

Exploring Metadata in Enterprise Architecture by Applying Conceptual Graphs—Formal Concept Analysis (CGFCA) to Moregraph

SALEEM, Mahpara <<http://orcid.org/0009-0003-1671-4796>>, POLOVINA, Simon <<http://orcid.org/0000-0003-2961-6207>> and FALLON, Richard <<http://orcid.org/0000-0001-5747-8839>>

Available from Sheffield Hallam University Research Archive (SHURA) at:

<https://shura.shu.ac.uk/37926/>

This document is the Accepted Version [AM]

Citation:

SALEEM, Mahpara, POLOVINA, Simon and FALLON, Richard (2026). Exploring Metadata in Enterprise Architecture by Applying Conceptual Graphs—Formal Concept Analysis (CGFCA) to Moregraph. In: CROITORU, Madalina, STUMME, Gerd and LÓPEZ-RODRÍGUEZ, Domingo, (eds.) Conceptual Knowledge Structures: Third International Joint Conference, CONCEPTS 2026, Montpellier, France, August 31– September 4, 2026, Proceedings. Lecture Notes in Computer Science (16811). Springer Nature Switzerland, 362-372. [Book Section]

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

Exploring Metadata in Enterprise Architecture by applying Conceptual Graphs – Formal Concept Analysis (CGFCA) to Moregraph

Mahpara Saleem¹[0009–0003–1671–4796], Simon Polovina^{2,3}[0000–0003–2961–6207],
and Richard Fallon⁴[0000–0001–5747–8839]

¹ Mahaz Tech Enterprise Limited (MTE), Manchester, UK
mahpara@mahaztechenterprise.com

² Department of Computing and Digital Technologies,
Sheffield Hallam University, Sheffield, UK

³ Simile Logics Limited, Sheffield, UK
S.Polovina@shu.ac.uk, simon@similelogics.ltd

⁴ Red Rooster IT, Vienna, Austria
richard@fallon.at

Abstract. Enterprise Architecture (EA) is increasingly driven by heterogeneous metadata drawn from cloud-based APIs, ERP systems, and platform reference architectures. This paper revisits Moregraph, a metadata-driven EA platform that combines OData API metadata parsing, force-directed graph visualisation, and extends its specification with Conceptual Graph – Formal Concept Analysis (CGFCA). Moregraph’s specification is expressed as a Conceptual Graph (CG), which captures the overarching architectural concepts of metadata-driven EA. The CG is expressed in the Conceptual Graph Interchange Format (CGIF), which is part of the interoperable family of logics in the ISO/IEC 24707:2018 Common Logic standard. The CGIF encodes thirteen architectural concepts that include [Application: Moregraph], [Solution Architecture], [Business Data Graph], [SAP Graph], and [Microsoft Graph], connected by seventeen relational triples. The CGFCA algorithm transforms these triples into a 13×17 binary Formal Concept Analysis (FCA) context in Burmeister format, identifies seven direct pathways from the single input concept [Application: Moregraph] to the single output concept [Enterprise Architecture], all mediated by [Technical Debt]. We show that the combination of CG and FCA in CGFCA provides a principled foundation for navigating, validating, and analysing complex EA metadata landscapes.

Keywords: Enterprise Architecture · Conceptual Graphs · Formal Concept Analysis · Metadata · OData · CGIF · CGFCA · SAP Graph · Microsoft Graph

1 Introduction

Enterprise architects managing large SAP or Microsoft deployments face a fundamental challenge: thousands of data entities are published as machine-readable OData metadata through SAP Graph [17] and Microsoft Graph [7], yet the architecturally significant relationships among them—Technical Debt, Solution Architecture alignment, Clean Core compliance [22, 26, 16]—remain undocumented in any formal, machine-readable form [20]. Bridging this gap requires a formal knowledge representation capable of encoding typed architectural relationships and an automated means of deriving structural insights from it.

Conceptual Graphs (CGs) [18, 19, 10] provide a logic-based notation with direct first-order semantics, standardised in CGIF (ISO/IEC 24707:2018 [5]). Formal Concept Analysis (FCA) [23, 4] derives hierarchical concept lattices from binary cross-tables. CGFCA builds on Wille’s Concept Graphs [24] to bring CG and FCA together [1, 2], enabling lattice-based structural classification beyond conventional graph traversal. This paper revisits Moregraph [11], a metadata-driven EA platform, and extends its specification with CGFCA. Moregraph’s specification is expressed as a CG in CGIF, enabling interoperability between CGs and other logics [3].

