Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 112, Issue 19, June/July 2019, Pages 60–74
https://doi.org/10.7250/csimq.2019-19.04

Discussing Hierarchic Viewpoints Theoretical Considerations and a Practical Example

Dierk Jugel[*], Christian M. Schweda, and Alfred Zimmermann

Herman Hollerith Zentrum, Reutlingen University, Danziger Str. 6, 71034 Böblingen, Germany

dierk.jugel@hhz.de, christian.schweda@hhz.de, alfred.zimmermann@reutlingen-university.de

Abstract. Enterprise Governance, Risk and Compliance (GRC) systems are key to managing risks threatening modern enterprises from many different angles. Key constituent to GRC systems is the definition of Controls that are implemented on the different layers of an Enterprise Architecture (EA). Controls become part of a "Concern" of the EA, which allows to use an EA viewpoint to cover Control compliance assessments. In this article we explore this relationship further, derive a metamodel linking Control and EA, and elicit how this linkage give rise to a hierarchic understanding of the viewpoint concept for EAs. We complement these considerations with an expository instantiation in a cockpit for Control compliance applied in an international enterprise in the insurance industry.

Keywords: Governance, Compliance, Control, Enterprise Architecture, Cockpit, Viewpoint, Concern.

1 Introduction

Modern enterprises face threats that originate from different sources. Different varieties of cyber security attacks are on the rise, as recent analyses of the threat landscape show [1]. In addition to cyber security-related threats, environmental factors also pose a risk to modern enterprises operating on a global scale. Architectural risks result from the current set-up of the enterprise and its supporting IT. Finally, legal risks arise from the variety of laws and regulations originating from different sources. Regulations differ in target audiences, with the General Data Privacy Regulation (GDPR) [2] with a broad target and the supervisory requirements for IT in insurance undertakings [3] with a narrow target issued by the German regulator for the financial industry. The Enterprise Governance, Risk and Compliance (GRC) [4] system is established in modern enterprises to diligently handle aforementioned types of risks via cultural,

[*] Corresponding author

Reference: D. Jugel, C. M. Schweda, and A. Zimmermann, "Discussing Hierarchic Viewpoints Theoretical Considerations and a Practical Example," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 19, pp. 60–74, 2019. Available: https://doi.org/10.7250/csimq.2019-19.04

Additional information. Author's ORCID iD: A. Zimmermann – https://orcid.org/0000-0003-3352-7207. PII S225599221900112X. Received: 8 February 2019. Accepted: 20 June 2019. Available online: 19 July 2019.

organizational, procedural or technical means, so-called Controls. Objectives of these controls [5] are to

- avoid a risk by changes in the business model;
- reduce the probability of a risk; or to
- limit the impact of a risk.

All types of risks lead to Controls that operationalize the externally imposed rules and regulations. The Controls are implemented into different ‘elements’ of the enterprise, e.g.

- additional checks within business processes,
- additional logic within business applications, or
- additional components within the technical infrastructure.

In this sense, a control is implemented in the enterprise as a whole or a relevant element within the Enterprise Architecture (EA). In the GRC system the Controls are not only designed, but the enterprise is regularly assessed with respect to compliance with and effectiveness of these Controls. In larger enterprises, these assessments are conducted on different levels: detailed for subject matter experts and aggregated to provide high-level indications for the senior management. Translating to the terminology of architecture modeling, these different levels of assessments are different concerns and viewpoints on the EA. The relationship between different controls and the resulting viewpoints give raise to a research question on the nature of concerns, viewpoints and views:

How can hierarchies of controls be reflected as GRC-related concerns and viewpoints in an architecture description?

Preparing our considerations on the research question, we relate our work to the foundations of GRC, control modeling and control assessments in Section 2 to provide context for the subsequent considerations on GRC concerns. Section 3 revisits our previously presented approach to integrate control objectives into EA as discussed in [6]. This approach provides a metamodel to model controls and control assessments, from which we derive a characteristics of GRC-related concerns in architecture descriptions. The key characteristic employing hierarchies is developed into a theory of hierarchic concerns, viewpoints and views exhibited in Section 4. In Section 5, we apply the theory in a practical example to show its utility. Section 6 summarizes how this article’s key assumptions address the research question and points to future research regarding a more formal perspective on concerns and viewpoints advocated by key assumptions.

2 Related Work

In this Section we introduce approaches that are relevant for our work. First we present approaches to the topic “EA viewpoints”, before we go into GRC approaches.

2.1 EA Viewpoints

The Open Group Architecture Framework (TOGAF) [7] defines a process for developing enterprise architectures as part of the so-called Architecture Development Method (ADM). The ADM process consists the design, planning, implementation, and maintenance of enterprise architectures. While TOGAF identifies key concepts for modeling an enterprise architecture, it does not include a modeling language. To close this gap, the modeling language ArchiMate [8] was developed. In ArchiMate, enterprise architectures can be broken down into business layer, application layer, technology layer and the further extensions focusing on strategy, motivation and implementation aspects. In order to select the appropriate standard viewpoint for the respective situation, ArchiMate also provides a classification scheme. The basic idea of viewpoints originates from the ISO Std. 42010 [9].

The ISO Std. 42010 [9] describes a systematic approach for creating architectural descriptions of systems. A system is an entity whose architecture is of interest. Due to the very generic definition of a system and the fact that different systems are used in different application domains, this approach can be transferred to other domains in addition to software engineering. For instance, a system can be understood as a software or an enterprise whose enterprise architecture corresponds to the concept of an architecture in the sense of the standard. For this reason, the standard is increasingly being used and adapted in enterprise architecture management [7], [8]. A weakness of the standard is the very abstract description of the individual concepts, which opens up a wide range of interpretation and leaves some questions unanswered [10].

Lankes, Matthes and Wittenburg [11] introduce a layer concept to the view of ISO Std. 42010. According to this approach, a view consists of different layers that are related to each other. The so-called base map represents the bottom of the map and is a special layer that is the basis for all other layers. This makes it possible to overlay visual presets. For instance, it is possible to specify at a layer which symbol is used to represent a specific element of the architecture at which position and size. Another layer can refer to it and extend the symbol with an additional visual property in the form of a background color.

