

Managing Model Complexity through Structure

Lovis Justin Immanuel Zenz^a, Harald Hagel, Peter Hillmann^b and Andreas Karcher

*Institute for Applied Computer Science, University of the Bundeswehr Munich,
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany*

Keywords: Enterprise Architecture, Enterprise Modeling, Model Quality, Model Structuring, Stakeholder Orientation.

Abstract: Structuring models is crucial for ensuring ease of gathering contained information. While enterprise architecture frameworks and other references inspire some structure, they often fall short when models grow too large. Hence, managing such complexity necessitates further structuring. In this paper, we present a four-step systematic process for sustainably achieving such structuring in real-world scenarios. Our approach ensures stakeholder and perspective orientation, maintains compliance with stipulations and preserves any preexisting structure. Altogether, it balances structural rigor with long-term practical applicability.

1 NEED FOR STRUCTURE

In enterprise architecture (EA), models are employed as a manner of communication (Lankhorst, 2017; Hillmann et al., 2021; Hillmann et al., 2024). EA frameworks – like the *Unified Architecture Framework (UAF)* or the *Zachman Framework* – provide a means for structuring such models through viewpoints (Zachman, 1987; UAF Team, 2022b; UAF Team, 2022a). However, with complex models, we still see large views with many elements (Capability Planning Working Group, 2024). As an excessive number of elements within a single view can impede understanding, further structuring is necessary (Sánchez-González et al., 2012; Recker, 2013; Zimoch et al., 2017; Hillmann et al., 2022). Thereby, the crucial orientation of models towards their stakeholders is promoted (Lankhorst, 2017; Hillmann et al., 2025; Zenz et al., 2026).

In this paper, we present an approach for achieving such structuring. To this end, the remainder of this paper is structured as follows: Firstly, we outline an exemplary model structuring scenario in Section 2. Secondly, we identify requirements for an approach to structure models in Section 3. Thirdly, we assess existing approaches for structuring models in Section 4. Fourthly, we present our own approach in Section 5. Fifthly, we exemplarily instantiate our approach in Section 6. Finally, we summarize our findings and provide an outlook in Section 7.

2 STRUCTURING SCENARIO

To illustrate the need for structuring models, we consider the real-world-based IT service provider *DISP* as depicted in Figure 1. Due to *DISP*'s activities in the defense sector, all EA models must comply with the *NATO Architecture Framework (NAF)* (Architecture Capability Team, 2020). On the other hand, *DISP* is structured based on the processes it performs to develop, sell, deliver, and maintain its services. These *End-to-End Processes (E2Es)* are categorized into groups and hierarchically subdivided into *Process Chains (PCs)*, *Business Processes (BPs)*, and *Sub-Processes (SPs)*. Such a process-centric perspective differs from the capability-centrality of *NAF*. Several stakeholders are involved in or affected by any modeling effort at *DISP* – e.g., people responsible for some process (process owners) and people building upon information on it to perform their work (process users). The same stakeholder might fulfill different roles for different processes. E.g., for process P1 depending on process P0, P1's owner is one of P0's users. To remain competitive, new technologies and methods are continuously adopted at *DISP*. This necessitates regular updates of processes and thus the models describing them.

Unfortunately, many naturally grown models at

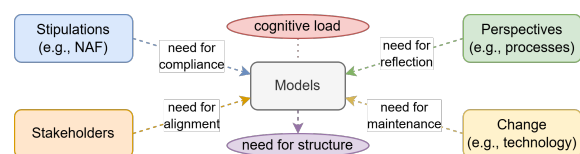


Figure 1: Scenario of an IT Service Provider named DISP.

^a <https://orcid.org/0009-0006-0219-6387>

^b <https://orcid.org/0000-0003-4346-4510>

DISP are monolithic and thus difficult to understand and maintain. E.g., there are automatically generated *NAF L7* views with more than 1,000 elements. The associated cognitive load inhibits the stakeholders' ability to gather information from and maintain such models. At the same time, there are some appropriately sized views which do not impose such inhibitions on stakeholders. As some stakeholders are familiar with these well-structured model parts, disregarding preexisting structure when structuring models would impede them.

Altogether, a model structuring approach (MSA) that considers all of these particularities is required. This scenario will be taken up throughout the remainder of this paper to rationalize requirements and thus our approach as well as to illustrate its instantiation.

