information

MDPI

Review

Foundations for a Generic Ontology for Visualization: A Comprehensive Survey

Suzana Loshkovska [1,*,†] and Panče Panov [2,*,†]

[1] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, 1000 Skopje, North Macedonia
[2] Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia
* Correspondence: suzana.loshkovska@finki.ukim.mk (S.L.); pance.panov@ijs.si (P.P.); Tel.: +389-70-390-842 (S.L.)
† These authors contributed equally to this work.

Abstract

This paper surveys existing ontologies for visualization, which formally define and organize knowledge about visualization concepts, techniques, and tools. Although visualization is a mature field, the rapid growth of data complexity makes semantically rich frameworks increasingly essential for building intelligent and automated visualization systems. Current ontologies remain fragmented, heterogeneous, and inconsistent in terminology and modeling strategies, limiting their coverage and adoption. We present a systematic analysis of representative ontologies, highlighting shared themes and, most importantly, the gaps that hinder unification. These gaps provide the foundations for developing a comprehensive, generic ontology of visualization, aimed at unifying core concepts and supporting reuse across research and practice.

Keywords: visualization; ontology; semantic modeling; visualization types; domain gaps; domain challenges

check for updates

Academic Editors: Domenico Fabio Savo, Gianluca Cima and Riccardo Rosati

Received: 12 August 2025
Revised: 1 October 2025
Accepted: 14 October 2025
Published: 18 October 2025

Citation: Loshkovska, S.; Panov, P. Foundations for a Generic Ontology for Visualization: A Comprehensive Survey. *Information* **2025**, *16*, 915. https://doi.org/10.3390/info16100915

1. Introduction

From Data Proliferation to Visualization. The current trend of digitalization in almost all life domains results in a vast amount of data. As the volume and complexity of data continue to increase, we face the challenge of efficiently representing data, as organizing data into tables, proved to be inefficient. In this context, visualization has become a helpful tool, focusing on mapping information to a graphical representation and limiting the amount of information presented to the user while telling them a story behind the information.

Knowledge Organization Systems as Foundations. Knowledge organization systems (KOS)—including taxonomies, thesauri, ontologies, and classification schemes—are structured vocabularies and models that organize concepts and their relationships to support consistent description, discovery, and reasoning across a domain [1]. The effectiveness of data visualization heavily depends on the underlying knowledge organization frameworks. These frameworks provide a structured method for understanding the complex process of creating and interpreting data visualizations. They define the steps in the visualization pipeline and the types of knowledge required to perform the transformation from the initial raw data to the final visual representation and its interpretation by the viewer.

Foundational knowledge organization models such as Shneiderman's *Task-by-Data-Type taxonomy* [2], Chi's *Data State Reference Model* [3], and Munzner's *nested model of visualization design and validation* [4] highlight the role of structured abstractions in bridging data, tasks, and views. Subsequent taxonomies and high-level frameworks [5,6] reinforced the importance of systematically organizing visualization knowledge to guide both human analysts and automated systems. Such a structured approach is essential not only for designing engaging and interpretable visualizations, but also for developing intelligent visualization systems that can support automation and adaptation.

Ontologies as the Semantic Backbone of Visualization. Ontologies, defined by Gruber as "a set of representational primitives with which to model a domain of knowledge or discourse" [7], serve as a formal map of a domain, identifying and naming the core concepts, specifying their types, and describing how they relate, thereby allowing software to reason automatically over that structure. Applied to the domain of visualization, they enable us to tag charts, marks, and interactions with explicit meaning, driving systems that can choose or adapt views and facilitating knowledge-driven visual analytics. As data becomes increasingly varied and complex, a shared semantic layer is essential for building adaptive visual tools and for reusing and combining visualization techniques across platforms.

Challenges: Standardization and Interoperability. The necessity of greater knowledge formalization within the field was introduced by Brodlie et al. [8], where they highlight the limitations of doing visualization without a formal knowledge framework. According to them, the ontology will provide a formal, machine-readable vocabulary for visualization concepts and enable automated discovery, selection, and composition of components or services. They also stress the need for a focus shift from applying ontologies to the visualized data to applying ontologies to the visualization process, its components, and services.

Creating a data-visualization ontology is challenging because the field spans many disciplines and lacks a common vocabulary. Practitioners often describe the same chart types, encodings, or tasks with different names, units, and data formats, leading to duplicated concepts and conflicting concept hierarchies [5,6]. Without standardization, it is hard to assess which definitions are authoritative, to merge descriptions from multiple studies, or to align new classes with existing knowledge graphs [3,4]. The challenge is amplified by visualization's interdisciplinary reach—computer science, statistics, medicine, education, finance, and more—each domain bringing its own specific terminology and perspective [9,10]. Unifying these perspectives and disambiguating overloaded terms is needed to achieve true interoperability; otherwise, mappings to pre-existing ontologies remain partial, and even the most polished visualizations fail to convey trustworthy information.

Furthermore, a variety of chart types, interaction styles, and analytic workflows exist, many of which overlap or combine in hybrid dashboards. As new frameworks (e.g., streaming libraries such as Vega-Lite [11]) emerge, the taxonomy of "what counts" continues to evolve; thus, many ontologies risk obsolescence without ongoing maintenance and updating. Deciding how to organize this ever-growing visualization catalogue—by data type, visual encoding, user task, or some other dimension—inevitably leaves edge cases and gray areas that do not fit neatly into a single category [12,13].

Finally, many crucial qualities of a visualization, such as clarity, aesthetic appeal, and task-specific "effectiveness", are subjective and context-dependent. These soft attributes do not translate cleanly into rigid classes or properties, yet omitting them would ignore what practitioners care about most. Ontology engineers therefore face a granularity dilemma: to model high-level ideas (easy to agree on but too coarse to use for reasoning tasks) or fine-grained details (more expressive but more complex to maintain and standardize). Balancing stability with adaptability, objectivity with subjectivity, and completeness with usability makes the quest for a generic, enduring visualization ontology uniquely challenging [14,15].

Prior Work and Open Gaps. Early efforts to design an ontology for visualization recognized the need to provide a common vocabulary and formalize knowledge of visualization. Duke et al. [16] propose the first roadmap for a shared visualization ontology. From a workshop of UK visualization researchers, they distilled a skeleton ontology with four top-level concept groups—task and use, representation, process, and data—each examined at conceptual, logical, and physical levels. The publication outlines these early categories, highlights the challenges in gaining community consensus, and positions the ontology as a foundation for service discovery, workflow composition, and education in visualization science.

Shu et al. [17] presented another ontology for visualization, aiming to add semantic descriptors to visualization services. Their ontology described components of visualization systems to improve the discovery and reuse of visualization services. This was followed by the *Top-Level Visualization Ontology (TLVO)* [18] introduced in 2004. Although the original TLVO does not appear to have an actively maintained repository or recent updates, it has given rise to various follow-up ontologies. For example, the same authors proposed an *Enhanced Visualization Ontology* that extended TLVO to represent the visualization pipeline and data models more effectively [18]. Despite the improvements, the ontology lacks concepts to support 3D visualization, evaluation metrics, and context of use.

To address coverage and accessibility, Polowinski et al. developed *Visualization Ontology (VISO)*, a comprehensive and modular ontology that formalizes knowledge of visualization for machine use [19]. The VISO-ontology has been made openly available, and subsequent research has extended the ontology in multiple directions. The *Visual Analytics Ontology for Machine Learning (VIS4ML)* builds directly on VISO's modular design principles, expanding them to capture concepts related to machine learning workflows, including models, hyperparameters, evaluation metrics, and explainability [20]. *SemViz* [21], on the other hand, explores the semantic enrichment of visualization pipelines by integrating them with domain ontologies.

In terms of theoretical development, pattern-based approaches such as the one proposed by Asprino et al. [22], formalize the process of visualization construction using ontological design patterns. These approaches, though developed independently, reinforce the conceptual contributions made by VISO in the direction of modular and reusable visualization knowledge models.

Another example in the accessibility domain is the family of multi-level visualization ontologies developed under the *OntoVis* framework. Within this stack, the *Upper Visualization Ontology (UVO)* serves as the top-level module, providing abstract classes for visual entities, data types, and tasks [23]. UVO underpins accessibility-oriented knowledge bases that describe how diagrams, charts, and other visual artefacts are composed, enabling non-visual interaction and alternative renderings. Follow-up work in 2021 refined the axioms and introduced new use cases such as natural-language access to statistical charts [24]. However, no public code repository has been released, and—apart from the 2021 update—there is no indication of active maintenance. Thus, OntoVis and its UVO module should be regarded as a research prototype rather than a continuously curated ontology.

While numerous papers present work on developing ontologies for different aspects of the visualization process, such as pipelines, tasks, techniques, and workflows, there is no recent comprehensive survey paper that systematically reviews, compares, and classifies these ontologies. No established classification framework or taxonomy is universally applied to compare these ontologies. No single review article provides an exhaustive survey of both general-purpose ontologies and significant, actively maintained visualization ontologies/taxonomies, with a detailed analysis of both their internal structure and real-world applications/adoption.

Our Contributions. The paper focuses on the paradigm of ontology for visualization, where knowledge structures are employed to enhance, guide, or automate the creation and interpretation of visual representations. The purpose of this article is to systematically identify, categorize, and analyze existing ontologies for visualization, examining their scope, underlying conceptualizations, formalisms, and target applications. We highlight common themes, recurrent concepts, and varying levels of granularity across these ontologies. Additionally, the paper identifies significant gaps and challenges in the current landscape, including issues of coverage, extensibility, and practical adoption.

Paper Organization. The structure of this paper is organized as follows. The next section describes the methods used and outlines the steps involved. In Section 3, we provide a detailed description and a comprehensive evaluation of selected ontologies. Furthermore, in Section 4, we summarize our findings, highlight limitations, and discuss discovered design patterns and challenges when designing a generic ontology for visualization. The final section concludes the paper with a summary and points for future developments.

2. Materials and Methods

2.1. Research Questions

To guide our investigation, we formulated a set of research questions addressing the scope, structure, and relevance of existing visualization ontologies. These questions aim to identify the key resources, distinguish general-purpose ontologies, uncover shared conceptual elements, and assess which ontology most closely aligns with a generic, reusable framework. The research questions are as follows:

- **RQ1** What are the existing ontologies that can be considered general-purpose for visualization?
- **RQ2** What common design patterns and modeling strategies emerge from existing ontologies, and how do they support interoperability and reuse?
- **RQ3** Which parts of the visualization process are well covered in current ontologies, and where are the most significant gaps?
- **RQ4** How widely have existing visualization ontologies been adopted, as measured by the availability of public artefacts and citation counts of their primary publications?
- **RQ5** How do current visualization ontologies align with Semantic Web standards and external domain ontologies, and what benefits does this bring?
- **RQ6** What technical and conceptual limitations hinder the broader adoption and applicability of existing visualization ontologies?

2.2. Search Strategy

The goal of compiling a comprehensive inventory of ontologies that support visualization—rather than merely visualizing ontologies themselves—dictated a two-phase search strategy.

2.2.1. Phase 1: Initial Web Search

We began with exploratory Web searches using simple keyword strings such as "visualization ontology", "ontology for visualization", and "ontology for visual analytics". This initial crawl was largely unproductive: most hits referred to *ontology-visualization* tools (graphical aids for building or debugging ontologies) rather than ontologies that model visualization knowledge. For example, a Google search for "ontology for visualization" returned about 8000 results, while the IEEE Digital Library (https://ieeexplore.ieee.org/Xplore/home.jsp, accessed on 5 August 2025) reported nearly 15,000. From these, we exported 1000 candidate entries and used ChatGPT (o3 mode) (https://openai.com/, accessed on 5 August 2025), the o3 model used was released on

25 April 2025) to assist in screening, which produced 16 suggestions that were later excluded after domain verification. Although the raw outputs of this preliminary phase were discarded, the process provided valuable insights that informed our subsequent strategy, including refining queries, selecting reliable sources, and narrowing the time window for the final search.

2.2.2. Phase 2: Structured Literature Search

To obtain a more reliable corpus of articles and digital resources, we turned to curated data sources and custom-made queries. Searches were executed against:

- **Digital Libraries and Indexes:** IEEE Xplore, ACM Digital Library (https://dl.acm.org/, accessed on 5 August 2025), Scopus (https://www.elsevier.com/products/scopus, accessed on 5 August 2025), Web of Science (https://www.webofscience.com/wos/author/author-search, accessed on 5 August 2025), DBLP (https://dblp.org/, accessed on 5 August 2025), Google Scholar (https://scholar.google.com/, accessed on 5 August 2025), ScienceDirect (Elsevier) (https://www.sciencedirect.com/, accessed on 5 August 2025), SpringerLink (https://link.springer.com/, accessed on 5 August 2025), and MDPI (https://www.mdpi.com/, accessed on 5 August 2025);
- **Open Ontology Repositories:** Linked Open Vocabularies (LOV) (https://lov.linkeddata.es/, accessed on 5 August 2025), Ontobee (https://ontobee.org/, accessed on 5 August 2025), and various GitHub/GitLab (https://github.com/, accessed on 5 August 2025; https://about.gitlab.com/, accessed on 5 August 2025) public projects supplied machine-readable artefacts in most popular knowledge organization formats, such as Ontology Web Language (OWL) (https://www.w3.org/TR/owl-features/, accessed on 5 August 2025), Resource Description Framework (RDF) (https://www.w3.org/TR/rdf11-concepts/, accessed on 5 August 2025), and Simple Knowledge Organization System (SKOS) (https://www.w3.org/TR/skos-reference/, accessed on 5 August 2025).
- **Gray Literature and Preprints:** arXiv (https://arxiv.org/, accessed on 5 August 2025), HAL (https://hal.science/?lang=en, accessed on 5 August 2025), and Zenodo (https://zenodo.org/, accessed on 5 August 2025) were screened to capture early or niche ontology proposals not yet presented in a journal publication.

To complement database queries, we leveraged various large-language-model assistants, including ChatGPT (o3 mode), Gemini, Undermind, and Perplexity, iteratively prompting them with exact phrases to surface gray-literature references and published preprints that conventional digital libraries and indices had not yet cataloged.

Initially, we conducted this search using the previously mentioned three principal search expressions: *ontologies for visualization*, *visual analytics ontologies*, and *semantic visualization* in the title and abstract of articles. Considering that visual analytics and semantic visualization have produced numerous hits that involve domain-specific visualizations, where visualizations are primarily used to represent results from domain-specific ontology searches, we revised our approach to creating queries. We applied Boolean operators to make queries by combining controlled vocabulary (e.g., “visualization” OR “infoVis”) with ontology terms (“ontology”, “knowledge graph”, “vocabulary”), and whenever possible, we asked full-text search in complete articles, which produced more accurate results.

