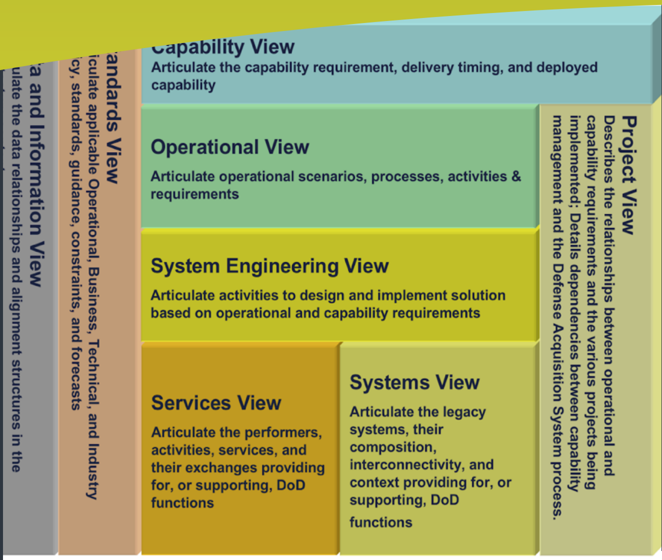
ISSUE
05
June
2020

Royal Academy of Engineering

Digitalization Transformation Guides



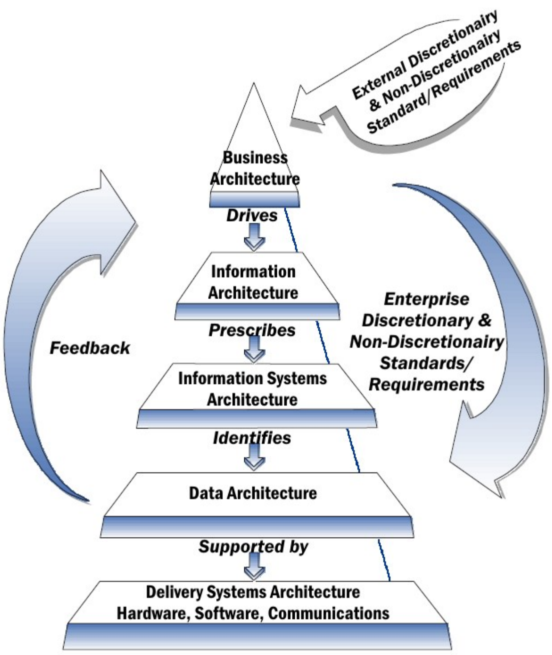
In this issue:

Solutions for a New Economy

Architecture Frameworks in the digital transformation landscape for modelling high-level models of digital systems and enterprise systems

Architecture Frameworks in the Digital Transformation Landscape

Enterprise Frameworks (EA) are often referred to as architecture frameworks (AF), particularly in a SE and MBSE context where the enterprise is viewed as a "system". AF assist systems engineering in developing a series of models using predefined guidelines and guide the engineer through the systems development lifecycle; requirements engineering, design and analysis, and verification and validation.



Loughborough University

Contents

Architecture Frameworks

Architecture development process is one of the most important parts of the SoS design and development lifecycle, where the structure that defines the connections and interactions between the constituent system is constructed. The process also identifies the types of the constituent systems that contribute to the SoS and their operational parameters and properties. Architecture development as part of the systems engineering (SE) process is important in the following methods and tasks:

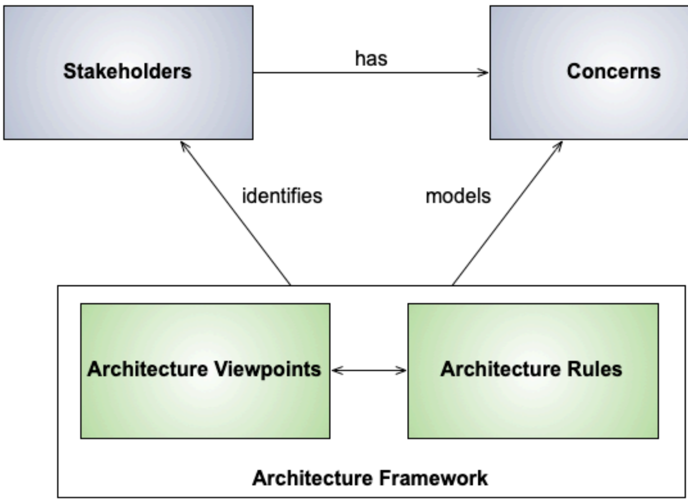
1. Requirements analyses, definition and management as basis for the development of the system
2. System design (in several phases and levels)
3. Change management to trace changes in the contract or in the development of the system
4. Integration to ensure connectivity to other systems
5. Verification and validation to proof that the requirements are fulfilled
6. Risk management.