3 STRUCTURE REQUIREMENTS

Before an MSA can be developed, relevant requirements must be identified. Inspired by the *Guidelines of Modeling* and *ISO/IEC 25010*, we extract the following six **MUST** and one **SHOULD** requirements from our scenario from Section 2 (Schütte, 2013; JTC 1 SC 7, 2023):

1. **Model Structuring:** With our approach, we intend to further structure models beyond the grid-like structuring provided by EA frameworks in order to reduce the number of elements per view and thus the associated cognitive load (Kear and Heap, 2007; Chen et al., 2011; Zenz et al., 2026). Hence, the MSA **must** provide a means for structuring models in a way that leads to acceptable view sizes – i.e., no more than 40 elements per view – while ensuring that each view remains coherent and meaningful.
2. **Stakeholder Orientation:** Stakeholder orientation is essential in EA modeling (Lucke and Lechner, 2011; Pankowska, 2015; Kurnia et al., 2021; Zenz et al., 2026). Hence, the MSA **must** ensure that resulting structures align with stakeholder demands.
3. **Perspective Orientation:** While stakeholder orientation ensures alignment with stakeholder demands, it does not account for the way the enterprise perceives itself. As many stakeholders are at least familiar with, and in some cases even shape, this self-perception, perspective orientation can promote greater understanding and acceptance of models by addressing this gap (Lucke and Lechner, 2011). Hence, the MSA **must** ensure that resulting struc-

tures reflect the enterprise's self-perception.

4. **Stipulation Compliance:** Models may need to adhere to specific stipulations – e.g., regulatory requirements or prescribed frameworks (Hillmann et al., 2025; Zenz et al., 2026). Hence, the MSA **must** ensure that resulting structures comply with such stipulations.
5. **Practical Suitability:** While abstract approaches might provide an ideal structuring in theory, actual added value is only achieved with practical applicability (Zenz et al., 2024). Hence, the MSA **must** be generically applicable and effective in real-world scenarios.
6. **Long-Term Maintainability:** While resource-intensive approaches may lead to optimal structures in one-time applications, efficient ones are more feasible in the long term. Hence, the MSA **must** ensure that resulting structures can be maintained with reasonable effort.
7. **Structure Preservation:** Both references – like EA frameworks – and the organizational structure may influence model contents (Architecture Forum, 2025; Zenz et al., 2025). Structuring models accordingly can aid in navigating and understanding them. E.g., structurally aligning models with organizations enables stakeholders familiar with the latter to intuitively navigate the former. Hence, the MSA **should** preserve corresponding preexisting structures.

4 EXISTING APPROACHES

Based on the requirements from Section 3, we assess the existing MSAs listed in Table 1. In doing so, we identify both inspirations for our own MSA and key shortcomings that we intend to avoid. For sake of contextualization and comparability, we also indicate fulfillment of requirements. This juxtaposition is not intended to be exhaustive, but rather to highlight the most relevant aspects.

Table 1: Structuring Concepts in Existing Approaches.

Approach	Decomposition Concept/-s	Abstraction Concept/-s
CFV	Stakeholder Groups	Solution Layers
BLM	Domains	Architecture Layers
CSDT	—	Hierarchy Levels Solution Layers
RAMI4.0	Life Cycle Stages Value Streams	Architecture Layers Hierarchy Levels
SCM	Life Cycle Stages	Solution Layers
SGAM	Value Streams	Architecture Layers Hierarchy Levels

Steen et al. present the *Classification Framework for Viewpoints (CFV)* – depicted in Figure 2 – in the context of EA modeling (Steen et al., 2004). It distinguishes between three stakeholder groups and three solution layers. Overall, the *CFV* promotes *Stakeholder Orientation* (Req. 2) through structuring in EA. However, its singular and generic consideration of stakeholders and solution layers leads to a lack of *Perspective Orientation* (Req. 3). Moreover, it poses the risk of insufficient *Model Structuring* (Req. 1).

Schüller et al. present the *Business Layer Model (BLM)* as a layer model for business process integration (Schüller et al., 2014). It specifies four architecture layers and two generic domains. Overall, the *BLM* highlights interrelationships in business process modeling facilitating both separation of concerns and business process integration. However, its high level of genericity leads to a lack of *Stakeholder & Perspective Orientation* (Req. 2 & Req. 3). Moreover, it poses the risk of insufficient *Model Structuring* (Req. 1). Furthermore, its specification of proprietary architecture layers results in a lack of *Stipulation Compliance* (Req. 4) and *Structure Preservation* (Req. 7).

Taibi et al. present a *Classification Schema for Development Technologies (CSDT)* (Taibi et al., 2014). It combines four solution layers with four hierarchy levels. Overall, the *CSDT* offers a structured Systems of Systems development approach. However, its specification of concrete viewpoints leads to a lack of *Stipulation Compliance* (Req. 4) and *Structure Preservation* (Req. 7). Furthermore, its hierarchy levels taking the form of system decomposition leads to insufficient *Practical Suitability* (Req. 5).

