

An Annotated Bibliography for ISO/IEC/IEEE 42010

version 4.4.b

October 15, 2020

Introduction

An annotated bibliography of papers, reports and books on topics pertaining to [ISO/IEC/IEEE 42010:2011](#) (revision of the former IEEE Std 1471:2000). Originally prepared for ISO/IEC JTC1/SC7 WG42, the Architecture Working Group of the Systems and Software Engineering Subcommittee of ISO. Currently the bibliography is undergoing a major update as ISO/IEC/IEEE 42010 is being revised. The current draft is a work-in-progress; it has many new entries, but these are not necessarily complete.

The bibliography includes 1) items which were inspirations for the Standard; 2) items citing or about the Standard or its development; and 3) items inspired by or built upon the Standard and its concepts.

Note: with version 4.x, we switch to producing the bibliography using the `biblatex` package. There may be errors.

Please send corrections and additions to r.hilliard@computer.org.

References

12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), 4–7 May 2015, Montréal, Québec, Canada. IEEE Computer Society, Apr. 2015.

A. Anwar et al. “A Formal Approach to Model Composition Applied to VUML”. In: *2011 16th IEEE International Conference on Engineering of Complex Computer Systems*. Keywords: graph grammars, process algebra, model composition, MBSE. Apr. 2011, pp. 188–197. DOI: [10.1109/ICECCS.2011.26](https://doi.org/10.1109/ICECCS.2011.26).

Annotations: -.

Abstract: Several approaches adopted by the software engineering community rely on the principle of multi-modeling which allows to separate concerns and to model a system as a set of less complex sub-models. Model composition is a crucial activity in Model Driven Engineering (MDE). It is particularly useful when adopting a multi-modeling approach to analyze and design software systems. In previous work, we have defined a view-based UML profile called VUML. In this paper, we describe a formal approach for model composition in which we consider the composition as an algebraic operator on the set of UML-compliant models. We specify the semantics of our composition operator by means of graph transformations. Furthermore, we present a composition scheme interpreted by a two-steps composition process based on two strategies of correspondence and merging. To illustrate our approach, we apply it to the composition of UML class models diagrams into one VUML model class diagram.

ArchiMate 3.0.1 Specification. Aug. 2017. URL: <http://www.opengroup.org/archimate/>.

Annotations: ArchiMate provides definitions of a number of architecture viewpoints, and provides a useful classification scheme for viewpoints.

Edin Arnautovic and Hermann Kaindl. "Aspects for crosscutting concerns in systems architectures". In: *Proc. of the Second Annual Conference on Systems Engineering Research (CSER-04)*. Keywords: aspects, concerns, crosscutting, systems. 2004, pp. 1–10.

Abstract: In this paper, we deal with crosscutting concerns in systems architecting. Examples of such concerns are safety or performance. They have a wide application or influence in the overall system. Designing for a separation of concerns means factoring out crosscutting concerns into separate modular units. For modular representation of crosscutting concerns, both for design and implementation, so-called aspects have been proposed in objectoriented software engineering. These are modules potentially able to encapsulate software or design artifacts treating an otherwise crosscutting concern. We argue for the use of aspects in architectures of general systems.

Colin Atkinson, Ralph Gerbig, and Christian Tunjic. "A Multi-Level Modeling Environment for SUM-Based Software Engineering". In: *VAO'13*. 2013.

Annotations: Single-Underlying-Model (SUM) based software engineering environments are founded on the principle of dynamically generating all required descriptions and visualizations of software systems on demand from a single underlying information source rather than by storing them decentrally as separately-persisted artifacts. It is possible to implement such environments using traditional two-level modeling tools, but for them to achieve their full potential we believe they need to be built on a multi-level modeling infrastructure that can support (a) the definition of model content across arbitrary ontological classification levels, (b) the addition of new domain-specific modeling features without the need to change the linguistic meta-model and (c) flexible support for various forms of content visualization and editing in a uniform, coherent and interchangeable way. In this paper we present such a multi-level modeling infrastructure and describe its use in the construction of a prototype SUM-based software engineering environment called nAOMi.

Paris Avgeriou et al. "Architectural knowledge and rationale: issues, trends, challenges". In: *SIGSOFT Software Engineering Notes* 32.4 (2007), pp. 41–46. DOI: [10.1145/1281421.1281443](https://doi.org/10.1145/1281421.1281443).

Ziv Baida. *Stakeholders and Their Concerns In Software Architectures*. Keywords: software architecture, information systems. 2001. DOI: [10.1.1.98.4250](https://doi.org/10.1.1.98.4250).

Annotations: -.

Abstract: An architect who writes an architecture document is not the only one involved in the process of building the architecture. And many more people are involved in the process of information systems development, which can be seen as a chain of activities, one of which is building the architecture. All these people are influenced by the decisions the architect makes when writing his architecture document. As a result, all these people must be taken into consideration when writing the architecture document. This paper is about identifying these people and the type of their involvement in the process.

Mario R. Barbacci. *Analyzing Quality Attributes*. Column in SEI newsletter, *The Architect*. Mar. 1999. URL: <http://www.sei.cmu.edu/library/abstracts/news-at-sei/architectmar99.cfm>.

Annotations: An eloquent argument for the need for specialized viewpoints in architectural description: "Unfortunately, in contrast to building architectures, we have yet to agree on what the appropriate software structures and views should be and how to represent them. One of the reasons for the lack of consensus on structures, views, and representations is that software quality attributes have matured (or are maturing) within separate communities, each with their own vernacular and points of view."

Alessandro Baroni et al. "Architecture Description leveraging Model Driven Engineering and Semantic Wikis". In: *Proceedings of Working IEEE/IFIP Conference on Software Architecture (WICSA 2014)*. Ed. by. Keywords: AD, wikis, agile, MBSE. 2014, pp. 251–254.

Annotations: -.

Abstract: A previous study, run by some of the authors in collaboration with practitioners, has emphasized the need to improve architectural languages in order to (i) make them simple and intuitive enough to communicate effectively with project stakeholders, and (ii) enable formality and rigour to allow analysis and other automated tasks. Although a multitude of languages have been created by researchers and practitioners, they rarely address both of these needs.

In order to reconcile these divergent needs, this paper presents an approach that (i) combines the rigorous foundations of model-driven engineering with the usability of semantic wikis, and (ii) enables continuous synchronization between them; this allows software architects to simultaneously use wiki pages for communication and models for model-based analysis and manipulation. In this paper we explain how we applied the approach to an industry-inspired case study using the Semantic MediaWiki wiki engine and a model-driven architecture description implemented within the Eclipse Modeling Framework. We also discuss how our approach can be generalized to other wiki-based and model-driven technologies.

Eduardo Barra, Anabel Fraga, and Juan Llorens. "The Symbiosis between View and Aspect". In: *First Workshop on Aspects in Architectural Description in Conjunction with Sixth International Conference on Aspect-Oriented Software Development*. Ed. by P. Clements et al. Keywords: early aspects, architecture viewpoint, architecture view, architecture description. 2007.

Len Bass et al. *Reasoning Frameworks*. Tech. rep. CMU/SEI-2005-TR-007. Software Engi-

neering Institute, Carnegie Mellon, 2005. URL: <http://www.sei.cmu.edu/publications/documents/05.reports/05tr007.html>.

Annotations: Reasoning frameworks have several properties similar to architecture viewpoints.

Abstract: Determining whether a system will satisfy critical quality attribute requirements in areas such as performance, modifiability, and reliability is a complicated task that often requires the use of many complex theories and tools to arrive at reliable answers. This report describes a vehicle for encapsulating the quality attribute knowledge needed to understand a system's quality behavior as a reasoning framework that can be used by nonexperts. A reasoning framework includes the mechanisms needed to use sound analytic theories to analyze the behavior of a system with respect to some quality attribute. This report defines the elements of a reasoning framework and illustrates the reasoning framework concept by describing several reasoning frameworks and how they realize these elements.

Adriatik Bedjeti et al. "Modeling Context with an Architecture Viewpoint". In: *2017 IEEE International Conference on Software Architecture (ICSA)*. Keywords: software architecture, software reusability, reusable work product, software architecture, context description viewpoint, contextual knowledge, software system, context, context modeling, organizations, software systems, standards organizations, stakeholders. Apr. 2017, pp. 117–120. DOI: [10.1109/ICSA.2017.26](https://doi.org/10.1109/ICSA.2017.26).

Annotations: -.

Marcel Bennis and Claus Lewerentz. "Towards Managing Software Architectures with Ontologies". In: *Graph Transformations and Model-Driven Engineering Essays Dedicated to Manfred Nagl on the Occasion of his 65th Birthday*. Ed. by Gregor Engels et al. Vol. 5765. Lecture Notes in Computer Science. Keywords: consistency, model checking. Springer, 2010.

Abstract: Software architectures are key enabling assets within organizations that develop complex software systems. Among other purposes, software architectures are useful to maintain intellectual control over a software product. We propose a method to continuously check the consistency between a specified architecture model and structural information reverse engineered from the code. We develop criteria that a design language for architectures should fulfill and show that an ontology based description has substantial benefits over the standard modeling languages MOF/UML/OCL. Using ontologies allows the explicit modelling of architectural styles as well as concrete system structures in a single architecture design language. The resulting specifications are modular, compositional and evolvable. Using ontologies we can apply an ontology reasoner to implement consistency checks. Our method integrates previously separate checks such as checking for allowed dependencies and coding style into a single framework and enables more powerful and flexible analyses.

John K. Bergery and Paul C. Clements. *Software Architecture in DoD Acquisition: A Reference Standard for a Software Architecture Document*. Technical Note CMU/SEI-2005-TN-020. CMU Software Engineering Institute, Feb. 2005. URL: <http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn020.pdf>.

F. Bergomi et al. "Beyond Traceability: Compared Approaches to Consistent Security

Risk Assessments". In: *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. Sept. 2013, pp. 814–820. DOI: [10.1109/ARES.2013.109](https://doi.org/10.1109/ARES.2013.109).

Abstract: As military and civil software-intensive information systems grow and become more and more complex, structured approaches, called architecture frameworks (AF), were developed to support their engineering. The concepts of these approaches were standardised under ISO/IEC 42010, *Systems and software engineering — Architecture description*. An Architecture Description is composed of Views, where each View addresses one or more engineering concerns. As mentioned in the standard, a multi-viewpoint approach requires the capacity to capture the different views, and maintain their mutual consistency. This paper addresses primarily the problem of integrating a model-based security risk assessment view to the mainstream system engineering view(s) and, to a lesser extent, the problem of maintaining the overall consistency of the views. Both business stakes and technical means are studied. We present two specific approaches, namely CORAS and Rinforzando. Both come with techniques and tool support to facilitate security risk assessment of complex and evolving critical infrastructures, such as ATM systems. The former approach offers static import/export relationships between artefacts, whereas the latter offers dynamic relationships. The pros and cons of each technical approach are discussed.

Jean Bézivin. "On the unification power of models". In: *Software & Systems Modeling* 4.2 (2005), pp. 171–188. DOI: [10.1007/s10270-005-0079-0](https://doi.org/10.1007/s10270-005-0079-0).

A. Bhave et al. "View Consistency in Architectures for Cyber-Physical Systems". In: *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*. 2011, pp. 151–160. DOI: [10.1109/ICCPs.2011.17](https://doi.org/10.1109/ICCPs.2011.17).

Annotations: Addresses consistency issues for a limited class of viewpoints, focused primarily on component/connector viewpoint and structural correspondences including what the authors refer to as control, physical, software, and hardware architectural styles: containment, attachment and binding. CC is used to form a base architecture, which is then decorated via various model (kinds). Introduces a notion, *strong consistency*, between views and architectures analogous to the Whole System principle in 42010.

Abstract: Current methods for modeling, analysis, and design of cyber-physical systems lack a unifying framework due to the complexity and heterogeneity of the constituent elements and their interactions. Our approach is to define relationships between system models at the architectural level, which captures the structural interdependencies and some semantic interdependencies between representations without attempting to comprehend all of the details of any particular modeling formalism. This paper addresses the issue of defining and evaluating consistency between architectural views imposed by various heterogeneous models and a base architecture (BA) for the complete system. This notion of structural consistency ensures that the model elements adhere to the cyber and physical types and the connections between components present in the BA, which serves as the unifying framework for model-based development. Consistency checking between a model and the underlying system architecture is formulated as a typed graph matching problem between the connectivity graphs of the corresponding architectural view and the system's BA. The usefulness of the approach to check system modeling assumptions is illustrated in the context of two heterogeneous views of a quad rotor air vehicle.

F. S. de Boer et al. "A Logical Viewpoint on Architectures". In: *8th International Enterprise Distributed Object Computing Conference (EDOC 2004), 20-24 September 2004, Monterey, California, USA, Proceedings*. IEEE Computer Society, 2004, pp. 73–83.

Annotations: Proposes to extend the IEEE 1471 conceptual model with “semantic models” and architecture signatures to bridge the gap between business process models and enterprise architectures.

Jan Bosch. “Software Architecture: The Next Step”. In: *Proceedings First European Workshop Software Architecture (EWSA 2004)*. Ed. by Flavio Oquendo, Brian Warboys, and Ron Morrison. Vol. 3047. Lecture Notes in Computer Science. Keywords: decisions. St Andrews, UK, May 21–22 2004: Springer Berlin/Heidelberg, 2004, pp. 194–199. DOI: [10.1007/978-3-540-24769-2_14](https://doi.org/10.1007/978-3-540-24769-2_14).

Abstract: This position paper makes the following claims that, in our opinion, are worthwhile to discuss at the workshop. 1) The first phase of software architecture research, where the key concepts are components and connectors, has matured the technology to a level where industry adoption is wide-spread and few fundamental issues remain. 2) The traditional view on software architecture suffers from a number of key problems that cannot be solved without changing our perspective on the notion of software architecture. These problems include the lack of first-class representation of design decisions, the fact that these design decisions are cross-cutting and intertwined, that these problems lead to high maintenance cost, because of which design rules and constraints are easily violated and obsolete design decisions are not removed. 3) As a community, we need to take the next step and adopt the perspective that a software architecture is, fundamentally, a composition of architectural design decisions. These design decisions should be represented as first-class entities in the software architecture and it should, at least before system deployment, be possible to add, remove and change architectural design decisions against limited effort.

