ISO/IEC/IEEE 42010
Annotated Bibliography

version 4.2

June 2, 2015

Introduction

An annotated bibliography of papers, reports and books regarding 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.

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 on 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

ArchiMate 2.0 Specification. Jan. 2012. URL: http://www.opengroup.org/archimate/.

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

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.

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."

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

John K. Bergey 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.

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.

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 Morri-

son. Vol. 3047. Lecture Notes in Computer Science. St Andrews, UK, May 21–22 2004: Springer Berlin/Heidelberg, 2004, pp. 194–199. DOI: 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.

John Brøndum and Liming Zhu. "Towards an Architectural Viewpoint for Systems of Software Intensive Systems". In: *Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge*. SHARK '10. Cape Town, South Africa: Association for Computing Machinery, 2010, pp. 60–63. DOI: 10.1145/1833335.1833344.

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.

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.

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). DOI: 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). DOI: 10.1016/j.scico.2010.11.008.

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.

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), pp. 38–45. DOI: 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), pp. 33–35. DOI: 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. Springer Berlin Heidelberg, 2013, pp. 87–104. DOI: 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.

Asesh 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.

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.”

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.

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/ECSA 2009)*. Ed. by Rick Kazman et al. IEEE Computer Society Press, 2009, pp. 31–40. DOI: 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.

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.

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

Mirco Franzago, Ivano Malavolta, and Henry Muccini. “Stakeholders, Viewpoints and Languages of a Modelling Framework for the Design and Development of Data-Intensive Mobile Apps”. In: *Workshop MOBILEng 2014* (2015). eprint: arXiv:1502.04014.

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)*. IEEE Computer Society, 2012.

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), pp. 136–169. ISSN: 1619-1366. DOI: 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.

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.

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-x. 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.

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". 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), pp. 1307–1322. DOI: 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.

K. Eric Harper and Jiang Zheng. "Exploring Software Architecture Context". In: *12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), 4–7 May 2015, Montréal, Québec, Canada*. Montreal, QC Canada: IEEE Computer Society, 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.

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), pp. 795–820. DOI: 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)*. 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.

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. *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)*. Helsinki, Finland: IEEE Computer Society, 2012.

Rich Hilliard et al. “Realizing architecture frameworks through megamodelling techniques”. In: *25th IEEE/ACM International Conference on Automated Software Engineering (ASE 2010)*. 2010. URL: http://megaf.di.univaq.it/.

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

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.

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), pp. 1232–1248. DOI: 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.

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.

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.

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.

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.

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.

Rainer Koschke and Daniel Simon. “Hierarchical Reflexion Models”. In: *Proceedings of the 10th Working Conference on Reverse Engineering*. WCRE ’03. 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 the Quality of Software Architectures (QoSA 2015)*. Montreal, QC Canada: Association for Computing Machinery, 2015, pp. 129–134.

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 2015)*. Montreal, QC Canada: Association for Computing Machinery, 2015, pp. 21–26.

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), 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 Soft-

ware Architecture Practice". In: *IEEE Software* 26.2 (March–April 2009), pp. 36–42. DOI: 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.

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.

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.

Zengyang Li, Peng Liang, and Paris Avgeriou. "Architecture Viewpoints for Documenting Architectural Technical Debt". In: *Software Quality Assurance in Large Scale and Complex Software-intensive Systems*. Elsevier, 2015.

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.

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.

Annotations: Overview of IEEE 1471 after its publication.

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

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.

Nicholas May. “A Survey of Software Architecture Viewpoint Models”. In: *Sixth Australasian Workshop on Software and System Architectures*. May 2005, pp. 13–24. URL: http://mercury.it.swin.edu.au/ctg/AWSA05/Papers/may.pdf.

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.

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*. 2011, pp. 285–294. DOI: 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: *12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), 4–7 May 2015, Montréal, Québec, Canada*. Montreal, QC Canada: IEEE Computer Society, 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.

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

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 Missikoff. 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.

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.

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*. 2009, pp. 1–8. DOI: 10.1109/SHARK.2009.5069109.

Dewayne E. Perry and Alexander L. Wolf. "Foundations for the study of Software Architecture". In: *ACM SIGSOFT Sofware 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.

Eltjo Poort and Eric van der Vliet. "Architecting in a Solution Costing Context: Early Experiences with Solution-Based Estimating". In: *12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), 4–7 May 2015, Montréal, Québec, Canada*. Montreal, QC Canada: IEEE Computer Society, 2015, pp. 127–130.

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

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 adapated 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)*. Helsinki, Finland: IEEE Computer Society, 2012.

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

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).

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.

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

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

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. “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: *12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), 4–7 May 2015, Montréal, Québec, Canada*. Montreal, QC Canada: IEEE Computer Society, 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*. 18–22 February 2008. Vancouver, BC Canada: IEEE Computer Society, 2008, pp. 167–176. DOI: 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), 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.

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.

Harald Storrle. “Structuring very large domain models: experiences from industrial MDSD projects”. In: *ECSA ’10: Proceedings of the Fourth European Conference on Software Architecture*. Copenhagen, Denmark: ACM, 2010, pp. 49–54. DOI: 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.

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.

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. Springer Berlin Heidelberg, 2010, pp. 71–85. DOI: 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 address 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*. Helsinki, Finland: ACM, 2012, pp. 163–166. DOI: 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.

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.

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.

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.

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

Annotations: Paper motivates introduction of concept of architectural perspectives, in contrast to 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.

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