Archival Knowledge Graphs: Balancing Semantic Richness and Accessibility of Hierarchical Structures

Remo Grillo[1,†], Lucia Giagnolini[1,†] and Paolo Bonora[1,†]

[1]*Alma Mater Studiorum – Università di Bologna, Bologna, 40125, Italy*

Abstract

Providing visualization and exploration environments for large-scale linked data that work well for non-expert users is especially demanding in cultural heritage contexts, where the user base does not consistently align with technological practices. This paper presents the Semantic Tree Advanced module, a specialized hierarchical navigation component for the ResearchSpace platform addressing performance and usability limitations in browsing RDF knowledge graphs. The standard tree visualization component in ResearchSpace employs synchronous query strategies that become computationally impractical for archival collections containing thousands of entities in complex relationship networks. The module implements dynamic SPARQL query generation, progressive data loading, contextual filtering, and persistent state management to enable efficient navigation while maintaining semantic precision. The architecture balances technical sophistication with familiar file-system metaphors, making sophisticated linked data structures accessible to non-technical users. The module has been tested on the Italian writer Valerio Evangelisti born-digital archive, modeled using BoDi (Born-Digital Archival Description Ontology) – an extension of Records in Contexts Ontology (RiC-O) – comprising approximately 60 million triples, a scale at which rationalizing access becomes essential.

Keywords

Knowledge Graphs, Born-Digital Archives, Data Exploration

1. Introduction

The Semantic Web introduced formal standards for representing, linking, and querying structured information across distributed environments through machine-readable, interoperable formats [1]. Within this framework, Linked Open Data (LOD) technologies employ the Resource Description Framework (RDF) and ontological vocabularies to encode explicit, typed relationships between entities, enabling graph-based representations that support flexible dataset integrations [2, 3]. These capabilities are particularly significant for cultural heritage institutions, where materials are embedded in complex contextual networks. Archives, libraries, and museums have increasingly adopted LOD to interconnect materials across institutional boundaries and to model heritage content and context [4].

Despite the advantages, a critical tension persists between the technical sophistication of Semantic Web technologies and their accessibility. The expressive power of LOD frequently imposes substantial cognitive and technical demands on users, who must understand SPARQL querying, graph structures, and ontology-driven data modeling to fully exploit its potential, yet the primary user communities of cultural heritage institutions (such as historians, literary scholars, students, and members of the general public) rarely possess such expertise, resulting in significant barriers to exploration and interpretation [5, 6, 7, 8]. This accessibility gap undermines the democratic ideals of open data: when only technically trained specialists can navigate linked datasets, the promise of openness becomes effectively constrained. The challenge becomes more pronounced at scale, as small experimental knowledge graphs containing hundreds or thousands of triples can be navigated through basic browsing interfaces or simple visualization tools, while contemporary cultural-heritage LOD initiatives increasingly operate at

IRCDL 2026: 22nd Conference on Information and Research Science Connecting to Digital and Library Science, February 19-20, 2026, Modena, Italy

[†] Remo Grillo is responsible for sections 3 and 4, Lucia Giagnolini is responsible for sections 1 and 2, Paolo Bonora supervised the project and all authors contributed to the conclusions and to the revision of the work.

remo.grillo@unibo.it (R. Grillo); lucia.giagnolini2@unibo.it (L. Giagnolini); paolo.bonora@unibo.it (P. Bonora)

0000-0003-3585-0198 (R. Grillo); 0000-0002-4876-2691 (L. Giagnolini); 0000-0001-8337-3379 (P. Bonora)

the scale of millions of triples, encompassing thousands of interlinked entities within deep hierarchical and relational structures [9, 10, 11]. In these contexts, conventional navigation metaphors break down: visualizations become congested and unreadable, and the semantic richness that makes LOD valuable simultaneously becomes a barrier to meaningful user interaction.

The archive of the Italian writer and activist Valerio Evangelisti (1952–2022) exemplifies these issues. This hybrid archive documents over four decades of activity, with its most substantial component consisting of born-digital materials: 389 floppy disks and three hard drives totaling approximately 2.1 TB, spanning from 1995 to 2022. This digital section documents Evangelisti's continuous experimentation with computing technologies since the 1980s, preserving working folders for novels, study collections, email backups from multiple accounts, software and video game collections, and extensive system files that situate creative production within specific technical environments. The archive's complexity extends beyond volume to encompass multiple contextual dimensions: individual items relate to political networks, academic production, literary creation across draft versions, and technological evolution from floppy disks to contemporary media and formats. Complementing this digital core are analog materials preserved at the Marco Pezzi Archive, including documents, photo albums, manuscript notebooks, and Super 8 reels, alongside approximately 6,000 volumes at the University of Bologna's Classical Philology and Italian Studies library, collectively capturing Evangelisti's intersecting roles as activist, historian, and internationally recognized author.

