Computing and Informatics, Vol. 45, 2026, 623–642, doi: 10.31577/cai_2026_3_623

BRIDGING THE GAP: A SYSTEMATIC REVIEW OF UML-BASED MODELING FOR NOSQL DATABASES

Mohammed elHabib Maicha

Laboratoire d'informatique et de mathematique
Université Amar Telidji, Laghouat
Algeria
e-mail: mh.maicha@lagh-univ.dz

Jaroslav Pokorný

Charles University, Prague
Czech Republic
&
Škoda Auto University, Mladá Boleslav
Czech Republic
e-mail: Jaroslav.Pokorny@matfyz.cuni.cz

Abstract. The increasing adoption of NoSQL databases to manage diverse and complex data structures necessitates robust modeling techniques. The Unified Modeling Language (UML), traditionally utilized for relational databases, has emerged as a promising tool for modeling NoSQL databases, despite their flexible and schema less nature. This paper presents a comprehensive exploration of UML's role in NoSQL data modeling through a systematic literature review (SLR). Key contributions include an extensive bibliometric analysis of UML applications across various types of NoSQL databases, as well as the identification of trends and gaps in current methodologies. The discussion highlights UML's strengths in the initial stages of conceptualizing data structures while addressing its limitations in representing the dynamic and distributed characteristics inherent to NoSQL systems. Novel insights are provided into the adaptation of UML for logical and physical modeling in document-oriented, graph-based, and columnar databases. The work concludes with actionable recommendations for enhancing UML-based methodologies and outlines future research directions to tackle unresolved challenges in this rapidly evolving field.

Keywords: Systematic literature review, Unified Modeling Language, NoSQL databases, data modeling, dynamic data structures

1 INTRODUCTION

In recent years, the surge in big data has driven the widespread adoption of NoSQL databases, which are renowned for their ability to manage vast volumes of unstructured and semi-structured data [1, 2].

Unlike traditional relational databases, NoSQL systems feature flexible and dynamic schemas tailored to diverse use cases, including document-oriented, graph-based, and key-value stores [3].

However, the absence of standardized modeling methodologies presents significant challenges for designing and maintaining NoSQL databases, which vary widely in structure and operation [4].

The Unified Modeling Language (UML), a mature and widely adopted modeling framework, offers a visual and systematic approach to representing data structures. Despite its extensive use in traditional database systems [5], the application of UML in NoSQL contexts remains underexplored and fraught with challenges. These challenges stem from the inherent schemaless nature of NoSQL databases, their distribution characteristics, and their support for polyglot persistence [6].

Existing literature highlights various attempts to apply UML to NoSQL databases; however, a comprehensive understanding of its efficacy, limitations, and potential remains lacking [7]. Most studies focus on specific types of databases or narrow aspects of UML's application, leaving critical gaps in evaluating UML's adaptability to modern database paradigms. Furthermore, there is little consensus on how UML can be extended to address the scalability, velocity, and variety dimensions of big data.

This study aims to bridge these gaps by presenting an exhaustive systematic literature review (SLR) on UML-based NoSQL data modeling. Our analysis examines the types of NoSQL databases explored in the literature, the representation levels used for modeling, and how existing methods address the four dimensions of big data: volume, velocity, variety, and veracity (4 Vs). This work not only consolidates existing knowledge but also provides a roadmap for future research by proposing innovative extensions to UML that meet the unique challenges posed by NoSQL systems.

2 POSITIONING OUR REVIEW WITHIN EXISTING SURVEYS AND STUDIES

The literature on big data modeling reveals a diverse range of approaches aimed at advancing methodologies and tools that facilitate the design of NoSQL databases [4, 8, 9].

Nevertheless, there is a paucity of publications that have explored this issue from the standpoint of meta-modeling. Furthermore, the majority of these studies present transformation solutions from traditional databases to specific NoSQL models tailored for particular case studies.

Table 1 presents academic works produced in the form of surveys or expert evaluations; however, none of them has undergone a systematic literature analysis on specific approaches or methods. While data modeling is mentioned in the work of Hecht and Jablonski [10], this concept refers to the structure, organization, and storage method of data in NoSQL systems, rather than the manner of modeling data using specific levels of representation and models. Similarly, by employing techniques such as ontology, semantics, RDF schemas, and the SPARQL language, Gil and Song [11] introduced the term "conceptual modeling" for big data. The strengths and weaknesses of various benchmark design approaches are discussed in this article.

Synthesis Study	Year	Context
[10]	2011	NoSQL databases
[11]	2016	Modeling and management of NoSQL databases
[3]	2018	NoSQL databases
[12]	2018	Methodology for designing NoSQL databases
[13]	2020	Database design methods in the era of big data
[14]	2020	Modeling and management of big data in databases
[7]	2021	Data modeling and NoSQL databases

Table 1. Surveys and reviews on data modeling in NoSQL databases

Two academic works published in 2018 [3, 12] provide an overview of four non-orthogonal modeling bases for distributed database systems: data partitioning, data models, the CAP theorem, and the consistency model. Once again, the data model in these works is simply an understanding of how data is organized, structured, and stored.

It was not until 2020 that two works [13, 14] conducted a survey specifically focused on the data model of NoSQL databases, resulting in significant findings despite the absence of a systematic analysis.

