmann, M., Schöne, K.M., Junginger, S., Boudinova, D.: Evaluating Enterprise Architecture Management Initiatives – How to Measure and Control the Degree of standardization of an IT Landscape (2009)
7. Calhau, R.F., Prince Sales, T., Oliveira, , Kokkula, S., Ferreira Pires, L., Cameron, D., Guizzardi, G., Almeida, J.P.A.: A System Core Ontology for Capability Emergence Modeling. In: Proper, H.A., Pufahl, L., Karastoyanova, D., Van Sinderen, M., Moreira, J. (eds.) Enterprise Design, Operations, and Computing, vol. 14367, pp. 3–20. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-46587-1_1, https://link.springer.com/10.1007/978-3-031-46587-1_1
8. Daoudi, S., Larsson, M., KTH Royal Institute of Technology, Stockholm, Sweden, Hacks, S., Stockholm University, Stockholm, Sweden, Jung, J., Frankfurt University of Applied Sciences, Frankfurt am Main, Germany: Discovering and Assessing Enterprise Architecture Debts. Complex Systems Informatics and Modeling Quarterly (35), 1–29 (Jul 2023). https://doi.org/10.7250/csimq.2023-35.01, https://csimq-journals.rtu.lv/article/view/csimq.2023-35.01
9. Daoudi, W., Doumi, K., Kjiri, L.: Adaptive Enterprise Architecture: Complexity Metrics in a Mixed Evaluation Method. In: Filipe, J., Śmia\lek, M., Brodsky, A., Hammoudi, S. (eds.) Enterprise Information Systems. pp. 505–523. Springer International Publishing, Cham (2022)
10. Daoudi, W., Doumi, K., Kjiri, L.: Adaptive Enterprise Architecture: Complexity Metrics in a Mixed Evaluation Method. In: Filipe, J., Śmia\lek, M., Brodsky, A., Hammoudi, S. (eds.) Enterprise Information Systems. pp. 505–523. Springer International Publishing, Cham (2022)
11. González-Rojas, O., López, A., Correal, D.: Multilevel complexity measurement in enterprise architecture models. International Journal of Computer Integrated Manufacturing **30**(12), 1280–1300 (Dec 2017). https://doi.org/10.1080/0951192X.2017.1307453, https://www.tandfonline.com/doi/full/10.1080/0951192X.2017.1307453

12. Group, T.O.: TOGAF® Standard, https://pubs.opengroup.org/togaf-standard/adm/chap01.html
13. Group, T.O.: ArchiMate® 3.1 Specification (2019), https://pubs.opengroup.org/architecture/archimate31-doc/toc.html
14. Hacks, S., Hofert, H., Salentin, J., Yeong, Y.C., Lichter, H.: Towards the Definition of Enterprise Architecture Debts. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW). pp. 9–16. IEEE, Paris, France (Oct 2019). https://doi.org/10.1109/EDOCW.2019.00016, https://ieeexplore.ieee.org/document/8907262/
15. Hammer, D.: The many aspects of an IT-architecture. In: Proceedings International Conference and Workshop on Engineering of Computer-Based Systems. pp. 304–311. IEEE Computer. Soc. Press, Monterey, CA, USA (1997). https://doi.org/10.1109/ECBS.1997.581891, http://ieeexplore.ieee.org/document/581891/
16. Hanschke, I.: IT Landscape Management. In: Strategic IT Management: A Toolkit for Enterprise Architecture Management, pp. 105–217. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-05034-3_4, https://doi.org/10.1007/978-3-642-05034-3_4
17. Holub, I.: Methodology for Measuring the Complexity of Enterprise Information Systems. Journal of Systems Integration **7**, 34–53 (Jul 2016). https://doi.org/10.20470/jsi.v7i3.260
18. Iacob, M., Jonkers, H., Quartel, D., Franken, H., van den Berg, H.: Delivering Enterprise Architecture with TOGAF and ArchiMate. Enschede BiZZdesign 2012 (2012), oCLC: 1073562791
19. Iacob, M.E., Monteban, J., Van Sinderen, M., Hegeman, E., Bitaraf, K.: Measuring Enterprise Architecture Complexity. In: 2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop (EDOCW). pp. 115–124. IEEE, Stockholm (Oct 2018). https://doi.org/10.1109/EDOCW.2018.00026, https://ieeexplore.ieee.org/document/8536112/
20. ISO: ISO/IEC/IEEE 42010:2022 (Nov 2022), https://www.iso.org/standard/74393.html
21. Jain, R., Chandrasekaran, A., Elias, G., Cloutier, R.: Exploring the Impact of Systems Architecture and Systems Requirements on Systems Integration Complexity. IEEE Systems Journal **2**(2), 209–223 (Jun 2008). https://doi.org/10.1109/JSYST.2008.924130, http://ieeexplore.ieee.org/document/4539770/