Considering that the survey is focused on all ontologies for visualization, we do not restrict the publication year.

2.3. Study Selection and Study Quality Assessment

In line with systematic review practice, we loosely adopted PRISMA terminology [25] to structure the process. After identification of possible candidates and removing duplicates, the results of the literature search were manually screened for relevance against the inclusion criterion of modeling knowledge that facilitates the creation, selection, or evaluation of visual representations of data. Furthermore, we examined the papers' references and cited URLs to identify supplementary material, online resources, or ontology repositories. For each ontology, we considered the publication period, the availability of artefacts, and implementation evidence. We did not conduct a formal quality assessment of the papers for two reasons: (i) lack of an established procedure for evaluating visualization-ontology publications; and (ii) our primary focus was on the ontologies themselves, since many analysis steps relied on artefacts or external resources beyond the articles.

2.4. Data Inclusion and Exclusion Criteria

The data for this analysis were manually extracted from publications and repositories that provide ontology descriptions and/or implementations. An initial pool of 21 resources was identified through keyword-based filtering, as summarized in Table A1. From this pool, we selected four ontologies for in-depth evaluation, based on inclusion and exclusion criteria designed to prioritize *domain-agnostic* and *practically reusable resources*. These include the Visualization Ontology (VISO) [19], the Visualization Knowledge Ontology (VisKo) [26], the Visual Analytics-Assisted Machine Learning Ontology (VIS4ML) [20], and the Audiovisual Analytics Vocabulary and Ontology (AAVO) [27].

Our selection emphasized *ontologies that capture general-purpose visualization constructs*—such as charts, workflows, encodings, and annotations—rather than those focused on specific application domains (e.g., molecular biology or geophysics). To ensure reusability, we required that selected ontologies provide at least one *publicly accessible artefact*, preferably in standard knowledge representation formats such as OWL (https://www.w3.org/OWL/, accessed on 5 August 2025) or RDF (https://www.w3.org/RDF/, accessed on 5 August 2025). We further prioritized *resources with demonstrable community visibility*, as indicated by references across multiple publications or repositories.

Ontologies with a narrow application scope, such as SBOL-VO (https://sbolstandard.org/sbol-visual-ontology/, accessed on 5 August 2025) for synthetic biology or GeoDataOnt [28] for geospatial data, were excluded. We also *omitted draft implementations available only as GitHub prototypes without peer-reviewed documentation*, since these do not meet the criteria for ontology resources.

Upon closer inspection of the available materials, we added three additional ontologies, despite their partial compliance with the initial requirements. OntoVis (UVO + VDO + DDO + VTO) [23] was included due to its modular architecture covering multiple aspects of visualization and its publicly accessible website. Automatic Visualization of Semantic Data (SemViz) [21] and the Industrial-Grade Visualization Ontology (VisuOnto) [29] were also considered, as both extend beyond domain-specific use cases and can be regarded as general-purpose visualization ontologies. Table A2 summarizes the adapted PRISMA selection protocol. A detailed rationale for the inclusion of the selected seven ontologies is provided in Table 1.

Table 1. General-purpose visualization ontologies retained after applying the inclusion criteria.

Ontology	Why It Satisfies the Prerogatives
VISO	Generic backbone for marks, encodings, data, tasks, and interaction; OWL on GitHub; reused by VizBoard and several recommendation prototypes; introduced at [19].
VisKo	Three public OWL/RDF ontologies (View, Operator, Service) capture generic pipelines and rendering services; reused in SciVis workflow research; UTEP tech-report [26].

Table 1. *Cont.*

Ontology	Why It Satisfies the Prerogatives
VIS4ML	Although centered on ML, its classes (Data, Task, View, Interaction, Workflow-Stage) are generic; OWL on GitLab; widely cited in VA-for-ML literature [20].
SemViz	Early (2007) generic vocabulary and 2D/3D viewers; RDF + demo code still online; referenced in linked-data visualization surveys [21].
AAVO	Formal OWL + SKOS linking data-mining tasks, visual encodings and sonification; artefacts on arXiv/GitHub; cited in audiovisual analytics research [27].
OntoVis	Upper-level OWL-2 vocabulary of graphic objects, layers, and spaces aimed at accessible diagrams; artefacts in CEUR-WS vol. 2859; referenced in accessibility and SVG-semantics research [23].
VisuOnto	Industrial yet domain-agnostic ontology of charts, workflow steps, and constraints; OWL published (SemIIM and IJCKG 2022); reused in Bosch manufacturing knowledge graphs [29].

2.5. Rationale for Surveying a Sparsely Documented Domain

Although the available body of work on evaluated ontologies is notably scarce—often limited to one or two publications per ontology with few maintained resources—this very scarcity serves as a compelling justification for conducting a structured survey.

First, the absence of a consolidated overview hinders scholarly progress. By mapping even a limited landscape, this survey establishes a verifiable baseline for the field. It exposes inconsistencies, identifies gaps in coverage, and makes missing resources explicit, thus providing a much-needed overview of the current state of development.

Second, the survey serves a future-oriented function: it supports reuse, encourages reinvigoration of abandoned efforts, and identifies promising ontologies for extension or integration. Many visualization ontologies were developed in isolated research contexts and have not benefited from sustained community support. Even though some reviewed publications are older, our goal is to extract stable, core ideas and lay the basic pillars for the development of a generic ontology for visualization.

Finally, we demonstrate a methodology for assessing underdocumented ontologies—one that combines literature review, semantic inspection, and resource verification.

2.6. Evaluation Protocol

Formal evaluation of ontologies typically involves assessing multiple quality dimensions, such as consistency, completeness, accuracy, modularity, reuse potential, and human interpretability. Established methodologies such as METHONTOLOGY [30], surveys of evaluation techniques [31], and quality frameworks like OQuaRE [32,33] and FOCA [34,35] emphasize verification of these dimensions through direct inspection of ontology artefacts (e.g., OWL/RDF files), reasoning tests for logical soundness, and application-based validation. While OWL files are available for several of the surveyed ontologies (e.g., VISO [19], VisKo [26], VIS4ML [20], AAVO [27]), supporting materials such as schema documentation, evaluation protocols, and worked examples are often scarce or fragmented. In some cases, legacy links have become inaccessible, further limiting reproducibility. As a result, only a subset of the quality dimensions can be meaningfully verified through artefact inspection.

Given these constraints, we adopted a pragmatic, literature-driven strategy. Our goal was not to perform a full quality evaluation in the sense defined by these frameworks, but rather to identify recurring gaps and patterns that could inform the design of a future generic ontology for visualization. To achieve this, we moved beyond the narrow scope of artefact-based testing and instead drew evidence from published descriptions, figures, and case studies, aligning them with OWL resources where available. This comparative focus on domain scope, primary application areas, pipeline stages, data types, visualization techniques, and user-related coverage allows us to highlight what existing ontologies capture well and, more importantly, what they leave out. In doing so, the assessment remains both practical under current resource limitations and directly relevant to the longer-term goal of developing a generic ontology of visualization.

3. Results

In this section, we present the results of our analysis for the seven selected ontologies. The section addresses the research questions from the previous part. We provide a detailed examination of the chosen ontologies, highlighting key features for comparison in the visualization domain.

3.1. Detailed Analysis of Selected Ontologies

3.1.1. VISO: The Visualization Ontology

Purpose. Introduced through a CHI'13 poster and earlier reports, VISO serves as the semantic backbone of the ontology-driven visualization approach (OGVIC). In this framework, VISO provides the vocabulary for abstract graphics, RVL specifies declarative data–visual mappings, and the Abstract Visual Model (AVM) captures the platform-independent graphic structure [19,36–38]. Beyond OGVIC, VISO was used as a knowledge base for context-aware component recommendation in the VizBoard prototype [39].

Core Concepts. The ontology adopts a modular design with four main components: *VISO/graphic*, *VISO/data*, *VISO/facts*, and a lighter *VISO/activity* module. This reflects insights from RVL, which distinguished attributes, relations, and mappings [38], and was extended by Voigt et al. (2013) with *user*, *system*, and *domain* modules for recommendation tasks [39]. In practice, *VISO/graphic* defines objects and representations (e.g., *Graphic Representation*, *Bar Chart*, *Legend*); *VISO/data* captures variables, roles, and scales; *VISO/facts* encodes constraints and effectiveness knowledge; and *VISO/activity* introduces interaction types (e.g., *Semantic Zoom*). A small *shape* add-on enumerates basic forms (*Circle*, *Triangle*, *Arrow*, *Star*).

Workflow Support. The *ACTIVITY* and *FACTS* modules model tasks, actions, and empirical constraints, supporting classification and suitability knowledge for semi-automatic design. While explicit task-ordering or pipeline axioms are absent, Voigt et al. (2013) demonstrated a semantics-based workflow in VizBoard (data upload, pre-selection, mapping, recommendation) where VISO provided the backbone [39].

Mappings (Data ↔ Visual). Mappings are expressed through *VISO/facts* (e.g., `appropriate_to_visualize`) that link data concepts (*VISO/data*) to attributes (*VISO/graphic*). RVL extends this with declarative constructs (*PropertyMapping*, *ValueMapping*) [38], and ranking algorithms integrate expert rules with empirical effectiveness data [39]. Mackinlay-style rankings [14] are intended to populate this knowledge base.

Technical Implementation. The active distribution resides on GitHub (https://github.com/viso-ontology/viso-ontology, accessed on 5 August 2025), The repository includes a top-level ontology importing *data*, *graphic*, and *facts*, with diagrams, RVL code-generation addenda, and CC BY-SA 4.0 licensing. Earlier entry points (`purl.org/viso`) are no longer maintained. VizBoard demonstrates integration with repositories and rating services for recommending components [39].

Axiomatization and Metrics. In the current repository snapshot (core modules: *graphic*, *data*, *facts*, *activity*, *main*, plus small codegen add-ons), VISO comprises 1003 triples, 93 classes, 15 object properties, 27 data-type properties, and 50 individuals. Module highlights are: *graphic* (569 triples; 46 classes; 41 individuals), *data* (215 triples; 41 classes;

2 individuals), *facts* (155 triples; 1 class; 3 individuals), and *activity* (25 triples; 4 classes). The repository also contains *anno* and *bibliography* bundles that add a large number of annotation triples and bibliographic individuals; these are excluded here for cross-ontology comparability. Public papers report few schema-level metrics, focusing instead on recommendation accuracy in VizBoard [39].

3.1.2. VisKo: Visualization Knowledge Ontologies

Purpose. VisKo is an ontology-driven framework for automating visualization pipelines. It encodes knowledge about views and toolkit operators, using inference to select and chain them so that scientists (e.g., in geoscience) can obtain visualizations without manual glue code [26,40]. It replaces narrative manuals with a machine-interpretable knowledge base from which pipelines can be synthesized and executed.

Core Concepts. VisKo is structured into three OWL modules: *VisKo-View* (graphic views such as isosurfaces, volumes, node–link networks), *VisKo-Operator* (abstract operators such as transformers, viewers, converters, filters, with defined input/output types and formats), and *VisKo-Service* (binding operators to concrete Web services and toolkits) [26,40]. The OWL packages (https://github.com/orgs/openvisko/repositories, accessed on 5 August 2025) instantiate operators (e.g., *Transformer*, *Viewer*, *Converter*, *Interpolator*) and link them to services in scientific toolkits, including GMT (`grdimage`), NCL (`gsn_csm_xy2_time_series`), VTK (`vtkContourFilter3D`, `vtkVolume`), and auxiliary converters (`pdf2png`, `fits2png`).

Workflow Support. Users issue a high-level *visualization query* specifying input format/type, desired view, and target viewer. VisKo searches the ontology to infer an operator path that transforms the input into a viewer-accepted format and includes a mapper for the requested view [26]. Concrete operator-path and instance files (e.g., `jsonGraph_OperatorPaths.owl`) demonstrate multi-step pipelines for force-directed graphs, bar charts, and data transformations.

Mappings (Data ↔ Visual). Mapping semantics are distributed across the *View* and *Operator* modules. Operators use the relation `mapsTo` (17 assertions) to link transformations with target view goals, covering 2D plots (*ContourMap*, *PointMap*, *TimeSeriesPlot*), projections (*SpherizedRaster*), and 3D visualizations (*SurfacePlot*, *IsoSurfaceRendering*, *VolumeRendering*), as well as bar charts. A limitation noted in the publications is that low-level view attributes (opacity, color, resolution) are not tied to corresponding operator parameters, leaving mappings underspecified.

Technical Implementation. The active distribution resides in the *ViSko* GitHub repository (https://github.com/openvisko/visko, accessed on 5 August 2025). Knowledge is encoded in RDF/OWL, with execution described using OWL-S (`Service`, `Process`, `Profile`) (https://www.w3.org/submissions/OWL-S/, accessed on 15 September 2025) and extended in *VisKo-Service* via relations such as `implementsOperator`, `supportedByToolkit`, and `supportedByOWLSService`. The repository also provides the Java-based API/server (Axis dependencies, build scripts), per-service OWL descriptors (e.g., `PDFToPNG.owl`, `vtkContourFilter3D.owl`), dataset profiles, and parameter bindings declared as OWL-S inputs/outputs to enable automatic invocation. A supporting wiki (https://github.com/nicholasdelrio/visko/wiki, accessed on 5 August 2025) explains server deployment and usage.

Axiomatization and Metrics. The OWL packages contain mainly instantiated ABox (Assertional Box represents the facts/instances, which are assertions about individuals in an ontology.) content, totaling ∼4562 triples. They define 61 operators (27 transformers, 17 viewers, 5 converters, 5 viewer sets, 3 reducers, 3 interpolators, 1 filter), 39 services (mirrored in OWL-S), and 9 toolkits (*vtk*, *gmt*, *ncl*, *ImageJ*, etc.). The ontologies support 22 input and 10 output formats (e.g., CSV, JSON, NETCDF, FITS, DICOM, PDF, PNG, XML) and 18 input and 14 output data types (e.g., CF *Variable_with_Time/LatLon*, VTK *vtkImageData3D*, *vtkPolyData*). Parameters are extensively modeled with 158 OWL-S `Input` and 39 `Output` instances, plus 266 declared parameter bindings and three demonstration endpoints. While publications describe TBox (Terminological Box represents the classes, properties, and axioms about them.) patterns for views, operators, formats, and services, they do not report schema-level metrics (axioms, Description Logic–DL expressivity). Reasoning is implicit, using property chains (`hasInput/hasOutput`) to recover valid operator paths [26,40]. All counts should be regarded as snapshots rather than stable release statistics.