In [7], the authors concentrated on the methodologies used in various studies. They cataloged related works aiming to unify most NoSQL databases under a uniform design methodology before proposing a methodological classification.

2.1 What We Specifically Drew from Selected Prior Works

To be explicit about how prior works influenced our SLR and taxonomy:

- **Atzeni et al. [4]**: We used the taxonomical discussion in Atzeni et al. to inform our classification of logical-level alternatives (document vs column vs graph) and to build part of the ‘target model’ coding categories. Atzeni et al. analyze modeling trade-offs (partitioning, consistency) but do not provide a reproducible SLR focused on UML-level representations – which justifies our narrower systematic focus.

- **McConnell et al.** [8]: This patent documents common denormalization and storage optimizations used in industry. We used it to cross-check the practical heuristics reported in academic studies and to motivate the extraction field ‘heuristic rules’ in our data-extraction form.
- **Mior et al.** [9]: NoSE is an important example of a schema-design approach for NoSQL applications. We summarize its approach (query-driven clustering of records) and used it as a comparative baseline when judging whether a study offers fully-automated transformations versus heuristics/manual procedures. NoSE informed our discussion of evaluation strategies and empirical baselines.

Terminology: RDF, ontologies and conceptual modeling. Briefly, an *RDF Schema* (RDFS) is a lightweight semantic meta-model describing classes and properties for RDF data; ontologies extend RDFS with richer axioms. We use the general term *conceptual modeling* to denote any representation intended to capture domain concepts and relationships independently of physical storage. When earlier surveys cite RDF/ontologies as conceptual-modeling techniques, we explicitly mark that linkage in our coding.

3 SPECIFIC DESIGN METHODS FOR NOSQL DATABASES BASED ON UML

In this section, we outline the research questions and the analysis protocol for conducting a systematic review. The guidelines proposed by Kitchenham [15] were employed to conduct this systematic literature review (SLR), which is a method for identifying, evaluating, and interpreting all relevant research related to a specific topic. Our three-phase SLR study is detailed in Figure 1.

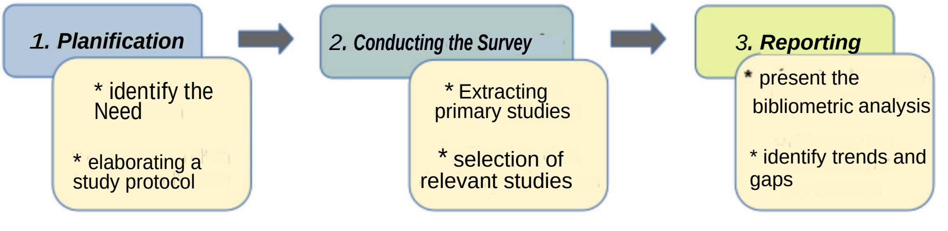


Figure 1. Phases of a systematic literature review (planning, conducting, reporting)

In the initial phase, our activities focused on confirming the presence of other systematic literature review studies related to the subject of interest. Simultaneously, we crafted the review protocol to address potential researcher biases and defined the study's objectives, justifications, and research questions. This phase also entailed a meticulous delineation of the strategic approach.

Moving to the subsequent phase, our efforts concentrated on systematically collecting primary studies from predetermined sources identified during the planning stage. We then applied stringent inclusion criteria to retain only those studies directly relevant to the domain of big data modeling based on UML.

In the final phase, our attention turned to a comprehensive analytical exploration. This stage sought to provide nuanced insights into the specific models utilized at each level of representation pertinent to the modeling of NoSQL databases.

Each phase is expounded upon in detail in the following sections.

4 PLANNING THE SYSTEMATIC LITERATURE REVIEW STUDY

The primary objectives of this phase are to determine the necessity of conducting an SLR study and to develop a review protocol.

Need for an SLR Study: The critical importance of conducting a modeling phase prior to developing applications based on big data is widely acknowledged in academic and expert circles. Esteemed scholars and industry experts have consistently emphasized the pivotal role of a thorough modeling process as an essential precursor to the development of effective applications within the domain of big data [16, 17].

Nevertheless, there are currently no widely embraced data modeling methodologies within the economic sector that have gained acceptance from the scientific community for NoSQL systems. Despite the inherent heterogeneity, an indeterminate number of representation levels, and the absence of widely adopted models at each level, novel proposals for data modeling in NoSQL databases continue to surface. Our primary objective is to delve into this domain, conducting comprehensive research, and subsequently offering a systematic literature review on endeavors utilizing established norms or standards in the modeling of NoSQL databases.

The SLR Protocol: Prior to commencing the systematic literature review, it is imperative to establish a research protocol to prevent potential conflicts of interest among researchers [15].

Consequently, our protocol is structured as follows: Firstly, we have delineated specific developmental objectives along with supporting arguments to explicate our contributions. Subsequently, in an effort to distill the existing evidence on big data modeling using UML, we have formulated five research questions.

Objectives and justifications: In a comprehensive literature analysis, the primary objective is to present information on the most relevant research in the field of NoSQL database modeling. Based on these findings, the secondary objective is to identify various approaches to big data modeling used in different studies to determine trends and gaps related to the three key concepts of big data: the source model, the target model, and the 4 Vs. The systematic literature review conducted within the scope of this research has compiled almost all studies on NoSQL database modeling based on UML into a single document, benefiting both industry and the academic world.