Nelis Boucké. “Composition and relations of architectural models supported by an architectural description language”. PhD thesis. Katholieke Universiteit Leuven, Oct. 2009.

Annotations: Describes a framework and formalization of relations and compositions between architectural models (and views).

Nelis Boucké, Alessandro Garcia, and Tom Holvoet. “Composing structural views in xADL”. In: *Early Aspects: Current Challenges and Future Directions*. Lecture Notes in Computer Science 4765. 2007, pp. 115–138.

Nelis Boucké and Tom Holvoet. “View composition in multi-agent architectures”. In: *International Journal of Agent-Oriented Software Engineering* (2007).

Nelis Boucké et al. “Characterizing Relations between Views”. In: *Proceedings 2nd European Conference on Software Architecture (ECSA 2008)*. Ed. by Ron Morrison, Dharini Balasubramaniam, and Katrina Falkner. Lecture Notes in Computer Science 5292. 2008, pp. 66–81.

Annotations: Presents a taxonomy of mechanisms for view relations.

H. Bowman et al. “A formal framework for viewpoint consistency”. In: *Formal Methods in System Design*. 2002, pp. 111–166.

Manfred Broy et al. “Toward a Holistic and Standardized Automotive Architecture Description”. In: *Computer* 42 (2009), pp. 98–101.

Annotations: Describes an architecture framework for the automotive enterprise. See also: ftp://ftp.software.ibm.com/software/plm/resources/AAF_TUM_TRI0915.pdf.

J. Brunel et al. "A Viewpoint-Based Approach for Formal Safety & Security Assessment of System Architectures". In: *11th Workshop on Model-Driven Engineering, Verification and Validation*. Keywords: systems, safety, security, viewpoints, MBSE. 2014, pp. 39–48. URL: <https://hal.archives-ouvertes.fr/hal-01070960/document>.

Annotations: –.

Abstract: We propose an model-based approach to address safety and security assessment of a system architecture. We present an integrated process where system engineers design the model of the system architecture, safety and security engineers specify the propagation of failures and attacks inside each component of the architecture using their dedicated tool. They also define the failure modes that have to be merged from both disciplines. The underlying analyses are then performed using Alloy. We instantiate this approach with the system engineering tool Melody from Thales, and the risk analysis supporting tool Safety Architect from All4Tec. We illustrate this work on a system that implements a landing approach of an aircraft.

Hugo Bruneliere et al. "Scalable Model Views over Heterogeneous Modeling Technologies and Resources". In: *Software and Systems Modeling* (2020). DOI: [10.1007/s10270-020-00794-6](https://doi.org/10.1007/s10270-020-00794-6).

Annotations: –.

Abstract: When engineering complex systems, models are typically used to represent various systems aspects. These models are often heterogeneous in terms of modeling languages, provenance, number or scale. As a result, the information actually relevant to engineers is usually split into different kinds of interrelated models. To be useful in practice, these models need to be properly integrated to provide global views over the system. This has to be made possible even when those models are serialized or stored in different formats adapted to their respective nature and scalability needs. Model view approaches have been proposed to tackle this issue. They provide unification mechanisms to combine and query various different models in a transparent way. These views usually target specific engineering tasks such as system design, monitoring and evolution. In an industrial context, there can be many large-scale use cases where model views can be beneficial, in order to trace runtime and design-time data, for example. However, existing model view solutions are generally designed to work on top of one single modeling technology (even though model import/export capabilities are sometimes provided). Moreover, they mostly rely on in-memory constructs and low-level modeling APIs that have not been designed to scale in the context of large models stored in different kinds of data sources. This paper presents a general solution to efficiently support scalable model views over heterogeneous modeling resources possibly handled via different modeling technologies. To this intent, it describes our integration approach between a model view framework and various modeling technologies providing access to multiple types of modeling resources (e.g., in XML/XMI, CSV, databases). It also presents how queries on such model views can be executed efficiently by benefiting from the optimization of the different model technologies and underlying persistence backends. Our solution has been evaluated on a practical large-scale use case provided by the industry-driven European MegaM@Rt2 project that aims at implementing a runtime

↔

design time feedback loop. The corresponding EMF-based tooling support, modeling artifacts and reproducible benchmarks are all available online.

Sabine Buckl, Sascha Krell, and Christian M. Schweda. "A Formal Approach to Architectural Descriptions – Refining the ISO Standard 42010". In: *Advances in Enterprise Engineering IV*. Ed. by Antonia Albani and Jan L.G. Dietz. Vol. 49. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2010, pp. 77–91. DOI: [10.1007/978-3-642-13048-9_6](https://doi.org/10.1007/978-3-642-13048-9_6).

Abstract: Architectural descriptions representing and modeling the architecture of a system or parts thereof are typically used in the engineering disciplines to plan, develop, maintain, and manage complex systems. Primarily originating from construction engineering, the means of architecting and architectural descriptions have been successfully transferred to related disciplines like software engineering. While a rich and formal theory on conceptual modeling exists as well as frameworks on how to approach architectural descriptions, e.g. the ISO standard 42010, only few attempts have yet been made to integrate the prescriptions and guidelines from these sources into a formal architectural description framework. In this paper, we establish such a framework against the background provided by the ISO standard 42010 by formally defining the terms *concern*, *view*, *viewpoint*, and *architectural description*. Further, an outlook discusses potential application areas of the framework.

Trosky B. Callo Arias, Pierre America, and Paris Avgeriou. "Defining Execution Viewpoints for a Large and Complex Software-Intensive System". In: *Proceedings WICSA/ECSA 2009*. 2009.

Trosky B. Callo Arias, Pierre America, and Paris Avgeriou. "Defining and Documenting Execution Viewpoints for a Large and Complex Software-Intensive System". In: *Journal of Systems and Software* (2011). Keywords: viewpoints. DOI: [10.1016/j.jss.2010.11.908](https://doi.org/10.1016/j.jss.2010.11.908).

Trosky B. Callo Arias et al. "A Top-down Strategy to Reverse Architecting Execution Views for a Large and Complex Software-Intensive System: An Experience Report". In: *Science of Computer Programming* (2011). Keywords: viewpoints. DOI: [10.1016/j.scico.2010.11.008](https://doi.org/10.1016/j.scico.2010.11.008).

Mariam Chaabane et al. "A modeling approach for Systems-of-Systems by adapting ISO/IEC/IEEE 42010 Standard evaluated by Goal-Question-Metric". In: *Science of Computer Programming* 184 (2019). Keywords: systems of systems. DOI: [10.1016/j.scico.2019.102305](https://doi.org/10.1016/j.scico.2019.102305).

Annotations: –.

Damien Chapon and Guillaume Bouchez. "On the link between Architectural Description Models and Modelica Analyses Models". In: *Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009*. 2009, pp. 784–789. URL: <http://www.ep.liu.se/ecp/043/092/ecp09430079.pdf>.

Annotations: Describes an integrated development environment (IDE) for physical system architecting using concepts of IEEE 1471.

Antonio Cicchetti, Federico Ciccozzi, and Thomas Leveque. "A hybrid approach for multi-view modeling". In: *Electronic Communications of the EASST 50* (2011). Keywords: multiple

views, separation of concerns, model-driven engineering, model synchronization, pp. 1–12. URL: <http://www.easst.org/eceasst/>.

Annotations: A hybrid to the usual Projective–Synthetic distinction; synthetic: each view is implemented as a distinct metamodel and the overall system is obtained as synthesis of the information carried by the different views; projective: end-users are provided with virtual views made up of selected concepts coming from a single base metamodel by hiding details not relevant for the particular viewpoint taken into account.

Abstract: Multi-view modeling is a widely accepted technique to reduce the complexity in the development of modern software systems. It allows developers to focus on a narrowed portion of the specification dealing with a selected aspect of the problem. However, multi-view modeling support discloses a number of issues: on the one hand consistency management very often has to cope with semantics interconnections between the different concerns. On the other hand, providing a predefined set of views usually results as too restrictive because of expressiveness and customization needs. This paper proposes a hybrid solution for multi-view modeling based on an arbitrary number of custom views defined on top of an underlying modeling language. The aim is to benefit from the consistency by-construction granted by well-defined views while at the same time providing malleable perspectives through which the system under development can be specified.

Paul C. Clements. *Comparing the SEI's Views-and-Beyond Approach for Documenting Software Architectures with IEEE Std 1471-2000*. Tech. rep. Software Engineering Institute, 2005.

Paul C. Clements et al. *Documenting Software Architectures: views and beyond*. Addison Wesley, 2003.

Paul C. Clements et al. *Documenting Software Architectures: views and beyond*. 2nd. Addison Wesley, 2010.

Paul Clements and Len Bass. “The Business Goals Viewpoint”. In: *IEEE Software* 27 (2010). Keywords: viewpoints, pp. 38–45. DOI: [10.1109/MS.2010.116](https://doi.org/10.1109/MS.2010.116).

Abstract: Architectures come about through forces and needs other than those captured in traditional requirements documents. A business goal expresses why a system is being developed and what stakeholders in the developing organization, the customer organization, and beyond aspire to achieve through its production and use. Business goals can provide the rationale for requirements and help identify missing or superfluous requirements. Business goals can also influence architectures directly, even without affecting requirements at all. A business goals viewpoint can help architects and organizations capture their business goals in a precise and unambiguous form, which in turn will help architects design systems that are more responsive to organizational needs.

Paul Clements et al. “Aspects in Architectural Description: report on a first workshop at AOSD 2007”. In: *SIGSOFT Software Engineering Notes* 32.4 (2007). Keywords: aspects, pp. 33–35. DOI: [10.1145/1281421.1281440](https://doi.org/10.1145/1281421.1281440).

Ryan Crichton et al. “An Architecture and Reference Implementation of an Open Health Information Mediator: Enabling Interoperability in the Rwandan Health Information Exchange”. In: *Foundations of Health Information Engineering and Systems*. Ed. by Jens Weber and Isabelle Perseil. Vol. 7789. Lecture Notes in Computer Science. Keywords: interoperability; national health information system architecture; enterprise service bus; health in-

formation exchange. Springer Berlin Heidelberg, 2013, pp. 87–104. DOI: [10.1007/978-3-642-39088-3_6](https://doi.org/10.1007/978-3-642-39088-3_6).

Abstract: Rwanda, one of the smallest and most densely populated countries in Africa, has made rapid and substantial progress towards designing and deploying a national health information system. One of the more challenging aspects of the system is the design of an architecture to support: interoperability between existing health information systems already in use in the country; incremental extension into a fully integrated national health information system without substantial re-engineering; and scaling, from a single district in the initial phase, to national level without requiring a fundamental change in technology or design paradigm. This paper describes the key requirements and the design of the current architecture using the ISO/IEC/IEEE 42010 standard architecture descriptions. The architecture takes an Enterprise Service Bus approach. A partial implementation and preliminary analysis of the architecture is given. Since these challenges are experienced by other developing African countries, the next steps involves creating a generic architecture that can be reused for health information exchange in other developing African countries.

Ashesh Das, Sandra Gorka, and Jacob Miller. “Designing Multidisciplinary Capstone Courses—A Knowledge Engineering Approach”. In: *Proceedings of the IEEE Southeastern Conference (IEEE SECON-09)*. March 5-8, 2009. Atlanta, Georgia, Mar. 2009.

Annotations: Uses IEEE 1471 concepts to conduct knowledge engineering on multidisciplinary course and curriculum design.

John Derrick, Howard Bowman, and Maarten Steen. “Viewpoints and Objects”. In: *Ninth Annual Z User Workshop*. Ed. by J. P. Bowen and M. G. Hinchey. Vol. 967. Lecture Notes in Computer Science. Springer-Verlag, Sept. 1995, pp. 449–468. URL: <http://www.cs.kent.ac.uk/pubs/1995/188/content.gz>.

Annotations: Tackles issues of inter-view consistency via unification in a multiple viewpoint setting based on RM-ODP.

Abstract: There have been a number of proposals to split the specification of large and complex systems into a number of inter-related specifications, called viewpoints. Such a model of multiple viewpoints forms the cornerstone of the Open Distributed Processing (ODP) standardisation initiative. We address two of the technical problems concerning the use of formal techniques within multiple viewpoint models: these are unification and consistency checking. We discuss the software engineering implications of using viewpoints, and show that object encapsulation provides the necessary support for such a model. We then consider how this might be supported by using object-oriented variants of Z.

A. van Deursen et al. “Symphony: View-Driven Software Architecture Reconstruction”. In: *Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture*. 2004, pp. 122–134.

Annotations: Symphony is a viewpoint-driven approach to reconstruction of software architectures.

Davide Di Ruscio et al. “Developing next generation ADLs through MDE techniques”. In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE 2010)*. Vol. 1. Keywords: ADLs, MDE. ACM, 2010, pp. 85–94.

Davide Di Ruscio et al. "Model-driven techniques to enhance Architectural Languages Interoperability". In: *15th International Conference on Fundamental Approaches to Software Engineering (FASE)*. Keywords: ADLs, MDE. 2012.

Annotations: -.

Hylke W. van Dijk. "Democratic Processing: Mastering the complexity of communicating systems". PhD thesis. Delft University of Technology, 2004.

Annotations: Uses IEEE 1471 conceptual framework as starting point for an ontology of complex communications and quality of service.

Remco M. Dijkman. "Consistency in multi-viewpoint architectural design". PhD thesis. University of Twente, 2006. URL: <http://www.utwente.nl/ewi/asna/research/Ph.D.\%20Theses/dijkman-thesis.pdf>.

Remco M. Dijkman et al. "An Approach to Relate Viewpoints and Modeling Languages". In: *Proceedings of the 7th International Enterprise Distributed Object Computing Conference (EDOC 2003)*. Brisbane, Australia, 2003, pp. 14–27. URL: http://wwwhome.cs.utwente.nl/~sinderen/publications/pubs_2003/edoc-dijkman03.pdf.

Annotations: This paper proposes the use of a basic viewpoint as a basis for defining and relating viewpoints for distributed application design.