The *Reference Architecture Model Industrie [sic] 4.0 (RAMI4.0)* – illustrated in Figure 3 – is a reference model for the Industry 4.0 domain (Adolphs et al., 2016). It defines six architecture layers as well as seven hierarchy levels and two generic value streams. The value streams are subdivided into two life cycle stages each. Overall, the *RAMI4.0* provides a sophisticated and multidimensional structuring concept

for the domain of Industry 4.0 that allows for sufficient *Model Structuring* (Req. 1) in this particular context. However, its singular focus on Industry 4.0 applications leads to insufficient *Practical Suitability* (Req. 5) in other domains. Furthermore, its definition of proprietary architecture layers results in a lack of *Stipulation Compliance* (Req. 4) and *Structure Preservation* (Req. 7).

Pfenning presents the *System Concretization Model (SCM)* as an interdisciplinary behavioral model for developing systems (Pfenning, 2017). It considers three spaces representing life cycle stages. The *Solution Space* is divided into three solution layers and each layer is subdivided into two sub-layers. Within each of the six sub-layers, further structuring is enabled through variation and decomposition. Overall, the *SCM* provides a comprehensive structuring concept for the domain of systems engineering that allows for sufficient *Model Structuring* (Req. 1) regardless of the context. However, its high level of genericity leads to a lack of *Stakeholder & Perspective Orientation* (Req. 2 & Req. 3).

The *Smart Energy Grid Architecture Model (SGAM)* is a reference model for the smart grid domain (CEN-CENELEC-ETSI Smart Grid Coordination Group, 2021). It defines five architecture layers. On each of these layers, six hierarchy levels are combined with five value streams. Overall, the *SGAM* provides a detailed structuring concept for the smart grid domain. However, its singular focus on smart grids leads to insufficient *Practical Suitability* (Req. 5) in other domains. Furthermore, its definition of proprietary architecture layers results in a lack of *Stipulation Compliance* (Req. 4) and *Structure Preservation* (Req. 7).

Altogether, the existing MSAs do not build upon structure provided by EA frameworks and other references, but either provide very little structuring or are solely applicable in peculiar contexts. However, the particular decomposition and abstraction concepts employed by these MSAs provide inspiration for our

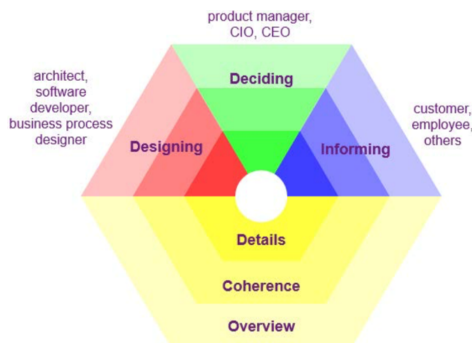


Figure 2: Classification Framework for Viewpoints.

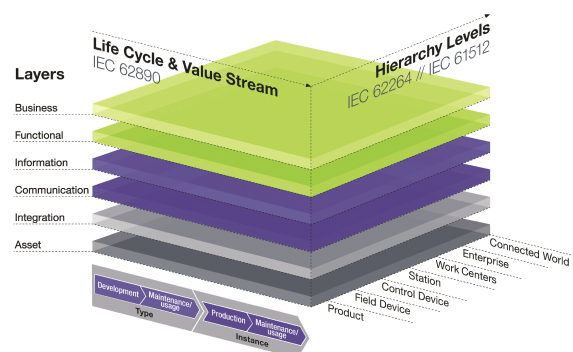


Figure 3: Reference Architecture Model Industrie 4.0 [sic].

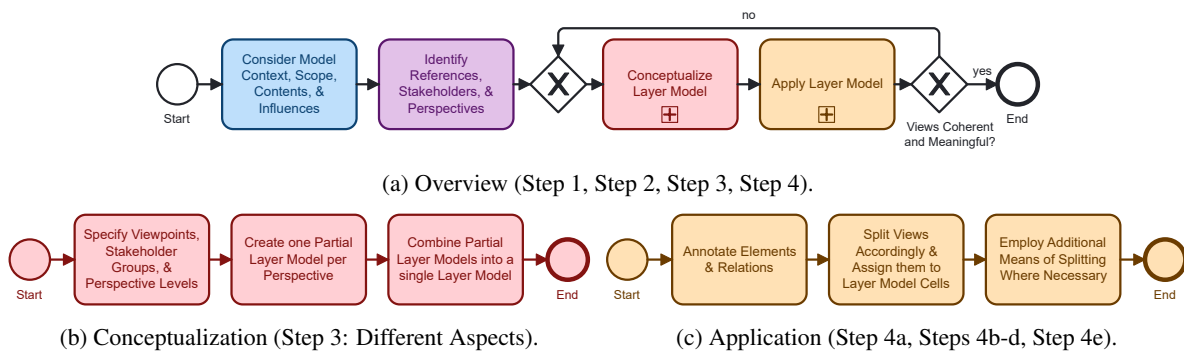


Figure 4: Model Structuring Approach.

own MSA. Moreover, the frequent lack of guidance highlights the need for an approach that systematically guides through the creation and more importantly the application of structuring concepts in a way that is suitable for real-world scenarios while remaining generically applicable.

