Author pre-print version. Paper accepted for the PoEM'26 Forum.
Final Paper to be published by CEUR.

Supporting the Assessment of Software Modernisation Rules and Regulatory ICT Requirements Through Enterprise Architecture Models*

Beāte Krauze[1,*], Ada Slupczynski[2], Simon Hacks[3], Philipp-Lorenz Glaser[4] and Dominik Bork[4]

[1]Institute of Information Technology, Riga Technical University, Kipsalas Str. 6a, LV-1048 Riga, Latvia
[2]Research Group Software Construction, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany
[3]Department of Computer and Systems Sciences, Stockholm University, Borgarfjordsgatan 12, 164 55 Kista, Sweden
[4]Business Informatics Group, TU Wien, Erzherzog-Johann-Platz 1, 1040 Vienna, Austria

Abstract

Enterprise modelling supports the analysis of organisational change by representing business objectives, processes and information systems. Using Enterprise Architecture (EA) models to assess software modernisation and Information and Communication Technology (ICT) regulatory requirements set out in applicable legislation requires determining what evidence the EA models provide and which conclusions that evidence supports. This paper presents an assessment process that identifies the required information, defines assumptions for translating requirements into graph queries and interprets the results in the context of each model. We examine the information required to assess 106 rules and execute eleven selected Cypher queries on 961 publicly sourced ArchiMate models. The results show that structural queries can be reused across models, but often identify patterns broader than the original requirements. Queries using element properties can define the assessment scope more precisely, but their applicability depends on metadata that is limited and inconsistent across repositories. These findings show that assessments should distinguish possible EA problems from missing model evidence and state when there is not enough information to assess a requirement. The proposed process clarifies which aspects of a requirement graph queries can assess and when additional evidence or human judgement is needed. It contributes to enterprise modelling by showing how requirements can guide EA model analysis and help identify the information models need to represent for assessment.

Keywords

Enterprise Architecture, Requirements assessment, Graph queries, ArchiMate, Information Systems, EU legislation, Modernisation

1. Introduction

Organisations modernise information systems to improve maintainability, interoperability, security and continuity. Decisions about replacing software applications, changing integrations or migrating technology must address organisational objectives, software modernisation needs and Information and Communication Technology (ICT) requirements set out in applicable legislation. Previous work analyses modernisation in relation to governance, resilience and Enterprise Architecture (EA) debt [1, 2, 3, 4]. EA models describe business activities, responsibilities, applications, information and technology together with their relationships. Such descriptions can help identify missing ownership information, incomplete dependencies or absent links between planned changes and objectives. A graph representation makes these patterns accessible to executable queries [5]. This raises the following question: *which aspects of*

Companion Proceedings of the 19th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling: PoEM Forum, Doctoral Consortium, Business Case & Tool Forum, Workshops (PoEM-C 2026) co-located with PoEM 2026, Mytilene, Greece, December 1-3, 2026

*Corresponding author.
beate.krauze@rtu.lv (B. Krauze); slupczynski@swc.rwth-aachen.de (A. Slupczynski); simon.hacks@dsv.su.se (S. Hacks); philipp-lorenz.glaser@tuwien.ac.at (P. Glaser); dominik.bork@tuwien.ac.at (D. Bork)
0009-0005-9630-2234 (B. Krauze); 0000-0001-5403-7720 (A. Slupczynski); 0000-0003-0478-9347 (S. Hacks); 0000-0002-0710-8052 (P. Glaser); 0000-0001-8259-2297 (D. Bork)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

software modernisation needs and regulatory ICT requirements can be assessed through EA models, and how do the available model information and query assumptions constrain the conclusions?

This paper presents a process for assessing software modernisation and regulatory ICT rules through EA models. It classifies 106 rules according to the information required for their assessment and executes eleven selected Cypher queries on 961 ArchiMate models. The evaluation examines what the queries identify and how missing model information and query assumptions affect the interpretation of results. The contribution to enterprise modelling is to show how requirements guide the selection of model evidence, the construction of queries and the interpretation of findings. The process makes explicit which aspects of a rule can be examined through graph queries and where additional evidence or interpretation is required. Accordingly, we formulate the following research questions:

1. RQ1: How can rules be operationalised as graph queries to support requirements assessment through EA models?
2. RQ2: What information and interpretation constraints affect their assessment across EA models?