3.1.3. VIS4ML: The Visual Analytics-Assisted Machine Learning Ontology

Purpose. VIS4ML was proposed as an ontology for *visual analytics-assisted machine learning* (VA-assisted ML), describing how interactive visualization supports the ML lifecycle [20]. Its goal is to standardize concepts and relations that characterize VA workflows in ML, highlight gaps where VA can replace ad hoc code, and guide human–AI teaming. VIS4ML thus provides both a shared conceptual foundation and a machine-readable schema for organizing prior work and informing tool design [20].

Core Concepts. The ontology defines two upper classes: *Process* and *IO-Entity* [20]. *Process* captures steps in ML workflows and distinguishes automated versus human-centered variants. *IO-Entity* represents artefacts consumed or produced by processes, grouped into *Data*, *Model*, and *Knowledge*. Directed relations link processes to their inputs, outputs, and successors, while decomposition links capture complex workflows. A companion glossary clarifies terminology and notation [41].

Workflow Support. VIS4ML aligns its classes with four canonical ML phases: *Prepare-Data*, *Prepare-Learning*, *Model-Learning*, and *Evaluate-Model* [20]. Examples include visual inspection during data preparation, feature/parameter design in learning preparation, iterative training with active human steering, and interpretability analyses in evaluation. Supplementary artefacts provide four modeled workflows (ActiVis, SOMFlow, decision-tree construction, TensorFlow graph visualization), a comparative handwriting-recognition pathway [42,43], and a requirements analysis enumerating intended tasks [44].

Mappings (Data ↔ Visual). VIS4ML does not define low-level visual grammars (marks, channels, encodings). Instead, connections between data and visualization are implicit in *Process* nodes representing inspection activities and their associated *IO-Entities*. Fine-grained mappings at the channel level are therefore not modeled [20].

Technical Implementation. VIS4ML is implemented in OWL 2 and authored in Protégé (https://protege.stanford.edu/, accessed on 5 August 2025). Public artefacts include the OWL file, an interactive browser (https://vis4ml.dbvis.de, accessed on 5 August 2025), and a GitLab repository (https://gitlab.dbvis.de/sacha/VIS4ML, accessed on 5 August 2025). Additional PDFs provide example workflows, the pathway comparison, glossary, and requirements analysis [41–44].

Axiomatization and Metrics. The ontology defines 68 classes, 12 object properties, and 433 axioms. However, structural metrics such as hierarchy depth, DL expressivity, or constraint sets (e.g., disjointedness, cardinality) are not reported [20]. Supplements provide goals and worked workflows but no formal competency questions, SHACL (https://www.w3.org/TR/shacl/, accessed on 5 August 2025) constraints, or reasoner benchmarks [42,44]. Thus, reasoning behavior and constraint coverage remain open for systematic assessment.

3.1.4. SemViz: Automatic Visualization of Semantic Data

Purpose. SemViz is an ontology-driven pipeline that converts Web data into ready-made visualizations without manual chart authoring [21]. It is framed under *Information Realization*, i.e., presenting semantically rich data in textual, graphical, or auditory forms suited to a user's context [45]. Visualization is treated as a mapping problem: ontologies capture expert knowledge about data semantics and representational affordances, then drive automatic tool and style selection. Targeting non-expert users, SemViz has been demonstrated on scenarios such as music charts and sports statistics [46].

Core Concepts. SemViz combines two ontology sets. For *representation selection*, Information Realization defines: (i) an *XML Entities Ontology* (generalizing XML elements/attributes as entities with semantics), (ii) a *Representation Artefacts Ontology* (visual/auditory/textual artefacts and properties such as position, color, pitch), and (iii) a *Target Environment Ontology* (user abilities, context, device characteristics) [45]. For *visualization mapping*, it uses a triad: a *Domain Ontology (DO)* for source semantics, a *Visual Representation Ontology (VRO)* encoding chart grammars, and a *Semantic Bridging Ontology (SBO)* containing weighted correspondences [21]. Two implementations instantiate these ideas: *VizThis*, a tree-centric XML→SVG/X3D tool with semantic assistance [47], and *SeniViz*, a graph-centric pipeline ranking VRO candidates with SBO weights [46].

Workflow Support. The pipeline can be described as six stages [21]: (1) extract tabular data, (2) analyze instances, (3) map to DO, (4) map DO→VRO via SBO, (5) combine/rank visualization plans, and (6) render with toolkits (e.g., ILOG Discovery, Prefuse). Information Realization adds an earlier stage matching the target environment to a modality [45]. VizThis provides semantic assistance, allows rule overrides and "locked" mappings, and supports virtual entities for common transformations [47].

Mappings (Data ↔ Visual). Mappings operate at the schema and instance levels. DO and VRO expose semantically equivalent attributes (e.g., `isQuantitative`), and SBO weights rank candidate mappings [21]. Instance cues (temporal, geographic) steer choices toward timelines or maps [46]. VizThis formalizes visualization-as-mapping with semantic bridges, value transformations, and constrained rules to balance automation with user control [47]. Earlier schema validation work supports descriptor-driven language transformations [48].

Technical Implementation. Ontologies are authored in Protégé and expressed in RDF/OWL (https://www.w3.org/TR/owl2-rdf-based-semantics/, accessed on 5 August 2025) [21]. Rendering uses ILOG Discovery [49], Prefuse (https://github.com/prefuse/Prefuse, accessed on 5 August 2025), and SVG/X3D via VizThis [50]. VizThis also provides data cleansing, AutoMap warnings, and a gallery of end-to-end examples [45–47].

Axiomatization and Metrics. Publications emphasize pipeline design and user evaluation rather than ontology profiling. Schema-level metrics (class/property counts, axioms, DL expressivity) are not reported for DO/VRO/SBO or Information Realization ontologies [21,45]. Qualitative evaluation includes a small user study (n = 6) on music-chart visualizations comparing automatic, cleansed, and manually adjusted mappings, showing trends and outlier detection [47], and further assessments for the graph-centric pipeline [46].

3.1.5. AAVO: Audiovisual Analytics Vocabulary and Ontology

Purpose. The *Audiovisual Analytics Vocabulary and Ontology (AAVO)* was proposed as an OWL ontology and SKOS vocabulary (https://www.w3.org/TR/skos-reference/, accessed on 5 August 2025) for organizing knowledge in *visual analytics*, understood as data-mining supported by interactive visual interfaces [27]. Its goal is a concise, machine-actionable conceptualization relating data, processing methods, and visualization techniques to enable queries, inferences, and support for audiovisual analytics.

Core Concepts. AAVO has two layers: a *core model* in OWL/SKOS and an *expansion* illustrating specializations. The core defines five minimal classes: *Visualization* (techniques generating a *Visual Representation* from *Data*); *Visual Representation* (an *Image* or *Animation*); *Data* (qualitative or quantitative values); *Dataset Type* (organization/semantics, aligned with Munzner's typology [51]); and *Processing* (transforming data, with *Preprocessing* as a subclass) [27]. The expansion instantiates hyponyms, e.g., *Temporal Series*, *Relational Data*, *Z-Score*, *Cleaning*, *MDS*, *Heat Map*, *Histogram*, *Scatter Plot*, *Timeline*. Candidate future additions include *Hypothesis*, *Analysis*, and *Task/Purpose/Application*. The SKOS layer captures synonyms and lexical variants (e.g., "element" → item/observation/row; node → vertex; edge → link) [27].

Workflow Support. AAVO models *Processing* as a transformation *Data* → *Data*, and *Visualization* as a technique producing a *Visual Representation* suitable for a *Dataset Type*. However, the ontology does not define explicit workflow relations (e.g., `hasNext`), so pipelines remain implicit [27].

Mappings (Data ↔ Visual). Data–visual mappings are coarsely specified by the suitability of a visualization for a dataset type and the fact that a visualization produces a representation. No mark/channel grammar or field-to-encoding properties are defined; the expansion lists techniques (e.g., *Heat Map*, *Scatter Plot*) rather than encoding rules [27].

Technical Implementation. AAVO is implemented using Semantic Web standards (RDF/RDFS, SKOS, OWL) with dereferenceable URIs [27]. The core and example expansion are provided as TTL (https://www.w3.org/TR/turtle/, accessed 15 September 2025)/RDF serialization. Python scripts generate both the OWL2 ontology and SKOS scheme, with source code available on GitHub (https://github.com/ttm/aavo, accessed on 5 August 2025). While the repository includes build scripts and example serialization, a compiled OWL file is not consistently published.

Axiomatization and Metrics. Publications do not report structural metrics (e.g., class counts, DL expressivity, constraints) [27]. Repository inspection shows the *full version* defines 27 classes, 8 object properties, 1 data property, and 106 axioms; the *minimal version* has 8 classes, 6 object properties, 1 data property, and 42 axioms.

3.1.6. OntoVis: A Hierarchical Visualization Ontology Stack

Purpose. OntoVis provides a hierarchical ontology stack that formalizes visualization semantics from low-level visual elements up to task semantics. It integrates four layers—*UVO* (Upper Visualization Ontology), *VDO* (Visualization Grammar), *DDO* (Data Domain), and *VTO* (Visualization Tasks)—to support accessible, queryable diagrams and natural-language/assistive interaction with charts [23,24].

Core Concepts. Each layer contributes a distinct vocabulary. *UVO* defines core lexicon classes (*Graphic_Object*, *Graphic_Space*, *Visual_Attribute*, *Visual_Layer*); *VDO* introduces grammar roles (*Graphic_Relation* with subtypes such as *Lineup_GR*, *Map_GR*, *Statistical_Chart_GR*) and syntactic roles (*Axis_SR*, *Container_SR*, *Modifier_SR*); *DDO* specifies data types (*Variable*, *Independent/Dependent Variable*, *Information_Type* with individuals for nominal, ordinal, quantitative); *VTO* models tasks with curated individuals (*Compare*, *Find Anomalies*, *Sort*, etc.). These vocabularies are instantiated in the public OntoVis deployment (https://ontovis.integriert-studieren.jku.at/ontovis-full/, accessed on 5 August 2025).

Workflow Support. A typical annotation flow is: (i) mark up chart objects/spaces with UVO; (ii) assert VDO relations and roles; (iii) type variables and measures with DDO; and (iv) expose task semantics through VTO. This process enables natural-language interfaces and non-visual interaction, as illustrated in accessibility scenarios (e.g., AUDiaL) [23,24].

Mappings (Data ↔ Visual). Mappings are defined via bridging relations between DDO and UVO/VDO, including `has_visual_attribute`, `has_graphic_object`, `has_graphic_space`, `has_syntactic_role`, and `has_information_type`. Suitability is expressed with `appropriateFor` (with inconsistencies in capitalization). Anchoring properties include `hasXCoordinate`, `hasYCoordinate`, `has_area`, `has_length`, `has_color/has_hue`, while task verbalizations are supported by `task_has_verbalization` [23,24].

Technical Implementation. OntoVis is published as an OWLDoc site covering all four modules, with browsable class/property indices and examples. Specifications are provided in RDF/OWL 2; worked use cases across layers are described in the literature [23,24].

Axiomatization and Metrics. The public deployment reports ∼94 classes, 33–34 object properties, 17–18 data properties, and 151 individuals. These layered catalogues are visible in the ontology browser. The publications emphasize modeling patterns and applications rather than detailed schema metrics or reasoner evaluations [23,24].

3.1.7. VisuOnto: An Industrial-Grade Visualization Ontology

Purpose. VisuOnto was introduced at ESWC 2022 [29] and later expanded in IJCKG 2022 [52] as a reusable ontology for industrial visual analytics at Bosch. It standardizes how teams describe visualization tasks, methods, and workflows, ensuring transparency and modularity. VisuOnto functions both as a shared vocabulary and as the backbone for executable knowledge graphs (KGs) that drive semi-automatic chart generation. It is also used in exploratory analysis and machine learning result visualization.

Core Concepts. VisuOnto [52] is structured around three upper classes: *Data*—covering *DataStructure* (vector, matrix, tensor), *DataSemantics* (independent feature, temporal sequence, ML result), and *Plot* (e.g., *LinePlot*, *ScatterPlot*); *VisualMethod*—visualization methods (e.g., *LineplotMethod*, *HeatmapMethod*) with admissible input data and rendering parameters; *VisualTask*—execution of methods on data, subdivided into *AtomicTasks* (e.g., *CanvasTask*, *PlotTask*, *DescriptionTask*) and *VisualPipeline* for ordered task sequences. Together, these support modeling from simple plots to multi-plot dashboards.

Workflow Support. Pipelines are explicitly encoded via *VisualPipeline*, which chains tasks through start, next, and end relations. This enables reasoning about dependencies (e.g., "what follows a given task?") and supports automated KG construction. Industrial case studies demonstrate reusability, with pipelines easily adapted from line plots to scatter plots or combined into dashboards.

Mappings (Data ↔ Visual). Mappings are expressed as constraints between *VisualMethod* and *Data* (e.g., *LineplotMethod* requires array input, *HeatmapMethod* requires a matrix). However, VisuOnto does not yet provide a semantic mapping layer from abstract data fields to perceptual encodings, as found in grammar-based ontologies.

Technical Implementation. Developed in OWL 2 EL for efficient reasoning, VisuOnto was authored in Protégé and validated on Bosch welding data (53.2 million records from 22 machines). Two use cases were reported: visualization of ML predictions with line and scatter plots, and exploratory analysis of welding statistics with heatmaps.

Axiomatization and Metrics. The ontology defines ∼504 axioms, 30 classes, 11 object properties, and 142 data-type properties. Formal constraints ensure pipeline correctness—for example, any *PlotMethod* must specify input data ($\exists$hasMethod.PlotMethod $\sqsubseteq \exists$hasInput) and is restricted to applicable types (PlotMethod $\sqsubseteq \exists$allowedData). These were validated with OWL 2 reasoners and SPARQL queries. Competency questions tested input/output data, method choice, plot counts, and pipeline order, all answerable with VisuOnto [29]. More detailed metrics (depth, DL expressivity, performance benchmarks) are not reported.

3.2. Evaluation of Ontologies

3.2.1. Ontology Purpose and Domain Coverage

Our first step in evaluation was to clarify each ontology's intended purpose, its scope of use, and the main application contexts. This provides both the original motivations behind development and the range of visualization scenarios where the ontology can be effectively applied. Table 2 summarizes concise purpose statements together with representative application areas drawn from the primary references.

Table 2. Concise overview of surveyed ontologies: purpose and primary applications.