5 CONCEPT FOR STRUCTURE

Considering the scenario from Section 2, the requirements from Section 3 and the qualities & shortcomings of existing MSAs from Section 4, we develop our own MSA. As depicted in Figure 4, it takes the form of a four-step process with two subprocesses:

1. **Consideration:** Collect relevant information for structuring models. Consider both desired and factual model specifics.
 - **Model Contexts & Scope:** Determine the intent behind the modeling effort and the intended use of the resulting *Pre-Model*. Clarify the scope of the *Pre-Model* in terms of included and excluded aspects. Specify the desired extent and level of detail of model contents.
 - **Model Contents & Influences:** Examine the current contents of the *Pre-Model*. Contrast the previously defined scope with the actual realization. Ascertain the influences that shaped the model contents and the *Pre-Model* as a whole.
2. **Identification:** Concretize relevant aspects for structuring models. Consider human, organizational, and stipulatory features.
 - **References:** List all references inspiring and regulating the *Pre-Model*. Determine both stipulations the *Pre-Model* must comply with and influences pre-structuring the *Pre-Model*.
 - **Stakeholders:** Pinpoint all stakeholders of the *Pre-Model*. Determine their roles, responsibilities, and interests regarding the *Pre-Model*.

- **Perspectives:** Analyze the enterprise's self-perception by establishing a thorough understanding of its internal structure and determining the features shaping this structure. Possible features include, but are not limited to, organizational elements (functions, projects, domains, federation), value streams, capabilities, processes, services, and products. Reduce the set of features to those that are (a) central to the enterprise's self-perception and (b) relevant to the modeling effort. If there are several levels of relevance, limit the selection to the most relevant level. This results in a set of *Perspectives* on the enterprise and *Pre-Model*.

3. **Conceptualization:** Construct a *Layer Model* consisting of one *Partial Layer Model* per *Perspective* as outlined in Figure 5. To this end, specify three dimensions, where the first two are generic and the last one is *Perspective-specific*:

- **Viewpoints:** For the first dimension, select already employed means of model structuring. Most likely, these are the viewpoints prescribed by an employed EA framework.
- **Stakeholder Groups:** For the second dimension, choose a meaningful grouping of all identified stakeholders. Ensure that these stakeholder groups are generically applicable to all *Perspectives*. E.g., such a grouping could consist of *Decision Makers*, *Middle Management*, *Perspective-Specific Leaders*, *Perspective-Specific Contributors*, and *Perspective-Specific Users*. As mentioned above, a single stakeholder might belong to multiple stakeholder groups.
- **Perspective Levels:** For the third dimension,

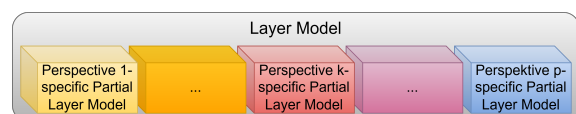


Figure 5: Generic Layer Model.

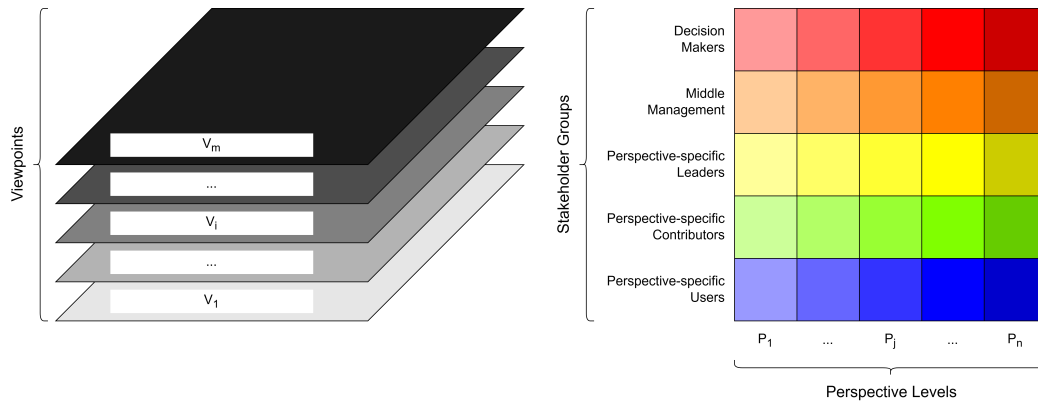


Figure 6: Generic Partial Layer Model.