Many tool vendors currently offer products and services for applying MBSE and the inclusive modelling languages such as the commonly used; Unified Modelling Language, UML [1], [2], and the Systems Modelling Language, SysML [3]–[5] and various architecture frameworks, such as DoDAF, MoDAF, NAF and UPDM. To name a few tool vendors and their products:

- IBM – Rational Rhapsody [6]
- Catia/ No Magic – Magic Draw [7]
- Modelio [8]
- Sparx Systems – Enterprise Architect [9]
- Visual Paradigm [10]
- Eclipse – Eclipse Papyrus [11]

AF are used to identify SoS stakeholders and their particular concerns, including functionality, cost and reliability. An extremely recent survey by MITRE [12] found that the general-purpose modelling language, SysML, was the most common language used to create systems architecture, with 14/17 responses identifying this particular language. SysML is an extension of UML and offers nine diagram types for architecture generation and supports the specification, design, analysis and verification and validation of systems and complex systems.

Architecture Frameworks offer the systems engineer or architect, a much broader set of viewpoints to express the SoS from varying perspectives. Frameworks such a DoDAF, MoDAF, NAF and TOGAF, which are regarded the most relevant frameworks in SoS engineering [13], enable the systems architect to express a concern of a stakeholder from a more abstract standpoint rather than by the detail technical aspects. For instance, a stakeholder may only be interested in a set of operational scenarios, which can be modelled using the operational viewpoint within DoDAF or MoDAF, rather than the technical *how* aspects of constituent, or set of CSs. The AF addresses the concerns of the stakeholders and the particular viewpoint within an AF models and frames these concerns accordingly, by a set of corresponding rules.



Brief History

Enterprise architectures (EA) are an important tool in aligning business and systems, resources and technologies, to achieve enterprise integration and to accomplish their mission. From the ANSI/IEEE Standard 1471-2000, architecture is defined as the "*fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution,"* [14].