Ontology	Purpose (Concise)	Primary Application Area
VISO	Semantic backbone for graphics, data, and effectiveness knowledge	Information visualization design; RVL/AVM integration
VisKo	Encodes operators, services, and views for pipeline composition	Geoscience, astronomy, image/volume processing
VIS4ML	Models VA-supported ML processes and artefacts	ML pipelines with human—AI interaction
SemViz	Semantic mapping from domain ontologies to visualization grammars	Linked-Data/Web visualization; automatic chart selection
AAVO	Links data, processing, and visual/audio outputs (OWL + SKOS)	Audiovisual analytics; multimodal dashboards
OntoVis (full)	Layered lexicon of visual entities, grammar, data, and tasks	Accessible graphics; assistive/NLI interfaces; queryable diagrams
VisuOnto	Models industrial visualization tasks, methods, and pipelines	Manufacturing dashboards; ML results (Bosch use cases)

To complement this, in Table 3, we illustrate the positions of the ontologies within major visualization subfields. This landscape view highlights, for example, that VisKo uniquely addresses *Scientific Visualization*, while most others are concentrated in *Information Visualization* and *Visual Analytics*. Only a few extend toward *Multimedia/Accessibility* (AAVO, OntoVis) or *Graph/Linked-Data visualization* (SemViz).

Table 3. Alignment of surveyed ontologies with major visualization subfields. *Note:* ✓ indicates explicitly covered subfield; "-" indicates not explicitly covered/not a primary focus.

Visualization Subfield	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
Information Visualization (InfoVis)	✓	-	-	✓	✓	✓	✓
Scientific Visualization (SciVis)	-	✓	-	-	-	-	-
Visual Analytics (VA)	✓	-	✓	-	✓	-	✓
Graph/Linked-Data Visualization	-	-	-	✓	-	-	-
Multimedia/Accessibility	-	-	-	-	✓	✓	-
Domain-specific (ML, Industry)	-	-	✓	-	-	-	✓

Finally, in Table 4, we provide a higher-level *scope* view. We distinguish between *general-purpose core ontologies* (VISO, VisKo, SemViz), *domain-specific systems* (VIS4ML, VisuOnto), and ontologies emphasizing *accessibility and media* (AAVO, OntoVis). The presented *categories are non-exclusive*: several ontologies span multiple roles (e.g., VisKo's workflow model is broadly applicable yet often embedded in domain pipelines; OntoVis offers general representational layers but is placed under accessibility/media due to its presentation focus; AAVO bridges visual and auditory analytics). For clarity, we assign each ontology to its *primary* scope in the table.

Table 4. Categorization of surveyed visualization ontologies by scope. *Note:* ✓ the ontology belongs to this category; "-" the ontology does not belong to the category.

Category	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
General-purpose core	✓	✓	-	✓	-	-	-
Domain-specific	-	-	✓	-	-	-	✓
Accessibility/Media	-	-	-	-	✓	✓	-

This tri-perspective comparison (purpose, subfield alignment, scope) situates each ontology within the broader visualization landscape and highlights remaining coverage gaps.

3.2.2. Visualization Process Coverage

Another key aspect when evaluating the overall domain coverage of a visualization ontology is how effectively it models tasks—both the analytic and interaction activities it supports and the procedural stages it formally encodes. Table 5 provides an overview of the task spectrum represented in each ontology, ranging from low-level interactions (e.g., filtering, zooming) to higher-level goals such as comparison, annotation, or explanation. This descriptive view shows whether an ontology is intended as a broad catalogue of visualization tasks (e.g., VISO, OntoVis) or focuses on narrower functions like machine learning pipelines (VIS4ML).

Table 5. Task coverage described in each ontology.

Ontology	Task Coverage Description
VISO	Describe and annotate graphics; overview, filter, zoom, compare; link views; provide suitability/effectiveness guidance (via facts).
VisKo	Compose pipelines: convert, subset/filter, resample/interpolate, map (operator→view), render, view.
VIS4ML	Cover full ML pipeline: prepare data; feature/parameter/split setup; model training with monitoring/steering; evaluation and interpretation.
SemViz	Label result schema, select chart type, bind template (rules/bridges), render automatically.
AAVO	Preprocess and transform data; visualize/sonify; monitor and report outcomes in audiovisual analytics.
OntoVis	Provide curated VTO tasks: compare, characterize distribution, find anomalies, compute aggregates, sort/rank; grounded in UVO/VDO/DDO primitives.
VAO	Support annotation tasks: segment regions, extract MPEG-7 descriptors, link prototypes, tag concepts; retrieve and explain multimedia content.

To complement this qualitative perspective, in Table 6, we present an evaluation of the coverage of the *visualization pipeline tasks*. These tasks span the process from *data preparation* (acquisition, cleaning, transformation), through *mapping and visual design* (data abstraction, encoding, and layout), to *view generation* (composition and rendering), and finally *user interpretation and analysis*. Each ontology is assessed on a three-level scale: *not covered*, *partly/implicitly supported*, or *explicitly modeled*. The results reveal, for example, that VISO and VisuOnto cover nearly the entire pipeline, although some tasks are only partly specified, whereas SemViz primarily emphasizes mapping and view construction with limited support for early preparation steps.

Table 6. Coverage of visualization-pipeline tasks in surveyed ontologies. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported; "-" = not covered.

Task	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
Data acquisition	-	△	△	△	△	-	✓
Data cleaning/preprocessing	-	-	△	-	✓	-	✓
Data transformation/wrangling	△	✓	✓	△	✓	△	✓
Data mapping/abstraction	✓	✓	✓	✓	△	✓	✓
Visual encoding design	✓	△	✓	✓	✓	✓	✓
Layout and composition	✓	△	△	△	△	✓	△
Rendering	△	✓	△	✓	△	✓	△
User interpretation/analysis	-	△	✓	△	△	△	✓

Finally, in Table 7, we present the extension of the assessment toward *supporting tasks* that contribute to effective visualization use, including interaction design, evaluation and refinement, annotation and storytelling, and ontology reuse or modularization. Here, ontologies like VISO, VisuOnto, and OntoVis provide richer coverage of interactive and modular design aspects, while others remain limited to core data-to-view mappings.

Table 7. Coverage of supporting tasks (interaction, evaluation, refinement) in surveyed ontologies. *Note:* ✓ = explicitly modeled; △ = partly supported; "-" = not covered.

Task	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
Interaction design	✓	△	✓	△	△	✓	✓
Evaluation and refinement	△	△	△	△	-	△	✓
Annotation/storytelling	-	-	△	-	-	-	△
Reuse/modularization	△	✓	△	-	△	✓	△

Together, these three perspectives—task spectrum, pipeline stages, and supporting tasks—provide a structured basis for comparing the functional scope of surveyed ontologies and identifying where important coverage gaps remain.

3.2.3. Data Types and Visualization Techniques

Data types and visualization techniques jointly determine an ontology's *generalizability*—that is, its ability to accommodate heterogeneous data structures and to prescribe or recommend diverse visual encodings. We analyzed the surveyed ontologies across three complementary perspectives.

In the first perspective (see Table 8), we assess ontology coverage of Shneiderman's classic data categories [2], including one-dimensional sequences, two-dimensional planar maps, volumetric data, temporal series, multidimensional tables, hierarchical trees, networks, text, and multimedia. This shows whether an ontology can handle the core data structures traditionally used in information and scientific visualization.

In the second perspective (Table 9), we extend this analysis by adding contemporary categories often absent from earlier frameworks such as uncertainty representation, streaming data, geospatial integration, interaction semantics, analytical models, multimodal representations (e.g., audio–visual), and provenance or narrative [9,15,53]. These dimensions expose gaps frequently overlooked but increasingly important in modern applications. The last row in the table is related to narrative and provenance. Provenance refers to the ability of an ontology to capture the origin, processing history, or lineage of datasets and visualizations. Narrative refers to modeling constructs that explicitly support storytelling or explanation, e.g., ordering tasks, highlighting elements, or annotating interpretive aspects of a visualization.

Table 8. Coverage of Shneiderman’s classic data-type categories across surveyed ontologies, showing the ability to represent traditional InfoVis and SciVis data structures. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported; “-” = not covered.

Data Category (Shneiderman)	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
1D/Linear sequences	△	△	✓	△	△	-	✓
2D/Planar (maps, layouts)	✓	✓	△	△	△	✓	△
3D/Volumetric/surfaces	△	✓	-	-	-	-	-
Temporal/time-series	△	△	✓	△	✓	-	✓
Multidimensional/tabular	✓	✓	✓	✓	✓	△	✓
Hierarchical/tree	✓	-	-	△	-	✓	△
Network/graph	✓	△	△	✓	-	✓	-
Text/documents	-	-	-	△	-	△	-
Multimedia (image, video, audio)	-	-	-	-	✓	△	-

Table 9. Coverage of extended data categories beyond Shneiderman’s typology across surveyed ontologies. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported; “-” = not covered.

Extended Data Category	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis	VisuOnto
Uncertainty	-	-	✓	-	-	-	-
Streaming	-	-	-	-	-	-	-
Geospatial	✓	✓	-	△	-	-	△
Interaction semantics	✓	△	✓	-	△	✓	✓
Analytical models	-	-	✓	-	-	-	✓
Multimodal	-	-	-	-	✓	✓	-
Narrative/Provenance	△	✓	△	△	-	✓	-

Finally, in the third perspective (Table 10), we group ontologies by the visualization technique families they explicitly model. This includes statistical charts, network layouts, scientific renderings, dimensionality reduction methods, and sonification techniques. The sub-rows list concrete techniques such as `BarChart`, `HeatMap`, or `VolumeRendering`, as defined in available OWL files or associated repositories.

Taken together, these perspectives reveal both the breadth of input data that an ontology can represent and the richness of visual encodings it supports. VISO spans the widest range of technique families, while VIS4ML extends beyond statistical charts into ML-focused dimensionality reduction. Task- or domain-specific ontologies, such as VisuOnto, remain narrower in scope. Importantly, most ontologies lack robust modeling for extended categories such as uncertainty, provenance, or multimodal data, highlighting opportunities for future integration and standardization.

Table 10. Visualization techniques explicitly modeled in ontology repositories, grouped by technique families. Sub-rows list concrete techniques where present. *Note:* — = technique from concrete family is not implemented.

Ontology	Technique Family	Concrete Techniques
VISO	Statistical	BarChart, PieChart, LineChart, ScatterPlot, Histogram
	Network/Relational	NodeLinkDiagram, ForceDirectedLayout
	Hierarchical	TreeDiagram, Treemap
	Scientific/Spatial	Map, ChoroplethMap
	Dimensionality Reduction/ML	—
	Audio/Sonification	—
VisuOnto	Statistical	LinePlotMethod, ScatterPlotMethod, HistogramMethod, PieChartMethod, HeatmapMethod
	Network/Relational	—
	Scientific/Spatial	—
	Dimensionality Reduction/ML	—
	Audio/Sonification	—
VIS4ML	Statistical	Scatterplot, HeatMap, ROC curve, ConfusionMatrix
	Network/Relational	—
	Scientific/Spatial	—
	Dimensionality Reduction/ML	Projection/DimensionalityReduction (e.g., PCA, t-SNE)
	Audio/Sonification	—
VisKo	Statistical	ScatterPlot, ContourPlot
	Network/Relational	—
	Scientific/Spatial	IsoSurface, VolumeRendering, 2D/3D SurfacePlot
	Dimensionality Reduction/ML	—
	Audio/Sonification	—
AAVO	Statistical	Histogram, Scatter Plot, Heat Map, Timeline
	Network/Relational	—
	Scientific/Spatial	—
	Dimensionality Reduction/ML	—
	Audio/Sonification	—
OntoVis (UVO+)	Statistical	Abstract Marks only (no concrete chart classes)
	Network/Relational	—
	Scientific/Spatial	—
	Dimensionality Reduction/ML	—
	Audio/Sonification	—
SemViz	Statistical	Bar, Line, Pie, Treemap, Node–Link (chart templates for SPARQL results)
	Network/Relational	—
	Scientific/Spatial	—
	Dimensionality Reduction/ML	—
	Audio/Sonification	—

3.2.4. User-Centric Features in Visualization Ontologies

Since visualization is inherently user-driven, it is important to check whether an ontology explicitly models human interaction, user roles, and analytical goals. These features indicate the ontology's potential to support human-in-the-loop systems, personalized recommendations, and adaptive visual interfaces.

Table 11 summarizes the surveyed ontologies according to three key user-centric dimensions. The first dimension, *interaction types*, captures whether the ontology supports common operations such as filtering, zooming, brushing, linking, annotation, or editing—capabilities that are central to exploratory and interactive analysis. The second dimension, *user roles*, reflects an ontology's ability to distinguish between different types of agents, such as creators (designers, developers, researchers), analysts (domain experts, decision-makers), and consumers (end-users, learners, systems). This distinction is particularly important in collaborative or multi-level environments. The third dimension, *tasks and goals*, assesses whether the ontology encodes higher-level analytical objectives, including analysis, exploration, envisionment, and interpretation. Examining coverage in this area helps determine how effectively an ontology can support reasoning about user intentions.

Table 11. Coverage of user-centric features across three dimensions: interaction types, user roles, and analytical tasks/goals. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported; — = not modeled.

Category/Feature	VISO	VisKo	VIS4ML	SemViz	AAVO	OntoVis (UVO+)	VisuOnto
Interaction types							
Filtering	✓	—	✓	—	—	—	✓
Zooming	✓	—	—	—	—	—	✓
Brushing	—	—	—	—	—	—	✓
Linking	△	—	—	—	—	—	✓
Annotation	—	—	—	—	—	—	✓
Editing	—	—	—	—	—	—	✓
User roles							
Creators (designer, developer, researcher)	✓	—	✓	—	—	—	✓
Analysts (domain analyst, decision-maker)	✓	—	✓	—	—	—	✓
Consumers (end-user, learner, system)	✓	—	✓	—	—	—	✓
Tasks and goals							
Analysis	✓	△	✓	—	—	✓	✓
Exploration	✓	—	✓	—	—	✓	✓
Envisionment	—	—	✓	—	—	✓	✓
Interpretation	✓	△	✓	—	—	✓	✓

Together, these dimensions provide a structured lens through which the user-adaptiveness and human-centered design potential of each ontology can be assessed. Notably, only a few ontologies—such as VISO, VIS4ML, and VisuOnto—show strong coverage across all three dimensions, highlighting their greater potential for user-centered applications. OntoVis, by contrast, provides richer modeling of analytical tasks and goals but lacks explicit representation of interaction types or user roles, while VisKo, SemViz, and AAVO remain much more limited in user-related features.