The semantic modeling of the digital section of Evangelisti's archive produced a knowledge graph exceeding 6 million RDF triples modeled through the Born-Digital Ontology (BoDi)[1], an extension of the Records in Contexts-Ontology (RiC-O). This scale reflects the representational density required to capture how files simultaneously participate in chronological sequences, thematic clusters, and production workflows, with the file system of the main hard drive reaching up to 18 levels of hierarchical depth. While this semantic richness enables granular search capabilities and supports both close reading of individual files and distant reading patterns across the collection, it intensifies the accessibility challenge: navigating 6 million triples demands knowledge of SPARQL querying, comprehension of the underlying ontological model, and familiarity with the archive's structure and contents. The Evangelisti case thus embodies the central problem of rendering large-scale LOD representations cognitively and practically accessible to users lacking specialized training in semantic technologies.

2. Exploring triples through ResearchSpace

In the digital ecosystem, archival description concerns both content representation and how researchers access and interpret data [12]. The challenge lies in rendering knowledge graphs comprehensible through familiar interfaces without losing conceptual coherence and semantic granularity. This requires platforms capable of enabling publication, exploration, and querying of RDF data while bridging semantic sophistication and user accessibility through SPARQL endpoints and visualization interfaces.

ResearchSpace[2] is an open-source platform for managing, exploring, and publishing cultural heritage knowledge graphs. Developed at the British Museum with support from the Andrew W. Mellon Foundation and Metaphacts, it aims "to help establish a community of researchers, where their underlying activities are framed by data sharing, active engagement in formal arguments, and semantic publishing" [13]. Since 2023, ResearchSpace has been maintained by Kartography CIC, with collaborative development through a public GitHub repository[3].

The platform's architecture is based on Semantic Web technologies, using RDF for data representation and SPARQL as the query language. ResearchSpace proposes a conception of knowledge graphs understood as "a continually changing informational structure that mediates between a human, the world and a computer. The graph itself is ontologically based and enhanced by human epistemology" [13]. ResearchSpace has been employed across research, preservation, and cultural heritage contexts. It

[1] http://w3id.org/bodi#
[2] https://researchspace.org/
[3] https://github.com/researchspace/researchspace

has been adopted by networks like LINCS[4] and Pharos[5], archaeological projects (Amara West[6], Deir el-Medina[7]), museums (British Museum, National Archives UK), and artistic/bibliographic projects including Late Hokusai[8], Sphaera[9], Belle da Costa Greene[10] correspondence, VeNiss[11] [14], RePIM[12] [15], and Florentia Illustrata [16].

The system is structured in integrable and configurable modules (templates), each offering specialized functionalities that can be adapted to individual project needs, allowing the combination of analysis, visualization, and curation tools in a customizable environment. Each template is built around one or more SPARQL queries that retrieve data for visualization, which can be defined and modified according to project objectives. The template specifies presentation modalities for extracted data, enabling extensive customization both functionally and graphically. Among these modules, Semantic Search and Semantic Forms enable context-based and faceted searches; Knowledge Patterns provide reusable ontological schemas for adding, retrieving, or modifying complex relationships between entities; Knowledge Maps allow visualization of network connections between actors, places, events, objects, and concepts. Other advanced modules include the Image Viewer for comparing and annotating high-resolution images; Semantic Narratives, which combine text, visualizations, tables, and knowledge maps that automatically update as data changes; dynamic timelines for visualizing temporal events; and geographic data integration with OpenStreetMap[13]. Templates are configurable according to project needs, and the platform enables creation of entirely new modules designed for specific research objectives, which can be shared with the ResearchSpace community for reuse and adaptation.

ResearchSpace was developed and tested adopting CIDOC CRM as the reference ontological framework for data modeling, as to integrate heterogeneous and variable data while preserving their specificity [12]. Recently, the platform has been adopted for testing by the Digital Humanities Advanced Research Centre (/DH.ARC) of the University of Bologna for the management of cultural heritage data projects. The Valerio Evangelisti Archive case study[14] represented a concrete opportunity to test the platform's flexibility beyond its established museum-oriented applications and CIDOC CRM implementations. The project enabled experimentation with the first ResearchSpace implementation based on an archival ontology, specifically BoDi as an extension of RiC-O, effectively allowing evaluation of the platform's capacity to support archival ontological frameworks and verification of its compatibility with ontological models other than CIDOC CRM, for which the system was originally designed. This implementation allowed active participation in platform experimentation while simultaneously evaluating its effectiveness with respect to archival data representation and access objectives at the scale of millions of triples.

3. The Semantic Tree Advanced Component

The interface for accessing data from the digital section of the Valerio Evangelisti Archive in ResearchSpace was designed to translate the conceptual model's requirements into operational navigation tools while addressing the performance and usability challenges inherent in large-scale archival knowledge graphs. The interface adopts a hierarchical structure that mirrors the file system organization (i.e., the way files and folders are naturally arranged on a computer).

This familiar metaphor serves as an intuitive representational tool for born-digital materials while

[4] https://rs.lincsproject.ca/resource/home
[5] https://artresearch.net/resource/About
[6] https://amara-west.researchspace.org/
[7] https://deir-el-medina-dev.kartography.net/
[8] https://latehokusai.researchspace.org/
[9] https://sphaera.mpiwg-berlin.mpg.de/
[10] https://bellegreene.itatti.harvard.edu/
[11] https://veniss.eu/
[12] https://repim.itatti.harvard.edu/
[13] https://kartography.org/researchspace.html
[14] The GitHub repository is available at https://github.com/LuciaGiagnolini12/ValerioEvangelisti_Project. The application is currently under development and not yet publicly available.

enabling the system to fulfill five essential conceptual requirements: (1) representation of digital materials in their physical, logical, and content-based components; (2) documentation of file integrity; (3) valorization and preservation of native metadata; (4) traceability of processual and curatorial provenance; (5) and restitution of multiple contexts, both hierarchical (vertical relationships within the folder structure) and transversal (horizontal connections across different entities of the archive).