The paper is structured as follows. Section 2 reviews the background and related work. Section 3 describes the research approach and assessment process, including the classification of rules according to their information needs and their operationalisation as graph queries. Section 4 presents the results of executing eleven selected queries on 961 ArchiMate models. Section 5 discusses the findings in relation to the research questions and examines the limitations and threats to validity. Section 6 summarises the conclusions and outlines further work.

The publicly available repository contains the supplementary materials: data set, rules, Cypher queries, source code, query outputs and analysis results. Zenodo: 10.5281/zenodo.23300069.

2. Background and related work

Research on regulatory compliance in information systems recognises the need to connect legal obligations with systems development and organisational processes [6, 7]. Legal requirements engineering examines how obligations can be identified, analysed and represented as requirements [8, 9], with their interpretation depending on the application context [10]. A systematic mapping documents goal and other modelling approaches to compliance [11]. More recent research identifies challenges concerning stakeholder involvement and connections between regulatory requirements engineering and other software lifecycle activities [12]. Research on privacy by design further highlights the need to preserve legal concepts and traceability across requirements and system specifications [13].

These contributions motivate traceability from a rule to the evidence used in its assessment. The analysed rules are based on ICT requirements gathered from the Directive (EU) 2022/2555, known as NIS2, the Regulation (EU) 2022/2554 or the Digital Operational Resilience Act (DORA) and the Regulation (EU) 2024/1689, the Artificial Intelligence Act (AI Act) [14, 15, 16] and on software modernisation literature [1, 3]. Together, they address governance, security, dependencies, strategic alignment, ownership, documentation and reassessment. Their source determines their rationale and applicability, while their operationalisation is examined using common criteria concerning the required model evidence.

Transforming EA models into graphs enables analysis of elements and their connections. The interpretation of model evidence also depends on why a model was created and who it serves. Schoonderbeek and Proper [17] explicitly represent model objectives and audiences in their ontology for EA modelling and model quality. This motivates interpreting missing properties or relationships against the declared scope and modelling conventions of the repository. Glaser et al. [18] describe the construction of EA knowledge graphs, providing a foundation for querying model elements and their connections. Fumagalli et al. [19] investigate the discovery of recurring structures in conceptual models, including ArchiMate. The EA ModelSet supplies a heterogeneous collection of models for examining such modelling practices [20]. Related work on data validation in data spaces separates policy specifications from executable validation functions, using knowledge graphs to retain traceability [21].

The authors have used a labelled property graph derived from ArchiMate models. ArchiMate provides a standard language for representing EA elements and their relationships [22]. Elements become nodes, ArchiMate relationships become typed edges and repository properties remain available as node or edge properties. Cypher queries can then examine element types, follow relationships up to a specified number of steps and inspect recorded properties. For example, a structural query identifies a `WorkPackage` with no path to a `Goal` within this limit. A query using properties selects production systems and examines their ownership information.

Furthermore, this paper examines the transition from a requirement to a query over EA evidence and the interpretation of its results. It makes the required information, applicability assumptions and query limitations explicit, distinguishing possible EA issues from missing evidence and insufficient information for assessment.

3. Research approach and assessment process

The analysed material comprises 106 items: 26 software modernisation rules and 80 regulatory ICT requirements. The modernisation rules were gathered from literature addressing software modernisation, governance and EA debt [1, 3]. The regulatory ICT requirements were derived from provisions concerning ICT in NIS2, DORA and the AI Act [14, 15, 16]. These provisions concern, among other topics, governance, security, ICT assets, dependencies and operational resilience. They were reformulated as requirements relevant to EA analysis, retaining references to their respective legal acts and articles in the supplementary catalogue. The two groups serve a shared research purpose: evaluating what information EA models provide for requirements assessment. Throughout this paper, the term *rule* refers collectively to software modernisation rules and regulatory ICT requirements. The complete set and its recorded sources are available in the supplementary repository.

All 106 rules were classified according to the information required to operationalise them. Eleven query cases, selected because the model set provided sufficient information to formulate executable queries, were executed, comprising seven modernisation rules and four ICT requirement rules. Applying them to the same graph representation of 961 ArchiMate models makes it possible to examine how these two forms of evidence affect query behaviour. Findings, affected models and involved element types were recorded.