Like Lankes et al. [11], Jugel [10], [12] also aims to concretize the ISO standard. For this purpose he presents an extended conceptualization and details the modeling language of a viewpoint, which is the counterpart to layers introduced by Lankes et al. [11] on the view side. Lankes et al. [11] leave open the question of how the individual layers are constructed by a viewpoint. According to Jugel [10], a viewpoint consists of a base language which describes the basic structure of a view corresponding to the base map from [11]. The base language of a view defined by a viewpoint can be extended by using so-called techniques. A technique describes a model-based approach to support stakeholders in doing a particular task. Depending on the degree of formalization, techniques are executed manually or automatically. An example of a technique is the calculation of a key performance indicator (KPI) for elements of the enterprise architecture. In addition to the actual calculation, the technique also displays the results as a layer in views. The execution of a technique therefore leads to an additional layer in the corresponding view.

In [12], Jugel goes one step further and introduces the so-called Architecture Cockpit. This approach puts different views side-by-side, whereby dependencies between views can be calculated and displayed.

2.2 Governance, Risk and Compliance

COBIT [13] provides a comprehensive framework for governance and management of support for Enterprise IT. In this context, IT-relevant goals of internal and external stakeholders are considered. COBIT provides a process framework complemented by internationally accepted IT process-related requirements. COBIT is based on five basic principles to ensure optimal value of IT. The key principles are the distinction between governance and management, the comprehensive, holistic approach, and the coverage of the entire enterprise. In the process model, governance processes take top priority. These processes set policies and monitor their compliance. The section below deals with management processes which deal with planning, procurement and implementation. These processes are monitored by other management processes and assessed against the given governance guidelines. These monitoring processes are related to performance and compliance, internal control and compliance with external requirements.

The ISO 2700x series considers Controls with the focus on information security. ISO 27001 [14] delineates requirements for the evaluation and treatment of information security risks tailored to the needs of businesses. It provides a framework for developing and maintaining an effective information security management system (ISMS). It will provide IT protection goals in terms of confidentiality, integrity and availability of information. An ISMS in practice consists of the governance view(-point), the risk view and the compliance view. These viewpoints are employed to determine the protective measures considering the different concerns of the enterprise's stakeholders. The governance perspective relates to the implementation and adherence to objectives; the risk perspective – on the identification, assessment and treatment of risks; and the compliance perspective – on the compliance with regulatory, contractual and legal requirements.

The MEMO approach [15] for enterprise modeling considers GRC as a key topic. MEMO addresses different stakeholders of and concerns with respect to the enterprise via an integrated set of modeling languages that cover the concerns and are based on one meta-language. MEMO ControlML [16] provides support to stakeholders in the effective and efficient conduct of an assessment of the internal control system and the surrounding organizational action system. The core of ControlML [16] is the Control Objective, which is a desirable condition for achieving an endangered business objective. Control Objectives can be derived from business goals and aggregated. They also determine Reference Objects that are objects to be controlled according the objective. The Reference Object is an abstract concept and represents any concept to describe an enterprise (e.g. business units, applications or technologies). ControlML is complemented by MEMO MetricML [17], which focuses on assessing controls. The Indicator concept described in MEMO MetricML defines the configuration of the indicator as well as the measured values and the date of measurement. The configuration consists of an algorithm for translating attribute's values of Reference Objects into a measurement and the frequency of calculation. The ControlML and MetricML can be combined to cover Control design and Control assessment related in an enterprise model.

Summarizing above considerations, controls, as part of GRC, can be considered a cross-cutting aspect targeting different elements of an EA. Such aspect in turn needs, in line with the presented approaches to EA management, be reflected in the underlying metamodel of a GRC-enabled EA management.

3 A Metamodel for Control Objectives

GRC systems are centered around the concept of the *ControlObjective*, which reflects a regulatory or compliance obligation to be fulfilled in the EA or a part thereof. In [6] we discussed how control compliance and the associated concept of the *ControlObjective* can be reflected in an EA metamodel. The metamodel draws from selected approaches from literature – foremost ControlML [16] and MetricML [17] – adapted to the EA context. The key concepts of the metamodel are introduced in Figure 1 and subsequently detailed.

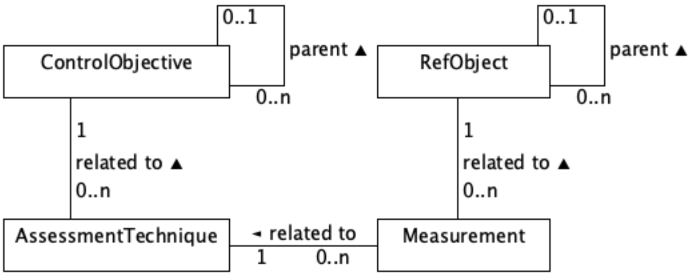


Figure 1. Metamodel to link ControlObjectives to EA Objects [6]

ControlObjectives – adapted from [16] – represent the functional objectives to control guidelines or regulations. Abstract specifications are operationalized into concrete, architecture-related objectives. *ControlObjectives* define what to control, but not how to assess their effectiveness.

The *AssessmentTechnique* – adapted from Indicator as presented in [17] – designates the procedure of assessing and of interpreting the results in terms of 'good' and 'bad'. The Indicator from [17] both represents the assessment process and the result thereof – represented in our case by the concept *Measurement*. The *AssessmentTechnique* also defines necessary calculations, intervals of measurement and thresholds for various effectiveness levels to derive a 'score'. In our setting the scores range from 'very good' to 'very poor' with an extra score for missing values.

In [6] we discuss the hierarchy of *ControlObjectives* – contrasting the assumptions from [16] – needed to reflect the concerns of the stakeholders. This hierarchy allows to group multiple *AssessmentTechniques* into a single high-level *ControlObjective*, which aligns with the notion of the ISO Std. 42010 [9], which names the concept of the technique as a means to address concerns and facilitate decision making.

Each *Measurement* is determined with respect to an element of the EA which is controlled by the corresponding *ControlObjective*. This element of the EA is represented by the concept *RefObject* – adapted from ReferenceObject [16]. Examples of *RefObject*s are operating entities (OEs) or business processes. A *Measurement* is unique for a given combination of *RefObject* and *AssessmentTechnique* at a given point in time. Different time-stamped Measurements may nevertheless exist for different points in time.

The hierarchy of *ControlObjectives* yields additional implications on the *AssessmentTechniques*, which can accordingly be distinguished by the way their corresponding *Measurements* are determined. In particular for grouped high-level *ControlObjective*s no direct assessments may exist, but their results may be derived from more granular *Measurement*s. In line with this we distinguish two types of *AssessmentTechnique*s:

- *DirectAssessmentTechnique*s acquire results by self-assessments or using technical tools for measuring.
- *DerivedAssessmentTechnique*s calculate results based on the results of already performed assessments. For such techniques the individual rules of calculation, e.g. using minimum rule, are specified.