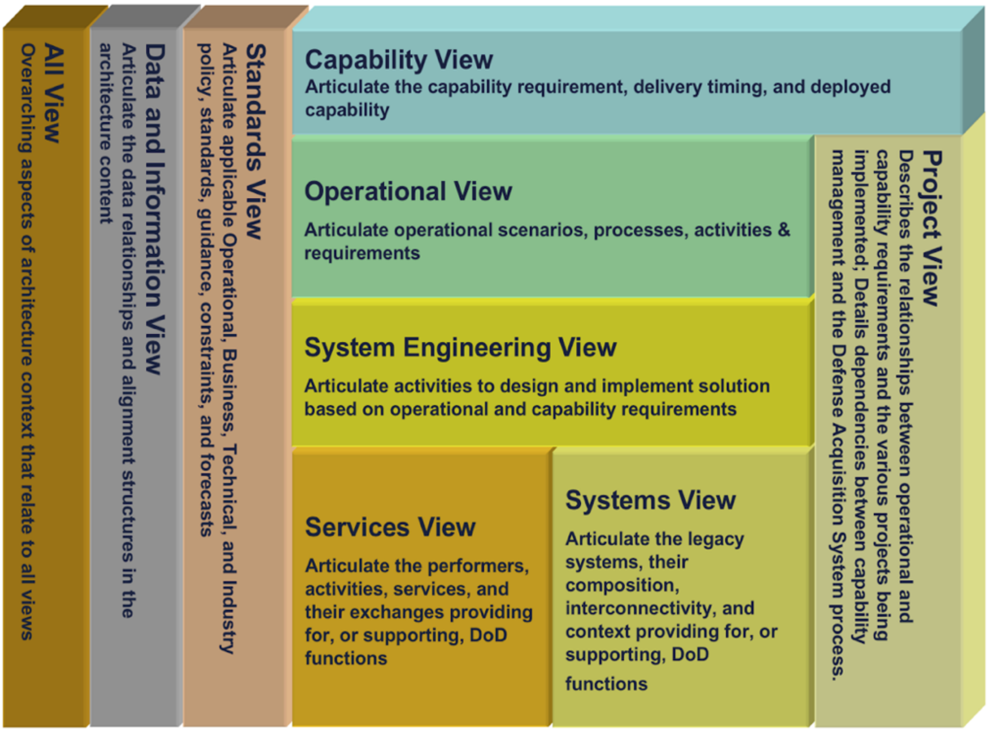
EA aligns an organisations software development processes and information technology (IT) with its business strategy. Frameworks of this type offer high-level outlines and models of the current enterprise landscape, allowing for future states to be explored and roadmaps to be created to achieve them. Numerous EA which offer various perspectives to describe the enterprise, its encompassing processes, information systems and transfers, infrastructure and organisational hierarchies. Some examples of EA include; the Zachman Framework [15]; The Open Group Architecture Framework (TOGAF) [16]; the Federal Enterprise Architecture Framework (FEAF); the Department of Defense Architecture Framework (DoDAF) [2], [13], [17], the Ministry of Defence Architecture Framework (MoDAF) [13], and the NATO Architecture Framework (NAF). Although these are the most common EA and AF, a host of domain and application specific frameworks also exist.

EA are often referred to as architecture frameworks (AF) [15], [18], particularly in a SE context where the enterprise is viewed as a "system". AF assist systems engineering in developing a series of models using predefined guidelines, depending on the framework. AF provide a certain set of perspectives called viewpoints that represent different aspects of a system or SoS [13]. These viewpoints themselves consist of multiple views or models, which are effectively individual models that allow the modelling of a single standpoint of an enterprise or system. Architecture frameworks provide a capability for implementing a systematic approach in understanding what aspects of a system would be relevant for a particular stakeholder or set of stakeholders, and a means to plan future phases of the organisation. A particular view can be adopted and adapted by the systems engineer or architect that best fulfils their set of requirements. Enterprise architecture frameworks are useful because they provide:

- Standard architecture definition and understanding
- A standard architecting process (although can be modified)
- Architecture Analysis
- Architecture Evolution
- Standard data structure to retain and relate information