3.1. Information classification

The 106 rules were classified using five categories that distinguish the information required for operationalisation (Table 1). The labels describe the information needed for the selected operationalisation. Table 1 summarises the five categories and provides one representative example for each category.

The classification also demonstrates that graph structure alone is sufficient for only a small part of the analysed rules. Most of the rules refer to concepts such as approval, validation, rationale, business value, technical debt, readiness or testing outcomes. These concepts may be described in the model, but they are not consistently represented using standard ArchiMate element types or relationships. A semantic enrichment stage could potentially classify explicit evidence in element names and descriptions.

3.2. Application of queries derived from the rules

This section applies eleven queries to a heterogeneous set of ArchiMate models. Seven queries operationalise modernisation rules concerning traceability, strategic alignment, service interaction, ownership, documentation and reassessment. Four additional queries operationalise selected regulatory ICT requirements concerning data security metadata, asset inventories, dependencies of critical functions and dependencies on external ICT providers.

The analysis was conducted on the ArchiMate models of the EA ModelSet, a curated dataset of EA models collected from public and community sources, including GitHub, GenMyModel, and other

Table 1
Categories used to classify the rules. Category labels: S = graph structure only; P = graph structure and repository properties; L = semantic interpretation of textual information; P/L = repository properties and semantic interpretation; N = information not reliably available in the analysed repositories. Abbreviations: Cat. = category; EA = enterprise architecture; IP = Internet Protocol; FQDN = fully qualified domain name; CPU = central processing unit; prod = production; UX = user experience.

Cat.	Required information	Executed cases	Example rule and assessment scope
S	ArchiMate elements, relationships and graph topology only	2	*Standardised service interaction model.* Shared data coupling can be detected structurally by identifying two application components that access the same data object. No element properties or semantic enrichment are required.
P	Graph structure and properties found in the analysed model repositories	5	*Operational ownership of production systems.* Production systems can be identified by properties such as `IpAddress-prod`, `FQDN-prod` or `CPU-prod`, while the query identifies elements lacking the selected ownership properties.
L	Semantic tags derived from existing element names, descriptions or documentation	12	*Business case justification for modernisation.* The query would need to identify whether a related element represents an approved cost-benefit analysis and whether cost, risk and expected value are quantified. These meanings cannot be obtained from the graph structure alone.
P/L	Existing repository properties together with semantic enrichment	4	*End-of-life technology risk mitigation.* Lifecycle properties can identify obsolete technology or technology planned for withdrawal, but semantic analysis is generally required to determine whether a related element represents an approved mitigation, replacement or retirement plan.
N	Information not represented reliably in the EA repository	3	*UX outcome preservation or improvement.* Determining whether task success, user satisfaction or interaction efficiency has improved requires empirical UX measurements that cannot be reliably inferred from an ArchiMate model.

manually identified sources [20]. We extracted all available files in the `.archimate` format from the dataset and used these files as input for the analysis.

The analysed dataset consists of 961 ArchiMate models. It is suitable for evaluating whether the queries derived from requirements and rules can be executed over a heterogeneous set of EA models, but it should not be interpreted as a representative sample of regulated EA practice. The EA ModelSet is a mined and curated dataset. Thus, its models differ in size, language, modelling purpose, modelling style, level of abstraction and degree of completion. The dataset contains reference EAs, examples, templates, exercises and models from other sources. As a consequence, the results indicate which EA patterns and metadata gaps can be detected in such a heterogeneous corpus, but they do not provide direct evidence of compliance for real organisations.

The assessment specification must describe the full requirement, even if the query examines only some of the relevant evidence. For example, a query that uses only owner properties may overlook ownership recorded through responsibility relationships. Documenting this limitation helps the assessor distinguish between ownership information that is missing from the model and information that is present but not examined by the query. The process in Figure 1 makes such restrictions explicit before findings are interpreted. An example is Q05, *operational ownership of production systems*. The source rule concerns accountability for production systems, whereas the executable query can assess only the corresponding repository representation. First, properties such as `IpAddress-prod`, `FQDN-prod` or `CPU-prod` identify eligible production `Node` elements according to a repository-specific convention. Next, the query evaluates whether selected ownership properties are recorded for those elements. An absent property produces a metadata finding, not a determination that the organisation lacks an accountable owner. If the production selector is not used, the query cannot establish its intended scope. Q10, which concerns dependencies of critical functions, illustrates the converse problem: without reliable

criticality properties, a structural query must use broader `BusinessFunction`, `BusinessProcess` and `Capability` candidates. This makes the query executable but does not establish that every returned element represents a critical function.