From a methodological standpoint, adopting the file system as a representational metaphor for born-digital archival materials reflects not merely an interface convention but the authentic organizational structure through which materials were created and managed. Beyond preserving the original organizational logic, the objective was to propose an expandable tree visualization capable of exposing underlying relationships while supporting exploration pathways that transcend the purely vertical dimension, enabling users to navigate simultaneously along hierarchical axes and through horizontal semantic connections.

Within the ResearchSpace ecosystem, the Semantic Tree module constitutes the predefined tool for representing hierarchical relationships among RDF entities. It is based on SPARQL queries that retrieve resources and their relationships, constructing a tree structure in which each node corresponds to an entity and branches represent relationships of dependency. The basic Semantic Tree offers static visualization: a single initial query retrieves the entire relationship graph, made available through expansion and compression functions. While effective for limited datasets, this approach presents critical performance limitations when applied to complex archival structures, represented through the lens of fine-grained, analytical conceptual models. Moreover, the exploratory dimensions remain strictly vertical. In the Evangelisti Archive case, with its file system reaching 18 hierarchical levels and encompassing thousands of entities across 6 million triples, the synchronous query strategy proved computationally untenable: initial loading times became prohibitive, memory consumption exceeded reasonable thresholds, and system responsiveness deteriorated significantly.

To address these requirements, an extended version, the Semantic Tree Advanced, was developed. The architectural redesign focused on four principal objectives: (1) optimizing query performance through progressive data loading, (2) maintaining semantic precision while reducing computational overhead, (3) preserving navigation state across dynamic view modifications, and (4) extending the navigation possibilities through integrated search functionality and the enhancement of transversal relationship.

The principal innovation consists of progressive and contextual data loading through dynamic SPARQL query generation. Instead of retrieving the entire relationship graph, only the root and first descendant level are displayed at initialization. When users expand a node, the system generates and executes a targeted query to retrieve specifically its child nodes, with each subtree generated exclusively in response to user interaction. This approach fundamentally transforms the performance profile: initial load times are reduced, memory consumption remains bounded regardless of dataset size, and query execution times scale with children per node rather than the entire collection. The architecture implements persistent state management, maintaining node expansion status across filtering and search operations without requiring complete tree reconstruction. The result is a dynamic view of the underlying dataset synchronised with the user's focus.

The Semantic Tree Advanced introduces an integrated contextual search function operating directly on the tree structure while maintaining performance at scale. Unlike traditional searches returning separate result lists, this search filters and reorganizes the hierarchy itself. When users insert a term, the system constructs a SPARQL query that identifies matching nodes, retrieves the complete hierarchical path from root to each match, and reconstructs the tree displaying only relevant branches while preserving structural context. Results are thus highlighted directly within the original hierarchy without modifying navigation logic or requiring complete data reloading.

This design addresses a fundamental limitation of traditional search systems, both archival and general-purpose, which typically return flat result lists separated from structural context. By maintaining results within their hierarchical position, users can simultaneously identify relevant materials and understand their provenance relationships and position within the collection. This contextualized approach significantly reduces cognitive load, enabling immediate assessment of whether a match