Abstract: The architectural design of distributed enterprise applications from the viewpoints of different stakeholders has been proposed for some time, for example, as part of RM-ODP and IEEE 1471, and seems now-a-days to gain acceptance in practice. However, much work remains to be done on the relationships between different viewpoints. Failing to relate viewpoints may lead to a collection of viewpoint models that is inconsistent and may, therefore, lead to an incorrect implementation. This paper defines an approach that helps designers to relate different viewpoints to each other. Thereby, it helps to enforce the consistency of the overall design. The results of this paper are expected to be particularly interesting for Model Driven Architecture (MDA) projects, since the proposed approach can be used for the explicit definition of the models and relationships between models in an MDA trajectory.

Edsger W. Dijkstra. *On the role of scientific thought*. Reprinted in *Selected writings on computing: a personal perspective* (1982). 1974. URL: <http://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html>.

Annotations: The use of concerns in IEEE 1471 derives from the phrase *separation of concerns* in software engineering. The earliest use of this phrase appears to be in this 1974 paper by Dijkstra: "Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects. We know that a program must be correct and we can study it from that viewpoint only; we also know that it should be efficient and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether, and if so: why, the program is desirable. But nothing is gained—on the contrary!—by tackling these various aspects simultaneously. It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by 'focussing one's attention upon some aspect': it does not mean ignoring

the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously."

Pierre Dissaux et al., eds. *Architecture Description Languages: IFIP TC-2 Workshop on Architecture Description Languages (WADL), World Computer Congress, Aug. 22-27, 2004, Toulouse, France*. Keywords: ADLs, AADL. Springer, 2004.

Annotations: –.

Hugo ter Doest et al. *Viewpoints Functionality and Examples*. Tech. rep. TI/RS/2003/091. Telematica Instituut, 2004. URL: <https://doc.telin.nl/dscgi/ds.py/Get/File-35434>.

Annotations: Describes ArchiMate's approach to the definition and presentation of enterprise architecture viewpoints, a classification of viewpoints; based upon the IEEE 1471 frame of reference.

Peter Eeles and Peter Cripps. *The Process of Software Architecting*. Addison Wesley, 2010. URL: <http://processofsoftwarearchitecting.com>.

Annotations: Defines a process for software architects, using the IEEE 1471 model as a foundation. Provides a viewpoint template and viewpoint catalog including: Requirements, Functional, Deployment, Validation, Application, Infrastructure, Systems Management, Availability, Performance, Security; and the work products (model kinds) used in each.

Alexander Franz Egyed. "Heterogeneous View Integration, and its Automation". PhD thesis. USC, 2000.

Walter J. Ellis et al. "Toward a Recommended Practice for Architectural Description". In: *Proceedings of 2nd IEEE International Conference on Engineering of Complex Computer Systems, Montreal, Quebec, Canada, October 21-25, 1996*. 1996.

Annotations: First account of the goals and requirements for IEEE 1471.

David E. Emery. "Architectural Frameworks: Defining the Contents of Architectural Descriptions". In: *Reliable Software Technologies—Ada-Europe '99*. Ed. by Michael González Harbour and Juan A. Puente. Vol. 1622. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pp. 64–75. DOI: [10.1007/3-540-48753-0_6](https://doi.org/10.1007/3-540-48753-0_6).

Abstract: This paper describes experiences with several architectural frameworks. An "architectural framework" specifies what is included in the description of an architecture, independent of the specific system being described. The three frameworks are the U.S. DoD C4ISR Architecture Framework, the associated Core Architecture Data Model and the emerging IEEE Recommended Practice on Architecture Description. From these experiences, we speculate on the further evolution of architecture frameworks and architectural descriptions.

David E. Emery, Rich Hilliard, and Timothy B. Rice. "Experiences Applying a Practical Architectural Method". In: *Reliable Software Technologies—Ada-Europe '96*. Ed. by Alfred Strohmeier. Lecture Notes in Computer Science 1088. Springer, 1996. URL: <http://web.mit.edu/richh/www/writings/index.html#Experiences>.

Annotations: One of the architectural methods motivating the development of IEEE 1471.

David Emery and Rich Hilliard. "Every Architecture Description Needs a Framework: Expressing Architecture Frameworks Using ISO/IEC 42010". In: *Proceedings of the 2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture (WICSA/ECISA 2009)*. Ed. by Rick Kazman et al. IEEE Computer Society Press, 2009, pp. 31–40. DOI: [10.1109/WICSA.2009.5290789](https://doi.org/10.1109/WICSA.2009.5290789).

David Emery and Rich Hilliard. "Updating IEEE 1471". In: *Proceedings of the 7th Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*. IEEE Computer Society, Feb. 2008, pp. 303–306.

Annotations: Overview of the joint IEEE and ISO revision.

David Emery, Rich Hilliard, and Mark Maier. *ISO/IEC/IEEE 42010: Frequently Asked Questions*. Nov. 2013. URL: <http://www.iso-architecture.org/42010/faq.html>.

Romina Eramo et al. "A model-driven approach to automate the propagation of changes among Architecture Description Languages". In: *Software & Systems Modeling* 11.1 (2012). Keywords: ADLs, MDE, pp. 29–53.

Boulanger F. et al. "Modeling Heterogeneous Points of View with ModHel'X". In: *Models in Software Engineering. MODELS 2009*. Ed. by S. Ghosh. Vol. 6002. Lecture Notes in Computer Science. Keywords: NFPs, multiple views, heterogeneous models, synthetic approach. Springer, 2010. DOI: [10.1007/978-3-642-12261-3_29](https://doi.org/10.1007/978-3-642-12261-3_29).

Annotations: Argues for the synthetic approach to view and model composition with examples of non-functional concerns from systems engineering.

Abstract: Non-functional properties (NFPs) concern various characteristics of a system (cost, power, QoS). These characteristics belong to different models of the system, built by different design teams, using different formalisms. Therefore, the design of a system includes a number of domain-specific modeling languages, used to express various NFPs. This paper focuses on the heterogeneity of the points of view on the system. We show that "multi-view" approaches which do not rely on a unique underlying model appear better-suited to express NFPs than model weaving or annotations. However, existing approaches in this category do not yet support model execution. We introduce a multi-view extension to Mod-Hel'X, a framework for executing heterogeneous models, and we show how it can be used for modeling non-functional characteristics of a system and expressing NFPs. A key point of this approach is that it relies only on the core concepts of ModHel'X, but uses them in new ways.

Rik Farenhorst and Remco C. de Boer. *Architectural knowledge management: supporting architects and auditors*. VU University, 2009.

Annotations: Two dissertations on architectural knowledge, built on the IEEE 1471 ontology. Yields useful insights into architectural decisions incorporated into ISO/IEC 42010 revision.

Klaus Fischer et al. "Viewpoint-Based Modeling—Towards Defining the Viewpoint Concept and Implications for Supporting Modeling Tools". In: *International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA-12), September 13–14, Wien, Austria*. Vol. 206. Lecture Notes in Informatics. 2012, pp. 123–136.

Annotations: -.

ystein D. Fjeldstad et al. "The Architecture of Collaboration". In: *Strategic Management Journal* 33 (2012), pp. 734–750. DOI: [10.1002/smj.1968](https://doi.org/10.1002/smj.1968).

Annotations: Using architecture concepts from IEEE 1471 to explore collaboration.

Pascal Fradet, Daniel Le Métayer, and Michaël Périn. "Consistency checking for multiple view software architectures". In: *Proceedings ESEC/FSE'99*. Springer, 1999.

R. Edward Freeman. *Strategic Management: a Stakeholder Approach*. Boston: Pittman, 1984.

Annotations: First introduction of *stakeholder* into management thinking.

Cristina Gacek et al. "On the definition of software system architecture". In: *Proceedings of the First International Workshop on Architectures for Software Systems*. Seattle, WA, 1995.

Annotations: One of the sources motivating the introduction of the notion of *stakeholder* into IEEE 1471.

Matthias Galster and Paris Avgeriou. "A Variability Viewpoint for Enterprise Software Systems". In: *Proceedings of the Joint 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA)*. Keywords: viewpoints. IEEE Computer Society, 2012.

Matthias Galster and Paris Avgeriou. "Empirically-grounded reference architectures: a proposal". In: *Joint ACM SIGSOFT Conference on Quality of Software Architectures (QoSA) and Architecting Critical Systems (ISARCS)*. Keywords: reference architecture, variability. 2011, pp. 153–158.

Annotations: –.

Matthias Galster, Paris Avgeriou, and Dan Tofan. "Constraints for the design of variability-intensive service-oriented reference architectures – An industrial case study". In: *Information and Software Technology* 55.2 (2013). Keywords: reference architecture, variability.

Annotations: –.

David Garlan et al. *An Activity Language for the ADL Toolkit*. Tech. rep. CMU Computer Science Department, 2000. URL: <http://repository.cmu.edu/compsci/694>.

Jeff Garland and Richard Anthony. *Large Scale Software Architecture: A Practical Guide Using UML*. John Wiley and Sons, 2002.

Annotations: Defines fourteen architecture viewpoints for use with UML.

Holger Giese and Alexander Vilbig. "Separation of non-orthogonal concerns in software architecture and design". In: *Software & Systems Modeling* 5.2 (2006). Keywords: separation of concerns; software architecture; consistency; behavior synthesis; design by contract, pp. 136–169. ISSN: 1619-1366. DOI: [10.1007/s10270-005-0103-4](https://doi.org/10.1007/s10270-005-0103-4).

Abstract: Separation of concerns represents an important principle for managing complexity in the design and architecture of large component-based software systems. The fundamental approach is to develop local solutions for individual concerns first, and combine them later into an overall solution for the complete system. However, comprehensive support for the integration of interdependent, possibly conflicting concerns related to synchronization behavior is still missing. In our work, we propose a sound

solution for this complex type of composition, employing well-known UML description techniques as well as a rigorous formal model of component synchronization behavior. Based on this foundation, we describe a constructive synthesis algorithm which reliably detects conflicting concerns or generates a maximal synchronization behavior for software components with multiple interactions. An optimized implementation of the algorithm has been integrated into a CASE tool to illustrate feasibility and scalability of the presented technique to the example of a moderately large case study.

Simon Giesecke, Jasminka Matevska, and Wilhelm Hasselbring. "Extending ANSI/IEEE Standard 1471 for Representing Architectural Rationale". In: *Proceedings of the 4th Nordic Workshop on the Unified Modeling Language and Software Modeling (NWUML'06), Grimstad, Norway*. Ed. by Merete Skjelten Prinz Andreas; Tveit. Agder University College, 2006. URL: http://grimstad.hia.no/nwuml06/Papers/Giesecke_Matevska_Hasselbring.pdf.

Thomas Goldschmidt, Steffen Becker, and Erik Burger. "Towards a tool-oriented taxonomy of view-based modelling". In: *Modellierung 2012*. Ed. by Elmar J. Sinz and Andy Schrr. Bonn: Gesellschaft fr Informatik e.V., 2012, pp. 59–74.

Annotations: –.

J. Gordijn, J.M. Akkermans, and J.C. van Vliet. "Business Modelling is not Process Modelling". In: *Conceptual Modeling for E-Business and the Web*. Vol. 1921. Lecture Notes in Computer Science. Springer, 2000, pp. 40–51.

Annotations: Suggests constructs distinct from process modeling toward the definition of a 'business' or 'commerce' viewpoint.

J. Gordijn, H. de Bruin, and J.M. Akkermans. "Scenario Methods for Viewpoint Integration in e-Business Requirements Engineering". In: *Proceedings of the 34th Hawaii International Conference On System Sciences*. IEEE, 2001. URL: <http://csdl2.computer.org/comp/proceedings/hicss/2001/0981/07/09817032.pdf>.

Annotations: Multiple viewpoint modeling for commerce-related architectural concerns.

K.A. de Graaf et al. "An exploratory study on ontology engineering for software architecture documentation". In: *Computers in Industry* (2014). DOI: [10.1016/j.compind.2014.04.006](https://doi.org/10.1016/j.compind.2014.04.006).

Danny Greefhorst, Henk Koning, and Hans van Vliet. "The many faces of architectural descriptions". In: *Information Systems Frontiers* 8 (2006), pp. 103–113. DOI: [10.1007/s10796-006-7975](https://doi.org/10.1007/s10796-006-7975). URL: <http://www.cs.vu.nl/~hans/publications/y2006/facesISF.pdf>.

Annotations: Surveys 23 architecture frameworks and proposes 9 dimensions for classifying frameworks: Type of information, Scope, Detail level, Stakeholder, Transformation, Quality attribute, Meta level, Nature and Representation.

Cyber Physical Systems Public Working Group. *Framework for Cyber-Physical Systems*. Tech. rep. Release 1.0. Keywords: cyber-physical systems, concerns, aspects, facets. NIST, 2016.

Annotations: "Aspects are high-level groupings of cross-cutting concerns. Concerns are interests in a system relevant to one or more stakeholders. The identified aspects are[:]"

Functional, Business, Human, Trustworthiness*, Timing, Data, Boundaries, Composability, [and] Lifecycle” “Trustworthiness includes security, privacy, safety, reliability, and resilience.”

“Facets are views on CPS encompassing identified responsibilities in the system engineering process. There are three identified facets: Conceptualization, Realization and Assurance.”

Paul Gruenbacher, Alexander Egyed, and Nenad Medvidovic. “Dimensions of Concerns in Requirements Negotiation and Architecture Modeling”. In: *Proceedings of the 2nd Workshop on Multi-Dimensional Separation of Concerns (MDSOC)*. 2000. URL: <http://www.alexander-egyed.com/publications/>.

Qing Gu. “Guiding Service-Oriented Software Engineering – A View-based Approach”. Keywords: viewpoints. PhD thesis. Vrije Universiteit Amsterdam, 2011. URL: http://www.cs.vu.nl/en/Images/Q\%20Gu\%2006-10-2011_tcm75-259548.pdf.

Qing Gu et al. “3D Architecture Viewpoints on Service Automation”. In: *Journal of Systems and Software* 86.5 (May 2013). Keywords: viewpoints, pp. 1307–1322. DOI: [10.1016/j.jss.2012.12.035](https://doi.org/10.1016/j.jss.2012.12.035).

Annotations: Introduces three viewpoints: Decision, Degree and Data for framing service automation concerns in architecting service-based applications.

M. Guessi, F. Oquendo, and E. Y. Nakagawa. “Ark: a constraint-based method for architectural synthesis of smart systems”. In: *Software and Systems Modeling* (2019). DOI: [10.1007/s10270-019-00764-7](https://doi.org/10.1007/s10270-019-00764-7).