To demonstrate the end-to-end workflow, we convert Moregraph’s CG to CGIF and translate it into a Burmeister FCA context using the open-source CharGer tool¹, whose CGFCA source code is also demonstrated in Moregraph. The analysis identifies a single input concept [**Application: Moregraph**], a single output concept [**Enterprise Architecture**], and seven direct pathways between them, all passing through [**Technical Debt**] and [**Solution Architecture**]. The derived 13×17 FCA binary context exposes the full transitive reachability structure of the CG.

Beyond [11], this paper contributes: (i) the complete 13×17 FCA formal context; (ii) five structural observations identifying input/output concepts, root and leaf attributes as formal invariants not derivable by graph traversal alone; and (iii) a concept lattice confirming the top/bottom ordering—all of direct practical value to enterprise architects validating metadata traceability.

2 Background

A **Conceptual Graph** (CG) [18, 19, 10] consists of [**concept**] nodes and (**relation**) nodes with first-order semantics; CGIF [5] serialises them as typed tokens, enabling interoperability with other logics [3]. **Formal Concept Analysis** (FCA) [4, 23] derives concept lattices from binary formal contexts (G, M, I) ordered by extent inclusion [25], supporting ontology construction and information retrieval [21, 12, 14, 13]; **CGFCA** [1, 2] formalises the CG-to-FCA mapping, building on Wille’s Concept Graphs [24]. **OData** [9, 8] is an OASIS standard for REST-based data services; SAP Graph and Microsoft Graph expose large API

¹ <https://github.com/Conceptual-Graphs/CharGer>

estates via OData [17, 7]. **Enterprise Architecture** (EA) disciplines [6, 22] decompose information systems into Business, Application, and Data Architecture layers forming the Solution Architecture; SAP’s **Clean Core** strategy [16] mandates customisations within approved extension points, making metadata traceability essential for Technical Debt control.

3 The Moregraph Platform

Moregraph is a Next.js web application backed by PostgreSQL via the Prisma ORM [11], comprising four subsystems: the *OData Parser* (accepts EDMX, auto-detects V2/V4, extracts entity types and typed navigation properties in four categories: containment, external reference, internal navigation, and referential constraint); the *Graph Store*; the *Force-directed Visualiser* (physics simulation grouped by service, with entity panel and pie chart); and the *CGFCA Tool* (three-tab browser interface to upload CGIF/CSV/draw.io XML, run the algorithm client-side, and download .cxt and pathway .txt files).

4 CG Representation of the Moregraph EA

We used Moregraph’s CG specification [11], authored in CharGer and exported to CGIF [5], as the EA knowledge base for CGFCA. The CG encodes thirteen architectural concepts and seventeen relational triples (Figure 1), with [Application: Moregraph] as the unique input and [Enterprise Architecture] as the unique output.

Table 1 lists the thirteen concept types (left) and seventeen relational triples (right). Two concepts carry individual referents—[Application: Moregraph] and [Ref. Solution Architecture: api.sap.com]—while the remaining eleven are generic. In the CGIF encoding, *in-support-of* is n -ary (last referent = target EA, preceding referents = sources), compressing five triples into one token; *drawing-on* grounds Clean Core compliance in the SAP reference architecture at api.sap.com.

5 CGFCA: Triples to Binaries

5.1 Binary Context Generation

The CGFCA Tool accepts CGIF, CSV triples, or draw.io XML; for CGIF it scans [Type: referent] tokens then (relation refs...) tokens, treating the last referent as target and all preceding as sources. After parsing, *input concepts* (never a target) and *output concepts* (never a source) are identified. The formal context is then constructed as follows:

- **Objects G :** all CG concepts (13 for the Moregraph CG).