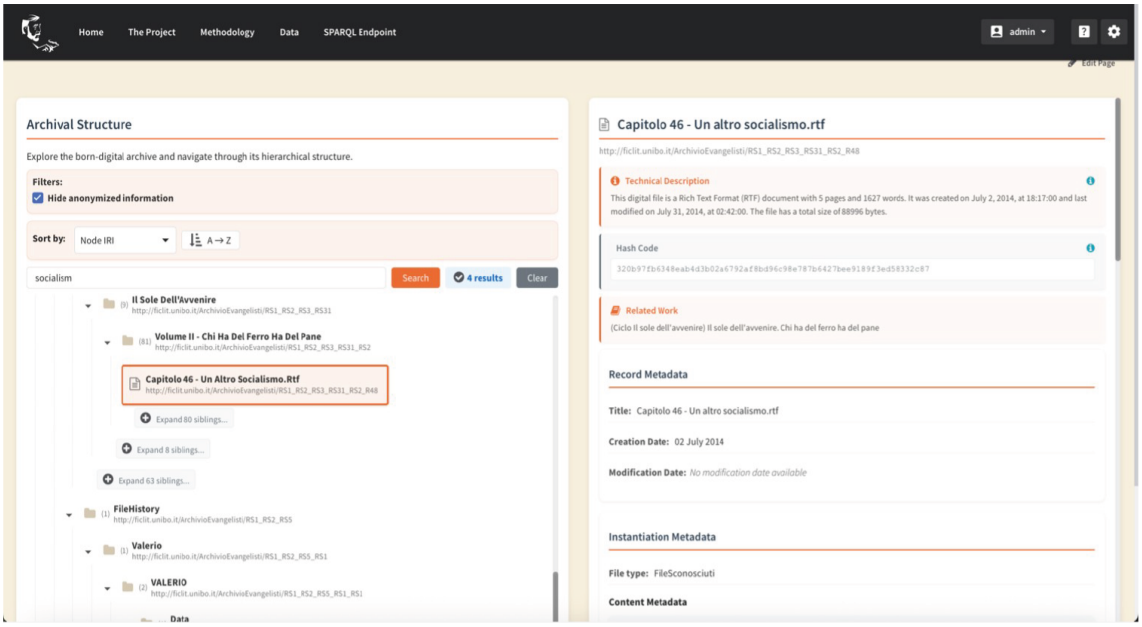

Figure 1: The Valerio Evangelisti Archive project featuring the Semantic Tree Advanced component. Results are shown in their archival context and the tree is "pruned" to focus on results, compressing sibling nodes. On the right, the metadata section highlights information depth for the found resource.

represents an isolated document or part of a larger documentary cluster, without requiring mental reconstruction of archival relationships or separate navigation steps.

More traditionally, the module integrates reversible and combinable filters that modulate displayed nodes through query-level exclusion rather than post-retrieval filtering. Filters modify SPARQL query patterns ensuring excluded nodes are never retrieved from the database, thereby maintaining performance consistency. In the Evangelisti use case, filters were implemented to exclude materials anonymized for privacy reasons, with the system dynamically adjusting queries while preserving descendant counts and structural integrity information. Furthermore, a traditional dynamic sorting functionality was implemented to enable node reorganization according to user-defined criteria (title, creation date, IRI, descendant elements) with operations applied recursively through query-level ORDER BY clauses, ensuring sorting occurs during data retrieval rather than post-processing.

While hierarchical visualization reflects physical-logical ordering, transversal connections restore the complexity of thematic, temporal, or contextual relationships binding documents. The "related nodes" system was indeed introduced to overcome strictly hierarchical navigation through horizontal exploration pathways. Each correlation criterion (e.g. elements with the same creation date, software, hash code, or connected to the same literary work) is associated with a dynamically generated SPARQL query utilizing the selected node's IRI as a parameter. The system identifies semantically connected entities and highlights them within the existing tree view without reconstruction.

From the informational structure perspective, the tree identifies at the highest level a general grouping of born-digital materials, functioning as a container for three principal support typologies: contents of the main computer used by Evangelisti, the external hard drive, and floppy disks copies. Each grouping then replicates the organization of each support according to the file system configuration. Node selection permits visualization of detailed descriptive information including: overview information with content and context descriptions, element counts, dimensions, dates, integrity information, complete lists of technical metadata, and references to corresponding literary works.

This ensemble of technical implementations configures an environment that addresses fundamental performance and usability challenges of large-scale archival knowledge graph navigation, balancing technical sophistication with familiar interface metaphors while maintaining responsiveness at the scale of millions of triples.

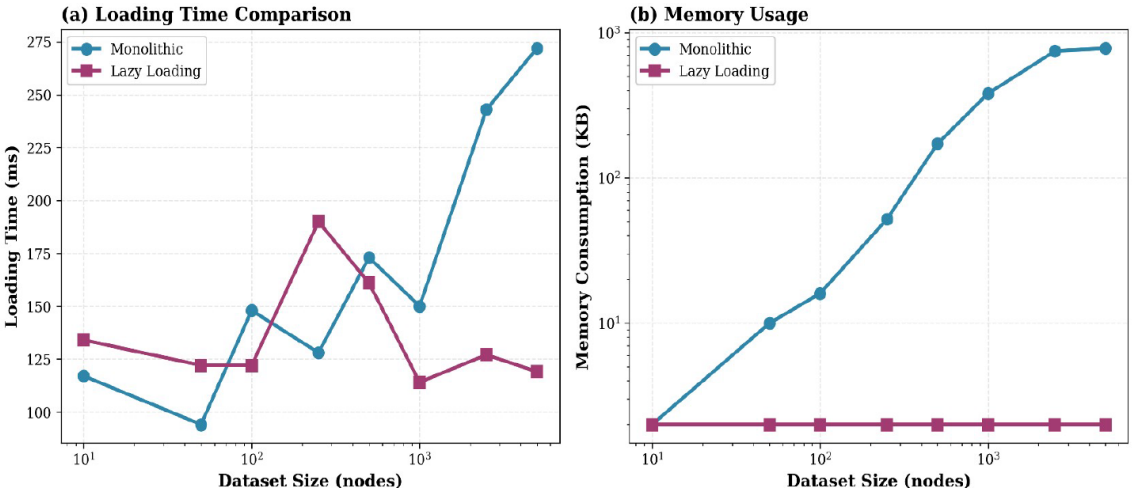

Figure 2: Initial loading time (a) and memory usage (b). For small trees (under ~1000 triples) the two strategies are comparable, but as the number of nodes grows, lazy loading scales with lower latency and constant memory consumption.