3.2.5. Level of Abstraction

Visualization ontologies differ significantly in the level of abstraction they support. This dimension reflects how deeply an ontology models the visualization process—from low-level graphical primitives, to mid-level chart structures, and up to high-level analytical workflows. Recognizing these levels is essential when assessing the potential of ontologies for rendering engines, visualization tools, or reasoning systems.

Low-level abstraction refers to ontologies that describe basic graphical elements such as marks, encodings, spatial positioning, or perceptual channels. These models enable precise control over the visual grammar of diagrams. For instance, VISO defines detailed classes for `Graphic` elements and spatial `Activity` structures [19], while OntoVis models accessible diagrams using concepts like graphic objects, layers, and spaces [24].

Mid-level abstraction captures predefined chart types and layout structures. SemViz represents chart families such as treemaps and node–link diagrams via its visual representation schema [21], but does not model higher-level analytic workflows. VisuOnto formalizes layout composition for dashboards and multi-plot figures [52]. VIS4ML also defines visualization methods bound to machine learning results, and VISO bridges between low- and mid-level views by modeling layout patterns and encodings [19]. AAVO provides mid-level constructs that link dataset types to visualization forms and offers partial connections to analytical goals in multimodal systems [27].

High-level abstraction focuses on semantic tasks, analytical workflows, and the reasoning behind visualization use. Ontologies in this group model processes such as analysis, exploration, or interpretation. VIS4ML covers machine learning workflows and the role of visualization in interactive model tuning [20]. AAVO connects visual and auditory encodings to analytical goals in multimodal systems [27]. VISO includes user tasks in its activity schema, while VisKo emphasizes service workflows in scientific visualization [26]. VisuOnto similarly models high-level pipelines of atomic and composite tasks [52].

Table 12 synthesizes these distinctions across seven representative ontologies. Most ontologies emphasize only one or two abstraction levels. VISO and OntoVis are among the few that span all three, reflecting their ambition as foundational ontologies. Others, such as SemViz and AAVO, remain more limited, covering mainly mid-level constructs with little or only partial support for high-level reasoning. The absence of vertical integration in many others highlights an opportunity for future work: to unify visual form, layout logic, and analytic purpose within a single, coherent ontology.

Table 12. Level of abstraction coverage in visualization ontologies. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported (e.g., not implemented in released OWL/SKOS); — = not covered.

Ontology	Low-Level	Mid-Level	High-Level	Reference
VISO	✓	✓	✓	[19]
VisKo	—	✓	✓	[26]
VIS4ML	—	✓	✓	[20]
SemViz	✓	✓	—	[21]
AAVO	—	✓	△	[27]
OntoVis	✓	✓	✓	[24]
VisuOnto	—	✓	✓	[52]

3.2.6. Validation and Quality Assurance

Validation and quality assurance of visualization ontologies can be considered across several dimensions. Core quality criteria include *consistency* (absence of contradictory axioms), *completeness* (coverage of requirements), *accuracy* (faithfulness of domain modeling), and *clarity* (human-understandable definitions). Evaluation is typically realized through formal reasoning checks, answering competency questions (CQs), workflow validation, case studies or tool demonstrators, and—less frequently—user studies [7,31,54].

In Table 13, we represent what has been reported in the literature, showing that coverage is highly uneven: only a few ontologies (e.g., VIS4ML, VisuOnto) explicitly apply CQs or reasoner-based validation, while others rely mainly on tool demonstrators or workflow feasibility checks (e.g., VisKo, AAVO). Notably, SemViz is the only ontology reporting user studies, though these remain rare overall. Most omit systematic user studies or metric-based assessments, limiting transparency of quality assurance.

Table 13. Evaluation approaches reported across the seven selected ontologies. *Note:* ✓ = explicitly reported; "-" = not reported.

Ontology	Competency Questions	Workflow Validation	User Studies	Tool Demonstrators	Ontology Metrics
VISO	-	-	-	✓	-
VisKo	-	✓	-	✓	-
VIS4ML	✓	✓	-	✓	-
SemViz	-	✓	✓	✓	-
AAVO	-	-	-	✓	-
OntoVis (UVO+)	-	✓	-	✓	✓
VisuOnto	✓	✓	-	✓	✓

In Table 14, we show the quality of available OWL/RDF artefacts, profiling structural metrics such as class/property counts, axioms, and DL expressivity. These measures are essential for reproducibility but are available only for a subset of ontologies with public resources (e.g., VIS4ML, AAVO, OntoVis). By contrast, VisuOnto's counts are known only from publications, as no dereferenceable OWL file has been released [52].

Table 14. Availability of OWL artefacts and repositories for surveyed ontologies (assessed on 5 August 2025).

Ontology	OWL Repository/URL	Key Notes
VISO	https://github.com/viso-ontology, accessed on 5 August 2025	OWL file available; structural counts not yet reported.
VisKo	https://github.com/orgs/openvisko/repositories/, accessed on 5 August 2025	Multiple OWL modules (Operator, View, Alternate); supports service pipelines; counts vary by package.
VIS4ML	https://gitlab.dbvis.de/sacha/VIS4ML/, accessed on 5 August 2025	OWL file and interactive browser; models VA–ML workflow integration.
AAVO	https://github.com/ttm/aavo/, accessed on 5 August 2025	Full and minimal OWL/SKOS versions; includes build scripts and TTL examples.
OntoVis	https://ontovis.integriert-studieren.jku.at/ontovis-full/, accessed on 5 August 2025	Integrated multi-layer ontology stack.
VisuOnto	Not publicly available. Results as reported in [52]	Structural metrics reported in paper only (30 classes, 11 object properties, 142 data properties).

In (Table 15), we present a cross-ontology comparison by including additional metrics such as lexical metrics (label coverage, definition coverage, synonym support) and reuse of external IRIs. This reveals relatively strong annotation and modular reuse in VIS4ML and VISO, but very low coverage of synonyms and limited external alignments across nearly all surveyed resources. Such gaps hinder semantic interoperability and long-term reusability. To ensure reproducibility, we define the structural metrics also reported in Table 15:

- $D_{\max}$ (maximum depth): the longest path from the root class to a leaf class in the subclass hierarchy.
- D_{avg} (average depth): mean depth of all classes.
- Average fan-out: average branching factor of non-leaf classes.
- Width balance (CV): coefficient of variation of the distribution of class counts across hierarchy levels.

Table 15. Cross-ontology structural and lexical metrics (raw values only; *Note:* — = not reported/not available; # = Number).

Metric	**VisuOnto**	**OntoVis**	**VIS4ML**	**AAVO (Full)**	**VisKo (Packages)**	**VISO**
# Classes	30	94	68	27	∼55	**93**
# Object properties	11	34	11	8	12	**15**
# Data properties	142	18	0	1	6	**27**
Total axioms/triples	504	—	433	106	∼4562 triples	**1003**
Logical axioms	—	—	269	0	—	—
Individuals	—	151	0	0	∼119 (61 operators, 39 services, 10 views, 9 toolkits)	**50**
DL expressivity	EL (OWL 2 EL)	—	—	—	—	—
Max depth ($D_{\max}$)	∼3	—	5	0	3	**5**
Avg depth (D_{avg})	∼1.8	—	3.51	0.0	1.4	**1.12**
Avg fan-out	∼2.1	—	0.99	0.0	0.9	**0.63**
Width balance (CV)	∼0.6	—	2.45	0.00	0.6	**2.24**
Label coverage %	∼100%	—	0%	100%	0%	—
Definition coverage %	Low (0–10%)	—	95.6%	0%	10–20%	—
Synonym coverage %	0%	—	0%	0%	0%	0%
Naming hygiene	Good	—	Moderate–High	High	Moderate	Low–moderate
Reuse/external IRIs	Low	—	Low	Low	High (OWL-S, GMT, VTK, NCL)	Moderate–High
Modularity/cohesion	Low–moderate (3 core areas)	Layered (UVO + VDO + DDO + VTO)	—	—	View/Operator/Service	Modular (multi-module)

We note that triple counts are not strictly comparable across ontologies. For example, VisKo reports ∼4562 triples, dominated by ABox individuals (61 operators, 39 services, 10 views, 9 toolkits), whereas VISO reports 1031 triples, mostly TBox axioms. Such caveats are important when interpreting values across heterogeneous artefacts.

Although we do not present a full OQuaRE assessment, our extracted structural and lexical metrics can be read through OQuaRE's lens [32,33]: *Maintainability* is suggested by modular organization (e.g., multi-module cores; view/operator/service splits; layered designs) and by hierarchy balance (CV); *Operability* is demonstrated by accessible OWL artefacts, documentation, and demonstrators (see Table 14); *Compatibility* emerges where external vocabularies are reused (e.g., workflow and graphics stacks in service-oriented models; moderate–high reuse in core vocabularies); and *Functional adequacy* is reflected in coverage of visualization-pipeline stages (see Table 16).

Table 16. Application of Chi's Data–State Model (DSM) taxonomy to selected ontology pipelines. *Note:* ✓ = explicitly modeled; △ = partly/implicitly supported; — = not covered.

Ontology/Case	**Within Value**	**Data Transformation**	**Analytical Abstraction**	**Visualization Transformation**	**Visualization Abstraction**	**Visual Mapping**	**Within View**
VisuOnto—ML predictions	—	✓	✓	✓	✓	✓	—
VisuOnto—Dataset overview	—	✓	—	✓	✓	✓	—
VIS4ML	—	✓	✓	✓	△	△	—
VISO	—	—	—	✓	✓	△	—
VisKo	—	✓	△	✓	✓	△	—
SemViz	✓	✓	—	✓	✓	✓	—
AAVO	—	✓	—	△	△	—	—
OntoVis	—	—	—	✓	✓	✓	—

To assess functional adequacy, we applied Chi's Data–State Model (DSM) taxonomy [3] to each ontology. Mapping the surveyed ontologies to Chi's Data–State Model (DSM) taxonomy assesses how well they capture the visualization pipeline. VisuOnto provides the broadest coverage, spanning transformation, abstraction, and mapping stages, while other ontologies (e.g., VISO, OntoVis) remain confined to mid-level visual abstractions. AAVO provides some extensions toward multimodal (audio–visual) channels as described in publications [27], though these are only partly reflected in the released OWL artefacts. This confirms that analytic workflow coverage and user-facing interpretive steps remain inconsistent across the field.

Taken together, these evaluations show that while most ontologies demonstrate conceptual soundness, they differ greatly in methodological rigor and transparency. Only a few combine formal validation, structural metrics, and pipeline-level mapping, underscoring the need for stronger and more standardized quality-assurance practices in visualization-ontology engineering.

3.2.7. Interoperability Features

Another important requirement of ontologies is to support seamless integration across heterogeneous tools, platforms, and data domains, ensuring consistent understanding and reuse of visualization knowledge. Table 17 synthesizes five critical dimensions that affect an ontology's ability to integrate with external systems, vocabularies, and workflows.

Table 17. Interoperability features of selected visualization ontologies.

Ontology	Standards Compliance	Modular Design (No. of Modules)	Interdisciplinary Links/Reused Vocabularies	Extensibility Mechanism
VISO	OWL 2 DL, RDF	5	FOAF (graphic modules)	Modular imports; declarative mapping composition (RVL)
VisKo	RDF, OWL-S	3	PML-P, OWL-S	Operator/service extension
VIS4ML	OWL 2	1	Self-contained with no external vocabularies reused	Extensible via Process/IO-Entity hierarchies (pillars)
SemViz	RDF, OWL	3	OMEN framework	Extendable schema
AAVO	OWL, SKOS	2 variants of same	Dublin Core, SKOS	SKOS expansion; OWL subclassing
OntoVis (UVO+)	RDF, OWL 2	4	Dublin Core	Layered hierarchy
VisuOnto	OWL 2 EL	reported as Modular	—	Workflow templates

The first column, *Standards compliance*, indicates which W3C languages each ontology adopts (e.g., RDF, OWL, OWL 2 DL, SKOS). All reviewed ontologies use at least one of these standards, enabling compatibility with Linked-Data ecosystems. For example, VISO, VIS4ML, and OntoVis adopt OWL 2, ensuring expressive modeling and compatibility with reasoning engines [19,20,24].

The logic expressiveness of the underlying knowledge representation language is another important factor affecting interoperability. Ontologies implemented in OWL DL or OWL 2 can represent complex semantics—such as class disjointedness, cardinality constraints, and property chains—allowing advanced reasoning and validation. However, greater expressivity increases computational complexity. Simpler models, such as those using RDFS or OWL Lite, provide faster performance but may limit semantic accuracy. In this review, VISO [19] and VIS4ML [20] employ OWL 2 with sufficient expressivity for reasoning tasks, while VisKo [26] focuses on RDF/OWL-S service descriptions rather than OWL DL axioms. By contrast, SemViz [21] relies primarily on RDF and a simplified OWL schema, closer to OWL Lite or RDFS, favoring simplicity and tool compatibility. This spectrum of logical expressiveness determines how well an ontology can encode detailed visualization semantics while remaining usable across platforms.

The second column, *Modular design*, lists the number of sub-ontologies or components, reflecting internal structure and reusability. VISO is notable for its multi-module organization. VisKo separates View, Operator, and Service modules [19], while OntoVis demonstrates a layered architecture (UVO, VDO, DDO, VTO) that supports incremental extensibility [24].

In the third column, *Interdisciplinary links/reused vocabularies*, we observe reuse of external vocabularies such as Dublin Core (https://www.dublincore.org/specifications/dublin-core/, accessed on 5 August 2025), FOAF (http://xmlns.com/foaf/spec/, accessed on 5 August 2025), OWL-S (https://www.w3.org/submissions/2004/SUBM-OWL-S-20041122/, accessed on 5 August 2025), SKOS (https://www.w3.org/2004/02/skos/, accessed on 5 August 2025), or MPEG-7 (https://www.iso.org/standard/34229.html, accessed on 5 August 2025). These links bridge visualization semantics with broader domain ontologies. For example, AAVO adopts SKOS to structure audiovisual descriptors at a conceptual level, while VisKo connects to OWL-S to describe visualization services [26,27].

The fourth column, *Extensibility mechanism*, describes how new concepts can be incorporated. AAVO uses SKOS-based expansions, VISO and SemViz rely on modular imports [21,27], and OntoVis supports layering that allows extensions without disrupting existing structures [24].

Finally, the practical impact of interoperability is demonstrated through reuse. VISO has been adopted in the RVL code-generation project [19], VisKo supports scientific service pipelines via GMT and VTK [26], and VisuOnto has been integrated into Bosch's industrial knowledge graph [52].

While these adoption cases illustrate external reuse, Table 18 focuses on interoperability between the surveyed ontologies themselves, highlighting where mappings are seamless and where manual alignment is still required, for example due to service–view mismatches (D), primitive–chart grounding issues (E), or domain gaps such as multimedia vs. chart-centric models (F).