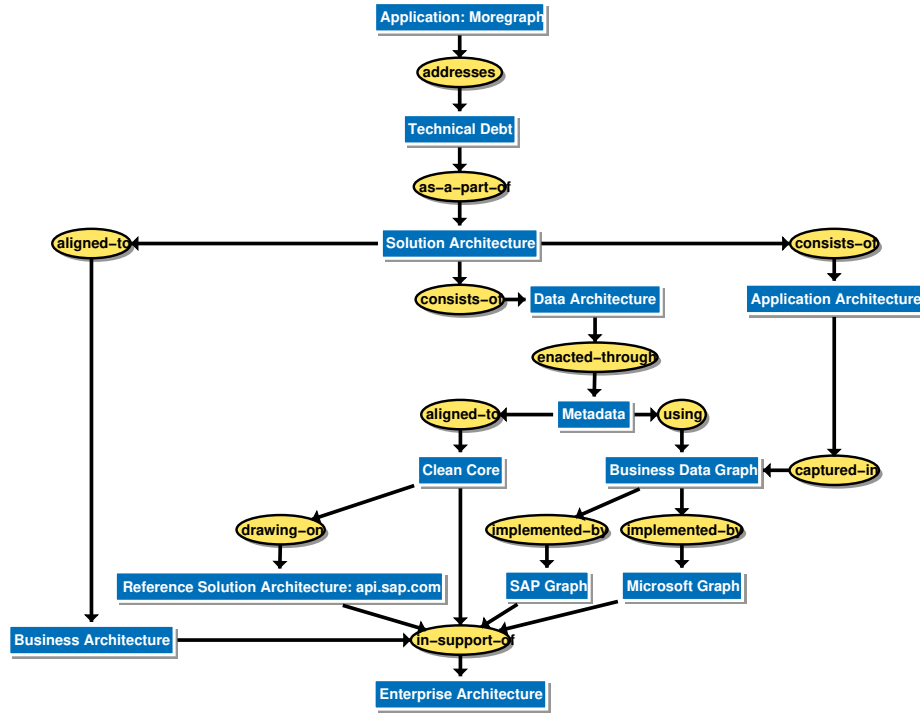


Fig. 1: CG representation of the Moregraph specification.

- **Attributes M :** all triples, labelled “source relation”. The 17 triples yield 17 attribute columns. Two pairs share the same source-relation label with distinct targets; these are disambiguated by appending the target in parentheses (e.g., *SA consists-of (AA)* and *SA consists-of (DA)*). ConExp treats column headers as unique strings; the appended targets ensure all 17 attribute labels are distinct in the exported .cxt file.
- **Incidence I :** $g I m$ iff concept g is reachable as a target from triple m through the transitive relational closure.

Transitivity is computed by the `ADDBINARY` procedure, called once per triple: starting from the triple’s direct target, it follows outgoing triples recursively, marking each visited concept as reachable; a path-membership check prevents revisiting nodes, giving worst-case $O(|G| \cdot |M|)$ complexity. The result is written as a Burmeister .cxt file: header B, object and attribute counts, names, and a binary matrix of X/. symbols.

5.2 FCA Context for the Moregraph CG

Table 2 shows the complete 13×17 binary FCA context generated from the Moregraph CGIF, matching the .cxt file produced by the CGFCA Tool. Rows

Table 1: Concept types (left) and relational triples (right) in the Moregraph CG. Abbreviations in the right half are introduced in the left.

Concept	Referent	Relation	Source Target	
		addresses	MG	TD
		as-a-part-of	TD	SA
Application (MG)	Moregraph	consists-of	SA	DA
Ref. Solution Arch. (RSA)	api.sap.com	consists-of	SA	AA
Enterprise Architecture (EA)	(generic)	aligned-to	SA	BA
Solution Architecture (SA)	(generic)	enacted-through	DA	M
Application Architecture (AA)	(generic)	captured-in	AA	BDG
Business Architecture (BA)	(generic)	using	M	BDG
Data Architecture (DA)	(generic)	aligned-to	M	CC
Metadata (M)	(generic)	drawing-on	CC	RSA
Business Data Graph (BDG)	(generic)	implemented-by	BDG	MS
Microsoft Graph (MS)	(generic)	implemented-by	BDG	SAP
SAP Graph (SAP)	(generic)	in-support-of	MS	EA
Technical Debt (TD)	(generic)	in-support-of	SAP	EA
Clean Core (CC)	(generic)	in-support-of	BA	EA
		in-support-of	RSA	EA
		in-support-of	CC	EA

are concepts (objects); columns are triples (attributes) labelled by source and relation. **X** marks reachability—direct or transitive. Five structural observations follow: (1) Row 1 (Application: Moregraph) no **X**—unique **input**; (2) Row 13 (EA) **X** in every column—unique **output**; (3) Col. 1 (MG addresses) covers all twelve other rows—**root attribute**; (4) Cols. 3, 9, 15–17 (*in-support-of*) cover only Row 13—five **leaf attributes**; (5) Col. 12 (CC drawing-on) covers only Rows 12–13, confirming RSA reachable exclusively via *drawing-on*.