Research questions: The aim of this literature review is to explore how different models applied in traditional databases (especially UML) are employed in NoSQL databases. To achieve this, five research questions (RQs) have been defined:

RQ1. What types of NoSQL databases are addressed by researchers?
RQ2. What levels of representation are presented in methodologies for modeling NoSQL databases?
RQ3. What considerations are taken into account for the 4 Vs with these methods?
RQ4. How have these methods been evaluated?
RQ5. What are the trends and gaps in the modeling and management of big data?

5 LITERATURE REVIEW PROCESS

To compile an exhaustive list of studies on the topic of NoSQL database modeling, we followed a systematic literature review process involving several key steps: identifying primary studies, selecting potentially relevant studies, applying inclusion criteria, and extracting and synthesizing data.

5.1 Identification of Primary Studies: Search Strings and Sources

We used a set of structured queries (Boolean combinations of the core keywords) across multiple sources. The principal search string was:

```
("NoSQL" OR "NoSQL databases" OR "No-SQL") AND
("UML" OR "Unified Modeling Language") AND
("data model" OR "data modeling" OR "schema")
```

Searches were run on the following sources on **20 October 2025**:

- Google Scholar,
- Semantic Scholar,
- Connected Papers (manual),
- **Scopus** (institutional access),
- **Web of Science** (institutional access).

The raw harvest returned a total of **2 045** records (Google Scholar + Semantic Scholar + Connected Papers: 1 680; Scopus: 220; Web of Science: 145). After deduplication (removing 365 duplicates across sources), **1 680** unique records remained for screening.

5.2 Screening and Selection

Title and abstract screening reduced the set to **150** potentially relevant papers. We then retrieved full-texts for the 150 candidates. Full-text assessment eliminated papers that did not match our inclusion criteria (studies that lack a UML-based modeling contribution, pure surveys without UML focus, non-English full-texts where translation was not possible, or non-accessible artefacts). The full-text assessment resulted in **50** full-texts assessed and **18** final included primary studies. The selection procedure is summarized in Figure 2.

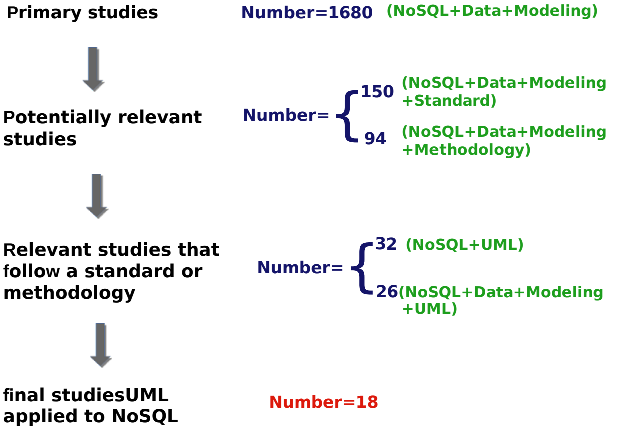


Figure 2. Selection procedure (identification, screening, eligibility, inclusion). Numbers: 2 045 identified; 1 680 after deduplication; 150 screened; 50 full-texts assessed; 18 included.

5.3 Inclusion and Exclusion Criteria

We applied the following inclusion criteria:

- Studies presenting an explicit use of UML (or UML-derived constructs) for modeling NoSQL databases (document, graph, columnar, key-value).
- Studies presenting a method, approach, tool, or concrete transformation from conceptual/logical models to NoSQL logical/physical schemas.
- Peer-reviewed articles, conference papers, and journals in English.

Exclusion criteria included non-UML-centered works, purely conceptual position papers without modeling artefacts, and publications that describe NoSQL systems without any modeling contribution.

5.4 Data Extraction and Synthesis

For each included study we extracted: bibliographic metadata, target NoSQL type, UML constructs used, representation levels (conceptual, logical, physical), whether the study addresses 4 Vs, type of validation/evaluation, and availability of artefacts. The extraction was performed independently by the two authors and cross-checked for consistency.

6 TERMINOLOGY CLARIFICATION

Throughout this manuscript we use the following terminology deliberately:

Methodology: an overarching framework or systematic approach (for example, a model-driven engineering (MDE) workflow that includes modeling, transformations, and code generation).

Method: a concrete technique or procedure used inside a methodology (for example, a transformation rule that maps UML associations to nested document structures).

This distinction is used when we code and categorize primary studies (some papers propose a methodology; others only describe individual mapping methods).

7 ANALYSIS RESULTS

In this section, we address the research questions through several key activities:

Bibliometric Analysis: We gather information on authors and publication details, including the impact of selected studies (e.g., H-index and Scientific Journal Rankings (SJR)), publication years, and the journals and conferences where the studies were published.

Comprehensive Literature Review: We categorize studies based on three key concepts: inputs, outputs, and characteristics of big data in a conceptual table. For input sources, we analyze the source model, UML constructs, and data types. For target outputs, we scrutinize data abstraction levels and proposed data models at the conceptual, logical, and physical levels for the targeted NoSQL system. Additionally, we assess support for each of the 4 Vs characteristics of big data.

Discussion of Trends and Gaps: We identify trends and gaps in the field of big data modeling and management.