Data: oversized view *original*

$list \leftarrow roots \leftarrow \{element \in original.elements : \nexists otherElement \in original.elements : element.extends(otherElement)\};$

while !*list.isEmpty()* **do**

$parent \leftarrow list.pop();$

$children \leftarrow \{element \in original.elements : element.extends(parent)\};$

if !*children.isEmpty()* **then**

$list.addAll(\{child \in children : \exists element \in original.elements : element.extends(child)\});$

$createView(parent :: children, \{relation \in original.relations : relation.source \in parent :: children \wedge relation.target \in parent :: children\});$

end

end

remove non-root elements from *original*;

Algorithm 1: Splitting of Taxonomy Views.

determine the levels of each selected *Perspective* and bring them into a logical order. E.g., order levels of a hierarchic *Perspective* from top to bottom or levels of a chronological one from first step to last.

Afterwards, combine these three dimensions to form one *Partial Layer Model* per *Perspective* as illustrated in Figure 6. In doing so, build upon the same *Viewpoints*, concretize the generic *Stakeholder Groups* for each *Perspective*, and limit the *Perspective Levels* to those of the respective *Perspective*. When finally combining all *Partial Layer Models* into one *Layer Model*, this ensures that each of the former reflects the specifics of its *Perspective* while the latter maintains consistency across all *Perspectives*. Each unique combination of *Viewpoint*, *Stakeholder Group*, and *Perspective Level* defines a *Cell* in the (Partial) *Layer Model*.

4. **Application:** Repeat the following steps for each *Partial Layer Model* to apply the *Layer Model* to the *Pre-Model*:

(a) **Annotation:** Annotate each element and relation of the *Pre-Model* with its relevance to each *Stakeholder Group*, and *Perspective Level*. Such an annotation takes the form of $\langle PLM \rangle - \langle SG \mid PL \rangle : \langle true \mid false \rangle$, where $\langle PLM \rangle$ is the name of a *Partial Layer*

Model. $\langle SG \mid PL \rangle$ denotes a *Stakeholder Group* or *Perspective Level* of the respective *Partial Layer Model*. $\langle true \mid false \rangle$ indicates whether the annotated element is relevant to this particular *Stakeholder Group* or *Perspective Level*.

- (b) **For Viewpoints:** Assuming pre-existence of at least one monolithic view per viewpoint that is intended to be instantiated, categorize existing views based on the *Viewpoint* they instantiate. This results in a set of views per *Viewpoint*.
- (c) **For Stakeholder Groups:** For each *Viewpoint* and view within, create one view per *Stakeholder Group* with all elements and relations of the original view that are annotated as relevant to it. Remove the original view. This results in a set of views per *Viewpoint*, a set of views per *Stakeholder Group*, and a set of views per *Viewpoint & Stakeholder Group*.
- (d) **For Perspective Levels:** For each *Viewpoint* and *Stakeholder Group* combination and view within, create one view per *Perspective Level* with all elements and relations of the original view that are annotated as relevant to it. Remove the original view. This results in a set of views per *Viewpoint*, a set of views per *Stakeholder Group*, a set of views per *Perspective*

Level, a set of views per *Viewpoint* & *Stakeholder Group*, a set of views per *Viewpoint* & *Perspective Level*, a set of views per *Stakeholder Group* & *Perspective Level*, and a set of views per *Viewpoint*, *Stakeholder Group*, & *Perspective Level*.

- (e) **Finalization:** Ensure that resulting views are coherent, meaningful, and manageable in size. In case of incoherent or meaningless views, reconsider the annotation of elements. If necessary, redo the *Conceptualization* in terms of *Stakeholder Groups* & *Perspective Levels* and reapply the *Layer Model*. Only affected annotations need to be considered. E.g., if a *Stakeholder Group* is removed, all annotations regarding it need to be removed and if a *Perspective Level* is added, elements and relations need to be annotated for it. Similarly, only affected views need to be generated again. In case of undersized views, consider merging views that belong to the same *Cell*. In case of oversized views, conduct further splitting. E.g., split taxonomy views as illustrated in Algorithm 1 and process views by establishing subprocesses.

This systematic structuring of the *Pre-Model* results in an *Applied Layer Model*. As with (*Partial*) *Layer Models*, a unique combination of *Viewpoint*, *Stakeholder Group*, and *Perspective Level* defines a *Cell* in the *Applied Layer Model*, as outlined in Figure 7. Each of these *Cells* contains either no, one, or multiple appropriately sized views.

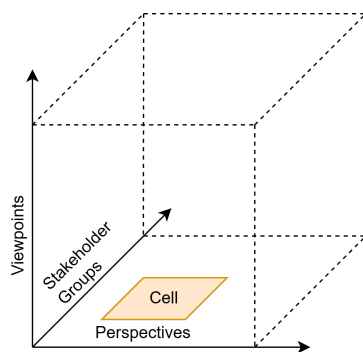


Figure 7: Concept of a Cell in the Layer Model.