The metamodel from [6] reflects this by sub-typing *AssessmentTechnique* (see Figure 2).

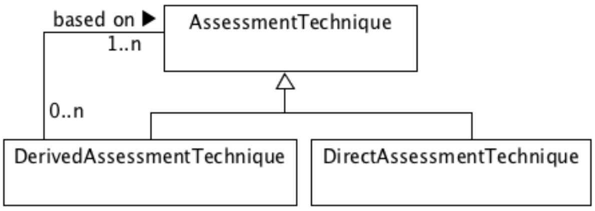


Figure 2. Specializing AssessmentTechniques [6]

A kind of parameterization of *ControlObjectives* is discussed in [6] with respect to the area of interest within IT. A concern as standardization can be considered broadly, i.e. covering the whole EA, but – as described along the example – can also be narrowed down to a specific part of the EA. This yields a differentiation of types of *ControlObjectives* as follows:

- A *DirectControlObjective* targets the EA as a whole.
- A *TypedControlObjective* is dependent on *EAObject* that reflects the facet under consideration. This *EAObject* is an instance of a previously determined type, e.g. IT Domain.

The metamodel introduced by us in [6] reflects this distinction by sub-typing *ControlObjective* (see Figure 3).

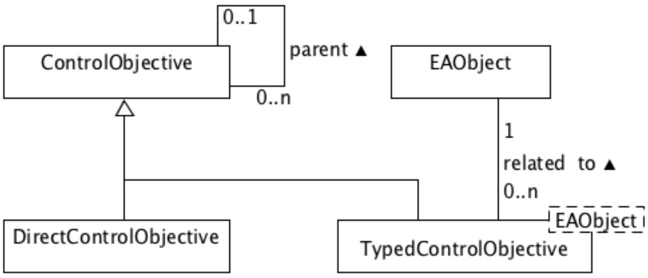


Figure 3. Specializing ControlObjectives [6]

ControlObjectives in the sense of a GRC system can be considered to represent ‘concerns’ that a stakeholder has with respect to the EA or a part thereof. The notion of the concern therein is consistent with the notion of concern as expressed by the ISO Std. 42010 [9]. One key exception to that is, that concerns in the ISO Std. are not considered hierarchical, whereas the *ControlObjectives* shown in the metamodels above, are considered hierarchical. This gives raise to the idea of concern being a hierarchic concept – an assumption, for which we seek additional evidence from research and practical application in the following Sections 4 and 5.

4 The Notion of Concerns, Viewpoints and Views

The ISO Std. 42010 [9] provides a framework for architectural descriptions that allows to relate stakeholders and their concerns to architecture viewpoints and corresponding views. It further discusses both the nature of view and viewpoint, and identifies the relationship between these two as the relationship between “a map” and “a legend” [9]. The viewpoint provides the conventions for creating and interpreting a corresponding view. A view in the terminology of the ISO Std. 42010 [9] is composed of one or more architecture models, which are abstractions of the architecture useful to answer questions about the architecture, i.e. reflecting qualities of the system under consideration. The construction of the model in turn is guided by a model kind, which provides meta-model, modeling language or template to create a model. The relationships between model and model kind again resembles the one between “a map” and “a legend”. This yields a type-instance relationship between viewpoint and model kind (both types) on the one, and view and model (both instances) on the other hand, as displayed in Figure 4.

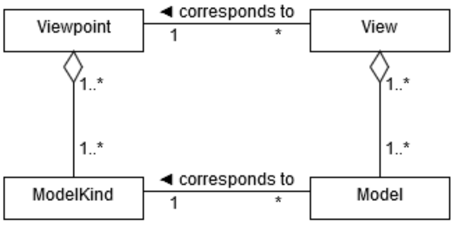


Figure 4. Basic structure of Viewpoint and View according to [9]

The work of Lankes, Matthes and Wittenburg [11] sparks some discussions around the true nature of the differentiation between views and models. They adopted the layering concept from conventional cartography into architecture descriptions. This concept allows to have a model to be enriched with additional information by adding another layers of symbols or a different color-coding to the already existing graphical work product. In this sense, a model can be re-used in another model – a kind of relationships that, according to the ISO Std. 42010 [9], exists between model and view. Conversely, also the differentiation between model kind and architecture viewpoint is not that clear in the light of the layering principle, that allows to re-use a model kind as part of another model kind and adding and additional layer. Figure 5 shows these compositional relationships of layered models and model kinds put in the context of the ISO Std. 42010 [9].

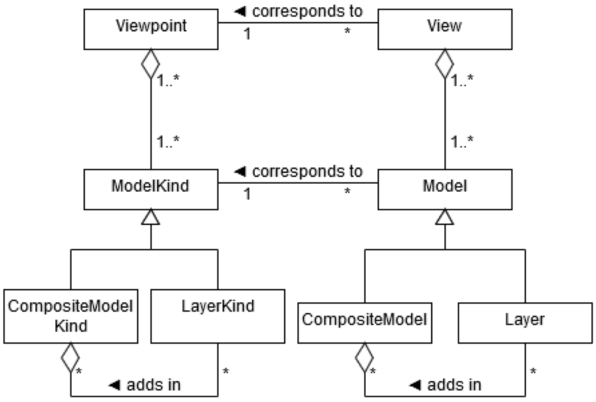


Figure 5. Compositional relationships of layered models and model kinds in context of [9]

One might argue that the differentiation might be still maintained as architecture viewpoints and views are – according to the ISO Std. 42010 [9] – bound to cover the whole architecture under consideration, whereas the layered models [11] are not subject to that restriction. The application of architecture cockpit concept of Jugel [12] gives rise to a more inductive definition of architecture view. A cockpit displays architecture models side by side and can be configured to be an architecture view, covering the whole architecture of the system. To any such cockpit view, additional architecture models covering the whole architecture from a different perspective can be added. In such sense, an architecture view represented in a cockpit can be extended by another architecture view, resulting in a cockpit, which is an architecture view in itself. Switching to the side of the cockpit's configuration, the same composite relationship holds, yielding the compositional structure of cockpit viewpoints and views put in the context of the ISO Std. 42010 [9] shown in Figure 6.