6 Experimental Results

6.1 Pathway Analysis

Table 3 lists the seven direct pathways from *Application: Moregraph* to *Enterprise Architecture*. Every pathway traverses MG $\xrightarrow{\text{addresses}}$ TD $\xrightarrow{\text{as-a-part-of}}$ SA, making *Technical Debt* and *Solution Architecture* mandatory mediators. Pathway 4 (via **drawing-on**) confirms Clean Core compliance traceability to the SAP reference architecture; no cyclic dependencies were detected.

6.2 Structural Observations from the Binary Context

Table 2 supports lattice observations without full lattice computation. Under the Ganter–Wille ordering [4], *Enterprise Architecture* (Row 13, all **X**) generates the **bottom** concept; *Application: Moregraph* (Row 1, all dot) generates the **top**

Table 2: Complete 13×17 FCA formal context from the Moregraph CGIF.[illegible]

Table 3: Seven direct pathways from *Application: Moregraph* to *Enterprise Architecture* (CGFCA output).

#	Pathway (relation labels omitted for brevity)
1	MG → TD → SA → DA → M → BDG → MS → EA
2	MG → TD → SA → DA → M → BDG → SAP → EA
3	MG → TD → SA → DA → M → CC → EA
4	MG → TD → SA → DA → M → CC → RSA → EA
5	MG → TD → SA → BA → EA
6	MG → TD → SA → AA → BDG → MS → EA
7	MG → TD → SA → AA → BDG → SAP → EA

concept. Cols. 3, 9, 15–17 (*in-support-of* leaf attributes) have extent $\{EA\}$ only; Cols. 1–2 cover the most objects and are the most generative. *Reference Solution Architecture* (Row 12) is reachable exclusively via *Clean Core drawing-on*, giving it the smallest non-empty intent. Figure 2, computed in ConExp², confirms this ordering: EA at the bottom node, Application: Moregraph at the top, with the seven pathways visible in the branching structure. The lattice additionally reveals which concepts share identical structural positions (same lattice node), a classification not expressible as a formal invariant by directed-graph traversal.

7 Related Work

The theoretical relationship between CGs and FCA was studied by Andrews and Polovina [1], who provided a foundational mapping from CG triples to FCA formal contexts and compared the approach to Wille’s Concept Graphs [24]—a distinct diagrammatic reasoning system grounded directly in FCA, which Wille intentionally named to distinguish it from Sowa’s Conceptual Graphs. CGFCA was further described in [2]. Rudolph and Stumme [14] showed that concept lattices can be constructed from CG projection graphs; Priss and Old [13] modelled semantic aspects of CGs using FCA incidence structures. The CGFCA algorithm is complementary: it constructs a formal context directly from the triple structure, capturing transitive reachability and pathway topology rather than via projection. While pathway analysis from input to output concept is achievable through conventional directed-graph traversal, the FCA formal context additionally enables lattice-based structural classification: the concept lattice identifies each architectural element’s structural role—input/output concepts, leaf and root attributes, and relative generativity—as formal invariants of the knowledge representation, beyond what graph traversal alone provides [25, 4].

The Moregraph platform was introduced in [11]. The present paper adds: (1) the complete 13×17 FCA formal context in Burmeister format; (2) five structural observations identifying root/leaf attributes and input/output concepts as formal invariants; and (3) a concept lattice confirming the top/bottom

² <https://conexp.sourceforge.net/>

ordering. A companion paper [15] extends the platform with a Metadata-to-Governance Transformation (MGT), mapping OData metadata to governed EA artefacts in SAP LeanIX.

In EA tooling, Lankhorst [6] advocates for model-driven EA using ArchiMate; Moregraph differs in grounding its model in machine-readable OData metadata rather than manually authored diagrams. Priss [12] surveys FCA in information science, underscoring its value for hierarchical knowledge organisation—a central requirement in EA.