Types of Frameworks

DoDAF

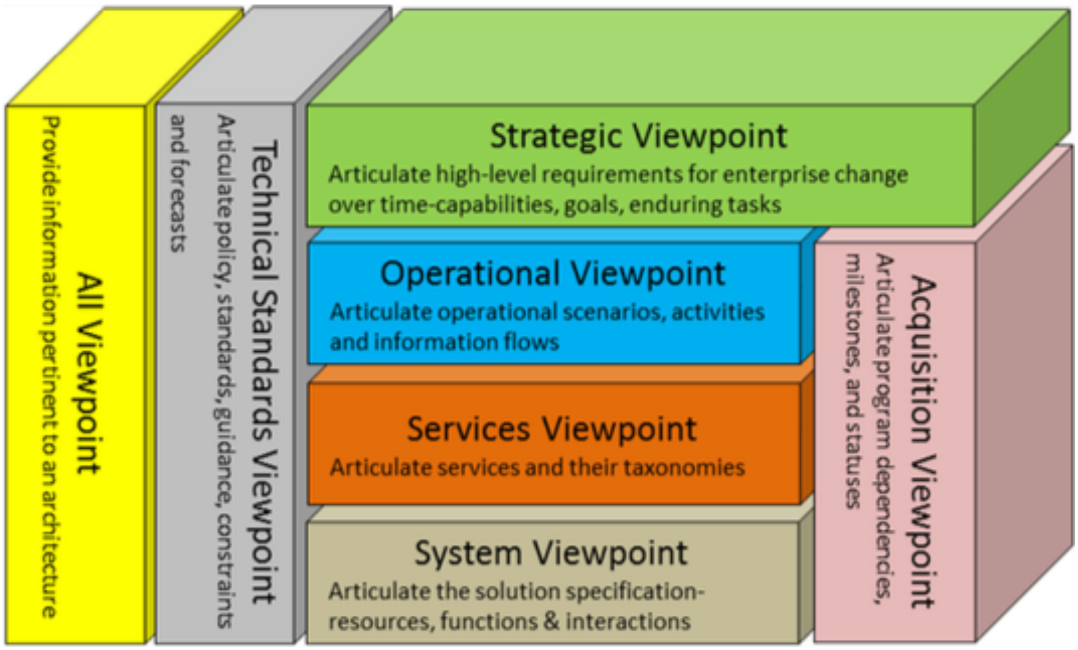


Viewpoint	Description
SV-1: Systems Interface Description	The identification of systems, system items, and their interconnections.
SV-2: Systems Resource Flow Description	A description of Resource Flows exchanged between systems.
SV-3: Systems-Systems Matrix	The relationships among systems in a given Architectural Description. It can be designed to show relationships of interest, (e.g., system-type interfaces, planned vs. existing interfaces).
SV-4: Systems Functionality Description	The functions (activities) performed by systems and the system data flows among system functions (activities).
SV-5a: Operational Activity to Systems Function Traceability Matrix	A mapping of system functions (activities) back to operational activities (activities).
SV-5b: Operational Activity to Systems Traceability Matrix	A mapping of systems back to capabilities or operational activities (activities).

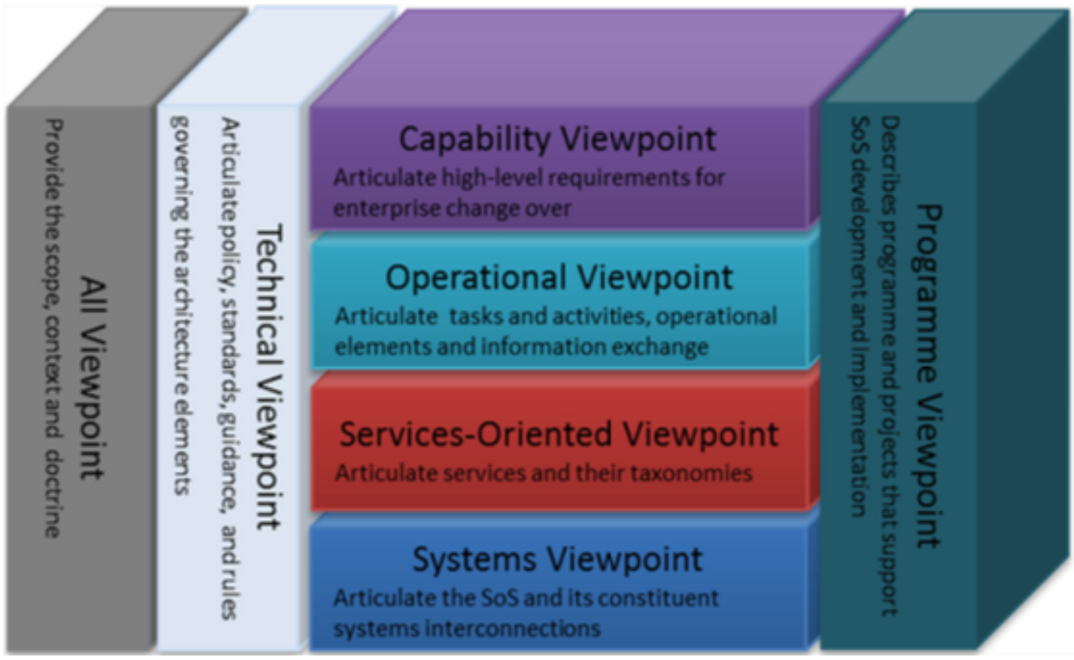
SV-6: Systems Resource Flow Matrix	Provides details of system resource flow elements being exchanged between systems and the attributes of that exchange.
SV-7: Systems Measures Matrix	The measures (metrics) of Systems Model elements for the appropriate timeframe(s).
SV-8: Systems Evolution Description	The planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation.
SV-9: Systems Technology & Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future system development.
SV-10a: Systems Rules Model	One of three models used to describe system functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.
SV-10b: Systems State Transition Description	One of three models used to describe system functionality. It identifies responses of systems to events.
SV-10c: Systems Event-Trace Description	One of three models used to describe system functionality. It identifies system-specific refinements of critical sequences of events described in the Operational Viewpoint.