Table 18. Interoperability among selected visualization ontologies. Symbols: ✓ = interoperable; △ = partially interoperable/mapping required; "-" = unlikely. Reason codes: A = shared Semantic Web stack; B = conceptual overlap (labels, schema alignment); D = service vs. view mismatch; E = primitive–chart grounding; F = domain gap (e.g., multimedia vs. chart-centric).

Row↓/Col→	**VISO**	**VisKo**	**VIS4ML**	**SemViz**	**AAVO**	**OntoVis**	**VisuOnto**
VISO		△D	△B	✓A,B	△F	△E	△B
VisKo	△D		△D	△D	-F	△E	△A,B
VIS4ML	△B	△D		△B	△F	△E	✓A,B
SemViz	✓A,B	△D	△B		△F	△E	△A,B
AAVO	△F	-F	△F	△F		△E,F	△F
OntoVis	△E	△E	△E	△E	△E,F		△E
VisuOnto	△B	△A,B	✓A,B	△A,B	△F	△E	

3.2.8. Adoption and Community Support

To evaluate an ontology's practical value, we considered both its *community support* and *adoption*.

For community support, we inspected the **availability of public artefacts**, including OWL files, SKOS/RDF serializations, documentation, and source code repositories. The presence of such artefacts in accessible repositories (e.g., GitHub, institutional archives) demonstrates transparency and facilitates reuse, integration, and extension. The availability of artefacts is summarized earlier in Table 14 and complemented here by Table 19 (evidence of real-world applications and demonstrators).

Table 19. Real-world applications and demonstrators operationalizing each ontology.

Ontology	Example Application/Deployment	Reference
VISO	Declarative mapping to Abstract Visual Model (AVM) via RVL prototypes	[19]
VisKo	OpenVisKo toolkit with operator/view ontologies and example pipelines	[26]
VIS4ML	Ontology browser/demonstrator; applied in teaching and VA-assisted ML research	[20]
AAVO	OWL/SKOS artefacts with proof-of-concept for audiovisual analytics	[27]
OntoVis	Accessible/NLI pipelines (e.g., AUDiaL); public OWLDoc vocabulary for annotation	[24]
VisuOnto	Industrial analytics at Bosch (welding dashboards; ontology-driven pipelines)	[52]

For adoption, we measured the *citation counts* of the primary peer-reviewed publications associated with each ontology. While citation counts are not a perfect metric, they remain a widely accepted proxy for community recognition and scholarly impact. These values are reported in Table 20. Citation numbers were retrieved from Google Scholar in August 2025 and should be interpreted as *indicative values only*, as they may drift over time. Taken together, these two perspectives provide a complementary view: the first measures technical openness and reusability, while the second reflects academic visibility and influence.

Table 20. Primary ontology publications and their citation counts (Google Scholar, August 2025).

Ontology	Primary Reference(s)	Citations
VISO	[19]	32
VIS4ML	[20]	159
SemViz	[21]	100
VisKo	[26,40]	0
AAVO	[27]	0
OntoVis (UVO+)	[23]	9
VisuOnto	[52]	4

4. Discussion

4.1. Limitations of the Study

The main limitation of this study lies in the availability and quality of ontology artefacts. For many visualization ontologies, published papers are limited to a single introductory article, and corresponding ontology files in standard format, such as OWL/RDF, are either unavailable, hosted on obsolete platforms, or referenced via broken links. Even when artefacts are nominally accessible (e.g., VISO, VisKo, VIS4ML, AAVO, OntoVis), they are often incomplete or insufficiently documented, preventing the systematic use of formal validation procedures such as reasoning consistency checks, competency question testing, or the application of established evaluation frameworks (e.g., FOCA [34,35], PRISMA [25]). Whenever possible, available artefacts were directly inspected to extract structural metrics and confirm reported features, supplementing the literature-based analysis.

Despite these constraints, a literature-driven evaluation remains valuable. By relying on published descriptions, diagrams, and case studies, we were able to identify observable features of the resources—including domain scope, visualization-pipeline stages coverage, data types, visualization techniques coverage, and user-centric aspects—that provide indicative measures of design breadth and applicability. Although less rigorous than a complete artefact-based assessment, this approach yields a structured and comparable synthesis of available evidence and highlights the need for future work to ensure stable artefact publication, long-term preservation, and richer documentation.

4.2. Gaps in Evaluated Ontologies

Beyond artefact availability, the surveyed ontologies exhibit conceptual and practical gaps that limit their broader utility. These span data coverage, user modeling, validation, interoperability, and adoption.

A first gap is the *narrow support for diverse data types and structures*. Several ontologies are domain-specific—SemViz focuses on SPARQL results, AAVO on audiovisual data, OntoVis on relational/statistical data—and neglect other more general datatypes such as trees, graphs, temporal streams, or multidimensional scientific data. This restricts their applicability in complex, real-world contexts (see Tables 8 and 9).

Another gap is the *inconsistent and incomplete modeling of user-centric tasks*. As summarized in Table 11, only a few ontologies (e.g., VISO, VIS4ML, VisuOnto) explicitly formalize user roles, interaction types, or analytic goals. Others take a limited approach: AAVO omits user modeling, while OntoVis defines analytic tasks but not their assignment to specific users. Without explicit representation of user intent and feedback, these ontologies cannot fully support human-centered or task-driven applications.

Validation and empirical grounding are also sparse. Most ontologies have been demonstrated only in small-scale prototypes or conceptual use cases (e.g., RVL mappings for VISO, Bosch cockpit for VisuOnto), with little evidence of large-scale deployment or benchmarking. This makes it difficult to assess whether class hierarchies reflect practical workflows or whether reasoning performance scales (cf. Table 13).

Conceptual inconsistencies persist across projects. As shown in Table 21, core notions such as *Chart*, *Visual Representation*, *Mapping*, or *Interaction* are modeled in divergent ways—for example, SemViz encodes charts as families of templates, VisuOnto treats them as dashboard-level chart types, and OntoVis represents them through *Graphic_Relation* individuals. These differences slow interoperability, requiring costly schema alignment and risking semantic ambiguity.

Table 21. Conceptual inconsistencies across selected visualization ontologies.

Concept	Ontology	Definition/Representation
Chart	SemViz	Encoded as chart family (e.g., bar, treemap) under Visual Representation Ontology
	VisuOnto	Dashboard-level chart types (LinePlot, ScatterPlot, Heatmap)
	OntoVis	No concrete chart classes; instead represented through *Graphic_Relation* individuals such as *Statistical_Chart_GR*.
Visual Representation	VISO	Graphical marks, encodings, and layout properties
	VIS4ML	Views of ML model stages (e.g., ROC, confusion matrix) used for interpretability
	AAVO	Visual Representation = Image or Animation; expansion mentions auditory outputs (sonification) but not implemented in OWL
	OntoVis	UVO layer defines *Graphic_Object*, *Visual_Attribute*, *Visual_Layer*, and related classes.
Mapping	VISO	Channel mappings between data and visual encodings
	VisKo	Transformation operators between services
	OntoVis	Bridging properties between DDO variables and UVO/VDO entities (e.g., `has_information_type`, `has_visual_attribute`).
Interaction	VISO, VIS4ML, VisuOnto	Filtering, steering, and other partial interaction support
	SemViz, OntoVis	Not modeled explicitly (OntoVis provides syntactic roles but no interaction semantics).
User Roles/Tasks	VISO	Generic user role (Analyst) and limited activity classes (Filtering, Zoom)
	VIS4ML	Roles in ML workflows (Analyst, Developer/Creator)
	VisuOnto	Analysts and system users in dashboard contexts
	OntoVis	Defines *Visualization_Task* and task individuals (e.g., Compare, Distribution, Sort), but does not model user roles.

Finally, the ontology *adoption levels remain low*. As Table 20 shows, VIS4ML and SemViz have gained notable attention, while others remain little cited (e.g., VisKo, AAVO, VisuOnto). Citation levels do not always align with reusability: SemViz is well cited but lacks an accessible OWL artefact, while VisKo and AAVO provide repositories but have little uptake. VISO sits in between with moderate citations and good artefact availability, suggesting stronger long-term potential.

In summary, the ontologies reviewed reveal partial coverage, inconsistent vocabularies, limited user modeling, sparse validation, and uneven adoption. Addressing these gaps will require a more unified and extensible ontology of visualization—one that integrates diverse data types, models complete analytic workflows, incorporates user perspectives, and supports semantic interoperability across domains.

4.3. Emerging Ontology Design Patterns

Although visualization ontologies differ in scope and maturity, our analysis uncovered several recurring modeling strategies that can serve as design patterns for future ontology developments. These patterns exemplify best practices in ontology engineering and provide a foundation for creating more comprehensive, adaptable, and interoperable resources.

Pipeline-Based Abstraction. A widely observed strategy is to conceptualize visualization as a pipeline of sequential stages—from data acquisition and transformation, through encoding and rendering, to presentation and interaction. This abstraction is explicit in ontologies such as VIS4ML, VISO, VisKo, and OntoVis, where tasks like `data preparation`, `mapping`, `view generation`, and `interaction` are represented (see Tables 6 and 7). By modeling the pipeline, these ontologies make it possible to trace the provenance of data, link transformations to resulting visual forms, and reconstruct workflows. This not only supports reproducibility but also enables comparative analysis and adaptation of visualization processes across applications.

Modular Architecture. Several ontologies employ a modular structure, decomposing the domain into loosely coupled sub-ontologies. For instance, VISO separates concepts into modules such as *Graphic*, *Data*, *Interaction*, *User*, *Activity*, *Task*, and *Context*. OntoVis similarly adopts a layered design (UVO, VDO, DDO, VTO). AAVO also distinguishes between visual and auditory outputs as modular components of its multimodal framework. As summarized in Table 17, modularization facilitates selective reuse, incremental extension, and collaborative development. For example, HCI experts can refine the interaction module independently from data scientists who extend the data module. This separation of concerns lowers barriers to adoption while supporting more scalable ontology engineering.

Reuse of Semantic Web Standards and General Vocabularies. Several of the surveyed ontologies demonstrate alignment with existing Semantic Web standards and general vocabularies, which enhances interoperability and shortens development time. VISO and AAVO reuse vocabularies like FOAF, Dublin Core, and SKOS, while VisKo incorporates OWL-S for service composition. This reuse creates opportunities for cross-domain integration, enabling richer queries that combine visualization semantics with external resources (e.g., metadata, provenance, or media descriptors). It also ensures compatibility with existing reasoning engines and Linked-Data infrastructures.

Community-Driven Evolution. A growing trend is the move toward open, collaborative ontology development. For example, VISO, AAVO, VIS4ML, VisKo, and OntoVis all maintain public repositories (e.g., GitHub, GitLab) with OWL artefacts, build scripts, or documentation (see Table 19). Such openness supports version control, issue tracking, and external contributions, reflecting practices from open-source software engineering. Even where artefacts are incomplete, public repositories encourage reuse, experimentation, and feedback from broader communities.

These patterns—pipeline abstraction, modular architecture, reuse of Semantic Web standards, and community-driven evolution—demonstrate early signs of convergence across otherwise independent efforts. Together, they improve extensibility and interoperability, while also aligning visualization ontologies with broader semantic infrastructures and collaborative research practices. Future ontology projects in the domain of visualization should adopt and refine these principles to support full analytic workflows, diverse data types, and user-centered tasks, thereby advancing toward a robust and reusable *generic ontology of visualization*.

4.4. Toward a Generic Ontology of Visualization

The notion of a *generic ontology for visualization* emerges from the need to establish a domain-independent semantic backbone that systematically describes visualization processes, techniques, and artefacts. Unlike domain-specific ontologies, which are often tailored to narrow application contexts (e.g., biomedical imaging or geospatial analysis), a generic ontology seeks to represent *all phases of the visualization pipeline* in a unified and reusable manner, addressing the uneven coverage highlighted in Table 6 (F1). The underlying design principles are summarized in Figure 1.

P1: Pipeline-based modularity. A generic ontology should be conceived as a set of interconnected modules that reflect key aspects of the visualization pipeline, such as *Data*, *Graphic*, *Interaction*, *User*, and *Task*. Such modularization (cf. Section 3.2.7) would enable selective reuse and independent evolution across communities (e.g., HCI, visualization, data science), while maintaining overall semantic coherence (F5, F11). Moreover, it can provide a three-level abstraction stack—from low-level graphics to chart/view structures to workflow reasoning (F4).

P2: Separation of structure and interpretation. Building on the modular view outlined in P1, a generic ontology should also distinguish between the *structural representation* of a visualization and its *interpretive label*. For example, a histogram can be described structurally as a rectangle-based layout, while its identification as a “histogram” arises from analytic intent (e.g., summarizing frequency distributions). This principle directly addresses the conceptual divergences highlighted in Table 21, and it provides the foundation for constraint and quality rules (appropriateness, expressiveness; F7), intent-aware recommendations through an explicit mapping layer (F2), and vocabulary harmonization (F6).

P3: Layered integration with domain ontologies. A generic ontology should be conceived as a backbone that complements, rather than replaces, domain-specific ontologies. Its role is to provide the common layer for concepts such as charts, mappings, tasks, interactions, and user context, while domain ontologies contribute area-specific semantics. This layered integration can be achieved through modular imports and OWL/RDF alignment hooks (e.g., `owl:equivalentClass`, `skos:closeMatch`). Such an approach directly addresses shortcomings in prior resources—limited artefacts and validation, incomplete modeling of users and tasks, uneven data-type support (Tables 8 and 9), and fragmented vocabularies (Table 21). By linking a generic layer with domain layers, the ontology ensures broad interoperability while preserving domain-level precision (F5–F6), and it supports reproducible evaluation and community adoption (F10–F12), as reflected in artefact and adoption patterns (Tables 13 and 20).

P4: Pattern-oriented engineering. A generic ontology should follow recurring design patterns that have proven effective in prior visualization and Semantic Web efforts. These include pipeline abstraction, modular architecture, reuse of Semantic Web standards, and community-driven evolution. Such patterns provide reusable components and operationalize key features: F2 (explicit mapping), F4 (three abstraction levels), F8 (data and measurement/scale foundations, including uncertainty), F9 (provenance and justification across data→mapping→view and interaction histories), and F10 (evaluation hooks with CQs, scenarios, and regression tests), while reinforcing F12 (metadata and multilinguality).

P5: Alignment with emerging AI paradigms. A generic ontology should also act as a bridge to knowledge graphs (KGs) and large language models (LLMs). Ontologies provide the structured vocabulary and relations that LLMs lack, reducing hallucinations and harmonizing chart synonyms (e.g., "bar chart," "column chart," "histogram") [55].