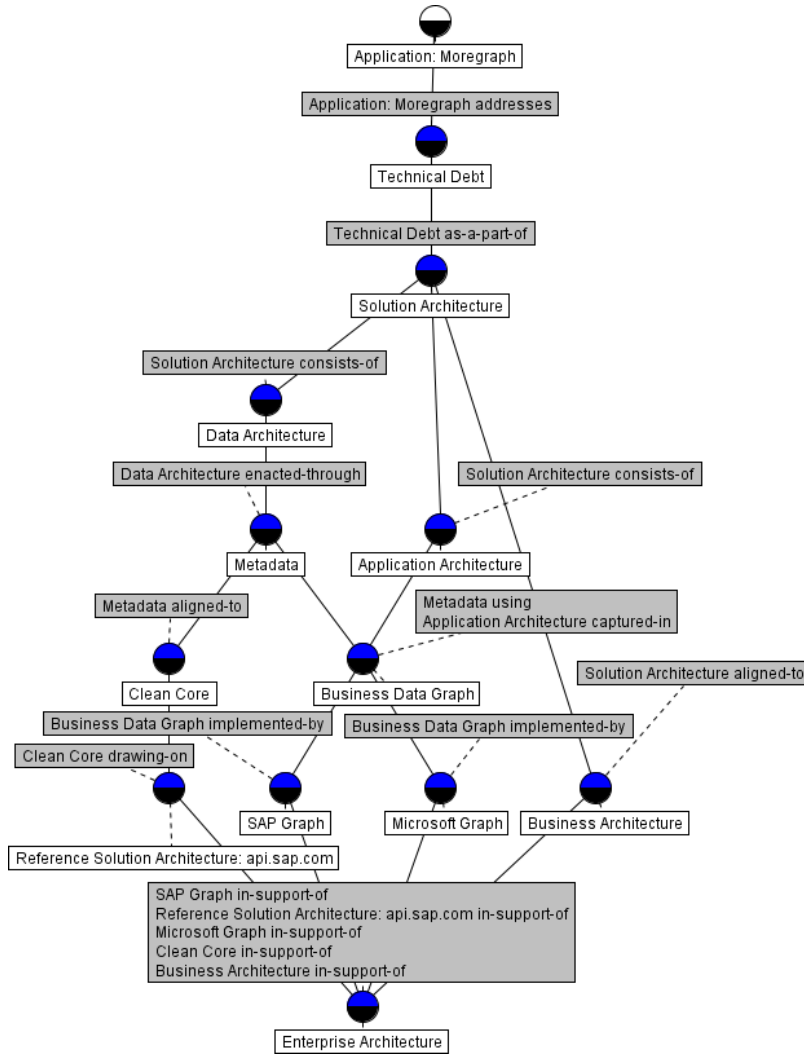


Fig. 2: Moregraph CG formalised as a concept lattice (CGFCA and ConExp).

8 Conclusion

We have presented Moregraph, a metadata-driven EA platform combining OData metadata parsing, force-directed visualisation, and a browser-based CGFCA tool, applied to a thirteen-concept, seventeen-triple CG in CGIF. The resulting 13×17 Burmeister FCA context identifies seven direct pathways from *Application: Moregraph* to *Enterprise Architecture*, all mediated by *Technical Debt* and *Solution Architecture*, with no cyclic dependencies. The formal context reveals EA is transitively reachable from every triple and that *addresses* is the root attribute—non-trivial invariants that emerge automatically, beyond what directed-graph traversal alone exposes. A companion paper [15] extends this into a Metadata-to-Governance Transformation (MGT) for SAP LeanIX; future work will render the concept lattice interactively within Moregraph.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Andrews, S., Polovina, S.: A Mapping from Conceptual Graphs to Formal Concept Analysis. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) *Conceptual Structures for Discovering Knowledge*, The 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK. pp. 63–76. Springer, Heidelberg (2011)
2. Andrews, S., Polovina, S.: Exploring, Reasoning with and Validating Directed Graphs by Applying Formal Concept Analysis to Conceptual Graphs. In: Croitoru, M., Marquis, P., Rudolph, S., Stapleton, G. (eds.) *Graph Structures for Knowledge Representation and Reasoning*, 5th International Workshop, GKR 2017, Melbourne, VIC, Australia, August 21, 2017, Revised Selected Papers. pp. 3–28. No. 10775 in *Lecture Notes in Artificial Intelligence*, Springer (2018)
3. Bridges, S., Schiffel, J., Polovina, S.: OpenSEA: A Framework for Semantic Interoperation Between Enterprises. In: Bessis, N., Xhafa, F. (eds.) *Next Generation Data Technologies for Collective Computational Intelligence*, pp. 61–86. No. 352 in *Studies in Computational Intelligence*, Springer (2011). <https://doi.org/10.1007/978-3-642-20344-2>, <https://shura.shu.ac.uk/3819/>
4. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, Heidelberg (1999)
5. ISO/IEC: Information technology—common logic (CL): A framework for a family of logic-based languages. ISO/IEC 24707:2018. Tech. rep., International Organization for Standardization (2018)
6. Lankhorst, M.: *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Heidelberg, 4th edn. (2017)
7. Microsoft Corporation: Microsoft Graph REST API v1.0 reference. Tech. rep., Microsoft Corporation (2024), available at <https://learn.microsoft.com/en-us/graph/api/overview>
8. OASIS: Open data protocol (OData) version 2.0. Tech. rep., OASIS Open (2014), oASIS Standard
9. OASIS: OData version 4.01. part 1: Protocol. Tech. rep., OASIS Open (2020), oASIS Standard