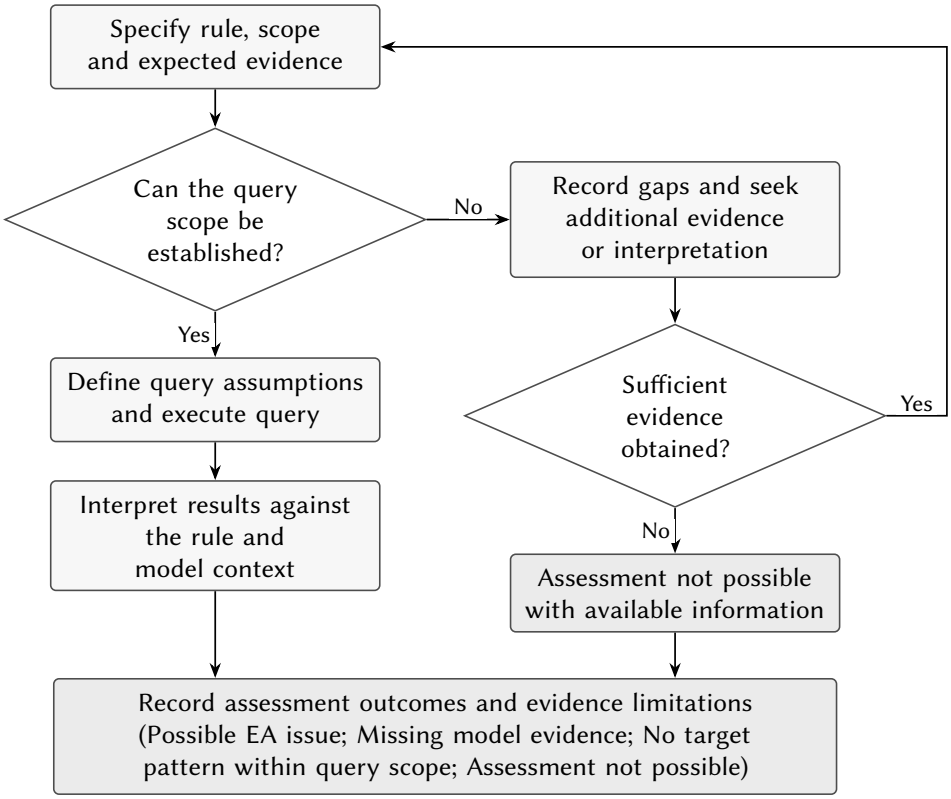


Figure 1: Process for requirements assessment through EA models

A further relevant characteristic of the dataset is its inconsistent use of properties. Only 110 of the 961 analysed models contain element properties, with 938 distinct property keys and 55,445 property occurrences. Property names are specific to individual models and multilingual. For example, ownership, criticality, lifecycle status, location and review information may be represented by different property keys, including keys in different languages. These properties may be present in only a few models or may be absent entirely. This affects the interpretation of all queries that use properties, since the absence of a finding may indicate that the queried property convention is not used in the model. Similarly, a finding may not represent a confirmed EA or regulatory deficiency.

Some queries use properties to identify the elements to assess, such as systems marked as production. If these properties are absent, the queries cannot identify the relevant elements. Other queries select elements by type or documentation and then identify missing properties, such as ownership metadata. These queries can therefore return findings even from models without element properties.

The models were parsed directly from XML. For each model, the parser extracted the model identifier, name, version and source file. For each ArchiMate element, it extracted the local identifier, name, ArchiMate type, documentation and properties. For each ArchiMate relationship, the parser extracted the relationship identifier, type, source and target identifiers and properties. Deprecated ArchiMate type names were normalised before import (e.g., `RealisationRelationship` to `RealizationRelationship` or `UsedByRelationship` to `ServingRelationship`). All 961 files were parsed successfully. The import produced 103,352 ArchiMate elements and 135,335 valid ArchiMate relationships. In addition, 566 dangling relationships were identified and excluded because their source or target element could not be resolved within the same model.