Generic Ontology for Visualization — Principles and Feature Set

Definition. A domain-independent semantic backbone that formalizes the full visualization pipeline—data, mappings, views, interaction, tasks, and user context—so heterogeneous systems can reason over and reuse visualization knowledge across application areas.

Guiding Principles.

P1 **Pipeline-based modularity:** Interconnected modules (*Data*, *Graphic*, *Interaction*, *User*, *Task*) with clear interfaces to enable selective reuse and independent evolution.

P2 **Separation of structure and interpretation:** Structural classes (marks, channels, layouts) are independent of analytic intent (e.g., "histogram" emerges from task + data semantics).

P3 **Layered integration:** The generic backbone aligns with domain ontologies via imports/mappings, ensuring general interoperability and domain-level precision.

P4 **Pattern-oriented engineering:** Reuse recurring design patterns (pipeline abstraction, modularity, Web standards alignment) to maximize extensibility and reasoning utility.

P5 **Alignment with emerging AI paradigms:** Provide a semantic bridge to knowledge graphs and LLMs by offering controlled vocabularies and formal relations that support disambiguation, constraint-guided generation, and explainability.

Feature Set (must-have).

F1 **Complete pipeline coverage:** Explicit constructs for data prep/transform, mapping/abstraction, view/layout/rendering, and user interpretation.

F2 **Explicit mapping layer:** Roles linking data semantics and measurement scales to visual encodings/composition (incl. multiview), enabling constraints and recommendations.

F3 **User–task–interaction triad:** User roles, analytic goals, and interaction semantics (filter, zoom, brush, link, annotate) with effects on data/view state.

F4 **Three abstraction levels:** Low-level graphics, mid-level chart/view structures, high-level analytical workflows.

F5 **Interoperability by design:** OWL/RDF compliance, modular imports, alignment hooks (e.g., `owl:equivalentClass`, `skos:closeMatch`).

F6 **Vocabulary harmonization:** Resolve divergences (Chart/Visual Representation/Mapping) with a consistent category scheme and synonyms.

F7 **Constraint and quality rules:** Appropriateness/expressiveness constraints, effectiveness hints (e.g., channel rankings), and validation patterns.

F8 **Data and scale foundations:** Attributes, datasets (table/matrix/graph), value domains, measurement scales, units, uncertainty/error annotations.

F9 **Provenance and justification:** Trace data→mapping→view decisions and interaction-driven state changes.

F10 **Evaluation hooks:** Competency questions (CQs), example scenarios, expected inferences, and regression tests over time.

F11 **Versioning and evolution:** Stable IRIs, deprecation policy, semantic versioning, change logs.

F12 **Metadata and multilinguality:** Labels, definitions, scope notes, examples, usage notes in multiple languages.

Concise one-liner. A modular, three-level OWL ontology that fully models the visualization pipeline, separates form from intent, aligns with domain ontologies, and bridges to AI paradigms (KGs, LLMs) to enable reasoning, interoperability, and reuse across heterogeneous systems.

Figure 1. Generic Ontology for Visualization—Principles and Feature Set.

This principle has three main implications: it enables *disambiguation* by aligning chart synonyms and normalizing user queries [19,52]; it supports *constraint-guided generation* by enforcing valid data–encoding mappings and preventing impossible combinations [21,56]; and it enhances *explainability* by grounding recommendations and outputs in ontology-defined rules for transparency [20,27].

Examples such as VIS4ML's ML workflows [20] and VISO's rule-based mappings [19] show the potential of ontology-driven automation. Coupled with LLMs, these mechanisms can enable reliable chart suggestions, adaptive pipelines, and educational explanations, while LLM-based interfaces can, in turn, inform and enrich the ontology [57,58].

Objective and status. The overarching objective is to develop a reusable OWL backbone that facilitates reasoning, interoperability, and integration across heterogeneous visualization systems and domains. At present, the ontology remains under development, with the aim of operationalizing **P1–P5** and implementing the broader **Feature Set** (F1–F12). These principles collectively address the evidence summarized in Tables 6 and 21, and related tables—ranging from pipeline modularity and structural separation (P1–P2), through layered integration and design patterns (P3–P4), to emerging AI alignment with LLMs (P5). Together, they define the contours of a generic ontology envisioned as both a semantic backbone for visualization and a foundation for adaptive, trustworthy, and explainable visualization assistants.

5. Conclusions

This survey presented a comprehensive analysis of seven representative ontologies for visualization, synthesizing their design approaches, levels of abstraction, coverage of user-related features, interoperability, adoption, and limitations. Our aim was to assess the current landscape and to identify the essential building blocks for constructing a *generic ontology of visualization* that supports interoperability, machine reasoning, and semantically grounded visualization systems.

Our findings reveal that while several ontologies cover important aspects of the visualization pipeline, no single model spans the full life cycle from data preparation to user interaction and analytic goals. Critical aspects such as user intent, interaction techniques, and task context remain underrepresented. Moreover, fragmented conceptual vocabularies and inconsistent terminology slow integration and reuse across systems. When comparing adoption indicators (citations) with artefact availability (Tables 14, 19, and 20), we observe that many ontologies remain isolated research prototypes with limited follow-up work. At the same time, recurring design patterns—such as pipeline-based abstraction, modular architecture, and alignment with external Semantic Web standards—provide a promising foundation for future reuse and unification.

For researchers, ontology engineers, and visualization developers, these results highlight the need to: (1) strengthen interoperability by aligning vocabularies across ontologies, (2) provide more complete modeling of user roles, tasks, and interactions, and (3) improve validation and empirical grounding in real-world systems. Addressing these gaps will enable ontologies not only to capture visualization semantics but also to support reasoning, recommendation, and adaptive systems.

Overall, this survey underscores the importance of developing a generic ontology for visualization that: *closes coverage gaps* in data, tasks, and users; *standardizes terminology* through shared vocabularies and mappings; and *formalizes reusable design patterns* that can be integrated into future tools. The long-term goal is a community-driven, extensible ontology that links to larger knowledge graphs, supports automated reasoning, and underpins next-generation visualization technologies. Such a unified framework would serve both as a shared semantic backbone for visualization research and as an enabler of innovation in data-driven insights across disciplines.

Author Contributions: Conceptualization, S.L. and P.P.; methodology, S.L. and P.P.; validation, S.L. and P.P.; formal analysis, S.L. and P.P.; investigation, S.L. and P.P.; resources, S.L. and P.P.; data curation, S.L. and P.P.; writing—original draft preparation, S.L. and P.P.; writing—review and editing, S.L. and P.P.; supervision, P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Slovenian Research and Innovation Agency (ARIS) through the research program Knowledge Technologies (P2-0103).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors utilized four AI-assisted platforms—ChatGPT (OpenAI, ’o3’ version/GPT-4), Gemini (Google, free version, accessed 13 June 2025), Undermind (Deep Scientific Research, PRO version), and Perplexity AI (accessed 13 June 2025)—as a supplementary aid in the literature selection and analysis phase. These tools were employed strictly for two purposes: to screen queries for obvious erroneous results (‘wrong hits’) and to broaden the scope by identifying gray literature and preprints. The human authors were solely responsible for the complete verification of all suggested references and maintain full accountability for the content and integrity of the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Acronym	Definition
AAVO	Audiovisual Analytics Vocabulary and Ontology
ABox	Assertion Box
ACM	Association for Computing Machinery
AVM	Abstract Visual Model
CQ	Competency Questions
DBLP	Digital Bibliography and Library Project
DL	Description Logic
DO	Domain Ontology
DOI	Digital Object Identifier
FOAF	Friend of a Friend vocabulary
HCOME	Human-Centered Collaborative Ontology Engineering
HTML	HyperText Markup Language
IEEE	Institute of Electrical and Electronics Engineers
InfoVis	Information Visualization
LOV	Linked Open Vocabularies
ML	Machine Learning
MQ	Ontology Metrics (Quality)
NCL	NCAR Command Language
OMEN	Ontology-based Multimedia Environment
OntoVis	Ontology for Visualization (UVO + VDO + DDO + VTO stack)
OWL	Web Ontology Language
PML-P	Proof Markup Language – Provenance
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QVT	Query/View/Transformation

RDF	Resource Description Framework
RDFS	RDF Schema
RVL	Rule-based Visual Language
SBO	Semantic Bridging Ontology
SemViz	Automatic Visualization of Semantic Data
SHACL	Shapes Constraint Language
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
SWRL	Semantic Web Rule Language
TBox	Terminology Box
TD	Tool Demonstrator
TLVO	Top-Level Visualization Ontology
URI	Uniform Resource Identifier
UVO	Upper Visualization Ontology
VAO	Visual Annotation Ontology
VDO	Visual Description Ontology
VIS4ML	Visual Analytics for Machine Learning Ontology
VisKo	Visualization Knowledge Ontology
VisuOnto	Industrial-Grade Visualization Ontology
VISO	Visualization Ontology
VRO	Visual Representation Ontology
VTO	Visualization Task Ontology
VTK	Visualization Toolkit
W3C	World Wide Web Consortium
WF	Workflow Validation

Appendix A

Table A1. Candidate visualization ontologies grouped by period of emergence.

Ontology	Year	Scope/Role (Very Short)	Reference
2004–2008: Early and foundational efforts			
Building an Ontology of Visualization	2004	Foundational attempt to formalize visualization concepts.	[16]
VisIOn—Interactive Visualization Ontology	2004	Early catalogue of software-visualization systems.	[59]
Ontology Construction for Scientific Viz	2006	Concept paper on visualization ontology in science.	[60]
SemViz	2007	"Semantic visualization" toolkit with ontologies; 2D/3D viewers.	[21]
2010–2015: Domain ontologies and prototypes			
Enhanced Visualization Ontology	2010	Extends earlier ontology with richer process nodes.	[18]
VisKo—Visualization Knowledge Ontologies	2011	Semantic planning of pipelines for Earth-science data.	[26]
Chart Ontology	2011	Maps SPARQL result sets to chart types.	[56]
Label Ontology	2011	Semantic typing variables (used by Chart Ontology).	[56]
Unifying Visualization Ontology	2011	Upper-level OWL, unifying existing vis models.	[36]
Ontology of 3-D Techniques	2012	OWL of 3-D visualization techniques for city models.	[61]
VISO—Visualization Ontology	2013	Generic backbone (marks, channels, tasks, data, inter.).	[19]
VUMO—Visualization Use Model Ontology	2014	Urban-mobility events → visual forms.	[62]
2016–2022: Modern frameworks and extensions			
AAVO—Audiovisual Analytics Ontology	2017	Vocabulary linking data-mining tasks, visual/audio views.	[27]
VAO—Visual Analytics Ontology	2018	enables ontology-guided visual analytics; supports interactive exploration, filtering, and spatio-temporal visualization.	[63]
VIS4ML	2019	Formalizes VA-assisted ML workflows.	[20]
Chen–Ebert IVAS Ontology	2019	Ontological framework for VA design/eval.	[64]
SBOL-VO—SBOL Visual Ontology	2020	Glyph catalogue for synthetic-biology circuits.	[65]
UVO—Upper Visualization Ontology	2021	Foundation-level vocabulary of graphic objects.	[24]
STMaps	2021	Dual ontologies for spatio-temporal analytics.	[66]
VisuOnto (Bosch)	2022	Industrial chart-workflow ontology.	[29]
2023–Present: Recent additions			
CH Heritage Viz Ontology	2023	Cultural-heritage visualization ontology.	[67]

Table A2. Adapted PRISMA-style selection process for visualization ontologies. The stages are simplified because many artefacts are from gray literature and repositories; counts are approximate.

Stage	Criteria/Actions	Resulting Artefacts

Identification	Sources consulted:
	Libraries: IEEE Xplore, ACM Digital Library, Scopus, Web of Science, DBLP, Google Scholar, ScienceDirect (Elsevier), SpringerLink, Taylor and Francis Online, and MDPI;
	Open Ontology Repositories: Linked Open Vocabularies (LOV), Ontobee, and GitHub/GitLab public projects;
	Gray Literature and Preprints: arXiv, and Zenodo;
	Full text search with any combination of keywords ontology, visualization, generic.
		more than 100 **manuscripts** initially identified

Screening	Deduplication and relevance check: removed overlapping or fragmentary versions; retained those with some public artefacts.	21 **ontology** candidates
Eligibility	Full-text inspection and artefact availability: required peer-reviewed description or repository with ontology files/documentation.	4 representative **ontologies**
Included	Ontologies analyzed in detail: VISO, VisKo, VIS4ML, AAVO, OntoVis (UVO+), VisuOnto, SemViz. With additional analysis of documentation content and links, three additional cases were considered.	7 **ontologies** in final comparisons

References

1. Zeng, M.L. Knowledge organization systems (KOS). *KO Knowl. Organ.* **2008**, *35*, 160–182. [CrossRef]
2. Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proceedings of the IEEE Symposium on Visual Languages, Boulder, CO, USA, 3–6 September 1996; pp. 336–343. [CrossRef]
3. Chi, E. A taxonomy of visualization techniques using the data state reference model. In Proceedings of the IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings, Salt Lake City, UT, USA, 9–10 October 2000; pp. 69–75. [CrossRef]
4. Munzner, T. A Nested Model for Visualization Design and Validation. *IEEE Trans. Vis. Comput. Graph.* **2009**, *15*, 921–928. [CrossRef]
5. Tory, M.; Möller, T. Rethinking Visualization: A High-Level Taxonomy. In Proceedings of the IEEE Symposium on Information Visualization, Austin, TX, USA, 10–12 October 2004; pp. 151–158. [CrossRef]
6. Brehmer, M.; Munzner, T. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2376–2385. [CrossRef]
7. Gruber, T. Ontology. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: New York, NY, USA, 2018; pp. 2574–2576. [CrossRef]
8. Brodlie, K.W.; Duce, D.A.; Herman, I.; Duke, D.J. Do You See What I Mean? *IEEE Comput. Graph. Appl.* **2005**, *25*, 6–9. [CrossRef]
9. Keim, D.A.; Kohlhammer, J.; Ellis, G.; Mansmann, F. *Mastering the Information Age: Solving Problems with Visual Analytics*; Eurographics Association: Goslar, Germany, 2010. [CrossRef]
10. Card, S.K.; Mackinlay, J.D.; Shneiderman, B. (Eds.) *Readings in Information Visualization: Using Vision to Think*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999.
11. Satyanarayan, A.; Moritz, D.; Wongsuphasawat, K.; Heer, J. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 341–350. [CrossRef]
12. Heer, J.; Mackinlay, J.; Stolte, C.; Agrawala, M. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Trans. Vis. Comput. Graph.* **2008**, *14*, 1189–1196. [CrossRef]
13. Lhuillier, A.; Hurter, C.; Telea, A. State of the Art in Edge and Trail Bundling Techniques. *Comput. Graph. Forum* **2017**, *36*, 619–645. [CrossRef]
14. Mackinlay, J.D. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* **1986**, *5*, 110–141. [CrossRef]
15. Munzner, T. *Visualization Analysis and Design*, 1st ed.; A K Peters/CRC Press: Boca Raton, FL, USA, 2014. [CrossRef]