Viewpoint	**Description**
OV-1: High-Level Operational Graphic	The high-level graphical/textual description of the operational concept.
OV-2: Operational Resource Flow Description	A description of the Resource Flows exchanged between operational activities.
OV-3: Operational Resource Flow Matrix	A description of the resources exchanged and the relevant attributes of the exchanges.
OV-4: Organizational Relationships Chart	The organizational context, role or other relationships among organizations.
OV-5a: Operational Activity Decomposition Tree	The capabilities and activities (operational activities) organized in a hierarchal structure.
OV-5b: Operational Activity Model	The context of capabilities and activities (operational activities) and their relationships among activities, inputs, and outputs; Additional data can show cost, performers or other pertinent information.
OV-6a: Operational Rules Model	One of three models used to describe activity (operational activity). It identifies business rules that constrain operations.
OV06b: State Transition Description	One of three models used to describe operational activity (activity). It identifies business process (activity) responses to events (usually, very short activities).
OV-6c: Event-Trace Description	One of three models used to describe activity (operational activity). It traces actions in a scenario or sequence of events.

MoDAF – Ministry of Defence Architecture Framework



NAF – NATO Architecture Framework



Other Architecture Frameworks, which have also been explored are The Open Group Architecture Framework (TOGAF) [refs], the Zachman Framework [refs], IEEE 1471, ISO 42010 [ref] and The Unified Profile of DoDAF and MoDAF (UPDM), which is essentially a profile developed by OMG to combine the views from the two respective frameworks [refs].

Limitations of Architecture Frameworks

Unfortunately, the seamless integration of architecture modelling and model simulation is something that is highly desired by all practitioners, however this capability is limited. Some tools claim [19]–[21] to have an in-built simulation capability; however, this has been found to be limited, and often non-existent, particularly in systems architecture modelling tools. Also, there is a requirement [13], [22], [23] to trace key relationships, parameter dependencies, and important information or data flows between parts. This is also a great shortcoming in current MBSE methods.