7.1 Bibliometric Analysis

The objective of this analysis is to consolidate research efforts on NoSQL database modeling through bibliometric scrutiny. This provides a comprehensive overview of authorship and publication data, enabling readers to understand the development of studies in this area over time. Table 2 illustrates journal relevance through H-index and SJR metrics. Key findings from this analysis include:

- The number of studies on NoSQL database modeling based on UML has steadily increased over the years.
- Prior to 2013, no relevant studies were identified.
- The year 2018 saw the highest number of studies on big data modeling.

Table 3 presents the distribution of primary studies by the number of authors and by country. The 18 selected academic works were authored by 54 researchers from 22 public and private research institutions across 15 countries.

7.2 Systematic Literature Review Results (RQ1–RQ3)

This section addresses research questions *RQ1*, *RQ2*, and *RQ3* as outlined in the SLR planning section.

> **RQ1.** What types of NoSQL databases are addressed by researchers?

Our analysis revealed two distinct categories: studies that do not target any specific database ("All") and those tailored to a specific type ("OneType"). Figure 3 summarizes distribution. Document-oriented databases attracted the most attention (9 studies). Four studies focused on graph databases, two on columnar stores, and none specifically targeted key-value stores in our final corpus.

> **RQ2.** What levels of representation are presented in NoSQL database modeling methodologies?

We classified representation levels as conceptual, logical, and physical. Document oriented studies predominantly target the logical level; several works provide conceptual and physical mappings, but only three studies explicitly span all three levels.

> **RQ3.** What considerations are taken into account for the 4 Vs with these methods?

To answer this question, we define non-functional requirements as meta properties such as scalability and consistency that can impact database design. Our analysis indicates that only six studies addressed non-functional requirements related to the 4 Vs. Table 5 shows the level of support for different criteria among the studied methods. Volume was the most frequently addressed concept, with over

Paper	Year	Journal/Proceeding	SJR 2020	H-Index
[18]	2016	International Conference on Conceptual Modeling (LNCS)	0.249	400
[19]	2018	International Conference on Model and Data Engineering (LNCS)	0.249	400
[9]	2017	IEEE TKDE	1.36	174
[4]	2020	Computer Standards & Interfaces	0.56	63
[20]	2014	Frontiers in AI and Applications	0.16	54
[21]	2018	BPM / Conference Proceedings	0.21	49
[22]	2017	Int. Journal of Applied Engineering Research	–	40
[23]	2013	IEEE Big Data Conf.	–	26
[24]	2021	Int. Journal of Data Warehousing & Mining	0.16	23
[25]	2016	Int. Journal of Web Information Systems	0.19	18
[26]	2018	International Journal of Adv. Computer Science	0.19	18
[27]	2020	DEXA Conf.	0.17	15
[28]	2015	IEEE Smart City	–	10
[29]	2014	WARTIA Workshop	–	10
[30]	2015	Information Integration	–	8
[31]	2018	WEBIST Proceedings	0.12	3
[32]	2018	Innovate-Data Proceedings	0.12	2
[33]	2022	Computing and Informatics	0.21	29

Table 2. Journals and conferences where relevant studies were presented

Works	Number of Authors	Countries
[18]	3	France, Spain
[19]	5	Spain
[9]	3	Canada, Qatar
[20]	4	Italy
[20]	2	Germany
[21]	4	Israel
[22]	3	South Korea
[23]	2	India
[24]	3	France
[25]	2	Brazil
[26]	6	Malaysia, Nigeria, Spain
[27]	3	France, Colombia
[28]	4	China
[29]	3	China
[30]	2	Brazil
[31]	3	Spain
[32]	2	Malaysia
[33]	3	Algeria

Table 3. Distribution of primary studies by the number of authors and by country

Database Type	Conceptual Level	Logical Level	Physical Level
Document-Oriented	✓	✓	✓
Graph-Based	✓	✓	
Columnar	✓		✓
Key-Value			

Table 4. Comparative table for RQ2: Levels of representation by database type

four studies incorporating it into their design methods; for instance, [24] and [33] discussed volume alongside other Vs of big data. This combined version integrates key elements from both sections while ensuring clarity and coherence throughout.

7.3 Evaluation Types and Trends (RQ4–RQ5)

RQ4. How have these methods been evaluated?

We identified three evaluation types across the corpus:

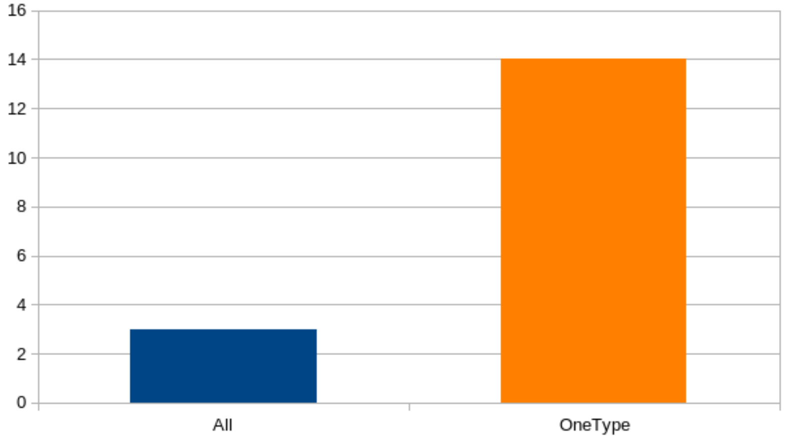
Experimentation: empirical experiments on workloads/benchmarks;