The parsed models were loaded into a local Neo4j database [23] as a labelled property graph, similar to the graph construction in [18]. Each model was represented as a `Model` root node with its elements

represented as an `ArchimateElement` and connected to the model by a `CONTAINS` relationship. Each element received an additional label corresponding to its ArchiMate type (e.g., `BusinessProcess`). Relationships were imported as typed Neo4j relationships, using their ArchiMate types, between the corresponding element nodes. Element and relationship properties were stored as Neo4j properties to support direct access from Cypher queries.

The queries derived from requirements and rules were implemented in Cypher [24] and executed once over the entire imported graph. The analysis was run using a local Python command-line tool that connects to the local Neo4j instance through the Neo4j driver. The analysis generated a CSV file with one row per finding. For each query, the recorded results include whether findings were detected, the number of findings, the number of affected models, the involved element types and representative findings. Given the dataset's characteristics and the query assumptions, the results should be treated as possible findings that require further interpretation.

The seven selected modernisation rules and four selected regulatory ICT requirement rules were implemented as deterministic Cypher queries. Structural queries use only ArchiMate element and relationship types, whereas queries that use properties additionally rely on element properties specific to each repository. The queries examine aspects of the original rules that are represented in EA models, but their results alone do not establish compliance. Their assumptions must be adapted to the property names and modelling conventions used in the assessed repository.

Rules requiring semantic judgement or external evidence were not operationalised in this execution. Examples include determining whether a business case is approved, whether user experience has improved after modernisation or whether a mitigation plan is adequate. Such information is not reliably represented through standard ArchiMate elements or properties.

4. Results

Table 2 summarises the execution results. Findings were returned for nine of the eleven queries. Their number ranged from 12 findings across 2 models to 33,203 findings across 828 models. These values must be interpreted in relation to the scope of each query and the information available in the underlying models. A high number of findings does not necessarily indicate many EA deficiencies, and a zero result does not demonstrate that a requirement has been fulfilled.

The results reveal three main patterns:

Structural queries are executable but semantically broad. The structural queries could be executed across a large part of the corpus. The strategic alignment query identified 778 *Work Package* elements across 94 models that lacked a relationship path to a *Goal*. The shared data query returned 900 findings across 75 models, corresponding to 450 unique pairs of *Application Component* elements that access the same *Data Object*. The query concerning dependencies of critical functions returned 13,574 findings in 443 models, while the query concerning dependencies on external providers returned 6,304 findings in 466 models.

These queries detect relevant structural patterns, but their interpretation remains broad. A *Work Package* is not necessarily a modernisation initiative, and a *Goal* is not necessarily an enterprise strategic objective. Similarly, shared access to a data object may represent a potentially problematic dependency, but it may also represent an intentional EA decision. The two dependency queries treat the scoped business and service elements as critical functions or services from external providers because the graph structure alone does not provide sufficiently precise information about applicability.

Queries dependent on properties have limited applicability. The business-to-application traceability and data security metadata queries returned no results because the properties used to select relevant elements were not associated with the expected ArchiMate element types. These results therefore indicate insufficient applicability information rather than the fulfilment of the underlying rules.

Table 2
Execution results for the eleven selected query cases. Q01–Q11 identify the query cases. Category labels: S = graph structure only; P = graph structure and repository properties. Abbreviations: Cat. = category; IT = information technology; ICT = information and communication technology.

Rule	Cat.	Findings	Models	Detected elements or patterns
Q01 Business capability to application service traceability	P	0	0	No eligible business elements were identified through the selected criticality properties.
Q02 Strategic alignment of modernisation scope	S	778	94	Work Packages without a relationship path to a Goal.
Q03 Standardised service interaction model	S	900	75	Pairs of Application Components accessing the same Data Object. The result contains 450 unique component pairs.
Q04 Dual accountability for modernisation initiatives	P	856	100	Work Packages without the selected business owner or IT owner properties.
Q05 Operational ownership of production systems	P	12	2	Nodes identified as production without the selected owner properties.
Q06 Continuous maintenance of knowledge artefacts	P	22,149	287	Documented elements without explicit review or update metadata.
Q07 Periodic modernisation reassessment cycle	P	12	2	Nodes identified as production without review or update metadata.
Q08 Data security metadata by classification level	P	0	0	No classified data or information assets were detected with incomplete selected security metadata.
Q09 ICT and information asset metadata completeness	P	33,203	828	ICT and information assets missing selected inventory metadata or a mapped supported business function.
Q10 Mapping dependencies of critical functions	S	13,574	443	Business Functions, Business Processes and Capabilities missing one or more required mapped dependency types.
Q11 Mapping dependencies on external ICT providers	S	6,304	466	Application Services and Technology Services missing one or more types of information about dependencies on external providers.