Annotations: –.

Milena Guessi, Flavio Oquendo, and Elisa Yumi Nakagawa. “Variability viewpoint to describe reference architectures”. In: *1st International Conference on Dependable and Secure Cloud Computing Architecture*. 2014, pp. 1–6.

Christophe Guychard et al. “Conceptual interoperability through Models Federation”. In: *Semantic Information Federation Community Workshop*. Keywords: complex systems, viewpoints. Miami, United States, Oct. 2013. URL: <https://hal.archives-ouvertes.fr/hal-00905036>.

Abstract: Successful architecting of complex systems requires reconciling heterogeneous viewpoints expressed by the stakeholders involved in the development process, including domain and technical experts, users and managers. Most of the time, each concern is analyzed by experts using well-fitted specific tools to produce their point of view on a solution. This results in a set of models with various technical spaces, formalisms and paradigms. Ensuring global consistency, maintaining traceability and building cross-concerns views in that context is challenging. In order to address this issues, we initiated the development of a tooling that provides support for building conceptual views expanding upon existing models and tools. It has been applied to uses cases such as: model composition across technical spaces, heterogeneous (meta)models alignment and keeping models in sync. In this paper, we introduce the models federation approach to conceptual interoperability that drives the development of our innovative modeling engine.

K. Eric Harper and Jiang Zheng. “Exploring Software Architecture Context”. In: *WICSA*

2015. Keywords: viewpoint, concerns, decisions. IEEE Computer Society, Apr. 2015, pp. 123–126.

Annotations: “extend[s] the decision Forces Viewpoint to capture detailed design context descriptions, and add features for tagging the architecture description elements to facilitate identification of commonality, classification, and specialization.”

Manfred Hauswirth, Mehdi Jazayeri, and Markus Schneider. “A phase model for e-commerce business models and its application to security assessment”. In: *Proceedings of the 34th Hawaii International Conference on System Sciences*. Jan. 2001. URL: <http://lsirpeople.epfl.ch/hauswirth/papers/EC-Security/EC-Security.pdf>.

R. Heckel et al. “A view-based approach to system modeling based on open graph transformation systems”. In: *Handbook of Graph Grammars and Computing by Graph Transformation*. Ed. by Grzegorz Rozenberg. Keywords: viewpoints, graphs, views. World Scientific, 1999, pp. 639–668.

Annotations: –.

U. van Heesch et al. “Platform design space exploration using architecture decision viewpoints—A longitudinal study”. In: *Journal of Systems and Software* 124 (Feb. 2017), pp. 56–81.

Annotations: -.

Uwe van Heesch. “Architecture decisions: the next step : understanding, modeling, supporting and reviewing architecture decisions”. PhD thesis. University of Groningen, Nov. 2012. URL: <http://irs.ub.rug.nl/ppn/35303827X>.

Abstract: Software architecture is the result of a set of architecture decisions. Unfortunately, there is currently no commonly accepted approach to architecture decision modeling. Existing approaches do not satisfy all stakeholder concerns in decision description; they do not optimally support the architecting process, and they do not integrate well with the rest of the architecture documentation, which is usually arranged in multiple architectural views. This dissertation reports on multiple empirical studies conducted to understand better the decision making process in practice. The core contribution is a framework for architecture decisions, following the conventions of the international architecture description standard ISO/IEC/IEEE 42010. The framework consists of five interrelated viewpoints, each of which being dedicated to satisfying different stakeholder concerns in architecture decisions. The viewpoints of the framework can be used individually, or in combination, to describe the architecture decisions made in a software project. To find out if decision viewpoints can support designers in making rational decisions, we conducted a comparative multiple-case study with four groups of senior software engineering students. The results confirm that students who create decision views according to the viewpoint definition explore and evaluate candidate architectural solutions more systematically than student groups who do not use the decision framework. Finally, this dissertation reports on a lightweight decision-centric architecture evaluation method, which uses viewpoints from the decision framework. The method uncovers and evaluates the rationale behind the most important architecture decisions made in a software project, considering all relevant forces that must be addressed by the decisions.

Uwe van Heesch, Paris Avgeriou, and Rich Hilliard. “A Documentation Framework for Architecture Decisions”. In: *The Journal of Systems & Software* 85.4 (Apr. 2012). Keywords: viewpoints, pp. 795–820. DOI: [10.1016/j.jss.2011.10.017](https://doi.org/10.1016/j.jss.2011.10.017).

Abstract: In this paper, we introduce a documentation framework for architecture decisions. This framework consists of four viewpoint definitions using the conventions of ISO/IEC/IEEE 42010, the new international standard for the description of system and software architectures. The four viewpoints, a Decision Detail viewpoint, a Decision Relationship viewpoint, a Decision Chronology viewpoint, and a Decision Stakeholder Involvement viewpoint satisfy several stakeholder concerns related to architecture decision management.

With the exception of the Decision Stakeholder Involvement viewpoint, the framework was evaluated in an industrial case study. The results are promising, as they show that decision views can be created with reasonable effort while satisfying many of the stakeholder concerns in decision documentation.

Uwe van Heesch, Paris Avgeriou, and Rich Hilliard. "Forces on Architecture Decisions – A Viewpoint". In: *Proceedings of the Joint 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA)*. Keywords: viewpoints. IEEE Computer Society, 2012.

Angenita Heijmans. "An Architectural Viewpoint for Conceptualization". MA thesis. Radboud University Nijmegen, Aug. 2002. URL: <http://www.cs.ru.nl/onderwijs/afstudereninfo/scripties/2002/509.Heijmans.pdf>.

Rogardt Heldal et al. "Descriptive vs Prescriptive Models in Industry". In: *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS 2016)*. 16.

Annotations: -.

Rich Hilliard. *All About ISO/IEC/IEEE 42010*. Apr. 2014. URL: <http://www.slideshare.net/dJdU/all-about-isoiecieee-42010-2014r5>.

Rich Hilliard. "IEEE Std 1471 and Beyond". In: *Workshop on Software Architecture Representation, 16–17 January 2001*. Software Engineering Institute, 2001. URL: <http://www.sei.cmu.edu/publications/documents/01.reports/01sr010.html>.

Annotations: Discussion of some open issues with respect to the use of IEEE 1471, after its standardization.

Rich Hilliard. "ISO/IEC 42010 née IEEE Std 1471". In: *Documenting software architectures: views and beyond*. Ed. by Paul Clements et al. 2nd. Addison Wesley, 2011, pp. 400–405.

Rich Hilliard. *ISO/IEC/IEEE 42010 Annotated Bibliography*. This document, because what architect doesn't love self-reference? 2018. URL: <http://www.iso-architecture.org/42010/>.

Rich Hilliard. *Impact Assessment of IEEE Std 1471 on The Open Group Architecture Framework*. Tech. rep. The Open Group, 2000. URL: <http://web.mit.edu/richh/www/writings/index.html#IEEE-1471-TOGAF>.

Annotations: Discusses impact of adopting IEEE 1471 on The Open Group's Architecture Framework (TOGAF).

Rich Hilliard. "Understanding Architectural Perspectives". Unpublished note. Mar. 2005. URL: <http://web.mit.edu/richh/www/writings/index.html#hilliard-up>.

Annotations: Response to Woods, Emmerich and Rozanski's "Using architectural perspectives" in light of the conceptual framework of IEEE 1471.

Rich Hilliard. "Using aspects in architectural description". In: *Early Aspects: Current Challenges and Future Directions*. Vol. 4765. Lecture Notes in Computer Science. Springer, 2007, pp. 139–154.

Abstract: This paper sketches an approach to using aspects for architectural description within the conceptual framework of IEEE 1471. I propose a definition of architectural aspect within that framework and examine its consequences and motivations. I show that architectural aspects can be accommodated within the current conceptual framework of IEEE 1471 without modification; and outline extensions to the framework which could be candidates for further standardization work, or incorporated into aspect-oriented architectural methods.

Rich Hilliard. "Viewpoint Modeling". In: *First ICSE Workshop on Describing Software Architecture with UML*. Position paper. May 2001.

Rich Hilliard. "Views and viewpoints in software systems architecture". In: *First Working IFIP Conference on Software Architecture*. Position paper. San Antonio, Feb. 1999. URL: <http://web.mit.edu/richh/www/writings/index.html#Hilliard99>.

Rich Hilliard and Timothy B. Rice. "Comments on C4ISR Architecture Framework". MITRE Corporation memo D510-M-013, dated 5 June 1997. June 1997. URL: <http://web.mit.edu/richh/www/writings/index.html#C4ISR-Cmnts>.

Annotations: The C4ISR Architecture Framework was a proposed approach to documenting architectures for the DoD. This memo provides detailed comments on version 1.0 of the framework. Subsequent versions of the framework are now known as the DoD Architecture Framework (DoDAF). The latest version has not fixed the problems cited in this memo.

Rich Hilliard, Timothy B. Rice, and Stephen C. Schwarm. "The Architectural Metaphor as a Foundation for Systems Engineering". In: *Proceedings of Sixth Annual International Symposium of the International Council on Systems Engineering*. 1996. URL: <http://web.mit.edu/richh/www/writings/index.html\#Hilliard-Rice-Schwarm96>.

Annotations: An early attempt to apply some of the ideas of IEEE 1471 to systems engineering.

Rich Hilliard et al. "On the Composition and Reuse of Viewpoints across Architecture Frameworks". In: *Proceedings of the Joint 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA)*. Keywords: viewpoint. Helsinki, Finland: IEEE Computer Society, 2012.

Rich Hilliard et al. "On the Composition and Reuse of Viewpoints across Architecture Frameworks". In: *10th Working IEEE/IFIP Conference on Software Architecture (WICSA) & 6th European Conference on Software Architecture (ECSA) - WICSA/ECSA 2012*. Keywords: ADLs, MDE. 2012.

Annotations: -.

Rich Hilliard et al. "Realizing architecture frameworks through megamodeling techniques". In: *25th IEEE/ACM International Conference on Automated Software Engineering*

(ASE 2010). Keywords: viewpoints, ADLs, MDE. 2010. URL: <http://megaf.di.univaq.it/>.

Annotations: Describes tools to support definition of architecture frameworks and their viewpoints based on 42010 model.

Abstract: -.

Christine Hofmeister, Robert L. Nord, and Dilip Soni. *Applied Software Architecture*. Addison-Wesley, 2000.

Annotations: One of the architecture methods motivating IEEE 1471's approach.

Christine Hofmeister, Robert L. Nord, and Dilip Soni. "Describing software architectures with UML". In: *Proceedings of the First Working IFIP Conference on Software Architecture*. Ed. by Patrick Donohoe. Kluwer Academic Publishers, 1999, pp. 145–160.

Christine Hofmeister et al. "A general model of software architecture design derived from five industrial approaches". In: *The Journal of Systems and Software* 80.1 (2007), pp. 106–126. DOI: [10.1016/j.jss.2006.05.024](https://doi.org/10.1016/j.jss.2006.05.024).

IEEE Std 1471, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Oct. 2000.

Annotations: Withdrawn, with the publication of ISO/IEC/IEEE 42010:2011.

ISO/IEC 42010:2007, Systems and software engineering — Recommended practice for architectural description of software-intensive systems. ISO. July 2007.

Annotations: Withdrawn, with the publication of ISO/IEC/IEEE 42010:2011.

ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description. Dec. 2011, pp. 1–46.

Anton Jansen, Paris Avgeriou, and Jan Salvador van der Ven. "Enriching software architecture documentation". In: *Journal of Systems and Software* 82.8 (Aug. 2009). Keywords: architectural knowledge, design decisions, design rationale, documentation, software architecture, pp. 1232–1248. DOI: [10.1016/j.jss.2009.04.052](https://doi.org/10.1016/j.jss.2009.04.052).

Anton Jansen and Jan Bosch. "Software Architecture as a Set of Architectural Design Decisions". In: *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture*. WICSA '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 109–120. DOI: [10.1109/WICSA.2005.61](https://doi.org/10.1109/WICSA.2005.61).

Anton Jansen, Jan Bosch, and Paris Avgeriou. "Documenting after the fact: Recovering architectural design decisions". In: *Journal of Systems and Software* 81 (2008). Keywords: software, architecture recovery, design decisions, pp. 536–557.

Annotations: -.

Abstract: Software architecture documentation helps people in understanding the software architecture of a system. In practice, software architectures are often documented after the fact, i.e. they are maintained or created after most of the design decisions have been made and implemented. To keep the architecture documentation up-to-date an architect needs to recover and describe these decisions.

This paper presents ADDRA, an approach an architect can use for recovering architectural design decisions after the fact. ADDRA uses architectural deltas to provide the architect with clues about these design decisions. This allows the architect to systematically recover and document relevant architectural design decisions. The recovered architectural design decisions improve the documentation of the architecture, which increases traceability, communication, and general understanding of a system.

Mehdi Jazayeri and Ivana Podnar. "A Business and Domain Model for Information Commerce". In: *Proceedings of the 34th Hawaii International Conference on System Sciences*. 2001. URL: <http://lsirpeople.epfl.ch/podnar/papers/HICSS34.pdf>.

Henk Jonkers et al. "Towards a Language for Coherent Enterprise Architecture Descriptions". In: *Proceedings of the 7th International Enterprise Distributed Object Computing Conference (EDOC 2003)*. Brisbane, Australia: IEEE Computer Society, 2003, pp. 28–39. DOI: [10.1109/EDOC.2003.1233835](https://doi.org/10.1109/EDOC.2003.1233835).

Mohamed M. Kandé. "A Concern-oriented Approach to Software Architecture". These n. 2796. PhD thesis. École Polytechnique Fédéral de Lausanne, 2003.

Mohamed M. Kandé et al. "Bridging the Gap between IEEE Std 1471, Architecture Description Languages and UML". In: *Journal on Software and Systems Modeling* 1.2 (2002), pp. 113–129.

Mika Katara and Shmuel Katz. "A concern architecture view for aspect-oriented software design". In: *Softw. Syst. Model* 6 (2007), pp. 247–265.