How to benefit from Architecture Frameworks

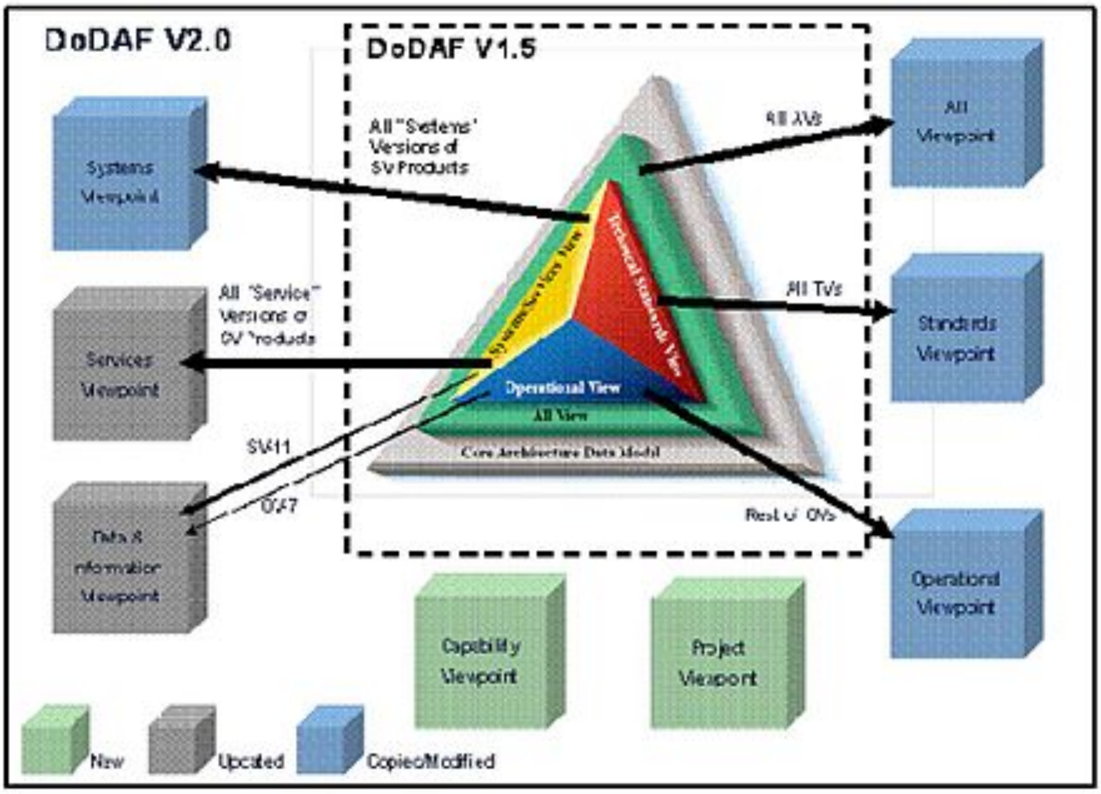
By enabling the architect or engineer to divide an architectural description into domains, viewpoints/ perspectives, and different levels of abstraction, it facilitates a greater understanding of a system. Design decisions can be made from effective architectural modelling where alternative solutions can be modelled and explored. Systematically, the development of an architectural description will naturally encourage the engineer to think about the connectivity and the operations of the elements which successfully bring about the behaviours, enabling the system to achieve its objectives. This comprehension enables the engineer to communicate designs with other stakeholders, promoting an evolutionary development lifecycle of the system or an enterprise.

Implementation of Architecture Frameworks

Selection of architectural viewpoints is key in the development of systems architecture modelling. Consisting of many different view packages and model types, it can be overwhelming in which to select for a given context or given scenario. Each viewpoint offers an alternative perspective. This should be the starting point in all view/model choices. Looking into the operational aspects of a system, service or product, the engineer must assess which operational views he/she is to implement. Typically, understanding the connectivity or relationships within a system set up is an initial task, therefore selection of the views which depicts interconnectivity (such as an OV-1 and OV-2 model) is a good starting point. Conjointly, a systems view could be used to address the connectivity between elements or subsystems, something like an SV-1.

Subsequently, understanding the flow of operations is likely to be the next objective of the engineer; of which can be achieved by implementing an OV-5 model that shows the flow of activities or actions by particular subsystems or actors. The operational views are complementary to the systems views as one shows the operations and the other describes the connectivity and the flow of data/information which makes those connections possible.

Realising the strengths of each model is central to architecture development; where time can be unproductive if the incorrect model type is selected and does not fill the requirements of that particular scenario. Thus, context is key. Plus, understanding of the capability of each view/model type within a particular framework.



References

[1] 'Unified Modeling Language™ (UML®) Resource Page', 2015. [Online]. Available: http://www.uml.org/.

[2] J. L. Hayden and A. Jeffries, 'On using SysML, DoDAF 2.0 and UPDM to model the architecture for the NOAA's Joint Polar Satellite System (JPSS) Ground System (GS)', *12th Int. Conf. Sp. Oper. SpaceOps 2012*, pp. 1–13, 2012.

[3] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML - The Systems Modeling Language*, 3rd Editio. Morgan Kaufmann, 2014.

[4] M. Mori, A. Ceccarelli, P. Lollini, B. Frömel, F. Brancati, and A. Bondavalli, 'Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile', *J. Softw. Evol. Process*, vol. 30, no. 3, pp. 1–20, 2018.

[5] S. Friedenthal, 'Evolving SysML and the System Modeling Environment To Support MBSE', *Insight*, vol. 19, no. 4, pp. 76–80, 2016.

[6] IBM, 'IBM Rational Rhapsody'. [Online]. Available: https://www.ibm.com/uk-en/marketplace/systems-design-rhapsody. [Accessed: 01-Apr-2020].