To maintain the *Layer Model*, update it (Steps 1 to 3) whenever the enterprise undergoes significant changes – e.g., when *References* are updated, new *Stakeholders* emerge or the roles of existing ones change, or the enterprise’s self-perception evolves leading to added, modified, or removed *Perspectives*. To maintain the *Applied Layer Model*, reapply the *Layer Model* to the *Pre-Model* (Step 4) whenever the

latter’s contents change in any way. Limit the reapplication to model contents affected by the change to minimize the associated effort.

The *Applied Layer Model* can further be employed to systematically retrieve relevant model contents for specific use cases. To this end, identify each user’s roles and their manifestations for each *Perspective*. This leads to a mapping of users to *Perspective*-specific *Stakeholder Groups*, which allows for automatic assignment of relevant *Cells* to each user. In combination with element and relation annotations as well as view categorizations, users can swiftly retrieve all relevant model contents by investigating their assigned *Cells*.

6 APPLICATION TO MODELS

Considering the enterprise in the scenario from Section 2, we identify the process hierarchy as its self-perception’s most prominent shaper. Accordingly, we build upon this process hierarchy as the sole *Perspective* and select *E2E Groups*, *E2Es*, *PCs*, *BPs*, and *SPs* for *Perspective Levels*. For *Stakeholder Groups*, we keep the generic *Decision Makers*, *Middle Management* from Section 5 while concretizing the remaining stakeholder group recommendations into *Process Owners*, *Process Contributors*, and *Process Users*. *Process Contributors* encompass both *Process Managers* and *Process Designers* while *Process Users* include *Application Developers* and *Technicians* as well as *End Users*. To comply with *NAF*, we select its viewpoints for *Viewpoints*. By combining these three selections, we obtain a *Partial Layer Model* for the process hierarchy. As there are no further *Perspectives*, this *Partial Layer Model* forms the complete *Layer Model* as depicted in Figure 8.

For applying this *Layer Model*, we act as described in Section 5. Based on the elements’ and relations’ annotations as well as original views’ categorizations, new views can automatically be generated through means of automatic layout (Eichelberger, 2005; Maier, 2012). When reaching the *Finalization* step, we split views by creating one view per process – i.e., for each *Cell* of *E2Es* one view per *E2E*, for each *Cell* of *PCs* one view per *PC*, and so forth. This ensures that resulting views are coherent, meaningful, and manageable in size.

For employing this *Applied Layer Model*, we extend our scenario by considering the exemplary use case of establishing a new *E2E*. In this context, two questions arise:

1. **For Process Modelers:** Which preexisting information model contents can I build upon when

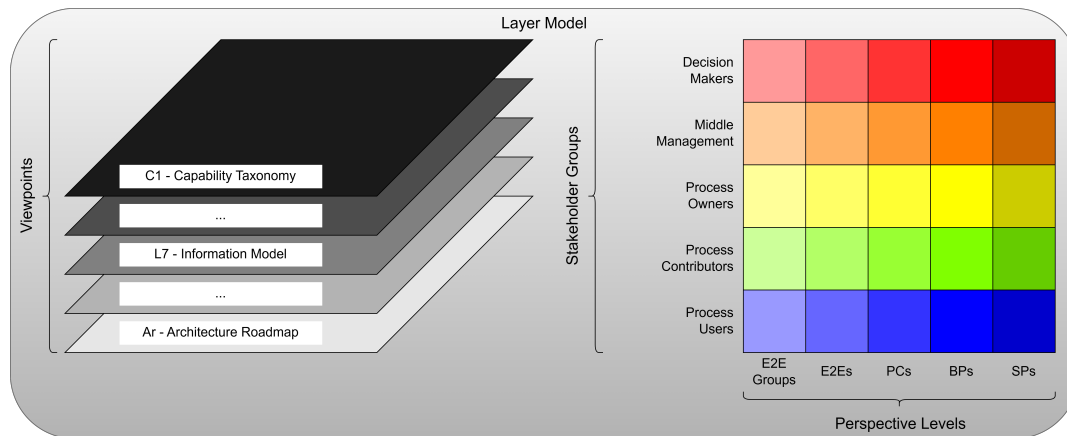


Figure 8: Layer Model for DISP.

modeling the new E2E within the process model?

2. **For Information Modelers:** Which additional contents do I need to add to the information model in order to adequately represent the new E2E?