22. Kruchten, P., Nord, R.L., Ozkaya, I.: Technical Debt: From Metaphor to Theory and Practice. IEEE Softw. **29**(6), 18–21 (Nov 2012). https://doi.org/10.1109/MS.2012.167, https://doi.org/10.1109/MS.2012.167, place: Washington, DC, USA Publisher: IEEE Computer Society Press
23. Laan, S.: IT Infrastructure Architecture - Infrastructure Building Blocks and Concepts. Lulu.com, 2nd edn. (2013)
24. Lankhorst, M.: Enterprise Architecture at Work. The Enterprise Engineering Series, Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-29651-2, http://link.springer.com/10.1007/978-3-642-29651-2
25. Lentz, J.L., Bleizeffer, T.M.: IT ecosystems: evolved complexity and unintelligent design. In: Proceedings of the 2007 symposium on Computer human interaction for the management of information technology. p. 6. ACM, Cambridge Massachusetts (Mar 2007). https://doi.org/10.1145/1234772.1234780, https://dl.acm.org/doi/10.1145/1234772.1234780

26. Manzur, L., Ulloa, J.M., Sánchez, M., Villalobos, J.: xArchiMate: Enterprise Architecture simulation, experimentation and analysis. SIMULATION **91**(3), 276–301 (Mar 2015). https://doi.org/10.1177/0037549715575188, http://journals.sagepub.com/doi/10.1177/0037549715575188
27. Marshall, A., Winkler, U., Gilani, W.: Business Continuity Management of Business Driven IT Landscapes (2011), https://api.semanticscholar.org/CorpusID:168205316
28. Martini, A., Bosch, J.: The Danger of Architectural Technical Debt: Contagious Debt and Vicious Circles. In: 2015 12th Working IEEE/IFIP Conference on Software Architecture. pp. 1–10. IEEE, Montreal, QC, Canada (May 2015). https://doi.org/10.1109/WICSA.2015.31, http://ieeexplore.ieee.org/document/7158498/
29. Martini, A., Bosch, J., Chaudron, M.: Architecture Technical Debt: Understanding Causes and a Qualitative Model. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications. pp. 85–92. IEEE, Verona, Italy (Aug 2014). https://doi.org/10.1109/SEAA.2014.65, http://ieeexplore.ieee.org/document/6928795/
30. Matook, S., Brown, S.A.: Characteristics of IT artifacts: a systems thinking-based framework for delineating and theorizing IT artifacts. Information Systems Journal **27**(3), 309–346 (May 2017). https://doi.org/10.1111/isj.12108, https://onlinelibrary.wiley.com/doi/10.1111/isj.12108
31. Mocker, M.: 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 (Jan 2009). https://doi.org/10.1109/HICSS.2009.506, https://ieeexplore.ieee.org/document/4755701, iSSN: 1530-1605
32. Onderdelinden, E., Hooff, B.v.d., Vliet, M.v.: IS Architecture Complexity Dynamics in M&A: Does Consolidation Reduce Complexity? ECIS 2023 Research Papers (May 2023), https://aisel.aisnet.org/ecis2023_rp/377
33. Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D.: The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. PLOS Medicine **18**(3), e1003583 (Mar 2021). https://doi.org/10.1371/journal.pmed.1003583, https://dx.plos.org/10.1371/journal.pmed.1003583
34. Perks, C., Beveridge, T. (eds.): Guide to Enterprise IT Architecture. Springer Professional Computing, Springer New York, New York, NY (2004). https://doi.org/10.1007/b98880, http://link.springer.com/10.1007/b98880
35. Rohloff, M.: Business Oriented Development of the IT Landscape: Architecture design on a Large Scale. In: Managing Worldwide Operations and Communications with Information Technology (2007)
36. Ross, J., Beath, C., Mocker, M.: Creating digital offerings customers will buy : find the sweet spot between what technologies can deliver and what your customers need (Jan 2019)
37. Sanden, W.v.d., Sturm, B., Intro (Amsterdam), Panfox Holding.: Informatiearchitectuur: de infrastructurele benadering. Panfox, Rosmalen, 2e [herz.] dr. edn. (2000), oCLC: 67354153

38. Schmidt, C.: Business Architecture Quantified: How to Measure Business Complexity. In: Simon, D., Schmidt, C. (eds.) Business Architecture Management, pp. 243–268. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-14571-6$_1$3, https://link.springer.com/10.1007/978-3-319-14571-6_13