Abstract: Although aspect-oriented programming is becoming popular, support for the independent description of aspect designs and for the incremental design of aspects themselves has been neglected. A conceptual framework for the design of aspects is presented, where aspects are viewed as augmentations that map an existing design into a new one with changes or additions. The principles of a Concern Architecture model are defined both to group designs of aspects, and to make explicit their dependencies and potential interferences in the design of a system with multiple aspects. The aspects are described generically, where any design element can be either required or provided. The required elements resemble formal parameters, and their binding to an existing design shows the context in which the provided parts are to modify that design. Overlap and a partial order among aspects and concerns are visualized in a Concern Architecture Diagram. An instantiation of the ideas as a UML profile is outlined, and the design of a digital sound recorder is used to demonstrate the utility of the approach.

Mika Katara and Shmuel Katz. "Architectural views of aspects". In: *AOSD '03 Proceedings of the 2nd international conference on Aspect-oriented software development*. Keywords: aspects, software. 2003, pp. 1–10.

Abstract: Support for the incremental design of aspects themselves has been neglected, even as the use of aspects in conjunction with underlying systems is gaining acceptance. The ways in which aspects can cooperate or interfere with each other need to be made explicit at the design level. An aspect architecture, a new software architecture viewpoint, is proposed, and its general principles are explained. An instantiation for extending UML is described, where designs of aspects provide maximal flexibility, and a new concern diagram is provided to show how aspects can be combined to treat different concerns of a system. An example shows aspect architecture views of a digital sound recorder.

J. Klein and H. van Vliet. "System-of-Systems Viewpoint for System Architecture Documentation". In: *ArXiv* (Jan. 2018). Keywords: computer science, software engineering, architecture description, architecture viewpoints. URL: <https://arxiv.org/abs/1801.06837>.

Alexander Knapp and Till Mossakowski. "Multi-view Consistency in UML: A Survey". In: *Ehrig Festschrift*. Ed. by R. Heckel and Taentzer G. Vol. 10800. Incs. Keywords: correspondences. Springer, 2018, pp. 37–60. DOI: [10.1007/978-3-319-75396-6_3](https://doi.org/10.1007/978-3-319-75396-6_3).

Annotations: –.

Hendrik Koning. "Communication of IT-Architecture". PhD thesis. Universiteit Utrecht, 2008.

Annotations: Builds upon the IEEE 1471 ontology to develop a set of 158 guidelines for improving the readability of IT architectures. Proposes a method to define IEEE 1471 viewpoints. Also surveys 23 architecture frameworks and presents 9 base dimensions that structure architecture descriptions: Type of information, Scope, Detail level, Stakeholder, Transformation, Quality attribute, Meta level, Nature and Representation.

Hendrik Koning, Rik Bos, and Sjaak Brinkkemper. "An Inquiry Tool for Stakeholder Concerns of Architectural Viewpoints: a Case Study at a Large Financial Service Provider". In: *Proceedings of the 10th International Enterprise Distributed Object Computing Conference Workshops*. Washington, DC, USA: IEEE Computer Society, 2006, p. 31. DOI: [10.1109/EDOCW.2006.19](https://doi.org/10.1109/EDOCW.2006.19).

Hendrik Koning and Hans van Vliet. "Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns". In: *Automated Software Engineering* 13.2 (2006), pp. 201–223. DOI: [10.1007/s10515-006-7736-6](https://doi.org/10.1007/s10515-006-7736-6).

Henk Koning and Hans van Vliet. "Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns". In: *Automated Software Engineering* 13 (2006), pp. 201–223. DOI: [10.1007/s10515-006-7736-6](https://doi.org/10.1007/s10515-006-7736-6).

Rainer Koschke and Daniel Simon. "Hierarchical Reflexion Models". In: *Proceedings of the 10th Working Conference on Reverse Engineering*. WCRE '03. Keywords: viewpoints; reverse engineering. Washington, DC, USA: IEEE Computer Society, 2003, pp. 36–47.

Max E. Kramer. "A Generative Approach to Change-Driven Consistency in Multi-View Modeling". In: *11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA), Montreal, QC*. Keywords: view, viewpoint, consistency. 2015, pp. 129–134. DOI: [10.1145/2737182.2737194](https://doi.org/10.1145/2737182.2737194).

Max E. Kramer et al. "Change-Driven Consistency for Component Code, Architectural Models, and Contracts". In: *18th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE), Montreal, QC*. 2015, pp. 21–26. DOI: [10.1145/2737166.2737177](https://doi.org/10.1145/2737166.2737177).

Philippe B. Kruchten. "Software architecture – a rational metamodel". In: *Proceedings 2nd International Workshop on the Architecture of Software Systems*. 1996.

Annotations: Key inspiration for the IEEE 1471 conceptual model and its documentation as a UML class diagram.

Philippe B. Kruchten. "The "4+1" View Model of architecture". In: *IEEE Software* 12.6 (Nov. 1995). Keywords: viewpoints, pp. 42–50.

Annotations: Leading example of a multiple view-based software architectural method, and a motivating case for IEEE 1471.

Philippe B. Kruchten. *The Rational Unified Process: an introduction*. Addison-Wesley, 1999.

Philippe Kruchten, Rafael Capilla, and Juan Carlos Dueñas. "The Decision View's Role in Software Architecture Practice". In: *IEEE Software* 26.2 (March–April 2009), pp. 36–42. DOI: [10.1109/MS.2009.52](https://doi.org/10.1109/MS.2009.52).

Annotations: Traces the historical evolution of thinking about software architecture representation and advocates a *decision viewpoint* cross-cutting other architectural views.

Patricia Lago, Paris Avgeriou, and Rich Hilliard. "Guest editors' introduction, Software Architecture: Framing Stakeholders' Concerns". In: *IEEE Software* 27.6 (November/December 2010), pp. 20–24.

Patricia Lago et al. "The Road Ahead for Architectural Languages". In: *IEEE Software* 32.1 (Feb. 2015). Keywords: ADLs, MDE, pp. 98–105.

Rikard Land. "An Architectural Approach to Software Evolution and Integration". PhD thesis. Mälardalen University, 2003. URL: <http://www.mrtc.mdh.se/publications/0590.pdf>.

Rikard Land. "Applying the IEEE Std 1471 Recommended Practice to a Software Integration Project". In: *International Conference on Software Engineering Research and Practice (SERP'03)*. Las Vegas, Nevada: CSREA Press, June 2003. URL: <http://www.mrtc.mdh.se/publications/0529.pdf>.

M. Lankhorst. *6 ways to organize your architecture models (parts 1 and 2)*. Bizzdesign blog. Mar. 2018. URL: <https://bizzdesign.com/blog/6-ways-to-organize-your-architecture-models-part-1/>.

Annotations: Ways to organize a model repository (not an AD): by business domain, information domains, technology stacks; by current- vs future-state; model content vs views of that content (assuming a projective stance!); define naming and other modeling conventions; establish an editorial board; define responsibilities, access rules, user groups and procedures.

Anne Lapkin. *Gartner defines the term 'enterprise architecture'*. Tech. rep. G00141795. Gartner, July 2006.

Annotations: Gartner builds on the IEEE 1471 definition of architecture to its relevance to Enterprise Architecture.

Anne Lapkin. *Gartner's Enterprise Architecture Process and Framework Help Meet 21st Century Challenges*. Tech. rep. G00133132. The Gartner Group, Nov. 2005. URL: http://www.gartner.com/resources/133100/133132/gartners_enterprise_architec_133132.pdf.

Annotations: Overview of Gartner's Enterprise Architecture Framework in which they "adopted an aspect-oriented approach to our framework, deliberately compatible with

IEEE 1471... [defining] three interdependent viewpoints: a business viewpoint, which is concerned with the processes and organization of the business; an information viewpoint, which is concerned with the information that runs the enterprise; and a technology viewpoint, which is concerned with the hardware and software components that support the enterprise. The aspect-oriented approach allows for the articulation of additional viewpoints, should the organization require them.”

N. Lassing, D. Rijsenbrij, and H. van Vliet. “Viewpoints on modifiability”. In: *International Journal of Software Engineering and Knowledge Engineering* 11.4 (2001), pp. 453–478.

H.W. Lawson, W. Rossak, and H. R. Simpson. “Working Group Report – CBS architecture”. In: *Proceedings of the 1994 tutorial and workshop on systems engineering of computer-based systems*. Los Alamitos, CA: IEEE Computer Society Press, 1994.

P. Linington. “Black Cats and Coloured Birds – What do Viewpoint Correspondences Do?” In: *4th International Workshop on ODP and Enterprise Computing (WODPEC 2007)*. IEEE Digital Library. Oct. 2007.

P.F. Linington. “How Viewpoints relate to Model Transformations, Configurations and Conformance”. In: *2008 12th Enterprise Distributed Object Computing Conference Workshops*. Keywords: viewpoints; context modeling; correspondences. Sept. 2008, pp. 452–457. DOI: [10.1109/EDOCW.2008.35](https://doi.org/10.1109/EDOCW.2008.35).

Abstract: The ODP Reference Model defines the way a system specification is structured into a number of inter-related viewpoints, but it does not constrain the specification process. However, applying ODP in a model-driven context requires some commitment to a series of transformation activities, and, in these, the viewpoints are used in different ways. In particular, the technology viewpoint can enter into transformations in a number of different ways, depending on the style and target of the system specification. This paper considers a number of different scenarios, comparing the transformations needed and the resulting assertions about conformance; as a result, it identifies a number of design choices, relating to different fields of application, and shows that these can usefully be made explicit in describing the available system design options.

Daniel Link et al. “The Value of Software Architecture Recovery for Maintenance”. In: *Proc. 12th Innovations in Software Engineering Conference*. Keywords: software, reverse architecting, concerns. 2019. DOI: [10.1145/3299771.3299787](https://doi.org/10.1145/3299771.3299787).

Ivan Logre. “Prserver la sparation des preoccupations durant lintgration de domaines htrognes dans les systmes logiciels”. Keywords: service oriented architecture, separation of concerns, variability, meta modeling. PhD thesis. Universit Cte d’Azur, 2017.

Annotations: –.

Abstract: The growing complexity of software engineering leads to the use of separation of concerns, i.e. enable to consider manageable sub-systems while keeping an overview of the whole system. The domain variability involved in these system design imply : (i) to compose multiple heterogeneous metamodels dedicated to each domain, (ii) to handle cross-domain consistency of the model produced in isolation, (iii) and to tame the multiplicity of concrete artefact available in the solution space of each domain. To adress these challenges, we offer in this thesis an approach encompassing three contributions :- an isolation-compliant composition which benefits from Service Oriented Architecture (SOA) integration. Each domain metamodel is embedded in a service exposing the rel-

evant behavior through an interface designed and used by domain experts ; - a business rule engine handling the interaction between domains and detecting cross-domain inconsistency to provide relevant feedback to resolve it ; - a feature-based characterization of the products variability allowing to concretize each sub-system toward concrete artifacts. The contributions are applied on the sensor data visualization use case. We validate (i) the need for domain isolation preservation on a dashboard design project, then we quantify (ii) the overhead of the service encapsulation, (iii) the impact of the externalization of domain interactions, (iv) the effort required from the experts and the integrator. Finally, we proceed to a user experiment to measure the gain during the concretization of a sub-system, and the impact on the user satisfaction on the resulting visualisation widgets.

Sihem Loukil et al. "AO4AADL: Aspect oriented extension for AADL". In: *Central European Journal of Computer Science* 3.2 (2013). Keywords: ADL, aspects, software, pp. 43-68.

Annotations: -.

Abstract: From the abstract: "Traditional ADLs do not normally provide appropriate formalisms to separate any kind of crosscutting concerns. This frequently results in poor descriptions of the software architectures and a tedious adaptation to constantly changing user requirements and specifications. AOSD (Aspect Oriented Software Development) deals with these problems by considering crosscutting concerns in software development...

In this paper, we propose a new aspect language called AO4AADL that adequately manipulates aspect oriented concepts at architecture level in order to master complexity and ensure scalability. The abstract nature of our proposed language allows the generation of aspect code for several programming languages and platforms."

David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Pearson, 2002.

Mark W. Maier. "Model Organization through Viewpoints and Views". In: *Proceedings of International Council on Systems Engineering Mid-Atlantic Regional Conference*. 2000, pp. 6.2-1-9.

Mark W. Maier. "System and Software Architecture Reconciliation". In: *Systems Engineering* 9.2 (2006), pp. 146-159.

Mark W. Maier, David Emery, and Rich Hilliard. "ANSI/IEEE 1471 and systems engineering". In: *Systems Engineering* 7.3 (2004), pp. 257-270.

Annotations: A technical overview of IEEE 1471 and discussion of its applicability to systems architecture.

Mark W. Maier, David Emery, and Rich Hilliard. "Software Architecture: Introducing IEEE Standard 1471". In: *Computer* 34.4 (Apr. 2001), pp. 107-109. DOI: [10.1109/2.917550](https://doi.org/10.1109/2.917550).

Annotations: Overview of IEEE 1471 after its publication.

Mark W. Maier and Eberhard Rechtin. *The art of systems architecting*. 2nd. CRC Press, 2000.

Mark W. Maier and Eric B. Wendoloski. *Weather Satellite Constellation As-Is and To-Be Ar-*

chitecture Description: An ISO/IEC/IEEE 42010 Example. AEROSPACE REPORT ATR-2020-00413. Keywords: AD, case study. The Aerospace Corporation, 2019.

Annotations: –.

Ivano Malavolta et al. “Providing Architectural Languages and Tools Interoperability through Model Transformation Technologies”. In: *IEEE Transactions on Software Engineering* 36.1 (2010). Keywords: ADLs, MDE, pp. 119–140.

Annotations: –.

Ivano Malavolta et al. “What Industry Needs from Architectural Languages: A Survey”. In: *IEEE Transactions on Software Engineering* 39.6 (2013). Keywords: ADLs, MDE, pp. 869–891.

Annotations: –.

Anders Mattsson et al. “Linking Model-Driven Development and Software Architecture: A Case Study”. In: *IEEE Transactions on Software Engineering* 35.1 (2009), pp. 83–93. DOI: [10.1109/TSE.2008.87](https://doi.org/10.1109/TSE.2008.87).

Anastasia Mavridou et al. “Architecture Diagrams: A Graphical Language for Architecture Style Specification”. In: *Electronic Proceedings in Theoretical Computer Science* 223 (2016), pp. 83–97. DOI: [10.4204/eptcs.223.6](https://doi.org/10.4204/eptcs.223.6).