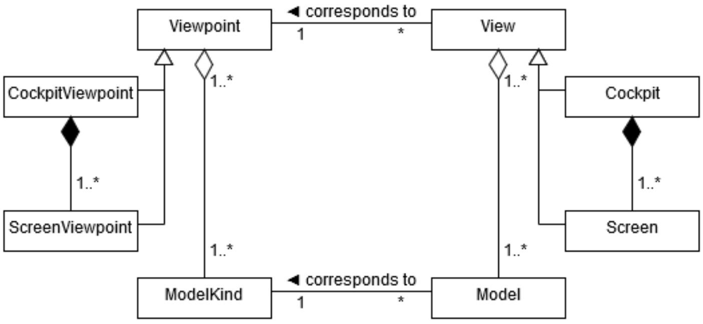


Figure 6. Compositional structure of cockpit viewpoints and views

Both the layering approach and the cockpit approach are built around mechanisms for ensuring consistency between the views and models, respectively. This is achieved by consistency rules designed into the viewpoint composition and model kind composition methods. The compositional nature of the architecture view and the constituting architecture model given rise to the first assumption of the article:

Hierarchy of views*: a view can be composed from one or more (mutually consistent) views.*

The hierarchy of views in turn has implications on the nature of the viewpoint, also visible along the layering principle and even more prominently with the cockpit concept. The configuration of the cockpit leading the visible view can be in line with the definition of the ISO Std. 42010 [9] considered to represent an architecture viewpoint. Removing a view from the cockpit leads to another configuration being a “prefix” of the preceding one. At the same time, the new configuration can be considered to represent a new viewpoint being a “prefix” or sub-viewpoint of the first one. As long as removal retains at least one view, each removal step yields another “smaller” viewpoint. Conversely, the configuration of the cockpit can be extended by configuring the creation of another view thereby also extending the viewpoint. This observation gives rise to the second assumption of the article:

Hierarchy of viewpoints*: a viewpoint can be composed from one or more (mutually consistent) viewpoints.*

The second assumption is also backed by the authors of [18], who establish the correspondence between viewpoints and modeling languages, showing that modeling languages can be considered to be part of larger modeling languages. In addition, [18] discusses the area-of-interest (in a system under consideration) that shapes what they consider to be the core of a concern. More precisely, they identify a concern with a set of characteristics of the architecture of the system that are of interest for a particular group of people. From that, they derive that an area-of-interest (within a concern) can be extended by adding additional characteristics of the system to be covered. This leads to the third assumption of our article, as follows:

Hierarchy of concerns*: a concern can be composed from one or more (mutually consistent) concerns.*

This last assumption also reverberates in the usage scenarios of the cockpit, in which the concern initially sparking the architecture discussion is furthered by identifying and involving additional stakeholders that add layers and additional views to the cockpit. In particular the last assumption gives a strong indication to abandon the differentiation between architecture viewpoint and model kind, as well as between architecture view and model, respectively. Instead, a hierarchy relationships for both architecture viewpoints and views can be introduced into the meta-model of the ISO Std. 42010 [9]. This allows to be more strict with the multiplicities on the relationships between concern and viewpoint. The ISO Std. 42010 [9] originally assumes that one or more viewpoints are necessary to address a concern. With the hierarchic viewpoint, any number of necessary viewpoints can always be composed into one “larger” viewpoint, which than conversely addresses the concern as a whole. With that, the relevant part of the meta-model exhibited in the ISO Std. 42010 [9] can be adapted by applying the discussed hierarchies as shown in Figure 7.

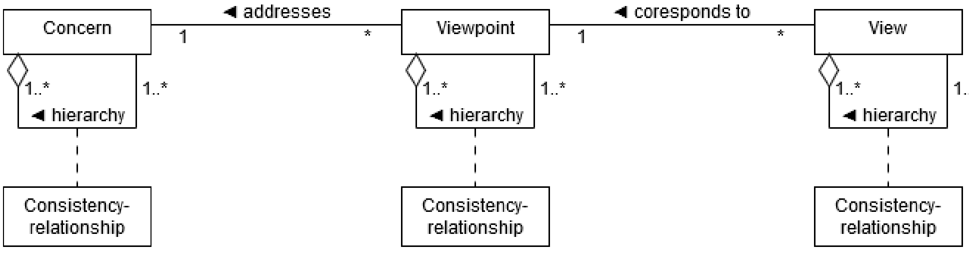


Figure 7. Hierarchic Concerns, Viewpoints and Views

The consistency relationships can be considered from the perspective on viewpoint relationships established by Nuseibeh et al. in [19]. Hierarchic viewpoints in the sense of Figure 7 are *partially overlapping*, i.e. all properties of the architecture covered by the “super”-viewpoint are also covered by the “sub”-viewpoint. The latter viewpoint adds additional detail, providing a lower level of abstraction. Such levels of abstraction are according to Dijkman et al. cf. [20] the reflection of different “positions” in the design or management process, only displaying information that is considered essential at a particular point of design or management.

Applying the same kind of consistency rules is justified by the work of Andrade et al. cf. [21], who identify consistency with the absence of *discrepancies*. Andrade et al. distinguish two types of discrepancies, namely *conflicts* and *inconsistencies*. Whereas inconsistencies derive from an accidental contradiction, a conflict indicates an intentional interference between the concerns addressed by the corresponding viewpoints. In that light two concerns are considered consistent, i.e. being related via an Consistency Relationship, if no conflict between the concerns exist. For the subsequent practical example, we assume that no conflicts of interest exist.

5 A Practical Example From Insurance Industry

We exemplify the concept of the hierarchical concerns, viewpoints and views along a real case study from the insurance industry. The insurance industry and the companies acting in that industry sector are exposed to a variety of risks through the core insurance and asset management activities. These risks are underwriting, operational, strategic but also credit, market, business, liquidity and reputational risk. Internal GRC systems as means to actively govern and manage these risks are therefore prevalent in the insurance industry. We take the perspective of an internationally operating insurance group to derive concerns for our example case reflecting typical stakeholders within the insurance group. The insurance group has a holding structure with over 60 Operating Entities (OEs) represented in more than 70 countries and serving more than 100 million customers. The IT necessary to support the business of the OEs is partly operated by a captive shared service provider, while certain OEs with special situations reserve the right to maintain a local IT. In this context not only efficient and effective but also resilient and above all secure information processing is a key capability for the organization.