The operational ownership and periodic reassessment queries each returned the same twelve *Node* elements, which were identified as production, in two models. Their assessment scope is more explicit because production status is represented through repository properties. Their applicability is nevertheless limited because only a small number of models use the required convention for identifying production through properties. These nodes lacked the selected ownership metadata and the selected review or update metadata. The overlap therefore reflects different information gaps affecting the same elements.

Metadata completeness queries produce broad findings. The dual-accountability query returned all 856 *Work Package* elements across 100 models because the selected business owner and IT owner properties were not applied to them. The documentation maintenance query returned 22,149 documented elements in 287 models without the selected review or update metadata. The asset inventory query returned 33,203 findings in 828 models. Most of these assets lacked owner, location, classification, support or lifecycle information, and 27,954 also lacked a nearby mapping to a *Business Function*, *Business Process* or *Capability*.

These findings primarily indicate differences in repository metadata practices. Missing owner properties do not prove that no owner exists, and missing review dates do not prove that no review has taken place. The relevant information may be represented through other properties, relationships or

external systems.

5. Discussion

Regarding RQ1, the analysis shows that regulatory ICT requirement rules and modernisation rules can be operationalised when the required evidence is represented using ArchiMate elements, relationships, paths or repository properties. Structural queries are comparatively portable because they rely on the ArchiMate metamodel. However, they often require broad scope assumptions. Queries that use properties allow more specific applicability conditions, but they depend on property names, formats and element assignments that are specific to each repository.

The results support a distinction between three types of query outcomes. A query may identify a *possible EA issue*, such as shared access to a data object. It may identify *missing evidence*, such as missing ownership or lifecycle metadata. Finally, a rule may be *not assessable* because the information required to determine its applicability or fulfilment is absent. This distinction is necessary because neither the presence nor the absence of a query finding constitutes direct evidence of regulatory compliance.

Regarding RQ2, the main limitations concern the model's scope, modelling conventions and data completeness. The analysed repositories differ in language, level of abstraction, relationship usage and property vocabularies. Only 110 of the 961 models contained properties, and these models used 938 distinct property keys. Applying an unchanged query that uses properties across different repositories can therefore produce false negatives, false positives or findings that primarily reflect missing metadata. For example, it could distinguish between modernisation initiatives, strategic objectives, mitigation plans or test evidence. Semantic enrichment could also help identify concepts explicitly represented in element names, documentation or properties. Gutschmidt and Nast [25] assess perceived semantic model quality using a large language model and compare its outputs with expert assessments. Galli et al. [26] use language processing and knowledge graphs to extract and structure obligations from the AI Act. Such support requires separate validation at each stage (interpreting source requirements and identifying relevant model evidence) and must not assume approvals, assessments or controls that are not documented.

Threats to validity. To mitigate threats to validity, we applied consistent classification criteria to all 106 rules and documented the source rules and query definitions in the supplementary repository. The same parsing procedure, graph representation and deterministic Cypher execution approach were used across all 961 models to ensure consistent processing. Query assumptions and applicability constraints were documented to distinguish possible EA issues from missing model information. Nevertheless, rule interpretation and mapping to EA elements involve researcher judgement, and the absence of independently validated assessment results prevents the measurement of precision and recall. Furthermore, the heterogeneous publicly sourced models limit generalisation to organisational EA practice.

6. Conclusions and further work

This paper presented a process for assessing software modernisation and regulatory ICT rules through EA models. We classified 106 rules by their information needs and executed eleven selected queries on 961 ArchiMate models. Structural queries are comparatively reusable but often broad. Queries using properties can be more specific, but depend on metadata availability and consistency. Other rules require semantic interpretation, external evidence or human judgement.