16. Duke, D.J.; Brodlie, K.W.; Duce, D.A. Building an Ontology of Visualization. In Proceedings of the IEEE Visualization 2004 (Poster Session), Austin, TX, USA, 10–15 October 2004. [CrossRef]
17. Shu, G.; Avis, N.J.; Rana, O.F. Bringing Semantics to Visualization Services. *Adv. Eng. Softw.* **2008**, *39*, 514–520. [CrossRef]
18. Pérez, A.M.; Pérez-Risquet, C.; Gómez, J.M. An Enhanced Visualization Ontology for a Better Representation of the Visualization Process. In *ICT Innovations 2010: Third International Conference, Ohrid, Macedonia, 12–15 September 2010. Revised Selected Papers*; Communications in Computer and Information Science; Gusev, M., Mitrevski, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 83, pp. 342–347. [CrossRef]
19. Polowinski, J.; Voigt, M. VISO: A Shared, Formal Knowledge Base as a Foundation for Semi-Automatic InfoVis Systems. In Proceedings of the CHI '13 Extended Abstracts on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013; pp. 1791–1796. [CrossRef]
20. Sacha, D.; Kraus, M.; Keim, D.A.; Chen, M. VIS4ML: An Ontology for Visual Analytics Assisted Machine Learning. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 385–395. [CrossRef]
21. Gilson, O.; Silva, N.; Grant, P.W.; Chen, M. From Web Data to Visualization via Ontology Mapping. *Comput. Graph. Forum* **2008**, *27*, 959–966. [CrossRef]
22. Asprino, L.; Colonna, C.; Mongiovì, M.; Porena, M.; Presutti, V. Pattern-based Visualization of Knowledge Graphs. *arXiv* **2021**, arXiv:2106.12857. [CrossRef]
23. Murillo-Morales, T.; Miesenberger, K. Ontology-based Semantic Support to Improve Accessibility of Graphics. In *Assistive Technology: Building Bridges*; Studies in Health Technology and Informatics; IOS Press: Amsterdam, The Netherlands, 2015; Volume 217, pp. 255–260. [CrossRef]
24. Murillo-Morales, T.; Miesenberger, K. Formalizing Visualization Semantics for Accessibility. In Proceedings of the 4th International Workshop on Digitization and E-Inclusion in Mathematics and Science (DEIMS 2021), Nihon University, Tokyo, Japan, 18–19 February 2021; pp. 47–54.
25. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef]
26. Rio, N.D.; da Silva, P.P. *VisKo: Semantic Web Support for Information and Scientific Visualization*; Technical Report UTEP-CS-11-58; University of Texas at El Paso: El Paso, TX, USA, 2011.
27. Fabbri, R.; de Oliveira, M.C.F. Audiovisual Analytics Vocabulary and Ontology (AAVO): Initial core and example expansion. *arXiv* **2017**, arXiv:1710.09954. [CrossRef]
28. Sun, K.; Zhu, Y.; Pan, P.; Hou, Z.; Wang, D.; Li, W.; Song, J. Geospatial data ontology: The semantic foundation of geospatial data integration and sharing. *Big Earth Data* **2019**, *3*, 269–296. [CrossRef]
29. Zheng, Z.; Zhou, B.; Soylu, A.; Kharlamov, E. Towards a Visualisation Ontology for Data Analysis in Industrial Applications. In Proceedings of the 1st International Workshop on Semantic Industrial Information Modelling (SemIIM 2022) Co-Located with the 19th Extended Semantic Web Conference. CEUR-WS.org, Hersonissos, Greece, 30 May 2022; CEUR Workshop Proceedings; Volume 3355, pp. 1–6.
30. Fernández-López, M.; Gómez-Pérez, A.; Juristo, N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In Proceedings of the AAAI Conference on Artificial Intelligence, 1997. Available online: https://aaai.org/papers/0005-ss97-06-005-methontology-from-ontological-art-towards-ontological-engineering/ (accessed on 5 August 2025).
31. Raad, J.; Cruz, C. A Survey on Ontology Evaluation Methods. In Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2015, Lisbon, Portugal, 12–14 November 2015; pp. 179–186. [CrossRef]
32. Duque-Ramos, A.; Fernandez-Breis, J.; Stevens, R.; Aussenac-Gilles, N. OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. *J. Res. Pract. Inf. Technol.* **2011**, *43*, 159–176.
33. Duque-Ramos, A.; Fernández-Breis, J.T.; Iniesta, M.; Dumontier, M.; Egaña Aranguren, M.; Schulz, S.; Aussenac-Gilles, N.; Stevens, R. Evaluation of the OQuaRE Framework for Ontology Quality. *Expert Syst. Appl.* **2013**, *40*, 2696–2703. [CrossRef]
34. Bandeira, J.; Bittencourt, I.I.; Espinheira, P.; Isotani, S. FOCA: A Methodology for Ontology Evaluation. *arXiv* **2017**, arXiv:1612.03353. [CrossRef]
35. Martin, J.; Axelsson, J.; Carlson, J.; Suryadevara, J. Evaluation of Systems Engineering Ontologies: Experiences from Developing a Capability and Mission Ontology for Systems of Systems. In Proceedings of the 2024 IEEE International Symposium on Systems Engineering (ISSE), Perugia, Italy, 16–19 October 2024; pp. 1–8. [CrossRef]
36. Voigt, M.; Polowiński, J. *Towards a Unifying Visualization Ontology (VISO)*; Technical Report TUD-FI11-01; Technische Universität Dresden, Institut für Software- und Multimediatechnik: Dresden, Germany, 2011.
37. Polowinski, J. Ontology-Driven, Guided Visualisation Supporting Explicit and Composable Mappings. Ph.D. Thesis, Technische Universität Dresden, Dresden, Germany, 2017.

38. Polowinski, J. Towards RVL: A declarative language for visualizing RDFS/OWL data. In Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, WIMS '13, Madrid, Spain, 12–14 June 2013. [CrossRef]
39. Voigt, M.; Franke, M.; Meißner, K. Using Expert and Empirical Knowledge for Context-aware Recommendation of Visualization Components. *Int. J. Adv. Life Sci.* **2013**, *5*, 27–41.
40. Del Rio, N.; Pinheiro da Silva, P. Capturing and Using Knowledge about the Use of Visualization Toolkits. In Proceedings of the AAAI Fall Symposium Series—Discovery Informatics: The Role of AI Research in Innovating Scientific Processes (FS-12-03), Arlington, TX, USA, 2–4 November 2012; AAAI Press: Menlo Park, CA, USA, 2012.
41. Sacha, D.; Kraus, M.; Keim, D.A.; Chen, M. VIS4ML Supplement: Glossary and Ontology Notation. 2018. Available online: https://gitlab.dbvis.de/sacha/VIS4ML/-/blob/master/vis4ml-glossary-ontology.pdf (accessed on 5 August 2025).
42. Sacha, D.; Kraus, M.; Keim, D.A.; Chen, M. VIS4ML Supplement: Four Example Workflows—Detailed Modeling. 2018. Available online: https://gitlab.dbvis.de/sacha/VIS4ML/-/blob/master/ExampleWorkflows.pdf (accessed on 5 August 2025).
43. Sacha, D.; Kraus, M.; Keim, D.A.; Chen, M. VIS4ML Supplement: Martins' Pathway—VA-Assisted Handwriting Recognition Workflow. 2018. Available online: https://gitlab.dbvis.de/sacha/VIS4ML/-/blob/master/Martins_Pathway.pdf (accessed on 5 August 2025).
44. Sacha, D.; Kraus, M.; Keim, D.A.; Chen, M. VIS4ML Supplement: Requirements and Goals for Using Visualizations to Assist Machine Learning. 2018. Available online: https://gitlab.dbvis.de/sacha/VIS4ML/-/blob/master/vis4ml-requirements-analysis.pdf (accessed on 5 August 2025).
45. Gilson, O.; Silva, N.; Grant, P.W.; Chen, M.; Rocha, J. Information Realisation: Textual, Graphical and Audial Representations of the Semantic Web. In Proceedings of the I-KNOW '06—Special Track on Knowledge Visualization and Knowledge Discovery, Graz, Austria, 6–8 September 2006.
46. Gilson, O.T. An Ontological Approach to Information Visualization. Ph.D. Thesis, Swansea University, Swansea, UK, 2008.
47. Gilson, O.; Silva, N.; Grant, P.W.; Chen, M.; Rocha, J. VizThis: Rule–based Semantically Assisted Information Visualization. Available online: https://www.researchgate.net/publication/228428180_VizThis_Rule-Based_semantically_assisted_information_visualization (accessed on 5 August 2025).
48. Gilson, O.; Grant, P.W.; Chen, M. Reducing the Complexity of Semantic Data Translation. In Proceedings of the International Semantic Web Doctoral Symposium (ISWDS 2005) at ISWC 2005, Galway, Ireland, 6–10 November 2005.
49. Baudel, T. Browsing through an information visualization design space. In Proceedings of the CHI '04 Extended Abstracts on Human Factors in Computing Systems, CHI EA '04, Vienna, Austria, 24–29 April 2004; pp. 765–766. [CrossRef]
50. Geroimenko, V.; Geroimenko, L. SVG and X3D: New XML Technologies for 2D and 3D Visualization. In *Visualizing the Semantic Web*; Geroimenko, V., Chen, C., Eds.; Springer: London, UK, 2006; pp. 124–133. [CrossRef]
51. Munzner, T. Visualization Analysis and Design. In Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses, Vancouver, BC, Canada, 10–14 August 2025; SIGGRAPH Courses '25. [CrossRef]
52. Zhou, B.; Tan, Z.; Zheng, Z.; Zhou, D.; Savkovic, O.; Kharlamov, E. Towards a Visualisation Ontology for Reusable Visual Analytics. In Proceedings of the 11th International Joint Conference on Knowledge Graphs (IJCKG 2022), Hangzhou, China, 27–28 October 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 99–103. [CrossRef]
53. Heer, J.; Shneiderman, B. Interactive dynamics for visual analysis. *Commun. ACM* **2012**, *55*, 45–54. [CrossRef]
54. Gangemi, A.; Catenacci, C.; Ciaramita, M.; Lehmann, J. Modelling Ontology Evaluation and Validation. In *The Semantic Web: Research and Applications. ESWC 2006*; Sure, Y., Domingue, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4011, pp. 140–154. [CrossRef]
55. Sharma, R. Ontologies, LLMs and Knowledge Graphs: A Discussion. Medium Blog. 2025. Available online: https://nachi-keta.medium.com/ontologies-llms-and-knowledge-graphs-a-discussion-cadeeabe1cc7 (accessed on 16 September 2025).
56. Leida, M.; Du, X.; Taylor, P.; Majeed, B. Toward Automatic Generation of SPARQL Result Set Visualizations: A Use Case in Service Monitoring. In Proceedings of the International Conference on e-Business (ICE-B), Seville, Spain, 18–21 July 2011; pp. 181–186. [CrossRef]
57. Neuhaus, F. Ontologies in the era of large language models—A perspective. *Appl. Ontol.* **2023**, *18*, 399–407. [CrossRef]
58. Lippolis, A.S.; Saeedizade, M.J.; Keskisärkkä, R.; Zuppiroli, S.; Ceriani, M.; Gangemi, A.; Blomqvist, E.; Nuzzolese, A.G. Ontology Generation Using Large Language Models. In *Proceedings of the 22nd European Semantic Web Conference (ESWC 2025), Portorož, Slovenia, 1–5 June 2025*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 321–341. [CrossRef]
59. Rhodes, P.; Kraemer, E.; Reed, B. VisIOn: An interactive visualization ontology. In Proceedings of the 44th Annual ACM Southeast Conference, ACMSE '06, Melbourne, FL, USA, 10–12 March 2006; pp. 405–410. [CrossRef]
60. Xie, L.; Zheng, Y.; Shen, B. Ontology Construction for Scientific Visualization. In Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), Hangzhou, China, 20–24 June 2006; Volume 1, pp. 778–784. [CrossRef]

61. Métral, C.; Ghoula, N.; Falquet, G. An ontology of 3D visualization techniques for enriched 3D city models. In Proceedings of the Usage, Usability, and Utility of 3D City Models (3U3D 2012), Online, 29–31 October 2012; EDP Sciences: Paris, France, 2012. [CrossRef]
62. Sobral, T.; Galvão, T.; Borges, J. VUMO: Towards an ontology of urban mobility events for supporting semi-automatic visualization tools. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1700–1705. [CrossRef]
63. Morshed, A.; Forkan, A.R.M.; Shah, T.; Jayaraman, P.P.; Georgakopoulos, D.; Ranjan, R. Visual Analytics Ontology-Guided I-DE System: A Case Study of Head and Neck Cancer in Australia. In Proceedings of the 4th IEEE International Conference on Collaboration and Internet Computing (CIC 2018), Philadelphia, PA, USA, 18–20 October 2018; pp. 424–429. [CrossRef]
64. Chen, M.; Ebert, D.S. An ontological framework for supporting the design and evaluation of visual analytics systems. *Comput. Graph. Forum* **2019**, *38*, 131–144. [CrossRef]
65. Quinn, J.Y.; Cox III, R.S.; Adler, A.; Beal, J.; Bhatia, S.; Cai, J.; Chen, J.; Clancy, K.; Galdzicki, M.; Hillson, N.J.; et al. SBOL Visual 2.0 Ontology: Symbolic Visual Notation for Genetic Designs. *ACS Synth. Biol.* **2020**, *9*, 1441–1452. [CrossRef]
66. Sevilla, J.; Casanova-Salas, P.; Casas-Yrurzum, S.; Portalés, C. Multi-Purpose Ontology-Based Visualization of Spatio-Temporal Data: A Case Study on Silk Heritage. *Appl. Sci.* **2021**, *11*, 1682. [CrossRef]
67. Sevilla, J.; Samper, J.J.; Fernández, M.; León, A. Ontology and Software Tools for the Formalization of the Visualisation of Cultural Heritage Knowledge Graphs. *Heritage* **2023**, *6*, 4722–4736. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.