5.1 Hierarchic Concerns

The requirements of the key capability, the demands derived from the company's business model and regulatory requirements are translated into harmonized Global Architecture and Global Security Standards which are mandatory for all OEs and governed centrally in the holding. These Standards mirror Controls that are designed specifically to purposefully mitigate the identified risks. In this context different stakeholders raise concerns with respect to the GRC system, subsequently summarized as follows:

Concern 1: Senior management in the holding needs to get an overview of Control compliance and effectiveness throughout the OEs to understand the overall risk exposure of the company and to enter into the planning dialogs with OE senior management resulting in OE-specific target setting.

Concern 2: Subject matter experts (SMEs) in the holding need to understand the status of Control compliance and effectiveness for a specific control area throughout the Group. Such control area may be as broad as a full relevant process, e.g. vulnerability management, but also further narrowed down within the IT, e.g. standardization of database management systems.

The experts use this information to perform 'what-if' analysis, to evolve the Controls, and to get in touch with OE counterparts to derive means of effective implementation.

Concern 3: Senior management of an OE needs to understand the Control compliance and effectiveness in their own OE also compared to the aspiration levels and current levels of assessment as achieved throughout the company. This allows senior management to leverage best-practices from other OEs to improve weak Controls.

Concern 4: Subject matter experts in individual OEs need to understand the defined control objective and their threshold values and see current effects of completed or ongoing measures in order to control the achievement of the specified goals.

Figure 8 shows how the different concerns within the practical example relate to each other. Concern 2 is a sub-concern of Concern 1, focusing on a specific sub-set of controls, but not on the full control set. Similarly, Concern 4 is a sub-concern of Concern 3 with a similar focus on a specific sub-set of controls instead the full control set. From a different angle, Concern 4 is also a sub-concern of Concern 2 even with a focus on the same sub-set of controls, as the scope of considered OEs is reduced from a comprehensive Group level view to a single OE. The same sub-concern logic applies for Concern 3 being a sub-concern of Concern 1.

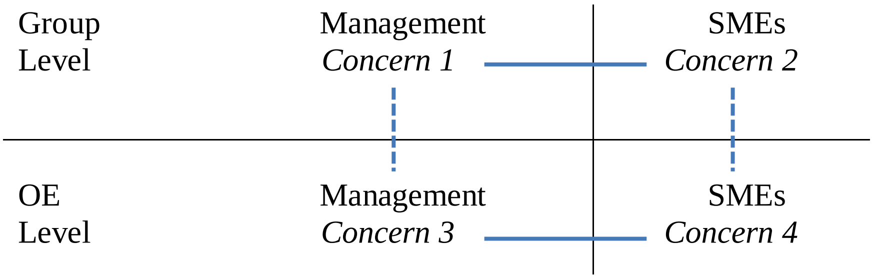


Figure 8. Overview about concerns

5.2 Control Objectives and Assessment Techniques

The concerns of the different stakeholders are focused on compliance with *ControlObjectives* and leverage different *AssessmentTechniques* in combination. Table 1 gives an overview of the combinations employed, precluding their detailed discussion in the following.

Table 1. Combinations of AssessmentTechnique Types and ControlObjective Types

	DirectAssessmentTechnique	*DerivedAssessmentTechnique*
DirectControlObjective	ExtV IntV	CSAE
TypedControlObjective	ITAge ITDebt	ArcDebt

The **number of internet-facing vulnerabilities** (ExtV) provides a number of vulnerabilities exposed via internet-facing IP addresses, taking into account the severity of the vulnerability and the time, for which this vulnerability has been exposed. The **number of internal vulnerabilities** (IntV) provides a corresponding assessment for the vulnerabilities being exposed on IP addresses being available from the internal network. The assessment techniques are direct, based on a technical vulnerability scanner.

IT Ageing and IT Debt relate to structuring concepts of the EA, typing the *ControlObjective* to the ‘areas’, in which the non-compliance is measured. Examples of such structuring elements are different hierarchical types of *ITDomain*s:

- *Infrastructure Domain*s reflect prevalent operating environments for the IT, e.g. data center, workplace and mobile.
- *Technical Domain*s reflect typical use cases for ‘commodity’ IT, e.g. operating system, database management system and application server.

The **IT Ageing** (ITAge) computes the distribution of IT Assets over the releases of a used technology. A ‘left-hanging’ distribution is thereby considered an indication for ageing, a ‘right-hanging’ distribution for actuality of the current IT Asset based with respect to that technology. A technology in turn is assigned to an *ITDomain* reflecting its prevalent operating environment and use case. The **IT Debt** (ITDebt) computes the distribution of IT Assets of Standard to non-standard technologies. The IT Debt is expressed in the amount of money needed to migrate from non-standard technologies to their standard.

The aforementioned *AssessmentTechnique*s are direct in terms of Section 3, i.e. their measurements are results of direct assessment. Based on these values the results of following two high-level *AssessmentTechnique*s are derived.

Cyber Security Attack Exposure (CSAE) provides a cumulated view on the exposure to cyber security related attacks resulting from organizational, procedural and technical vulnerabilities that can potentially be exploited by an attacker. The value of an OE's measurement is derived from the assessments of constituting control objectives. The score of the measurement is determined by applying a minimum operation to the scores of the constituting *AssessmentTechnique*s, reflecting a worst-case assumption with respect to exposure.

The **Architectural Debt** (ArcDebt) provides a cumulated view on potential costs and disadvantages that result from non-compliance to Global Architecture Standards and missing investments into IT rejuvenation. The value of the OE's measurement is derived from the assessments of constituting control objectives. The Architectural Debt for an *ITDomain* combines operating environments (as the top-level) and use cases (at the child-level), e.g. 'operating system on workplace'. The value is determined by applying a summation over the values of the constituting *AssessmentTechnique*s.

The **number of internet-facing vulnerabilities**, the **number of internal vulnerabilities** and the **Cyber Security Attack Exposure** all consider the OE as a whole, making them *DirectControlObjective*s in terms of Section 3. The **Architectural Debt** and its constituting **IT Ageing** and **IT Debt** are conversely *TypedControlObjective*s bound to the EA concepts *ITDomain* and *Technology* and can be assessed for any instance of these concepts, e.g. the aforementioned *ITDomain* 'operating system on workplace'.

Aforementioned *ControlObjective*s and *AssessmentTechnique*s can be described via a model (see Figure 9) instantiating the metamodel from Section 3. The *TypedControlObjective*s employed reflect their 'binding' to *ITDomain* and *Technology*, as discussed above, via a parameterization with the corresponding types. This allows to leverage for the actual instances of **Architectural Debt**, **IT Ageing** and **IT Debt** the existing relationships between the related *Technology* and *ITDomain* instances from the EA model.