The Valerio Evangelisti Archive project is available at: http://evangelisti.dharc.unibo.it/. At present, access to the application is restricted to authorized users with credentials, as the dataset is undergoing a privacy review by the Associazione Evangelisti - Il Sol dell'avvenire. Public access will be granted by the end of 2026, with the dataset published progressively upon completion of each review phase.

4. Results and Discussion

To assess the effectiveness of the architecture for navigating large-scale digital archives, we carried out a series of performance experiments along five dimensions: (1) initial loading, (2) node expansion, (3) contextual search, (4) semantic relationship traversal, and (5) cache behavior. All tests were run on the Valerio Evangelisti Archive data (≈6 million RDF triples) using ResearchSpace 4.0 and Blazegraph 2.1.6, on a typical consumer laptop[15].

4.1. Initial Loading Performance

Figure 2 compares traditional synchronous tree loading with the lazy loading approach of the Semantic Tree Advanced across eight dataset sizes, from 10 to 5,000 nodes. For small trees (10–500 nodes), the two strategies are broadly comparable in terms of latency, with synchronous loading sometimes slightly faster. However, once the tree reaches 1,000 nodes and beyond, the behavior diverges. This trend becomes more pronounced as the tree grows, since the synchronous approach must materialize the entire structure before interaction can begin.

A key property of the lazy loading design is that the initial view always consists of the same root nodes, independent of the total number of nodes available in the graph. In practice, this yields constant-time initialization with an average of 129 ms and a standard deviation of 20 ms. Synchronous loading, by contrast, must process all nodes in advance and therefore exhibits growth that approaches the underlying query limits as it increases. From a user's point of view, this means that the first interaction with the archive remains stable and predictable even as the underlying dataset grows.

4.2. Depth-Independent Node Expansion

We next examined whether node expansion time depends on the depth of the hierarchy, which is crucial for archives with deeply nested structures. Figure 3 reports expansion times for several depth levels.

[15]Apple MacBook Pro M1 Max, 32gb Ram.

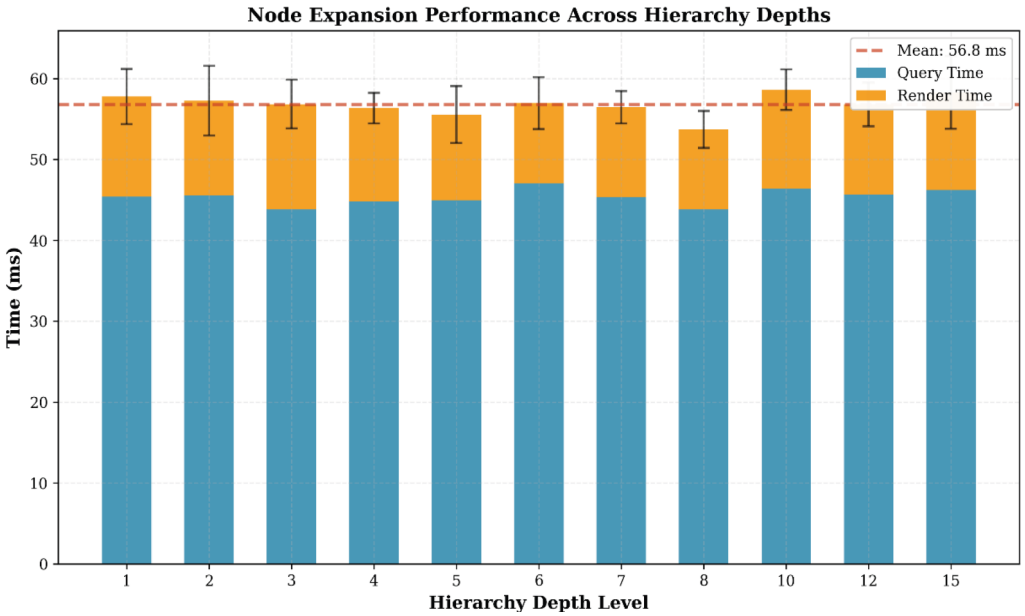

Figure 3: Node expansion time across hierarchy depths. Total expansion latency remains nearly constant (≈57 ms) with low variance, indicating an expansion cost effectively independent of depth.

Across all depths, total expansion time clustered tightly around a mean of 56.78 ms (SD = 1.47 ms), with observed values ranging between 53.70 ms and 58.63 ms. The variance is low (≈8.7%), and supports the classification of node expansion as effectively O(1) (its running time remains constant regardless of the size of the input) with respect to depth. Standard deviations per depth fall between 1.9 ms and 4.5 ms, which further indicates that no particular region of the hierarchy is significantly more expensive to open than any other.

From a user perspective, this means that expanding a node near the root costs almost the same as expanding a node buried deep within the archive. Users are therefore free to organize collections according to archival logic (which often favors deep, nested structures) without incurring a performance penalty at greater depths.

4.3. Contextual Search Scalability

To understand how contextual search behaves as result sets grow, we measured performance for nine search scenarios with result counts ranging from 1 to 502 nodes (Figure 4). The queries spanned from highly specific terms (e.g. an exact filename) to broad patterns (e.g. generic substrings and file-related terms).

All searches were completed successfully, and total search time scaled approximately linearly with the number of results, with an estimated growth rate of α=1.09. Even for the broadest query, which returned 502 results, the end-to-end time remained below 2.5 seconds.