The process makes query assumptions and evidence limitations explicit. Findings may indicate possible EA issues or missing model evidence, while insufficient information can prevent assessment. The absence of findings does not establish compliance.

Further work should validate the assessment process in organisational EA repositories with documented modelling conventions. Considering the distinction between verification, validation and evaluation of modelling methods [27], further work should examine query correctness, the suitability

of the assessment process and its practical usefulness. Experts could compare their assessments with query results to identify correct findings, misleading findings and missed deficiencies. Queries should also recognise different ways of recording the same evidence and indicate when there is not enough information to determine whether a rule applies.

The results of semantic enrichment should be compared with the information explicitly recorded in the model. Interpretations without clear support should be marked as uncertain. Linking EA models to asset inventories, vulnerability records and regulatory reporting data could extend assessment coverage. This is currently addressed by: 1) research on integrating software vulnerability data through enterprise knowledge graphs [28] and 2) research on ICT third party concentration risk using DORA reporting data [29].

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to review grammar and spelling and improve wording. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Slupczynski, B. Krauze, Software modernization: phase-specific ICT requirements, governance and resilience alignment, in: Proceedings of the 28th International Conference on Enterprise Information Systems (ICEIS 2026), volume 3, SciTePress, 2026, pp. 2182–2189. doi:10.5220/0014957400004018.

[2] A. Slupczynski, S. Hacks, Towards a knowledge base of terms on enterprise architecture debt, in: Enterprise Design, Operations, and Computing: EDOC 2023 Workshops, volume 498 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2024, pp. 194–210. doi:10.1007/978-3-031-54712-6_12.

[3] A. Slupczynski, B. Krauze, S. Hacks, Advancing enterprise architecture debt: a work system theory perspective on modernization and resilience, Complex Systems Informatics and Modeling Quarterly (2026) 67–96. doi:10.7250/csimq.2026-46.04.

[4] B. Krauze, An analysis of resilience in digital business ecosystems, in: Research Challenges in Information Science, volume 548 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2025, pp. 162–171. doi:10.1007/978-3-031-92471-2_13.

[5] M. Smajevic, S. Hacks, D. Bork, Using knowledge graphs to detect enterprise architecture smells, in: E. Serral, J. Stirna, J. Ralyté, J. Grabis (Eds.), The Practice of Enterprise Modeling, volume 432 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2021, pp. 48–63. doi:10.1007/978-3-030-91279-6_4.

[6] A. Cleven, R. Winter, Regulatory compliance in information systems research: literature analysis and research agenda, in: Enterprise, Business-Process and Information Systems Modeling, volume 29 of *Lecture Notes in Business Information Processing*, Springer, Berlin, Heidelberg, 2009, pp. 174–186. doi:10.1007/978-3-642-01862-6_15.

[7] S. Mishra, H. R. Weistroffer, A framework for integrating Sarbanes-Oxley compliance into the systems development process, Communications of the Association for Information Systems 20 (2007) 712–727. doi:10.17705/1CAIS.02044.

[8] P. N. Otto, A. I. Antón, Managing legal texts in requirements engineering, in: Design Requirements Engineering: A Ten-Year Perspective, volume 14 of *Lecture Notes in Business Information Processing*, Springer, Berlin, Heidelberg, 2009, pp. 374–393. doi:10.1007/978-3-540-92966-6_21.

[9] T. D. Breaux, A. I. Antón, Analyzing regulatory rules for privacy and security requirements, IEEE Transactions on Software Engineering 34 (2008) 5–20. doi:10.1109/TSE.2007.70746.

[10] A. Siena, G. Armellin, G. Mameli, J. Mylopoulos, A. Perini, A. Susi, Establishing regulatory compliance for information system requirements: an experience report from the health care

domain, in: Conceptual Modeling: ER 2010, volume 6412 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2010, pp. 90–103. doi:10.1007/978-3-642-16373-9_7.

[11] O. Akhigbe, D. Amyot, G. Richards, A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance, Requirements Engineering 24 (2019) 459–481. doi:10.1007/s00766-018-0294-1.

[12] O. Kosenkov, P. Elahidoost, T. Gorschek, J. Fischbach, D. Mendez, M. Unterkalmsteiner, D. Fucci, R. Mohanani, Systematic mapping study on requirements engineering for regulatory compliance of software systems, Information and Software Technology 178 (2025) 107622. doi:10.1016/j.infsof.2024.107622.