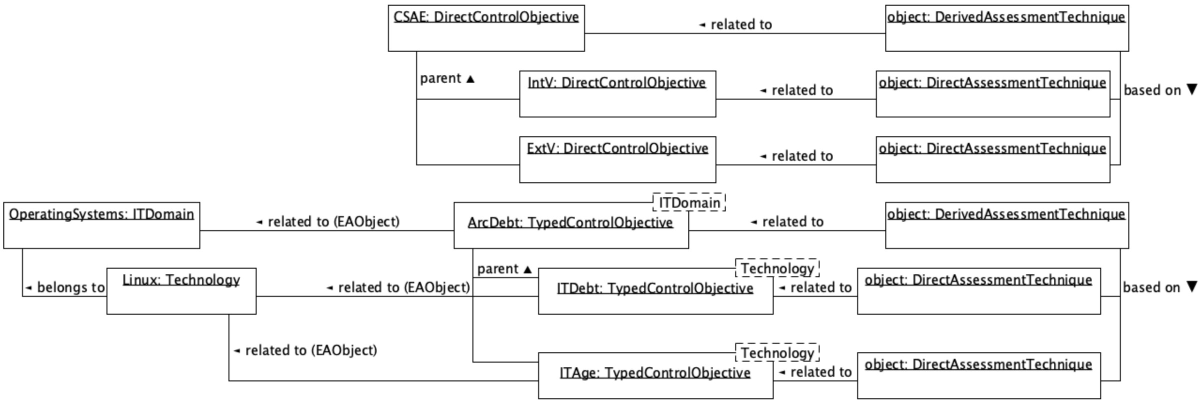

CSAE: DirectControlObjective
◂ related to
object: DerivedAssessmentTechnique
parent ▲
IntV: DirectControlObjective
◂ related to
object: DirectAssessmentTechnique
based on ▼
ExtV: DirectControlObjective
◂ related to
object: DirectAssessmentTechnique
ITDomain
OperatingSystems: ITDomain
◂ related to (EAObject)
ArcDebt: TypedControlObjective
◂ related to
object: DerivedAssessmentTechnique
parent ▲
Technology
◂ belongs to
Linux: Technology
◂ related to (EAObject)
ITDebt: TypedControlObjective
◂ related to
object: DirectAssessmentTechnique
based on ▼
◂ related to (EAObject)
Technology
ITAge: TypedControlObjective
◂ related to
object: DirectAssessmentTechnique

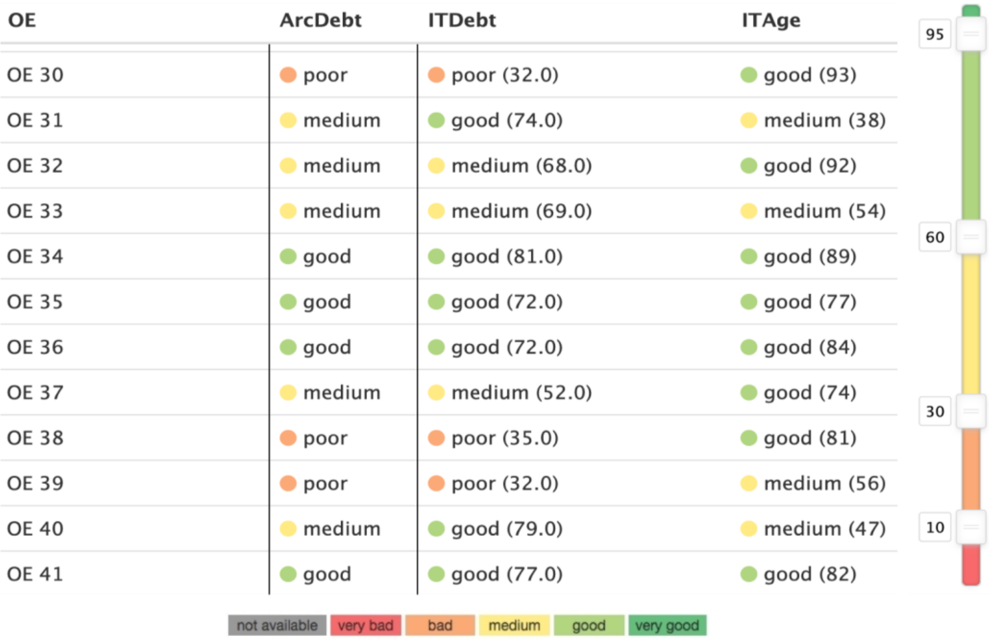
Figure 9. Instantiation of ControlObjectives and AssessmentTechniques

5.3 Hierarchic Viewpoints and Views

In this section we exemplary derive one viewpoint each for Concern 1 and Concern 3. We start with a tabular report illustrated by Figure 10 that corresponds to Concern 1. This view provides a comprehensive picture of the previously selected *ControlObjective* 'ArcDebt'. Since 'ArcDebt' is of type *TypedControlObjective*, it must be bound to an instance of type 'ITDomain'. In our case we do this with the domain 'OperatingSystems'.

Each line of the tabular report represents the current status of a particular OE. The first column displays the name of the representing OE, while column two displays the overall status of

‘ArcDebt’. Since ‘ArcDebt’ consists of the two subordinate control objectives ‘ITDebt’ and ‘ITAge’ (see Figure 9), they are displayed in columns three and four. The number in brackets shows the value measured in each case, which is classified into a score between very good and very poor using a related *AssessmentTechnique*. For ‘ArcDebt’ there is no measured value, because this score is calculated by a *DerivedAssessmentTechnique*. In this example, the formula to derive the score for ‘ArcDebt’ is the minimum score of ‘ITDebt’ and ‘ITAge’. For this reason the ‘ArcDebt’ score for ‘OE 32’ is medium, because medium (‘ITDebt’) is worse than ‘good’ (‘ITAge’). The Tabular Report also serves as entry point for user interactives. By clicking the column ‘ITDebt’ or ‘ITAge’ a slider is displayed to change the algorithm translating the measured values into scores. In the current example, the figure shows the slider corresponding to ‘ITDebt’.