The breakdown of the timings reveals a consistent pattern. SPARQL query execution itself is relatively stable, with a mean of 195 ms (SD = 44 ms) and little correlation with the size of the result set. The main contributor to growth is path reconstruction: reconstructing the hierarchical paths for each result (from the leaf to the root) accounts on average for 69% of the total time, and its cost increases roughly linearly with the number of results. This suggests that the query layer is not the bottleneck; rather, the cost is driven by the amount of navigation context that must be assembled for the user. Nonetheless, the resulting response times remain acceptable even for large sets of hits.

4.4. Horizontal Semantic Relationships

We then evaluated how the system supports exploration of semantic relationships within the archive. Five relationship types of increasing complexity were examined, ranging from simple equality of a hash

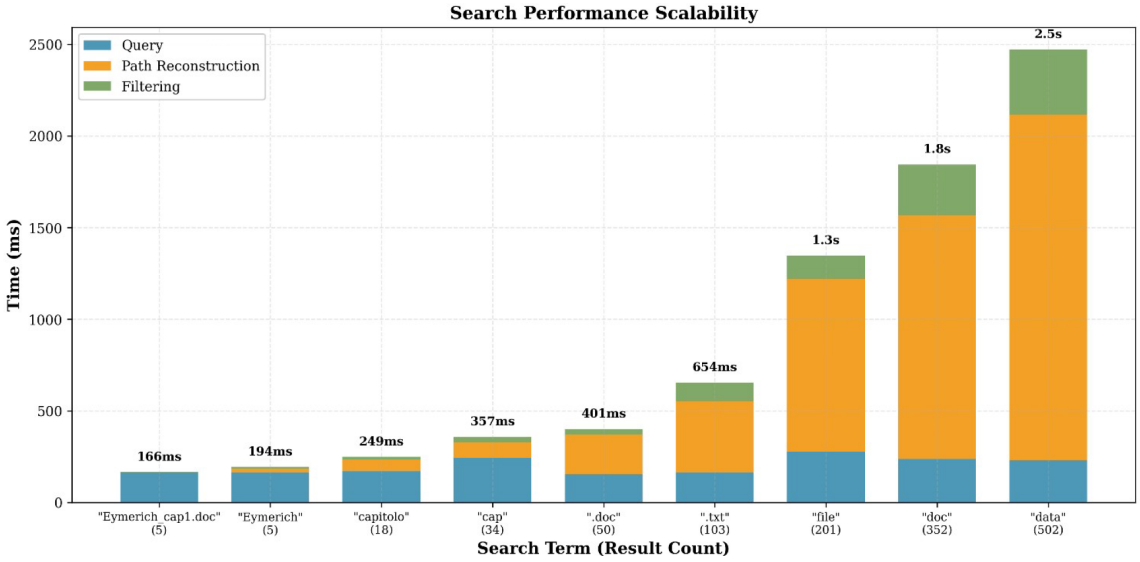

Figure 4: End-to-end contextual search time. Overall latency increases with the number of results and is dominated by path reconstruction, while query time remains relatively stable.

code (same format and file content) to more demanding groupings such as “same software” or “same file extension” (Figure 5).

For the simplest case, detecting items with the same hash code, queries completed in under 400 ms and returned only a handful of related nodes. Moderately complex relationships, such as grouping by literary work, produced dozens of related nodes and required around 2 seconds. The most demanding scenarios, for example “same file extension”, yielded over 300 related items and were completed in approximately 5.65 seconds. Other complex relationships (e.g. “same software”) with 150+ results remained under 5 seconds.

Across these more complex traversals, path reconstruction once again dominates the cost profile, accounting for roughly 63–73% of total time. Importantly, the overhead scales primarily with the number of retrieved nodes rather than with the conceptual complexity of the relationship. From a user perspective, this means that navigating semantic connection (following relations across works, software environments, or file properties) remains feasible and responsive even when the underlying graph is large and semantically rich.

4.5. Cache Effectiveness

Finally, we assessed the behavior of the caching mechanism. Figure 6 presents miss and hit times for cache sizes ranging from 5 to 500 cached expansions. Each configuration was measured five times to average out content-specific effects.

Cache misses, which trigger a full query, showed a very stable latency around 65.24 ms (SD = 3.5 ms) across all cache sizes. This indicates that enlarging the cache does not adversely affect baseline query performance. Cache hits, by contrast, were extremely fast, with an average of 3.56 ms (SD = 0.26 ms). The small dispersion ($\sigma < 0.35$ ms in all scenarios) suggests true constant-time retrieval.

The resulting speedup for cached expansions is substantial. On average, a cache hit is 18.3 times faster than a miss, with a 95% confidence interval between 17.0× and 19.6×. Memory consumption grows linearly with cache size at roughly 4 KB per cached entry, reaching about 2 MB for 500 cached nodes. Even at this stress-test level, memory use remains modest compared to the 782 KB required for a monolithic load of only 5,000 nodes, and the relative advantage grows as archives become larger and navigation more repetitive.