To answer these questions, we identify the relevant *Cells* in the *Layer Model*:

- **For Process Modelers:** Searching for information model contents, we investigate the *Viewpoint L7 - Information Model*. On this layer, we consider the *Stakeholder Group Process Contributors* as process modelers are a type of *Process Designers* and thus belong to this group. Finally, we focus on the *Perspective Level E2E* as we are interested in such a process.
- **For Information Modelers:** Checking information model contents against process model contents, we investigate the *Viewpoints L4 - Logical Activities* and *L7 - Information Model*. On this layer, we consider the *Stakeholder Group Process Users* as information modelers belong to this group. Finally, we again focus on the *Perspective Level E2E* as we are interested in such a process.

Based on targeted information retrieved from the identified *Cells*, both questions can be answered swiftly and the new E2E can be established efficiently. By updating the *Applied Layer Model* accordingly, future maintenance tasks can be facilitated similarly.

7 CONCLUSION

In this paper, we presented an approach for structuring models by application of a *Layer Model* consisting of one or more three-dimensional *Partial Layer Models*. By combining *Viewpoints*, *Stakeholder*

Groups, and *Perspective Levels*, the *Layer Model* ensures compliance with stipulations and preservation of preexisting structures while promoting stakeholder and perspective orientation. Furthermore, we demonstrated the practical suitability and long-term maintainability of our approach through an exemplary instantiation in a real-world scenario.

Unlike existing approaches, our approach considers not only stakeholder but also perspective orientation which enables alignment with the enterprise's self-perception. Combined with the desired view size reduction, this promotes understanding of stakeholders and perspectives. Moreover, the resulting reduction of cognitive load to a manageable level, provision of correct information, and identification of appropriate communication partners enables cross-organizational collaboration in terms of clear communication and purposeful coordination. For modelers, annotating individual elements and relations allows for a divide & conquer approach to model structuring, where multiple people work in parallel, each contributor's cognitive load remains at a manageable level and progress is easily tracked. Moreover, the possibility of automatic view generation massively reduces the associated effort and thus further promotes long-term maintainability. Altogether, this enables effective and efficient management of a model's complexity.

However, additional complexity arises when we look at model landscapes instead of singular models. Here, the interplay of different *Perspectives* needs to be considered more deeply when creating and applying the *Layer Model*. Future work could address this challenge and further enhance our approach in additional ways. In addition, means of automatic element and relation annotation could be investigated. For instance, elements and relations could be automatically annotated based on views they are added to. Here,

implications of the same view appearing in multiple *Cells* need to be considered – e.g., an element added to a view in one *Cell* might be relevant to other *Cells* as well. Altogether, such research might further increase the practical suitability and long-term maintainability of our approach, especially in scenarios with federated modeling or modeling that is subject to frequent changes.

ACKNOWLEDGEMENTS

We would like to express our gratitude to our student Christopher Lyko. Especially, we thank him for his support in collecting related work.

In preparing this work, ChatGPT was employed solely to suggest improvements in phrasing. No substantive content was inspired or generated by AI.