OE	ArcDebt	ITDebt	ITAge
OE 30	● poor	● poor (32.0)	● good (93)
OE 31	● medium	● good (74.0)	● medium (38)
OE 32	● medium	● medium (68.0)	● good (92)
OE 33	● medium	● medium (69.0)	● medium (54)
OE 34	● good	● good (81.0)	● good (89)
OE 35	● good	● good (72.0)	● good (77)
OE 36	● good	● good (72.0)	● good (84)
OE 37	● medium	● medium (52.0)	● good (74)
OE 38	● poor	● poor (35.0)	● good (81)
OE 39	● poor	● poor (32.0)	● medium (56)
OE 40	● medium	● good (79.0)	● medium (47)
OE 41	● good	● good (77.0)	● good (82)

Figure 10. Tabular Report to get an overview about existing OEs and their status of Control Objectives

Each *AssessmentTechnique* also defines a way of representing the corresponding *Measurement*s in a viewpoint. We employ the approach outlined in [10] according to which a technique can be applied to a viewpoint in terms of an additional layer adding/changing visual variables of existing symbols. For the Control Compliance Cockpit, we employ color-coding on the different layers. The legend at the bottom of the visualization reflects the scoring system of the respective *AssessmentTechnique* ranging from ‘very good’ (dark green) to ‘very bad’ (red), adding one more color for ‘not available’ (dark gray) *Measurement*s. The slider on the right side of the visualization directly influences the thresholds specified in the *AssessmentTechnique*. When these thresholds are adapted, the *AssessmentTechnique* re-calculates the scoring and the color-coding is adapted. Via this mechanism, subject matter experts are supported in ‘what-if’ analyses.

In contrast to the Tabular Report, which aims to provide an overview of the entire group, the Cluster View shown in Figure 11 provides detailed information about architecture elements of a selected OE and a Control Objective. The values of the measurements are alluded to a ‘KPI’ – a term used in the particular enterprise in this context.

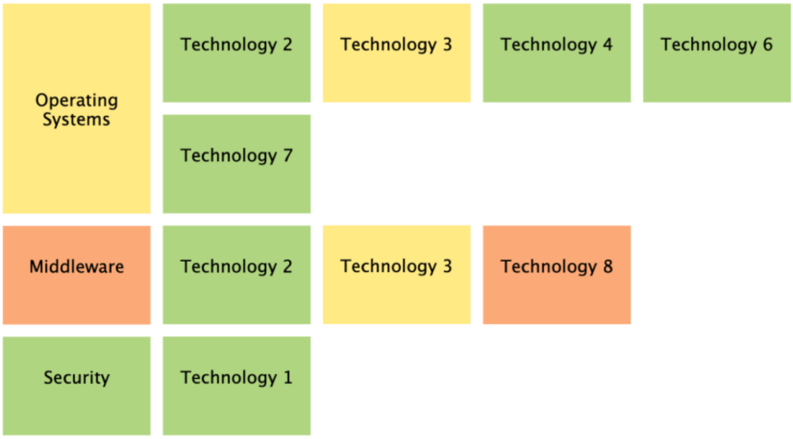


Figure 11. Cluster View about ‘OE 31’

In this visualization, technologies are grouped into so-called IT domains (e.g. operating systems). The elements on the far left represent IT domains. The *ControlObjective* used for the background color in this case is ‘ArcDebt’. This is a *TypedControlObjective* (see Figure 10), which is bound to the type ‘ITDomain’. The calculation of the score is based on the subordinate *ControlObjectives* ‘ITDebt’ and ‘ITAge’, both of which are also *TypedControlObjectives* and bound to the type ‘Technology’. In this example, the coloring of the technologies is done using ‘IT Debt’. For the sake of simplicity and because rectangles cannot be colored with two background colors at the same time, we assume that all *Measurements* to ‘ITAge’ have a ‘good’ score and therefore do not count for further consideration.

The score belonging to ‘ArcDebt’ consists of the two subordinate ControlObjectives. This is determined by *DerivedAssessmentTechniques* (see Figure 10). In this case, the score is derived using the min operation. This means the worse score of ‘ITDebt’ and ‘ITAge’ is used.

Both viewpoints, the Tabular Report and the Cluster View, can be configured by a set of parameters. Several views can be created from one viewpoint as a result of parameterization. An example shows the cockpit represented by Figure 12, which contains eight different Cluster Views corresponding to different parameterizations of the same viewpoint.

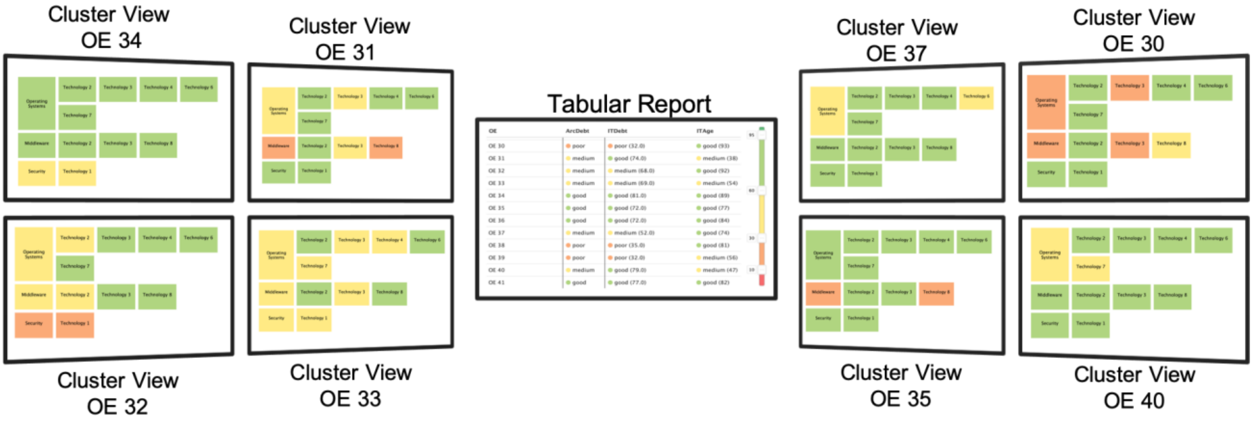


Figure 12. Aggregated Cockpit View

The parameterization of a viewpoint not only influences the resulting view, it also influences the associated Concern and its Stakeholder in the same way. For instance, if the viewpoint is parameterized with the ‘OE 31’, this view interests the responsible manager in the ‘OE 31’. Similarly, the stakeholder changes by selecting ‘OE 40’ by the corresponding manager.