Annotations: Proposes architecture diagrams to express architecture styles. An architecture diagram consists of a set of component types, a cardinality function and a set of connector motifs. Connector motifs are non-empty sets of generic ports that must interact. A connector motif defines a set of possible configurations, where a configuration is a set of connectors.

Abstract: Architecture styles characterise families of architectures sharing common characteristics. We have recently proposed configuration logics for architecture style specification. In this paper, we study a graphical notation to enhance readability and easiness of expression. We study simple architecture diagrams and a more expressive extension, interval architecture diagrams. For each type of diagrams, we present its semantics, a set of necessary and sufficient consistency conditions and a method that allows to characterise compositionally the specified architectures. We provide several examples illustrating the application of the results. We also present a polynomial-time algorithm for checking that a given architecture conforms to the architecture style specified by a diagram.

Nicholas May. “A Survey of Software Architecture Viewpoint Models”. In: *Proc. of the Sixth Australasian Workshop on Software and System Architectures*. Keywords: software, viewpoints. Mar. 2005, pp. 13–24.

Abstract: The documentation of software architecture is carried out in many different ways. One method is to break up the description into separate perspectives that address the different concerns that stakeholders have with software architecture. These perspectives, sometimes called viewpoints, can contain multiple diagrams to describe the complete system. Various models have been proposed that detail viewpoints and specify the stakeholders and concerns that they will satisfy. In this paper we survey five viewpoint models to determine the extent to which they cover the software architecture domain. We attempt to identify a set of viewpoints from different models can be combined to provide the widest possible coverage. We found that no model has complete coverage, but an

optimal set of viewpoints can be selected from the models. This optimal set, whilst not providing complete coverage, has a greater coverage than any of the individual viewpoint models.

Tom Mens, Jeff Magee, and Bernhard Rumpe. "Evolving Software Architecture Descriptions of Critical Systems". In: *Computer* 43.5 (2010), pp. 42–48. DOI: [10.1109/MC.2010.136](https://doi.org/10.1109/MC.2010.136).

M. M. Michel and G. H. Galal-Edeen. "Detecting inconsistencies between software architecture views". In: *2009 International Conference on Computer Engineering Systems*. Keywords: software architecture, architecture viewpoints, architecture views, consistency checking. Dec. 2009, pp. 429–434. DOI: [10.1109/ICCES.2009.5383227](https://doi.org/10.1109/ICCES.2009.5383227).

Abstract: Software architecture is widely presented in multiple architectural views in order to address different concerns of stakeholders and to minimize complexities of presenting architecture in one model. Using multiple architecture views raised the problem of inconsistencies as views are interrelated and overlap with one another. This paper proposes a framework to help checking inconsistencies between architectural views. We firstly define the types of inconsistencies that might occur between multiple architectural views and then we propose a framework for checking inconsistencies between them. At the end a case study is provided to evaluate the framework.

E. Miotto and T. Vardanega. "On the integration of domain-specific and scientific bodies of knowledge in Model Driven Engineering". In: *Proceedings of STANDRTS'09*. June 2009.

Annotations: composable: their semantics should not superimpose, or if this happens they should agree on the overlaps; compositional: there must exist a systematic way to assemble the semantics of concern specific views to obtain the semantics of the PIM as a whole.

Abstract: (excerpt) An effective MDE development environment should therefore subsume two distinct bodies of knowledge: the domain-specific one and its scientific complement. We are interested in devising a provably correct and affordable way to implement such an MDE environment.

Naeem Muhammad, Nelis Boucké, and Yolande Berbers. "Parallelism Viewpoint: A Viewpoint to Model Parallelism in Parallelism-Intensive Software Systems". In: *Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on*. Keywords: parallel programming;architecture viewpoint;multithreading;software architecture;software performance. 2011, pp. 285–294. DOI: [10.1109/ICECCS.2011.35](https://doi.org/10.1109/ICECCS.2011.35).

Abstract: The use of parallelism enhances the performance of a software system. Its excessive use, however, can degrade the performance. In this paper we propose a parallelism viewpoint to optimize the use of parallelism by eliminating unnecessarily used threads in legacy systems. The viewpoint describes the parallelism behaviour of the system, which can be used to analyze for overheads associated with threads. We illustrate the proposed viewpoint with the help of an industrial case, a parallelism-intensive electron microscope software system. We use the viewpoint to analyze threads suitable to be replaced with a small sized thread pool in this system. Results show that the viewpoint provides a profound insight into the thread-model of the system that is required to reduce the parallelism. In the thread pool analysis, we found that more than 50% threads are underused. They were replaceable with a pool of approximately 11% of these threads.

Juergen Musil et al. "An Architecture Framework for Collective Intelligence Systems". In: *WICSA 2015*. Keywords: viewpoint. IEEE Computer Society, Apr. 2015, pp. 21–30.

Annotations: "The framework defines a set of three architecture viewpoints for building new CIS solutions: CI context viewpoint, CI technical realization viewpoint, and CI operation viewpoint."

J. Muskens, R. J. Bril, and M. R. V. Chaudron. "Generalizing Consistency Checking between Software Views". In: *WICSA '05: Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 169–180. DOI: [10.1109/WICSA.2005.37](https://doi.org/10.1109/WICSA.2005.37).

Annotations: Shows how relational calculus can be very powerful means for cross-view analysis.

Sunia Naeem and Salma Imtiaz. "Architecture Coverage: Validating Optimum Set of Viewpoints". In: *ICSEA 2014: The Ninth International Conference on Software Engineering Advances*. 2014.

Annotations: –.

Abstract: There are various software architecture viewpoint models but none of them provides optimum coverage of software architecture domain. Software architecture coverage is the coverage of concepts that are required to effectively design and analyze software architecture. An optimum set of viewpoints can be selected from different software architecture viewpoint models that provide maximum coverage of software architecture domain than an individual architecture model. In this paper, an optimum set of viewpoints is selected by comparing five commonly used software architecture viewpoint models namely 4+1 RM-ODP, SEI, Siemens and Rational ADS via a common comparison framework. These architecture models are compared on the evaluation criteria, i.e., viewpoints, stakeholders and quality attributes. This evaluation criterion is based on IEEE Standard 1471 *Recommended Practice Architecture Description of Software-Intensive Systems*. The resulting optimum set is validated in industry via multiple case studies, and the results show that the optimum set of viewpoints provide greater coverage than any of the viewpoint alone.

E.Y. Nakagawa, F. Oquendo, and M. Becker. "RAModel: A reference model for reference architectures". In: *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA 2012, Helsinki, Finland, August 20-24, 2012 – Companion Volume*. Ed. by Tomi Mnnist et al. Keywords: reference architectures. ACM. 2012.

Annotations: –.

Mahmoud Nassar. "VUML : a Viewpoint oriented UML Extension". In: *18th IEEE International Conference on Automated Software Engineering (ASE 2003), 6–10 October 2003, Montreal, Canada*. 2003, pp. 373–376. DOI: [10.1109/ASE.2003.1240341](https://doi.org/10.1109/ASE.2003.1240341).

Annotations: -.

Dionisio de Niz and Peter H. Feiler. "Aspects in the Industry Standard AADL". In: *Proceedings of Workshop on Aspect-Oriented Modeling '07*. Keywords: aspects, concerns, systems, ADL. 2007, pp. 15–20.

Annotations: –.

Abstract: Aspect-Oriented Modeling is aimed at reducing the complexity of models by separating its different concerns. In model-based development of embedded systems this separation of concerns is more important given the multiple non-functional concerns addressed by embedded systems. These concerns can include timeliness, fault-tolerance, and security to name a few. The Architecture Analysis and Design Language (AADL) is a standard architecture description language to design and evaluate software architectures for embedded systems already in use by a number of organizations around the world. In this paper we discuss our current effort to extend the language to include new features for separation of concerns. These features not only include constructs to describe design choices but also routines to verify the proper combination of constructs from different concerns. This verification includes techniques and tools from the formal methods arena integrated into the AADL development tool providing a seamless design flow. We believe that work in this direction is fundamental to tackle the potential combinatorial explosion problem of verifying the merging of multiple concerns into a final system.

F. J. Nogueira Santos, F. M. Santoro, and C. Cappelli. "Crosscutting concerns at enterprise architecture level". In: *2011 IEEE International Conference on Systems, Man, and Cybernetics*. Keywords: enterprise, aspects, concerns, crosscutting. Oct. 2011, pp. 345–350. DOI: [10.1109/ICSMC.2011.6083689](https://doi.org/10.1109/ICSMC.2011.6083689).

Annotations: –.

Abstract: Aspect-Oriented approach was created to solve scattering and tangling issues generated by the implementation of crosscutting requirements in programming. This approach has been proposed in many research areas regarding the entire process of software development, from requirements and business process models to code generating. Enterprise Architecture (EA) allows structural organization of artifacts generated during Information Systems development and reflects relational view existent between the artifacts, but, in general, it disregards the crosscutting concerns existent between them. In this paper, we present a scenario of these concerns at EA level using the Zachman Framework.

R.L. Nord et al. *A Structured Approach for Reviewing Architecture Documentation*. Tech. rep. CMU/SEI-2009-TN-030. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2009.

OMG Systems Modeling Language (OMG SysML) version 1.1. formal/2008-11-01. Nov. 2008.

Annotations: "SysML has extended the concept of view and viewpoint from UML to be consistent with the IEEE 1471 standard. In particular, a viewpoint is a specification of rules for constructing a view to address a set of stakeholder concerns, and the view is intended to represent the system from this viewpoint. This enables stakeholders to specify aspects of the system model that are important to them from their viewpoint, and then represent those aspects of the system in a specific view. Typical examples may include an operational, manufacturing, or security view/viewpoint."

Henk Obbink et al. *Report on Software Architecture Review and Assessment (SARA)*. Tech. rep. version 1.0. The SARA Working Group, Feb. 2002. URL: <http://philippe.kruchten.com/architecture/SARAv1.pdf>.

Annotations: Final report of an industry group defining an approach to architecture evaluation. Uses IEEE 1471 conceptual framework in its foundation.

M. A. Ogush, D. Coleman, and D. Beringer. "A template for documenting software and firmware architectures". Draft version 1.3. Jan. 2000.

Oddrun Pauline Ohren. "Ontology for Characterising Architecture Frameworks". In: *EMOI-INTEROP 2004: Enterprise Modelling and Ontologies for Interoperability*. Ed. by Michele Misikoff. 2004. URL: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-125/>.

Valiallah Omrani and Seyyed Ali Razavi Ebrahimi. "Software Architecture Viewpoint Models: A Short Survey". In: *Advances in Computer Science: an International Journal* 2.6 (5 Nov. 2013). URL: <http://www.acsij.org/documents/v2i5/ACSIJ-2013-2-5-241.pdf>.

PPOOA: Processes Pipelines in Object-oriented Architectures. Keywords: ADFs, real-time, coordination, architecture styles. URL: <http://www.ppooa.com.es/>.

Claus Pahl, Simon Giesecke, and Wilhelm Hasselbring. "An Ontology-Based Approach for Modelling Architectural Styles". In: *Software Architecture: First European Conference, ECSA 2007, Proceedings*. Ed. by Flavio Oquendo. Vol. 4758. Lecture Notes in Computer Science. Springer, 2007, pp. 60–75. URL: <http://www.computing.dcu.ie/~cpahl/papers/ecsa07.pdf>.

Abstract: The conceptual modelling of software architectures is of central importance for the quality of a software system. A rich modelling language is required to integrate the different aspects of architecture modelling, such as architectural styles, structural and behavioural modelling, into a coherent framework. We propose an ontological approach for architectural style modelling based on description logic as an abstract, meta-level modelling instrument. Architectural styles are often neglected in software architectures. We introduce a framework for style definition and style combination. The link between quality requirements and conceptual modelling of architectural styles is investigated. The application of the ontological framework in the form of an integration into existing architectural description notations such as ACME and UML-based approaches, and also service ontologies is illustrated.

Marco Panunzio and Tullio Vardanega. "An architectural approach with separation of concerns to address extra-functional requirements in the development of embedded real-time software systems". In: *Journal of Systems Architecture* 60 (2014), pp. 770–781.

Annotations: –.

Abstract: A large proportion of the requirements on embedded real-time systems stems from the extra-functional dimensions of time and space determinism, dependability, safety and security, and it is addressed at the software level. The adoption of a sound software architecture provides crucial aid in conveniently apportioning the relevant development concerns. This paper takes a software-centered interpretation of the ISO 42010 notion of architecture, enhancing it with a component model that attributes separate concerns to distinct design views. The component boundary becomes the border between functional and extra-functional concerns. The latter are treated as decorations placed on the outside of components, satisfied by implementation artifacts separate from and composable with the implementation of the component internals. The approach was evaluated by industrial users from several domains, with remarkably positive results.

John van des Pas and Geert-Jan van Bussel. "Embedding Privacy in ICT architectures.

The citizen as public stakeholder in ICT architecture development". In: *Proceedings of the Amsterdam Privacy Conference*. Ed. by B. van der Sloot. Keywords: ICT, privacy, stakeholders, concerns, viewpoints. Oct. 2015.

Annotations: Discusses the role of architecture description in the formulation and protection of citizens' privacy concerns.

L. Passos et al. "Static Architecture-Conformance Checking: An Illustrative Overview". In: *IEEE Software* 27.5 (Sept. 2010). Keywords: software architecture, software metrics, static architecture, conformance checking, dependency-structure matrices, reflexion models, architectural constraint checking, pp. 82–89. DOI: [10.1109/MS.2009.117](https://doi.org/10.1109/MS.2009.117).

Abstract: In this article, the authors compare and illustrate the use of three static architecture-conformance techniques: dependency-structure matrices, source code query languages, and reflexion models. To highlight the similarities and differences between these three techniques, they describe how to apply some of the techniques' available supporting tools to specify and check architectural constraints for a simple personal information management system.

Patrizio Pelliccione et al. "Automotive Architecture Framework: The Experience of Volvo Cars". In: *Journal of Systems Architecture* (2017). Keywords: ADFs.

Annotations: -.

D. Perovich and M. C. Bastarrica. *Model-Based Formalization of Software Architecture Knowledge on Description and Design*. Tech. rep. MaTE Group, DCC, Universidad de Chile, Nov. 2013. URL: <http://mate.dcc.uchile.cl/research/tns/pb13.pdf>.