[13] O. Kosenkov, E. Zabardast, D. Fucci, D. Mendez, M. Unterkalmsteiner, Privacy by design: Aligning GDPR and software engineering specifications with a requirements engineering approach, Information and Software Technology 190 (2026) 107946. doi:10.1016/j.infsof.2025.107946.

[14] European Parliament and Council, Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS 2 directive), Official Journal of the European Union, L 333, pp. 80–152, 2022.

[15] European Parliament and Council, Regulation (EU) 2022/2554 on digital operational resilience for the financial sector, Official Journal of the European Union, L 333, pp. 1–79, 2022.

[16] European Parliament and Council, Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Official Journal of the European Union, L, 2024/1689, 2024. 12 July 2024.

[17] J. A. H. Schoonderbeek, H. A. Proper, Toward an ontology for EA modeling and EA model quality, Software and Systems Modeling 23 (2024) 535–558. doi:10.1007/s10270-023-01146-w.

[18] P.-L. Glaser, S. J. Ali, E. Sallinger, D. Bork, Model-based construction of enterprise architecture knowledge graphs, in: Enterprise Design, Operations, and Computing: EDOC 2022, volume 13585 of *Lecture Notes in Computer Science*, Springer, Cham, 2022, pp. 57–73. doi:10.1007/978-3-031-17604-3_4.

[19] M. Fumagalli, T. P. Sales, P. P. F. Barcelos, G. Micale, P.-L. Glaser, D. Bork, V. Zaytsev, D. Calvanese, G. Guizzardi, Mining frequent structures in conceptual models, Software and Systems Modeling 25 (2026) 549–577. doi:10.1007/s10270-025-01295-0.

[20] P.-L. Glaser, E. Sallinger, D. Bork, The extended EA ModelSet: a FAIR dataset for researching and reasoning enterprise architecture modeling practices, Software and Systems Modeling 25 (2026) 659–677. doi:10.1007/s10270-025-01278-1.

[21] A. Hmimou, P. Jovanovic, S. Nadal, O. Romero, A. Queralt, Operationalizing and automating data validation in data spaces, Data Science and Engineering 11 (2026) 303–323. doi:10.1007/s41019-025-00317-7.

[22] The Open Group, ArchiMate 4 specification, The Open Group, 2026. URL: https://pubs.opengroup.org/architecture/archimate4-doc/.

[23] Neo4j, Inc., The Neo4j operations manual, Neo4j, Inc., 2026. URL: https://neo4j.com/docs/operations-manual/current/.

[24] Neo4j, Inc., Neo4j Cypher manual, Neo4j, Inc., 2025. URL: https://neo4j.com/docs/cypher-manual/current/.

[25] A. Gutschmidt, B. Nast, Assessing model quality using large language models, in: E. Paja, J. Zdravkovic, E. Kavakli, J. Stirna (Eds.), The Practice of Enterprise Modeling, volume 538 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2025, pp. 105–122. doi:10.1007/978-3-031-77908-4_7.

[26] F. Galli, T. R. Dal Pont, G. Sartor, G. Contissa, Approaching the AI Act... with AI: LLMs and knowledge graphs to extract and analyse obligations, Computer Law and Security Review 60 (2026) 106230. doi:10.1016/j.clsr.2025.106230.

[27] J. Ralyté, G. Koutsopoulos, J. Stirna, Verification, validation, and evaluation of modeling methods: experiences and recommendations, Software and Systems Modeling 25 (2026) 1305–1320. doi:10.1007/s10270-025-01304-2.

[28] M. Egaña Aranguren, J. T. Fernández-Breis, B. Leon Balentzia, M. Rompe, A. García Castro,

A comprehensive view of software vulnerability risks through enterprise knowledge graphs, Computers and Security 163 (2026) 104815. doi:10.1016/j.cose.2025.104815.

[29] B. Krauze, L. Kruglova, Quantifying ICT third-party concentration risk using DORA reporting data, in: Research Challenges in Information Science. RCIS 2026 Forum and Doctoral Consortium, volume 597 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2027, pp. 78–89. doi:10.1007/978-3-032-35043-5_7.