39. Schneider, A., Zec, M., Matthes, F.: Adopting Notions of Complexity for Enterprise Architecture Management. In: 20th Americas Conference on Information Systems, AMCIS 2014 (Aug 2014)
40. Schneider, A.W., Reschenhofer, T., Schutz, A., Matthes, F.: Empirical Results for Application Landscape Complexity. In: 2015 48th Hawaii International Conference on System Sciences. pp. 4079–4088. IEEE, HI, USA (Jan 2015). https://doi.org/10.1109/HICSS.2015.490, http://ieeexplore.ieee.org/document/7070309/
41. Schuetz, A., Widjaja, T., Gregory, R.: Escape from Winchester Mansion – Toward a Set of Design Principles to Master Complexity in IT Architectures. In: International Conference on Information Systems (ICIS 2013): Reshaping Society Through Information Systems Design. vol. 2 (Dec 2013)
42. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., Paige, R.: Large-scale complex IT systems. Communications of the ACM **55**(7), 71–77 (Jul 2012). https://doi.org/10.1145/2209249.2209268, https://dl.acm.org/doi/10.1145/2209249.2209268
43. Storey, V.C., Kaul, M., Woo, C.: A Framework for Managing Complexity in Information Systems. J. Database Manage. **28**(1), 31–42 (Jan 2017). https://doi.org/10.4018/JDM.2017010103, https://doi.org/10.4018/JDM.2017010103, place: USA Publisher: IGI Global
44. Tieu, B., Hacks, S.: Determining Enterprise Architecture Smells from Software Architecture Smells. In: 2021 IEEE 23rd Conference on Business Informatics (CBI). pp. 134–142. IEEE, Bolzano, Italy (Sep 2021). https://doi.org/10.1109/CBI52690.2021.10064, https://ieeexplore.ieee.org/document/9610644/
45. Van Der Torre, L., Lankhorst, M.M., Ter Doest, H., Campschroer, J.T.P., Arbab, F.: Landscape Maps for Enterprise Architectures. In: King, R. (ed.) Active Flow and Combustion Control 2018, vol. 141, pp. 351–366. Springer International Publishing, Cham (2006). https://doi.org/10.1007/11767138$_2$4, http://link.springer.com/10.1007/11767138_24
46. Verdecchia, R., Kruchten, P., Lago, P., Malavolta, I.: Building and evaluating a theory of architectural technical debt in software-intensive systems. Journal of Systems and Software **176**, 110925 (Jun 2021). https://doi.org/10.1016/j.jss.2021.110925, https://www.sciencedirect.com/science/article/pii/S0164121221000224
47. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly **26**(2), xiii–xxiii (2002), https://www.jstor.org/stable/4132319
48. Wehling, K., Schaefer, I.: Towards an Expert System for Identifying and Reducing Unnecessary Complexity of IT Architectures (2017). https://doi.org/10.18420/IN2017$_1$52, https://dl.gi.de/handle/20.500.12116/3917
49. Wehling, K., Wille, D., Seidl, C., Schaefer, I.: Decision Support for Reducing Unnecessary IT Complexity of Application Architectures. In: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW). pp. 161–168 (2017). https://doi.org/10.1109/ICSAW.2017.47

50. Widjaja, T., Gregory, R., University of Virginia, USA: Monitoring the Complexity of IT Architectures: Design Principles and an IT Artifact. Journal of the Association for Information Systems **21**(3), 664–694 (May 2020). https://doi.org/10.17705/1jais.00616, https://aisel.aisnet.org/jais/vol21/iss3/4/
51. Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.M.: Using grounded theory as a method for rigorously reviewing literature. European Journal of Information Systems **22**(1), 45–55 (Jan 2013). https://doi.org/10.1057/ejis.2011.51, https://www.tandfonline.com/doi/full/10.1057/ejis.2011.51
52. Zachman, J.A.: A framework for information systems architecture. IBM Systems Journal **26**(3), 276–292 (1987). https://doi.org/10.1147/sj.263.0276, http://ieeexplore.ieee.org/document/5387671/
53. Zazworka, N., Seaman, C., Shull, F.: Prioritizing design debt investment opportunities. In: Proceedings of the 2nd Workshop on Managing Technical Debt. pp. 39–42. ACM, Waikiki, Honolulu HI USA (May 2011). https://doi.org/10.1145/1985362.1985372, https://dl.acm.org/doi/10.1145/1985362.1985372