Illustrative examples/case studies: small or domain-specific worked examples;

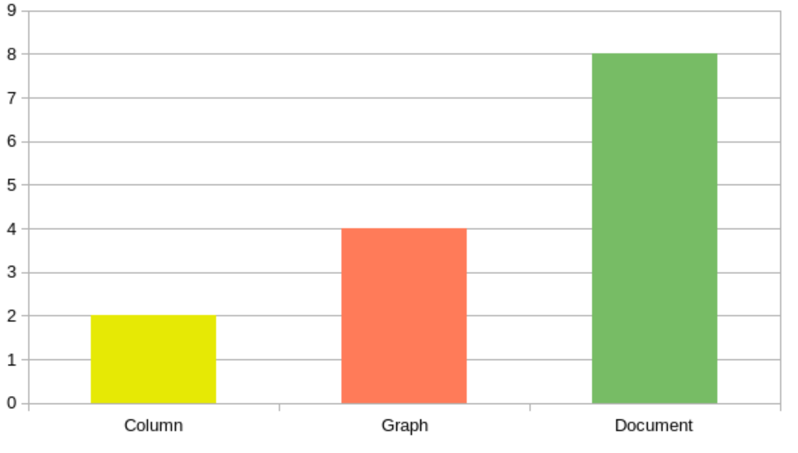
Comparison: comparisons with baselines such as NoSE or handcrafted schema.

Articles	Year	Inputs (e.g., UML blocks)	Outputs (NoSQL systems)	Big Data Characteristics
[23]	2013	Model-oriented – ER	Document-oriented and graph databases	–
[20]	2014	UML Notation	Graph Database	Value
[29]	2014	UML Class Diagram	Platform-independent model	–
[30]	2015	Extended ER, UML Notation	Document-oriented	Consistency
[28]	2015	UML Class Diagram	Document-oriented	Variability
[18]	2016	Meta-model of UML Class Diagram	Column-oriented	–
[25]	2016	Extended ER	Document-oriented	Volume and Value
[9]	2017	Conceptual Schema and Statistics	Column-store	Volume and Velocity
[22]	2017	UML Data Model – Peter Chen	Document-oriented	–
[31]	2018	Entity-Relationship Model	Column–oriented	–
[32]	2018	ER Model	Document-oriented	–
[21]	2018	UML Class Diagram	Graph-type Database	–
[19]	2018	Generic Data Model	Document and Column-oriented Database	–
[26]	2018	UML Notation	Document-oriented	Consistency
[27]	2020	UML Class Diagram and OCL Constraints	Graph Database	–
[4]	2020	Aggregated Data View	System-independent Data Model	Scalability and Consistency
[24]	2021	UML Class Diagram	Column, Document, and Graph-type Database	Volume, Variety, Velocity
[33]	2022	UML Use case and Class Diagrams	Document-oriented	The 4 V's

Table 5. A comparative study of UML-based NoSQL modeling approaches (full table)



a) Specific to the Database



b) NoSQL Type

Figure 3. a) Number of works per database treatment; b) Number of works per NoSQL type

As Figure 4 shows, most methods have only illustrative validations or no evaluation at all.

> **RQ5.** What are the trends and gaps in the modeling and management of big data?

From the corpus we distilled the following recurring observations and gaps:

- The majority of studies present results at conceptual, logical, and physical abstraction levels, but only a minority provide end-to-end transformations.

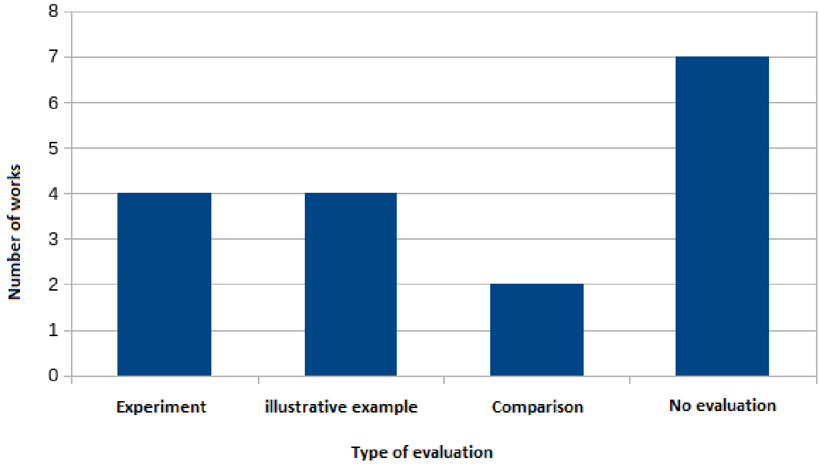


Figure 4. Number of studies by type of evaluation (experiment, illustrative example, comparison, none)

- UML class diagrams and ER models dominate conceptual modeling; document-oriented logical schemas are the most studied logical outputs.
- Heuristics and manual guidelines are common; few proposals provide fully automated transformation pipelines.
- Key-value stores are underrepresented.
- Non-functional aspects (velocity, real-time ingestion) are infrequently addressed.