D. Perovich, M.C. Bastarrica, and C. Rojas. "Model-Driven Approach to Software Architecture Design". In: *Sharing and Reusing Architectural Knowledge (SHARK '09)*. ICSE Workshop on. Keywords: model-driven architecture; architecture design methods; megamodeling. 2009, pp. 1–8. DOI: [10.1109/SHARK.2009.5069109](https://doi.org/10.1109/SHARK.2009.5069109).

Gilles Perrouin. "Architecting Software Systems Using Model Transformations and Architectural Frameworks". Keywords: software, product lines, architecture frameworks, MBSE. PhD thesis. Universite du Luxembourg, 2007.

Abstract: This thesis strives to propose a trade-off between automated and unsupported product derivation by providing a model-driven product line development method that allows developers to define product line members by transforming a coherent and layered set of product line models. Moreover, constraints on the possible transformations have to be specified in order to determine which products cannot be derived both for functional and technical reasons.

The first part of this thesis introduces the foundational concepts of our FIDJI method. In particular, it describes the notion of architectural framework as a set of models defining product line assets at analysis and design levels and which is instantiated in order to obtain product line members thanks to model transformations. This part then describes key methodological principles driving the choice of architectural framework models and how flexibility in product derivation can be achieved and controlled by constraints defined over the set of architectural framework models. [excerpt].

Dewayne E. Perry and Alexander L. Wolf. "Foundations for the study of Software Architecture". In: *ACM SIGSOFT Software Engineering Notes* 17.4 (Oct. 1992), pp. 40–52.

Annotations: Published version of their underground classic “Software Architectures”. Early motivation for use of multiple views in architecture description.

Nic Plum. *Architecture Description Viewpoints. Metamodel Description, Implementation and Model Changes*. Tech. rep. Keywords: architecture framework, TRAK, meta model, ADL. Eclectica Systems Ltd., Aug. 2019. URL: https://www.researchgate.net/profile/Nic_Plum/research.

Annotations: Defines a set of architecture viewpoints conforming to ISO/IEC/IEEE 42010:2011; a meta model representing those viewpoints how ADLs may implement those viewpoints.

Nic Plum, ed. *TRAK Enterprise Architecture Framework*. Keywords: architecture frameworks. 2018. URL: <https://trak.sourceforge.io>.

Nic Plum. “Using directed graphs to define viewpoints to keep a metamodel, an architecture framework and views using different modeling languages consistent”. In: *Engineering Reports 2.6* (2020). Keywords: viewpoints, consistency, correspondences.

Annotations: –.

Eltjo Poort and Eric van der Vliet. “Architecting in a Solution Costing Context: Early Experiences with Solution-Based Estimating”. In: *WICSA 2015*. Keywords: viewpoint, costing. IEEE Computer Society, Apr. 2015, pp. 127–130.

Annotations: Introduces the Delivery Breakdown Viewpoint to address concerns associated with cost estimation.

Rachel A. Pottinger and Philip A. Bernstein. *Merging Models Based on Given Correspondences*. Tech. rep. UW-CSE-03-02-03. correspondences, multiple models: University of Washington, 2003. URL: <ftp://ftp.cs.washington.edu/tr/2003/02/UW-CSE-03-02-03.pdf>.

Annotations: -.

Abstract: A model is a formal description of a complex application artifact, such as a database schema, an application interface, a UML model, an ontology, or a message format. The problem of merging such models lies at the core of many meta data applications, such as view integration, mediated schema creation for data integration, and ontology merging. This paper examines the problem of merging two models given correspondences between them. It presents requirements for conducting a merge and a specific algorithm that subsumes previous work.

H.A. Proper, A.A. Verrijn-Stuart, and S.J.B.A. Hoppenbrouwers. “On Utility-based Selection of Architecture-Modelling Concepts”. In: *Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*. Ed. by Sven Hartmann and Markus Stumptner. Vol. 43. CRPIT. Newcastle, Australia: Australian Computer Society, 2005, pp. 25–34.

Annotations: Surveys the principles of architectural modeling from three angles: Modeling, Utility and Communication and works through case studies of two viewpoint frameworks: Kruchten’s 4+1 and RM-ODP, using consideration of concerns adapted from IEEE 1471.

Aleksandar Radjenovic. “View Consistency in Architectural Models of Dependable Systems”. PhD. The University of York, Mar. 2008.

Maryam Razavian and Patricia Lago. "A Viewpoint for Dealing with Change in Migration to Services". In: *Proceedings of the Joint 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA)*. Keywords: viewpoints. Helsinki, Finland: IEEE Computer Society, 2012.

Eberhardt Rechtin. *Systems architecting: creating and building complex systems*. Prentice Hall, 1991.

William C. Regli, Mayk Israel, and Christopher T. Cannon. "Development and Specification of a Reference Architecture for Agent-Based Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.2 (2014). –, pp. 146–161.

Annotations: –.

Jose R. Romero and A. Vallecillo. "Well-formed Rules for Viewpoint Correspondences". In: *Proceedings of the 5th International Workshop on ODP and Enterprise Computing (WODPEC 2008)*. Munich, Germany, Sept. 2008. URL: <http://www.lcc.uma.es/av/Publicaciones/08/wodpec2008-correspondences.pdf>.

Douglas T. Ross. "Structured Analysis (SA): a language for communicating ideas". In: *IEEE Transactions on Software Engineering* SE-3.1 (Jan. 1977), pp. 16–34.

Annotations: Earliest reference to first-class viewpoints in software engineering literature.

David Rowe. "An Ontological Model of Computer Based Systems and Architectural Change". PhD thesis. University of Technology, Sydney, 2000.

Annotations: Uses IEEE 1471 as part of its foundations.

Nick Rozanski and Eóin Woods. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison Wesley, 2005.

Annotations: Adopts IEEE 1471 as a starting point. Defines a number of viewpoints and perspectives (cross-cutting viewpoints).

Nick Rozanski and Eóin Woods. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. 2nd. Addison Wesley, 2011.

Annotations: Adopts IEEE 1471 as a starting point. Defines a number of viewpoints and perspectives (cross-cutting viewpoints).

I. Ruchkin et al. "IPL: An Integration Property Language for Multi-model Cyber-physical Systems". In: *FM 2018: International Symposium on Formal Methods*. Ed. by K. Havelund et al. Vol. 10951. Lecture Notes in Computer Science. Keywords: model integration, correspondences. Springer, 2018, pp. 165–184.

Annotations: Proposes IPL which could serve as an approach to correspondences.

Abstract: Design and verification of modern systems requires diverse models, which often come from a variety of disciplines, and it is challenging to manage their heterogeneity – especially in the case of cyber-physical systems. To check consistency between models, recent approaches map these models to flexible static abstractions, such as architectural views. This model integration approach, however, comes at a cost of reduced expressiveness because complex behaviors of the models are abstracted away. As a result, it may be impossible to automatically verify important behavioral properties across multiple

models, leaving systems vulnerable to subtle bugs. This paper introduces the Integration Property Language (IPL) that improves integration expressiveness using modular verification of properties that depend on detailed behavioral semantics while retaining the ability for static system-wide reasoning. We prove that the verification algorithm is sound and analyze its termination conditions. Furthermore, we perform a case study on a mobile robot to demonstrate IPL is practically useful and evaluate its performance.

Ivan Ruchkin. "Integration of Modeling Methods for Cyber-Physical Systems". Keywords: cyber-physical systems, model checking, viewpoints. PhD thesis. Carnegie Mellon University, 2019. URL: https://kilthub.cmu.edu/articles/Integration_of_Modeling_Methods_for_Cyber-Physical_Systems/7970222.

Annotations: From : 6.2.2 Introduces integration views To relate models and views, I introduce the concept of an integration viewpoint. This concept is inspired by the ISO standard 42010 [128] for architectural description, which uses viewpoints as descriptions of each view (or persons) perspective. A similar interpretation of viewpoints has appeared in seminal works in software architecture and cyber-physical systems [32, 51, 83, 117]. In the context of model integration, a viewpoint consists of several entities that relate the elements in models and views. Intuitively, integration viewpoints serve two functions:

1. Define the aspects of the model that need to be extracted as view elements. For example, a timing viewpoint would describe what information is related to timing in models: delays, deadlines, execution times, and so on.
2. Create views from models. For example, a timing viewpoint would create a timing view from a model with timing-related information.

Abstract: Cyber-physical systems (CPS) incorporate digital (cyber) and mechanical (physical) elements that interact in complex ways. Many safety-critical CPS, such as autonomous vehicles and drones, are becoming increasingly widespread and hence demand rigorous quality assurance. To this end, CPS engineering relies on modeling methods, which use models to represent the system and design-time analyses to interpret/change the models. Coming from diverse scientific and engineering fields, these modeling methods are difficult to combine, or integrate, due to implicit relations and dependencies between them. CPS failures can lead to substantial damage or loss of life, and are often due to two key integration challenges: (i) inconsistencies between models contradictions in models that do not add up to a cohesive design, and (ii) incorrect interactions of analyses analyses performed out-of-order and in mismatched contexts, leading to erroneous analysis outputs. This thesis presents a novel approach to detect and prevent integration issues between CPS modeling methods during the design phase. To detect inconsistencies between models, the approach allows engineers to specify integration properties quantified logical statements that relate elements of multiple models in the Integration Property Language (IPL). IPL statements describe verifiable conditions that are equivalent to an absence of inconsistencies. To interface with the models, IPL relies on integration abstractions simplified representations of models for integration purposes. This thesis proposes two abstractions: views (annotated component-and-connector models, inspired by software architecture) and behavioral properties (expressions in model-specific property languages, such as the linear temporal logic). Combining these abstractions lets engineers relate model structure and behavior in IPL statements. To ensure correct interactions of analyses, I introduce analysis contracts a lightweight specification that captures inputs, outputs, assumptions, and guarantees for each analysis, in terms of the integration abstractions. Given these contracts, an analysis execution platform performs analyses in the order of their dependencies, and only in the contexts that guarantee correct outputs. My approach to integration was validated on four case studies of CPS modeling methods in

different systems: energy-aware planning in a mobile robot, collision avoidance in a mobile robot, thread/battery scheduling in a quadrotor, and reliable/secure sensing in an autonomous vehicle. This validation has shown that the approach can find safety-critical errors by specifying expressive integration properties and soundly checking them within practical constraints all while being customizable to heterogeneous models, analyses, and domains.

Mehrdad Sabetzadeh and Steve M. Easterbrook. "Analysis of Inconsistency in Graph-Based Viewpoints: A Category-Theoretic Approach". In: *18th IEEE International Conference on Automated Software Engineering (ASE 2003)*, 6–10 October 2003, Montreal, Canada. 2003, pp. 12–21. DOI: [10.1109/ASE.2003.1240290](https://doi.org/10.1109/ASE.2003.1240290).

Annotations: -.

Santonu Sarkar and Srinivas Thonse. "EAML – Architecture Modeling Language for Enterprise Applications". In: *IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04)*. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 40–47. DOI: [10.1109/CEC-EAST.2004.37](https://doi.org/10.1109/CEC-EAST.2004.37).

Jaap Schekkerman. *Another View at Extended Enterprise Architecture Viewpoints*. Sept. 2004.

Annotations: Extends the IEEE 1471 conceptual framework to Enterprise Architecture.

CCSDS Secretariat. *Reference Architecture for Space Data Systems*. Recommended Practice CCSDS 311.0-M-1. Keywords: reference architecture. The Consultative Committee for Space Data Systems, 2008. URL: <https://public.ccsds.org/Pubs/311x0m1.pdf>.

Annotations: Uses enterprise, functional, connectivity, communications, and information viewpoints and correspondences between them.

Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), 18–22 February 2008, Vancouver, BC, Canada. IEEE Computer Society, 2008.

Peter Shames and Joseph Skipper. *Toward a Framework for Modeling Space Systems Architectures*. Tech. rep. Jet Propulsion Laboratory, 2006. URL: <http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/39851/1/06-0876.pdf>.

Kari Smolander and Tero Päivärinta. "Practical Rationale for Describing Software Architecture, Beyond Programming-in-The-Large". In: *Proceedings of 3rd Working IEEE/IFIP Conference on Software Architecture (WICSA3)*. Ed. by Jan Bosch. 2002, pp. 113–125.

Kari Smolander et al. *Required and Optional Viewpoints — What Is Included in Software Architecture?* Working Papers 10. Keywords: software architecture, viewpoints, meta architecture. Telecom Business Research Center Lappeenranta, 2001.

Annotations: -.

Abstract: Software architecture is conventionally concerned with the structures in a high abstraction level describing the main constituents of a software system. However, there seems to be no common agreement of what these structures exactly are. This paper acts as an attempt to define the viewpoints from which we see the structures. The rationale of having different viewpoints is reflected and a framework of the relationship between software architecture design and information systems development (ISD) methods is presented. The framework is based on the ideas found from method engineering and meta-

modelling. It is noted that the selection of viewpoints is contingent and based on the prevalent architectural practices in an organization and on the requirements from the project at hand. The paper includes a brief case study on the experiences from three different telecom-area software organizations that end up using different sets of viewpoints in their software process.

Kari Smolander et al. "What is Included in Software Architecture? A Case Study in Three Software Organizations". In: *Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2002)*. 2002, pp. 131–139.

Mohamed Soliman, Matthias Riebisch, and Uwe Zdun. "Enriching Architecture Knowledge with Technology Design Decisions". In: *WICSA 2015*. Keywords: decisions, AK, meta model. IEEE Computer Society, Apr. 2015, pp. 135–144.

J. F. Sowa and J. A. Zachman. "Extending and formalising the framework for information systems architecture". In: *IBM Systems Journal* 31.3 (1992), pp. 590–616.

Annotations: Follow-on to Zachman, "[A framework for information systems architecture](#)", and key paper for enterprise architecture frameworks.

Hasan Sözer and Bedir Tekinerdogan. "Introducing Recovery Style for Modeling and Analyzing System Recovery". In: *WICSA 2008*. IEEE Computer Society, 2008, pp. 167–176. DOI: [10.1109/WICSA.2008.9](https://doi.org/10.1109/WICSA.2008.9).

Annotations: This paper describes an explicit viewpoint/style for recovery concern.