10. Polovina, S.: An Introduction to Conceptual Graphs. In: Priss, U., Polovina, S., Hill, R. (eds.) *Conceptual Structures: Knowledge Architectures for Smart Applications*. pp. 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
11. Polovina, S., Fallon, R., Saleem, M.: Moregraph: Metadata-driven enterprise architecture using conceptual structures. In: *Conceptual Knowledge Structures: CONCEPTS 2025*. Lecture Notes in Computer Science, vol. 15941. Springer (2025)
12. Priss, U.: Formal concept analysis in information science. *Annual Review of Information Science and Technology* **40**(1), 521–543 (2006)
13. Priss, U., Old, L.J.: Modelling semantic aspects of conceptual graphs with FCA. In: *Conceptual Structures: Leveraging Semantic Technologies, 17th International Conference on Conceptual Structures, ICCS 2009*. Lecture Notes in Artificial Intelligence, vol. 5662, pp. 193–206. Springer (2009)
14. Rudolph, S., Stumme, G.: Constructing lattices from conceptual graphs. In: *Conceptual Structures at Work: 12th International Conference on Conceptual Structures, ICCS 2004*. Lecture Notes in Artificial Intelligence, vol. 3127, pp. 139–155. Springer, Berlin, Heidelberg (2004)
15. Saleem, M., Polovina, S.: Advancing metadata-driven enterprise architecture management tools using conceptual structure. In: *Conceptual Knowledge Structures: CONCEPTS 2026*. Springer (2026), in press
16. SAP SE: Clean core strategy for SAP S/4HANA. SAP Community (2023), available at community.sap.com
17. SAP SE: SAP Graph: A unified API for SAP data. Tech. rep., SAP SE (2024), available at api.sap.com
18. Sowa, J.F.: *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA (1984)
19. Sowa, J.F.: Conceptual Graphs for Representing Conceptual Structures. In: *Conceptual Structures in Practice*. Chapman and Hall/CRC (2009), num Pages: 36
20. Sowa, J.F., Zachman, J.A.: Extending and formalizing the framework for information systems architecture. *IBM systems journal* **31**(3), 590–616 (1992)
21. Stumme, G., Maedche, A.: FCA-MERGE: Bottom-up merging of ontologies. In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*. pp. 225–230. Morgan Kaufmann (2001)
22. The Open Group: TOGAF standard, version 10. Tech. rep., The Open Group (2025), latest
23. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht (1982)
24. Wille, R.: Conceptual graphs and formal concept analysis. In: *Conceptual Structures: Fulfilling Peirce’s Dream: Fifth International Conference on Conceptual Structures, ICCS 1997*. Lecture Notes in Computer Science, vol. 1257, pp. 290–303. Springer, Berlin, Heidelberg (1997)
25. Wille, R.: Formal concept analysis as mathematical theory of concepts and concept hierarchies. In: *Formal Concept Analysis: Foundations and Applications*, Lecture Notes in Computer Science, vol. 3626, pp. 1–33. Springer, Berlin, Heidelberg (2005)
26. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* **26**(3), 276–292 (1987)