Critique and practical implications. We discuss limitations of existing methods (notably the lack of standardized notation for eventual consistency and sharding) and recommend that tool developers incorporate automation and support for polyglot persistence. We also emphasize the need for real-world validations.

8 CONCLUSIONS AND CONTRIBUTIONS

In this paper, we addressed five research questions through a systematic literature review, contributing to a clearer understanding of UML-based big data modeling. Our findings can assist both researchers and practitioners in enhancing their big data projects by providing valuable insights into existing methodologies.

The results reveal several critical insights: notably, over 70 % of studies do not validate their proposals with real-world datasets, highlighting a significant gap in empirical support. Additionally, the "velocity" characteristic of big data is often inadequately addressed. At the conceptual level, the Entity-Relationship model is the most commonly utilized, while at the logical level, document-oriented models dominate the research landscape. Furthermore, we identified a concerning lack of evaluations for proposed methods.

Our prior contribution and its relevance. As one of the included works (UML4NoSQL [32]) demonstrates, it is possible to propose a systematic transformation approach from UML class diagrams to document-oriented schemas. UML4NoSQL offers a mapping framework and a prototype implementation for document stores; however, as our SLR shows, broader empirical validation across datasets and NoSQL types remains necessary. We briefly summarize UML4NoSQL's core ideas:

1. annotate conceptual UML with stereotypes and constraints that capture denormalization choices;
2. apply deterministic mapping rules to produce families of logical schemas;
3. offer guidance to select a schema based on query patterns and 4 V considerations.

Future work. We recommend:

1. extending UML notations to better express non-functional concerns (e.g., stereotypes for sharding/consistency);
2. developing semi-automated tools for mapping and validation;
3. validating methods on large-scale, real-world datasets; and
4. addressing key-value modeling in UML-based approaches.

Acknowledgements

We thank the anonymous reviewers for their constructive suggestions, which helped improve the clarity and reproducibility of this manuscript.

REFERENCES

[1] Gandomi, A.—Haider, M.: Beyond the Hype: Big Data Concepts, Methods, and Analytics. International Journal of Information Management, Vol. 35, 2015, No. 2, pp. 137–144, doi: 10.1016/j.ijinfomgt.2014.10.007.

[2] Krommyda, M. K.—Kantere, V.: The Big Data Era: Data Management Novelties for Visualizing, Exploring, and Processing Big Data. Analyzing Future Applications of AI, Sensors, and Robotics in Society, IGI Global, 2021, pp. 87–103, doi: 10.4018/978-1-7998-3499-1.ch006.

[3] Davoudian, A.—Chen, L.—Liu, M.: A Survey on NoSQL Stores. ACM Computing Surveys (CSUR), Vol. 51, 2018, No. 2, Art. No. 40, doi: 10.1145/3158661.

[4] Atzeni, P.—Bugiotti, F.—Cabibbo, L.—Torlone, R.: Data Modeling in the NoSQL World. Computer Standards & Interfaces, Vol. 67, 2020, Art. No. 103149, doi: 10.1016/j.csi.2016.10.003.

[5] Roques, P.—Vallée, F.: UML 2 en Action: De l'Analyse des Besoins à la Conception J2EE. Éditions Eyrolles, 2004, https://www.editions-eyrolles.com/livre/uml-2-en-action (in French).

[6] Sharma, A.—Singh, G.—Rehman, S.: A Review of Big Data Challenges and Preserving Privacy in Big Data. In: Kolhe, M. L., Tiwari, S., Trivedi, M. C., Mishra, K. K. (Eds.): Advances in Data and Information Sciences (ICDIS 2019). Springer, Singapore, Lecture Notes in Networks and Systems, Vol. 94, 2020, pp. 57–65, doi: 10.1007/978-981-15-0694-9_7.

[7] Vera-Olivera, H.—Guo, R.—Huacarpuma, R. C.—Da Silva, A. P. B.—Mariano, A. M.—Holanda, M.: Data Modeling and NoSQL Databases – A Systematic Mapping Review. ACM Computing Surveys (CSUR), Vol. 54, 2021, No. 6, Art. No. 116, doi: 10.1145/3457608.

[8] McConnell, C. C.—Liu, W.—Shayandeh, S.—Goodwin, R. L.: Efficient Denormalization of Data Instances. Microsoft Technology Licensing, LLC, 2020, https://patents.google.com/patent/US20200151156A1/en (US Patent App. 16/740,081).

[9] Mior, M. J.—Salem, K.—Aboulnaga, A.—Liu, R.: NoSE: Schema Design for NoSQL Applications. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, 2017, No. 10, pp. 2275–2289, doi: 10.1109/TKDE.2017.2722412.

[10] Hecht, R.—Jablonski, S.: NoSQL Evaluation: A Use Case Oriented Survey. 2011 International Conference on Cloud and Service Computing, IEEE, 2011, pp. 336–341, doi: 10.1109/CSC.2011.6138544.

[11] Gil, D.—Song, I. Y.: Modeling and Management of Big Data: Challenges and Opportunities. Future Generation Computer Systems, Vol. 63, 2016, pp. 96–99, doi: 10.1016/j.future.2015.07.019.

[12] Asaad, C.—Baïna, K.: NoSQL Databases – Seek for a Design Methodology. In: Abdelwahed, E. H., Bellatreche, L., Golfarelli, M., Méry, D., Ordonez, C. (Eds.): Model and Data Engineering (MEDI 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11163, 2018, pp. 25–40, doi: 10.1007/978-3-030-00856-7_2.