Hasan Sözer, Bedir Tekinerdogan, and Mehmet Akşit. "FLORA: A Framework for Decomposing Software Architecture to Introduce Local Recovery". In: *Software – Practice and Experience* 30.10 (July 2009). Keywords: viewpoints, pp. 869–889.

Annotations: This paper discusses the decomposition of an architecture based on the recovery style as well as the automatic generation of the code based on the selected architectural decomposition.

David I. Spivak. "Categories as Mathematical Models". In: *Categories for the working philosopher*. Ed. by Elaine Landry. Keywords: models, modeling. Oxford University Press, 2017, pp. 381–401.

Annotations: Exploration of a number of modeling tenets, including: "A model is known only by its relationships with other models."

M. W. A. Steen et al. "Supporting Viewpoint-Oriented Enterprise Architecture". In: *8th International Enterprise Distributed Object Computing Conference (EDOC 2004)*. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 201–211. DOI: [10.1109/EDOC.2004.10008](https://doi.org/10.1109/EDOC.2004.10008).

Harald Storrle. "Structuring very large domain models: experiences from industrial MDSD projects". In: *ECSCA '10: Proceedings of the Fourth European Conference on Software Architecture*. Copenhagen, Denmark: ACM, 2010, pp. 49–54. DOI: [10.1145/1842752.1842766](https://doi.org/10.1145/1842752.1842766).

Abstract: View/Viewpoint approaches like IEEE 1471-2000, or Kruchten's 4+1-view model are used to structure software architectures at a high level of granularity. While research has focused on architectural languages and with consistency between multiple views, practical questions such as the structuring at a lower level of detail have not been dealt

with. This paper aims at filling this gap by reporting personal experiences from a very large scale industrial domain modeling project. There, structuring the logical view turned out to be a critical success factor. We explain the project and its setting, analyze the role and repercussions of model structuring, and examine the implications model structuring decisions have on other parts of the project. We then explain the model structure abstracted from a very large scale industrial modeling project. Finally, we discuss lessons learned.

Christoph F. Strnadl. *The Logical Syntax of (IT) Architectures*. 2020. arXiv: [2004.03719](https://arxiv.org/abs/2004.03719) [cs.LO].

Annotations: –.

Abstract: Despite several (accepted) standards, core notions typically employed in information technology (IT) architectures lack the precise and exact foundations encountered in logic, algebra, and other branches of mathematics. In this contribution we define the term “architecture” in a mathematically rigorous way. We motivate our particular choice by demonstrating (i) how commonly understood and expected properties of an architecture can be suitably defined or derived within our formalization, and (ii) how our concept is fully compatible with real life (business) architectures. Based on our fundamental definitions we further develop a rigorous notion of architectural *similarity* based on the notion of “homomorphisms” between architectures. We demonstrate the (theoretical) applicability by deriving some theorems on the characterization n-tier architectures.

S. Strohmeier and F. Gross. “Designing an architecture description language for nontechnical actors and purposes (NOTE-ADL)”. In: *Journal of Enterprise Information Management* 33.3 (2020). Keywords: ADLs, pp. 482–501. DOI: [10.1108/JEIM-06-2018-0120](https://doi.org/10.1108/JEIM-06-2018-0120).

Annotations: –.

Stanley M. Sutton Jr. and Isabelle Rouvellou. “Concern Modeling for Aspect-Oriented Software Development”. In: *Aspect-Oriented Software Development*. Ed. by Robert E. Filman et al. Addison-Wesley, 2004, pp. 479–505. URL: http://www.research.ibm.com/AEM/pubs/Cosmos--Chapter_21.pdf.

Annotations: Building on the definition of concern in IEEE 1471, the authors argue concerns must be first-class entities and concern modeling must be an explicit part of Aspect-Oriented Software Development.

Damian Andrew Tamburri et al. “Architecting in Networked Organizations”. In: *Proceedings Working IEEE/IFIP Conference on Software Architecture (WICSA 2014)*. Sydney, Australia, 7–11 April 2014, 2014.

Damien A. Tamburri, Patricia Lago, and Henry Muccini. “Leveraging Software Architectures through the ISO/IEC 42010 Standard: A Feasibility Study”. In: *Trends in Enterprise Architecture Research*. Ed. by Will Aalst et al. Vol. 70. Lecture Notes in Business Information Processing. Keywords: enterprise architecture. Springer Berlin Heidelberg, 2010, pp. 71–85. DOI: [10.1007/978-3-642-16819-2_6](https://doi.org/10.1007/978-3-642-16819-2_6).

Abstract: The state of the practice in enterprise and software architecture learnt that relevant architectural aspects should be illustrated in multiple views, targeting the various concerns of different stakeholders. This has been expressed a.o. in the ISO/IEC 42010 Standard on architecture descriptions. In the same vein, the research community observed that Architecture Description Languages, or ADLs, should be developed to ad-

dress stakeholders concerns concentrating on the use of viewpoints for their description. This notwithstanding, we notice today a proliferation of ADLs impervious to these guidelines. This imperviousness creates a gap between what the IT industry requires and what ADLs can provide. This gap makes it impossible for practitioners to choose and use the best-fit ADL for his/her requirements. To fill this gap, we must analyze the existing ADLs, and mine and make explicit their addressed concerns, views, viewpoints, and stakeholders. Such an explicit overview can provide practitioners with pragmatic information for selecting the most suitable ADL, and hence support them in the architecting process. This paper reports on initial results in this direction. Given a specific ADL (namely, DARWIN/FSP), it presents a feasibility study on the methodology mapping the concepts of the ISO/IEC 42010 on the DARWIN/FSP ADL.

Bedir Tekinerdogan, Christian Hofmann, and Mehmet Akşit. "Modeling Traceability of Concerns in Architectural Views". In: *Proceedings of the 10th international workshop on Aspect-oriented modeling*. 2007, pp. 49–56.

Bedir Tekinerdogan and Hasan Sözer. "Variability viewpoint for introducing variability in software architecture viewpoints". In: *Proceedings of the WICSA/ECSA 2012, Companion Volume*. Keywords: viewpoints, software architecture modeling, variability. Helsinki, Finland: ACM, 2012, pp. 163–166. DOI: [10.1145/2361999.2362033](https://doi.org/10.1145/2361999.2362033).

The Open Group Architectural Framework (TOGAF). 2009. URL: <http://www.opengroup.org/togaf/>.

Annotations: The Open Group's enterprise architecture framework.

Frédéric Thomas and Fabien Belmonte. "Using Topcased and a Viewpoint-based Framework to describe Safety Concerns of Railway Signalling Systems". In: *Topcased Days, Toulouse, France, February 2011*. 2011. URL: http://www.obeonetwork.org/images/ObeoNetwork/safety/abstract_obeo_safety_topcased.pdf.

Warren K. Vaneman. "Evolving Model-Based Systems Engineering Ontologies and Structures". In: *INCOSE International Symposium 28 (July 2018)*. Keywords: MDE, ontology, pp. 1027–1036. DOI: [10.1002/j.2334-5837.2018.00531.x](https://doi.org/10.1002/j.2334-5837.2018.00531.x).

Abstract: Model-Based Systems Engineering (MBSE) is a mysterious concept that means many different things to different stakeholders. MBSE was envisioned to manage the increasing complexity within systems, by replacing traditional document-based system engineering with a model-based approach. However, more than a decade after MBSE was introduced, many systems engineering efforts still default to a "document-like view" rather than integrated, "virtual," representation of the system. This paper suggests a revised definition for MBSE which supports system design and analysis, throughout all phases of the system lifecycle, and through the collection of modeling languages, model-based processes, structures, and presentation frameworks used to support the discipline of systems engineering in a model-based or model-driven context. To realize this definition, and an environment where the system is virtually represented, the long-sought ontology must be attained for better definition and structure within MBSE. This paper explores how current MBSE methods can be extended to include an ontology.

Pedro Filipe Oliveira Vicente. "A Reference Architecture for Integrated Governance, Risk and Compliance". Keywords: risk, governance, viewpoints, reference architecture. PhD thesis. Instituto Superior Tecnico Universidade Tecnica de Lisboa, 2018.

Annotations: Defines a Business Viewpoint and proposes a viewpoint-based approach to Governance, Risk and Compliance concerns.

Tiago Volpato et al. "Two Perspectives on Reference Architecture Sustainability". In: *11th European Conference on Software Architecture: Companion Proceedings (ECSA 2017)*. Ed. by Antnia Lopes and Rogrio de Lemos. Vol. 10475. Incs. Keywords: reference architecture. Springer, 2017, pp. 188–194.

Annotations: –.

Tim Weilkiens et al. *Model-Based System Architecture*. John Wiley & Sons, 2016. URL: <http://www.mbse-architecture.com>.

Annotations: Introduces *perspectives* following TRAK (cf. *Rozanski-Woods version of perspectives*). "Architecture views may aggregate into perspectives." "An architecture view may belong to any number of perspectives." Seems to limit architecture views to single stakeholder's concern.

Michael K. Wilkinson, Charles E. Dickerson, and Siyuan Ji. "Concepts of Architecture, Structure and System". In: *CoRR abs/1810.12265* (2018). URL: <http://arxiv.org/abs/1810.12265>.

Abstract: The current ISO standards pertaining to the Concepts of System and Architecture express succinct definitions of these two key terms that lend themselves to practical application and can be understood through elementary mathematical foundations. The current work of the ISO/IEC Working Group 42 is seeking to refine and elaborate the existing standards. This position paper revisits the fundamental concepts underlying both of these key terms and offers an approach to: (i) refine and exemplify the term 'fundamental concepts' in the current ISO definition of Architecture, (ii) exploit existing standards for the term 'concept', and (iii) introduce a new concept, Architectural Structure, that can serve to unify the current terminology at a fundamental level. Precise elementary examples are used in to conceptualise the approach offered.

Jeffery L. Williams and Jerrell T. Stracener. "First Steps in the Development of a Program Organizational Architectural Framework (POAF)". In: *Systems Engineering* 16.1 (2013). Keywords: system architecture; architecture framework; organization, pp. 45–70. DOI: [10.1002/sys.21218](https://doi.org/10.1002/sys.21218).

Abstract: The target audiences for this paper are systems engineers and architects involved in the design of complex systems such as program organizations. This is the first in a series exploring how the design of program organizations developed for the purpose of designing and developing aerospace and defense systems can be optimized. The objective of this paper is to lay the groundwork for an architecture framework for the development of a program organization. The draft standard ISO/IEC 42010 is used to define the structural requirements of the architecture framework. In addition, we use the Zachman Architecture Framework™ to organize the framework and the Department of Defense Architecture Framework Version 2.0 (DoDAF 2.0) to create the model environment for the Program Organizational Architecture Framework (POAF). This approach to defining the POAF ensured that we would have the data needed to support our objective to optimize the design of the program organization and hopefully reduce the number of defects inherent in the design. We also believe that we have sufficiently defined the characteristics of a POAF to spur more research in this area.

Martin Wirsing and Alexander Knapp. "View Consistency in Software Development". In: *Radical Innovations of Software and Systems Engineering in the Future*. Ed. by Martin Wirsing, Alexander Knapp, and Simonetta Balsamo. Vol. 2941. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2004, pp. 3–14. DOI: [10.1007/978-3-540-24626-8_24](https://doi.org/10.1007/978-3-540-24626-8_24).

Abstract: An algebraic approach to the view consistency problem in software development is provided. A view is formalised as a sentence of a viewpoint language; a viewpoint is given by a language and its semantics. Views in possibly different viewpoints are compared over a common view for consistency by a heterogenous pull-back construction. This general notion of view consistency is illustrated by several examples from viewpoints used in object-oriented software development.

Rebekka Wohlrab et al. "Improving the Consistency and Usefulness of Architecture Descriptions: Two Industrial Surveys". In: *IEEE International Conference on Software Architecture (ICSA 2019), Hamburg, Germany, March 25–29, 2019*.

Annotations: -.

Rebekka Wohlrab et al. "On Interfaces to Support Agile Architecting in Automotive: An Exploratory Case Study". In: *IEEE International Conference on Software Architecture (ICSA 2019), Hamburg, Germany, March 25–29, 2019*.

Annotations: -.

Eóin Woods. "Experiences Using Viewpoints for Information Systems Architecture: An Industrial Experience Report". In: *Software Architecture, Proceedings First European Workshop*. Ed. by Flavio Oquendo, Brian Warboys, and Ron Morrison. Lecture Notes in Computer Science 3047. St Andrews UK, May 21–22, 2004: Springer-Verlag, 2004, pp. 182–193.

Eóin Woods, Wolfgang Emmerich, and Nick Rozanski. "Using Architectural Perspectives". Unpublished draft, dated August 2004. 2004.

Annotations: Paper motivates introduction of concept of architectural perspectives, contrasting and supplementing IEEE 1471-style viewpoints.

Nesrine Yahiaoui, Bruno Traverson, and Nicole Levy. "A new viewpoint for change management in RM-ODP systems". In: *Workshop on ODP for Enterprise Computing (WODPEC 2005)*. Ed. by P. Linington et al. 2005, pp. 1–6.

Annotations: Proposes an approach to consistency between views using correspondence rules.

Takahiro Yamada. "Proposal for Defining a Generic Viewpoint in RM-ODP". In: *4th International Workshop on ODP and Enterprise Computing (WODPEC 2007)*. 2007. URL: http://www.inf.ufes.br/~jpalmeida/wodpec2007/cameraready/WODPEC_Yamada.pdf.

C. Yang et al. "An industrial case study on an Architectural Assumption Documentation Framework". In: *Journal of Systems and Software* 134.C (Dec. 2017). Keywords: ADFs, viewpoints, pp. 190–210.

Annotations: Describes an architecture framework for documenting assumptions constructed in the style of 42010 frameworks.

R. Youngs et al. "A standard for architecture description". In: *IBM Systems Journal* 38.1 (1999).

J. A. Zachman. "A framework for information systems architecture". In: *IBM Systems Journal* 26.3 (1987), pp. 276–292.

Annotations: A key paper underlying much work on enterprise architecture, and establishing an initial foundation for same.

Olaf Zimmermann et al. "Architectural Decision Guidance across Projects". In: *WICSA 2015*. Keywords: decision capture, templates. IEEE Computer Society, Apr. 2015.