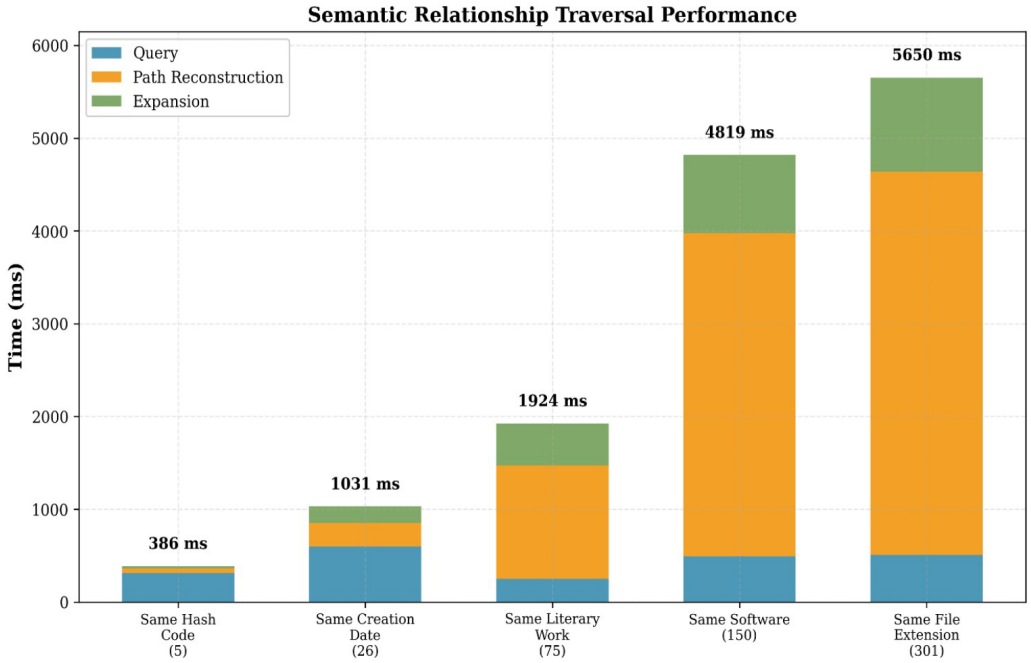

Figure 5: End-to-end performance for five semantic relationship types. Response time increases with the number of related nodes rather than relationship complexity, with path reconstruction accounting for the majority of cost (≈63–73% of total time). Even the most demanding case ("same file extension", 301 results) completes in about 5.7 seconds.

4.6. Architectural Validation

Taken together, the experiments give a coherent picture of the system's architectural behavior. Both the initial lazy-loading phase and subsequent node expansion act effectively as constant-time operations: initialization depends only on a small fixed set of root nodes, and expansion cost is essentially independent of depth. Users can therefore work with very large, deeply nested archives without progressively longer delays as they move through the hierarchy.

The measurements confirm this depth independence quantitatively. Across the tested range of depths, expansion times show only minor variation and a fitted growth exponent close to zero, so depth becomes a modeling choice rather than a performance constrain, an important property in archival domains where hierarchical depth carries semantic meaning.

Search and semantic traversal also remain within interactive time bounds: total search time grows roughly linearly with the number of results, and semantic relationship queries follow a similar pattern, so even broad queries over large result sets remain practically usable rather than being confined to carefully crafted edge cases.

Caching further improves responsiveness in iterative workflows. Revisiting already explored parts of the archive becomes almost instantaneous from the user's perspective, while memory consumption grows in a simple, predictable way with cache size and remains modest compared with synchronous loading.

At the same time, the experiments highlight a clear area for improvement: path reconstruction. In both contextual search and semantic traversal, reconstructing the full hierarchical path for each result up to the root consistently dominates latency. This feature is crucial (it turns raw matches into a clear navigable context) but its cost currently scales with the number of returned nodes. Techniques such as reusing shared path segments, incrementally updating already materialized branches, or batching ancestor retrieval offer promising directions for future optimisation.