[13] Roy-Hubara, N.—Sturm, A.: Design Methods for the New Database Era: A Systematic Literature Review. Software and Systems Modeling, Vol. 19, 2020, No. 2, pp. 297–312, doi: 10.1007/s10270-019-00739-8.

[14] Martinez-Mosquera, D.—Navarrete, R.—Lujan-Mora, S.: Modeling and Management Big Data in Databases – A Systematic Literature Review. Sustainability, Vol. 12, 2020, No. 2, Art. No. 634, doi: 10.3390/su12020634.

[15] Kitchenham, B.: Procedures for Performing Systematic Reviews. Technical Report No. TR/SE-0401, Keele University, Keele, UK, 2004, http://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf.

[16] Sivarajah, U.—Kamal, M. M.—Irani, Z.—Weerakkody, V.: Critical Analysis of Big Data Challenges and Analytical Methods. Journal of Business Research, Vol. 70, 2017, pp. 263–286, doi: 10.1016/j.jbusres.2016.08.001.

[17] Ohlhorst, F. J.: Big Data Analytics: Turning Big Data into Big Money. Wiley, 2012, doi: 10.1002/9781119205005.

[18] Daniel, G.—Sunyé, G.—Cabot, J.: UMLtoGraphDB: Mapping Conceptual

Schemas to Graph Databases. In: Comyn-Wattiau, I., Tanaka, K., Song, I. Y., Yamamoto, S., Saeki, M. (Eds.): Conceptual Modeling (ER 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9974, 2016, pp. 430–444, doi: 10.1007/978-3-319-46397-1_33.

[19] de la Vega, A.—García-Saiz, D.—Blanco, C.—Zorrilla, M.—Sánchez, P.: Mortadelo: Automatic Generation of NoSQL Stores from Platform-Independent Data Models. Future Generation Computer Systems, Vol. 105, 2020, pp. 455–474, doi: 10.1016/j.future.2019.11.032.

[20] Sedlmeier, M.—Gogolla, M.: Design and Prototypical Implementation of an Integrated Graph-Based Conceptual Data Model. In: Thalheim, B., Jaakkola, H., Kiyoki, Y., Yoshida, N. (Eds.): Information Modelling and Knowledge Bases XXVI. IOS Press, Frontiers in Artificial Intelligence and Applications, Vol. 272, 2014, pp. 376–395, doi: 10.3233/978-1-61499-472-5-376.

[21] Roy-Hubara, N.—Rokach, L.—Shapira, B.—Shoval, P.: Evaluation of a Design Method for Graph Database. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (Eds.): Enterprise, Business-Process and Information Systems Modeling (BPMDS 2018, EMMSAD 2018). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 318, 2018, pp. 291–303, doi: 10.1007/978-3-319-91704-7_19.

[22] Shin, K.—Hwang, C.—Jung, H.: NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen's Framework. International Journal of Applied Engineering Research, Vol. 12, 2017, No. 5, pp. 632–636, `https://www.ripublication.com/Volume/ijaerv12n5.htm`.

[23] Kaur, K.—Rani, R.: Modeling and Querying Data in NoSQL Databases. 2013 IEEE International Conference on Big Data, 2013, pp. 1–7, doi: 10.1109/BigData.2013.6691765.

[24] Abdelhedi, F.—Ait Brahim, A.—Zurfluh, G.: OCL Constraints Checking on NoSQL Systems Through an MDA-Based Approach. International Journal of Data Warehousing and Mining (IJDWM), Vol. 17, 2021, No. 1, pp. 1–14, doi: 10.4018/IJDWM.2021010101.

[25] Lima, C.—Mello, R. S.: On Proposing and Evaluating a NoSQL Document Database Logical Approach. International Journal of Web Information Systems, Vol. 12, 2016, No. 4, pp. 398–417, doi: 10.1108/IJWIS-04-2016-0018.

[26] Imam, A. A.—Basri, S.—Ahmad, R.—Watada, J.—González-Aparicio, M. T.—Almomani, M. A.: Data Modeling Guidelines for NoSQL Document-Store Databases. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, 2018, No. 10, doi: 10.14569/IJACSA.2018.091066.

[27] Gómez, P.—Casallas, R.—Roncancio, C.: Automatic Schema Generation for Document-Oriented Systems. In: Hartmann, S., Küng, J., Kotsis, G., Tjoa, A. M., Khalil, I. (Eds.): Database and Expert Systems Applications (DEXA 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12391, 2020, pp. 152–163, doi: 10.1007/978-3-030-59003-1_10.

[28] Feng, W.—Gu, P.—Zhang, C.—Zhou, K.: Transforming UML Class Diagram into Cassandra Data Model with Annotations. 2015 IEEE International Conference on Smart City (SmartCity), 2015, pp. 798–805, doi: 10.1109/SmartCity.2015.165.

[29] Li, X.—Ma, Z.—Chen, H.: QODM: A Query-Oriented Data Modeling Approach for NoSQL Databases. 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), 2014, pp. 338–345, doi: 10.1109/WARTIA.2014.6976265.