[7] Catia, 'Magic Draw'. [Online]. Available: https://www.nomagic.com/products/magicdraw. [Accessed: 01-Apr-2020].

[8] Modelio, 'Modelio'. [Online]. Available: https://www.modelio.org/. [Accessed: 01-Apr-2020].

[9] S. Systems, 'Enterprise Architect'. [Online]. Available: https://sparxsystems.com/. [Accessed: 01-Apr-2020].

[10] V. Paradigm, 'Visual Paradigm'. [Online]. Available: https://www.visual-paradigm.com/. [Accessed: 01-Apr-2020].

[11] Eclipse, 'Papyrus'. [Online]. Available: https://www.eclipse.org/papyrus/. [Accessed: 01-Apr-2020].

[12] T. A. McDermott, N. Hutchinson, M. Clifford, E. Van Aken, A. Slado, and K. Henderson, 'Benchmarking the Benefits and Current Maturity of Model-Based Systems Engineering across the Enterprise', 2020.

[13] M. Hause, 'The unified profile for DoDAF/MODAF (UPDM) enabling systems of systems on many levels', *2010 IEEE Int. Syst. Conf. Proceedings, SysCon 2010*, pp. 426–431, Apr. 2010.

[14] IEEE-SA Standards Board, 'IEEE Recommended Practice for Architectural Description of Software-Intensive Systems', *IEEE Std*, vol. 1471–2000, pp. 1–23, 2000.

[15] L. Urbaczewski and S. Mrdalj, 'A comparison of enterprise architecture frameworks', *Issues Inf. Syst.*, vol. 7, no. 2, pp. 18–23, 2006.

[16] T. Mueller, D. Schuldt, B. Sewald, and M. Morisse, 'Towards inter-organizational Enterprise Architecture Management - Applicability of TOGAF 9 . 1 for Network Organizations Completed Research Paper', *Proc. Ninet. Am. Conf. Inf. Syst.*, pp. 1–13, 2013.

[17] C. Piaszczyk, 'Model Based Systems Engineering with Department of Defense Architectural Framework', *Syst. Eng.*, vol. 14, no. 3, pp. 305–326, 2011.

[18] D. Ota and M. Gerz, 'Benefits and Challenges of Architecture Frameworks', *Sixt. Int. Command Control Res. Technol. Symp.*, 2011.

[19] S. Saravi *et al.*, 'A Systems Engineering Hackathon - A methodology involving multiple stakeholders to progress conceptual design of a complex engineered product', *IEEE Access*, 2018.

[20] T. Blockwitz *et al.*, 'Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models', *8th Int. Model. Conf. 2011*, pp.

173–184, 2012.
[21] Y. A. Feldman, L. Greenberg, and E. Palachi, ‘Simulating Rhapsody SysML Blocks in Hybrid Models with FMI’, *Proc. 10th Int. Model. Conf.*, pp. 43–52, 2014.
[22] A. Maté and J. Trujillo, ‘Tracing conceptual models’ evolution in data warehouses by using the model driven architecture’, *Comput. Stand. Interfaces*, vol. 36, no. 5, pp. 831–843, 2014.
[23] Y. Wang, J. H. Han, and P. Beynon-Davies, ‘Understanding blockchain technology for future supply chains: a systematic literature review and research agenda’, *Supply Chain Manag.*, 2018.

Useful information sources

- **This Guide has been funded under the RAEng Regional Engagement Award.**

Videos:

- https://www.youtube.com/watch?v=9TVc32M_gIY
- https://www.youtube.com/watch?v=Sd-2zG5L6t4
- https://www.youtube.com/watch?v=rhrjueWLPmE

Websites:

- https://dodcio.defense.gov/Library/DoD-Architecture-Framework/
- https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/architectural-frameworks-models-and-views
- https://www.visual-paradigm.com/guide/enterprise-architecture/enterprise-architecture-framework-in-a-nutshell/
- https://www.omg.org/

Guide written by:
Demetrios Joannou, Roy S. Kalawsky, Loughborough University
Contact details:
d.joannou@lboro.ac.uk
r.s.kalawsky@lboro.ac.uk