6 Conclusion and Outlook

In this article, we elaborated on the hierarchic nature of concern in the area of GRC. From a theoretic point, we built our argumentation on relevant pre-work revisited in Section 2, formulating our key assumptions in Section 3:

- Concerns can be hierarchic.
- Viewpoints can be hierarchic.
- Views can be hierarchic.

The application to the GRC domain in Section 4 supported our argument by displaying the hierarchic nature of control objectives and corresponding techniques for assessing control compliance and effectiveness. From a more practical point of view, the application exampled in Section 5 showed how hierarchic concerns, viewpoints and views can be applied in the GRC context.

The combination of theoretic argument and practical utility gives evidence supporting the key assumptions around the hierarchic concern, viewpoint and view, answering the research question exhibited in Section 1.

The assumptions and the derived metamodel also show limitations with respect to their validity, which become evident taking a closer look on the different places, where sub-super relationships (‘hierarchies’) are employed:

- Switching between viewpoints covering all operating entities (OE) of the organization or just a single OE.
- Switching between a direct assessment of a fine-grained control objective and a derived assessment for a more coarse-grained control objective.
- Switching the scope of a control objective by selecting a functionally ‘bigger’ or ‘smaller’ IT domain to be considered.

The first example might be considered a hierarchy, even in the narrow sense of that term, whereas the second example could be denoted as ‘abstraction’ and the third example can be considered a ‘specialization’. This opens another avenue of research to be pursued, seeking to understand and define the concepts concern, viewpoint and view with a more formal semantics. The work of [18] can be a starting point for that providing set-theoretic formalisms and correspondences in the context of the ISO Std. 42010 [9]. Additional support for that avenue of research stems from the concept of the *TypedControlObjective* as discussed in this article. This concept was used to parameterize an assessment by a respective structuring concept of the EA. In the implementation of the Control Compliance Cockpit, this parameterization also translated to a parameterization of the viewpoint and view, respectively. In line with the key argument of this article, this should also have implications on the concern.

A sound formal conceptualization of concern, viewpoint and view must also account for that observed parameterization, which – in a type theoretic sense – can be reflected as an instance of a *template class* with one (or more) formal parameters. Combining this with the operational semantics of viewpoints as presented in [11] and [12] would advocate to revisit this article’s key assumptions as an indication towards an algebra of viewpoints that can be composed, combined and added onto each other in order to fully reflect an arbitrary concern.

References

[1] European Union Agency for Network & Information Security (ENISA), “ENISA Threat Landscape Report 2017,” 2017.

[2] European Parliament, “Regulation 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data,” *Off. J. Eur. Union*, vol. L119, pp. 1–88, 2016.

[3] Bundesanstalt für Finanzdienstleistungsaufsicht (BaFIN), “Supervisory Requirements for IT in Insurance Undertakings,” 2019.

[4] P. E. Proctor, J. A. Wheeler, and K. Pratap, “Definition: Governance, Risk and Compliance,” 2015.

[5] Bundesamt für die Sicherheit in der Informationstechnik, “BSI Standard 200-3: Risk Analysis based on IT-Grundschutz – Version 1.0,” 2017.

[6] D. Jugel, C. M. Schweda, C. Bauer, J. Zamani, and A. Zimmermann, “A metamodel to integrate control objectives into viewpoints for EA management,” *CEUR Workshop Proc.*, vol. 2218, pp. 110–119, 2018.

[7] The Open Group, “TOGAF Version 9.1,” 2011.

[8] The Open Group, “ArchiMate 3.0 Specification,” 2016.

[9] International Organization Of Standardization, “ISO/IEC/IEEE 42010:2011 - Systems and software engineering – Architecture description,” 2011.

[10] D. Jugel, “Modeling Interactive Enterprise Architecture Visualizations: An Extended Architecture Description,” *Complex Syst. Informatics Model. Q.*, no. 16, pp. 17–35, 2018. Available: https://doi.org/10.7250/csimq.2018-16.02

[11] J. Lankes, F. Matthes, and A. Wittenburg, “Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000,” *Softw. Eng. 2005, Fachtagung des GI-Fachbereichs Softwaretechnik, Lect. Notes Informatics*, vol. 64, pp. 43–54, 2005.

[12] D. Jugel, “Eine integrative Methode zur Entscheidungsfindung im Unternehmensarchitekturmanagement,” University of Rostock, 2018.

[13] ISACA, “COBIT - A Business Framework for the Governance and Management of Enterprise IT,” 2013.

[14] International Organization Of Standardization, “ISO/IEC 27001: Information technology - Security techniques - Information security management systems - Requirements (2nd edition),” 2013.

[15] U. Frank, “The MEMO meta modelling language (MML) and language architecture (2nd edition),” Duisburg-Essen, 2011.

[16] D. Heise, S. Strecker, and U. Frank, “ControlML: A domain-specific modeling language in support of assessing internal controls and the internal control system,” *Int. J. Account. Inf. Syst.*, vol. 15, no. 3, pp. 224–245, Sep. 2014. Available: https://doi.org/10.1016/j.accinf.2013.09.001

[17] S. Strecker, U. Frank, D. Heise, and H. Kattenstroth, “MetricM: a modeling method in support of the reflective design and use of performance measurement systems,” *Inf. Syst. E-bus. Manag.*, vol. 10, no. 2, pp. 241–276, Jun. 2012. Available: https://doi.org/10.1007/s10257-011-0172-6

[18] S. Buckl, S. Krell, and C. M. Schweda, “A Formal Approach to Architectural Descriptions – Refining the ISO Standard 42010,” in *Advances in Enterprise Engineering IV. CIAO! 2010. Lecture Notes in Business Information Processing*, pp. 77–91, 2010. Available: https://doi.org/10.1007/978-3-642-13048-9_6

[19] B. Nuseibeh, J. Kramer, and A. Finkelstein, “A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification,” *IEEE Trans. Softw. Eng.*, vol. 20, no. 10, pp. 760–773, 1994. Available: https://doi.org/10.1109/32.328995

[20] R. M. Dijkman, D. A. C. Quartel, and M. J. van Sinderen, “Consistency in multi-viewpoint design of enterprise information systems,” *Inf. Softw. Technol.*, vol. 50, no. 7–8, pp. 737–752, Jun. 2008. Available: https://doi.org/10.1016/j.infsof.2007.07.007

[21] J. Andrade, J. Ares, R. Garcia, J. Pazos, S. Rodriguez, and A. Silva, “A methodological framework for viewpoint-oriented conceptual modeling,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 282–294, May 2004. Available: https://doi.org/10.1109/TSE.2004.1