[30] de Lima, C.—dos Santos Mello, R.: A Workload-Driven Logical Design Approach for NoSQL Document Databases. Proceedings of the 17th International Conference on Information Integration and Web-Based Applications & Services (iiWAS '15), 2015, doi: 10.1145/2837185.2837218.

[31] Suárez-Otero, P.—Suárez-Cabal, M. J.—Tuya, J.: Leveraging Conceptual Data Models for Keeping Cassandra Database Integrity. Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018), 2018, pp. 398–403, doi: 10.5220/0007236303980403.

[32] Hamouda, S.—Zainol, Z.: Document-Oriented Data Schema for Relational Database Migration to NoSQL. 2017 International Conference on Big Data Innovations and Applications (Innovate-Data), IEEE, 2017, pp. 43–50, doi: 10.1109/Innovate-Data.2017.13.

[33] Maicha, M. E.—Ouinten, Y.—Ziani, B.: UML4NoSQL: A Novel Approach for Modeling NoSQL Document-Oriented Databases Based on UML. Computing and Informatics, Vol. 41, 2022, No. 3, pp. 813–833, doi: 10.31577/cai_2022_3_813.

A INCLUDED STUDIES (TABLE AND SUPPLEMENTARY CSV)

A list of the included studies (title, authors, year, venue, DOI/URL, NoSQL target type, UML constructs used, representation levels, evaluation type) has been prepared and included as supplementary material. Table 6 shows the list of included references (bibliographic key + short metadata).

B SIMPLE QUALITY ASSESSMENT (QA)

To provide a sense of the methodological strength of included studies we applied a compact quality-assessment checklist. For each study we scored the following items (2 = yes/strong; 1 = partial; 0 = no/absent):

1. QA1 – Clear statement of objectives and research questions.
2. QA2 – Sufficient methodological description to reproduce the approach.
3. QA3 – Presence of empirical evaluation (experiments or real datasets).
4. QA4 – Availability of artefacts or worked examples (models, scripts).

We then computed a total quality score (0–8) and assigned a label (High: 7–8, Moderate: 4–6, Low: 0–3). Table 7 summarizes the QA scores.

Ref key	Short metadata (title, year, target NoSQL type)
[18]	UmlToGraphDB: mapping conceptual schemas to graph DBs (2016) – graph
[19]	Mortadelo: Automatic generation of NoSQL stores from PIM (2020) – document/column
[9]	NoSE: Schema design for NoSQL applications (2017) – column/store-focused
[20]	Integrated graph-based conceptual data model (2014) – graph
[21]	NoSQL design using UML conceptual model (2018) – graph
[22]	Modeling and querying data in NoSQL databases (2013) – document
[23]	(2013) Model-oriented - ER to document/graph
[24]	Document database logical approach (2016) – document
[25]	Data modeling guidelines for document stores (2018) – document
[26]	Automatic schema generation for document systems (2020) – document
[27]	Transforming UML class diagram into Cassandra data model (2015) – column (Cassandra)
[28]	QODM: Query-oriented data modeling (2014) – document
[29]	Workload-driven logical design for document DBs (2015) – document
[30]	Leveraging conceptual data models for Cassandra (2015) – column
[31]	ER-based approaches (2018) – column/document
[32]	UML4NoSQL: UML to document-oriented DBs (2022) – document
[4]	Data modeling in the NoSQL world (2020) – survey, cross-cutting
[33]	Comparative analysis / recent synthesis (2022) – document/4 Vs

Table 6. Included primary studies (bibliographic key and short metadata)

Ref	QA1	QA2	QA3	QA4	Label
[18]	2	1	0	1	Moderate
[19]	2	1	1	1	Moderate
[9]	2	2	2	1	High
[20]	1	1	0	1	Low
[21]	1	1	0	1	Low
[22]	1	1	0	1	Low
[23]	1	1	0	1	Low
[24]	2	1	0	1	Moderate
[25]	1	1	0	1	Low
[26]	1	1	0	1	Low
[27]	2	2	1	1	Moderate
[28]	1	1	0	1	Low
[29]	1	1	1	1	Moderate
[30]	1	1	0	1	Low
[31]	1	1	0	1	Low
[32]	2	2	1	2	High
[4]	2	1	0	1	Moderate
[33]	2	1	0	1	Moderate

Table 7. Simple quality assessment of included studies (summary)

Mohammed elHabib Maicha received his Ph.D. degree in computer science from the University of Laghouat, Algeria, in 2015. He is Assistant Professor in the Department of Computer Science at Amar Telidji University Laghouat and a member of the LIM laboratory. He is responsible for the Master's program in data science and artificial intelligence. He also serves as a reviewer for IEEE journals and conferences. His research interests include databases, conceptual modeling, Big Data, and NoSQL.

Jaroslav Pokorný received his Ph.D. degree in theoretical cybernetics from Charles University, Prague, Czechoslovakia, in 1984. He is Full Professor in the Faculty of Mathematics and Physics, Charles University in Prague. He has published more than 350 papers and books on data modeling, relational databases, query languages, XML technologies, and data organization. His current research interests include semi-structured data, Web technologies, database architectures, Big Data, and social networks. He is involved as a chair and co-chair in the organization of conferences, e.g. ADBIS-DASFAA, EDB, ISD, IDEAS, ADBIS, etc. He is a member of ACM and IEEE. He works also as the representative of Czech Republic in IFIP. His current research interests include databases, conceptual and data modeling.