REFERENCES

- Adolphs, P., Berlik, S., Dorst, W., Friedrich, J., Gericke, C., Hankel, M., Heidel, R., Hoffmeister, M., Mosch, C., Pichler, R., Rauschecker, U., Schulz, T., Schweichhart, K., Steffens, J., Taube, M., Weber, I., Wollschlaeger, M., and Mätzler, S. (2016). Reference Architecture Model Industrie 4.0 (RAMI4.0). Specification 91345, DIN.
- Architecture Capability Team (2020). NATO Architecture Framework – Version 4. Framework AC/322-D(2018)0002-REV1, NATO C3 Board.
- Architecture Forum (2025). TOGAF – Introduction and Core Concepts. Standard C220, The Open Group.
- Capability Planning Working Group (2024). Federated Mission Networking (FMN) – Spiral 6 Specification. Framework, NATO Military Committee.
- CEN-CENELEC-ETSI Smart Grid Coordination Group (2021). Smart Grid Reference Architecture. Systems Reference Deliverable 63200, IEC.
- Chen, C.-Y., Pedersen, S., and Murphy, K. L. (2011). Learners' Perceived Information Overload in Online Learning via Computer-Mediated Communication. *Research in Learning Technology*, 19(2):101–116.
- Eichelberger, H. (2005). Aesthetics and Automatic Layout of UML Class Diagrams. PhD Thesis, Julius-Maximilians-Universität Würzburg.
- Hillmann, P., Heiland, E., and Karcher, A. (2021). Automated Enterprise Architecture Model Mining. In Gurrola, J., editor, *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pages 194–201. IEEE.
- Hillmann, P., Kessler, M., Schnell, D., Mihelcic, G., and Karcher, A. (2024). Enterprise Architecture Governance of Excellence. In Filipe, J., Śmiałek, M., Brodsky, A., and Hammoudi, S., editors, *Proceedings of the 26th International Conference on Enterprise Information Systems*, volume 2, pages 603–613. SciTePress.
- Hillmann, P., Schnell, D., Hagel, H., and Karcher, A. (2022). Enterprise Model Library for Business-IT-Alignment. In *Signal, Image Processing and Embedded Systems Trends (SIGEM 2022)*, pages 157–170. AIRCC.
- Hillmann, P., Zenz, L. J. I., and Karcher, A. (2025). Critical Success Factors for Enterprise Architecture. In Filipe, J., Śmiałek, M., Brodsky, A., and Hammoudi, S., editors, *Proceedings of the 27th International Conference on Enterprise Information Systems*, volume 2, pages 793–803. SciTePress.
- JTC 1 SC 7 (2023). Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Product Quality Model. Standard 25010, ISO/IEC.
- Kear, K. and Heap, N. (2007). 'Sorting the Wheat from the Chaff'. *Journal of Computer Assisted Learning*, 23(3):235–247.
- Kurnia, S., Kotusev, S., Shanks, G., Dilnutt, R., and Milton, S. (2021). Stakeholder Engagement in Enterprise Architecture Practice. *Information and Software Technology*, 134:1–23.
- Lankhorst, M. (2017). *Enterprise Architecture at Work – Modelling, Communication, and Analysis*. The Enterprise Engineering Series. Springer, 4 edition.
- Lucke, C. and Lechner, U. (2011). Goal-Oriented Requirements Modeling as a Means to Address Stakeholder-Related Issues in EA. In *Wirtschaftsinformatik Proceedings 2011*, pages 714–723. AIS.
- Maier, S. (2012). A Pattern-based Approach for the Combination of Different Layout Algorithms in Diagram Editors. PhD Thesis, Universität der Bundeswehr München.
- Pankowska, M. (2015). Stakeholder Oriented Enterprise Architecture Modelling. In *Proceedings of the 12th International Conference on e-Business*, pages 72–79. SciTePress.
- Pfenning, M. (2017). Durchgängiges Engineering durch die Integration von PLM und MBSE [Continuous Engineering through the Integration of PLM and MBSE]. PhD Thesis D-386, Technische Universität Kaiserslautern.
- Recker, J. (2013). Empirical Investigation of the Usefulness of Gateway Constructs in Process Models. *European Journal of Information Systems*, 22(6):673–689.
- Sánchez-González, L., García, F., Ruiz, F., and Mendling, J. (2012). Quality Indicators for Business Process Models from a Gateway Complexity Perspective. *Information and Software Technology*, 54(11):1159–1174.
- Schüller, A., Epple, U., Elger, J., Müller-Martin, A., and Löwen, U. (2014). Geschäftsprozesse und technische Prozesse. *at - Automatisierungstechnik*, 62(9):665–675.
- Schütte, R. (2013). *Grundsätze ordnungsmäßiger Referenzmodellierung [Guidelines of Reference Modeling]*, volume 233 of *neue betriebswirtschaftliche forschung (nbf)*. Gabler, 1 edition.

- Steen, M. W., Akehurst, D., ter Doest, H. W., and Lankhorst, M. M. (2004). Supporting Viewpoint-Oriented Enterprise Architecture. In *Proceedings of the 8th International Conference on Enterprise Distributed Object Computing*, pages 201–211. IEEE.
- Taibi, D., Plociennik, C., and Dieudonné, L. (2014). A Classification Schema for Development Technologies. In *ICSEA 2014 – The 9th International Conference on Software Engineering Advances*, pages 577–583. IARIA.
- UAF Team (2022a). Unified Architecture Framework Modeling Language (UAFML). Specification, OMG.
- UAF Team (2022b). Unified Architecture Framework (UAF) Domain Metamodel. Specification, OMG.
- Zachman, J. A. (1987). A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3):276–292.
- Zenz, L. J. I., Heiland, E., Hillmann, P., and Karcher, A. (2024). Aligning Models with Their Realization through Model-based Systems Engineering. In Liu, S. and Kolyshkin, A., editors, *2023 International Conference on Advanced Enterprise Information System*, pages 63–69. IEEE.
- Zenz, L. J. I., Heiland, E., Hillmann, P., and Karcher, A. (2025). Reference Modeling for Automatic Target Recognition. In Ehlers, F., editor, *Maritime Autonomous Vehicles – Methods and Measurements*, volume 507 of *Radar, Sonar and Navigation*, pages 157–184. IET, 1 edition.
- Zenz, L. J. I., Hillmann, P., and Karcher, A. (2026). Stakeholder-Oriented Model Formatting. In *Proceedings of the 28th International Conference on Enterprise Information Systems*. SciTePress.
- Zimoch, M., Pryss, R., Schobel, J., and Reichert, M. (2017). Eye Tracking Experiments on Process Model Comprehension. In Reinhartz-Berger, I., Gulden, J., Nuncan, S., Guédria, W., and Bera, P., editors, *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing, pages 153–168. Springer.