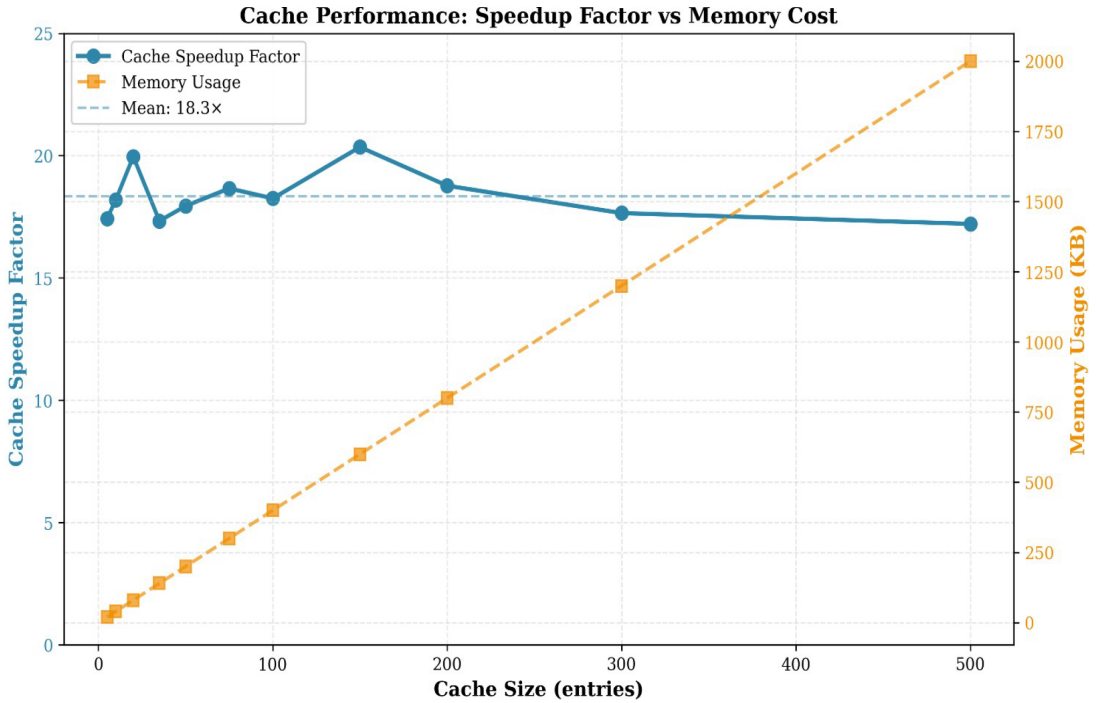

Figure 6: Cache speedup factor and memory usage as a function of cache size. Cache hits remain consistently fast (about 18× quicker than cache misses across all configurations) while memory consumption increases linearly.

4.7. Implications for Digital Archive Navigation

The measured characteristics enable genuinely scalable exploration of huge graphs. Archives with millions of triples can be opened and browsed without noticeable initial delays or browser slowdowns, even when users move across wide portions of the collection. This supports long-form research sessions where the archive is used as a continuous workspace rather than as a static reference.

Second, depth-independent expansion with O(1) behaviour provides robust support for deeply nested structures. Curators can adopt expressive hierarchical arrangements, reflecting provenance, institutional organization, or narrative structure, without worrying that users will encounter sluggish responses when working at greater depth.

Third, the system is designed to support interactive exploration of structured knowledge. Because both contextual search and semantic traversal remain within interactive bounds, users can follow relationships, inspect clusters of related items, and recognize patterns in how materials are connected. The ability to resolve relationship queries over hundreds of nodes within a few seconds makes semantic exploration a routine part of navigation, not an expensive analytical operation separated from the interaction.

Fourth, interaction remains responsive in iterative, back-and-forth workflows. The cache is especially valuable in this context: moving back to previously opened branches typically takes around 3.5 ms, which from the user's perspective feels instantaneous. This kind of responsiveness is important for exploratory research practices, where it is common to refine queries, retrace steps, and compare multiple parts of the archive.

Finally, predictable and bounded memory use ensures that the system remains stable over extended sessions. Since memory consumption does not grow with the total archive size and increases only linearly with the number of cached expansions, users can work for long periods without risking gradual exhaustion of client-side resources.

In combination, these properties show that lazy loading, together with progressive disclosure and caching, is not merely an implementation detail but a key enabling technology. It transforms large-scale digital archives into environments that are not only technically navigable, but also practically and

sustainably explorable in a semantically rich way.

While the Valerio Evangelisti Archive represents a born-digital personal collection, the demonstrated performance characteristics extend beyond this specific domain. Critically, the lazy loading mechanism is ontology-agnostic, operating on any RDF graph structure that encodes hierarchical relationships through parent-child predicates. This architectural independence extends applicability beyond archival contexts to any domain organizing knowledge through hierarchical taxonomies. Within the archival domain specifically, the queries' foundation on RiC-O ensures applicability to diverse archival typologies, such as traditional paper-based archives, institutional records, and hybrid collections, while the scalability achieved with millions of triples inherently validates efficiency on smaller datasets.[16]

5. Conclusions

This work addresses challenges related to the performance and usability of navigating large-scale archival knowledge graphs with a pronounced hierarchical structure. The Valerio Evangelisti Archive project served as a test case to evaluate whether progressive loading strategies can overcome the computational limitations of synchronous approaches while enabling multidimensional exploration.

The module implements dynamic SPARQL query generation through lazy loading, persistent state management, contextual search, and exploration of horizontal relations. This last component represents a critical architectural contribution: while hierarchical visualization reflects physical-logical ordering of born-digital materials, the transversal navigation system overcomes strictly vertical browsing by exposing horizontal connections based on semantic criteria. Performance testing confirms constant initialization times and depth-independent expansion behavior. Importantly, semantic relationship traversal completes within interactive thresholds even when retrieving hundreds of related nodes, demonstrating that horizontal exploration is computationally viable at this scale.

These results validate an approach that balances two complementary navigation paradigms. Vertical navigation provides familiar access patterns aligned with archival provenance, while horizontal navigation exposes complex contextual networks (thematic clusters, temporal groupings, production workflows, technological environments). The integration of these modalities addresses a fundamental limitation of conventional tree visualizations, which constrain users to single descent paths and obscure relational density. By enabling fluid movement between hierarchical and associative exploration, the system supports both systematic browsing and discovery of connections across contextual boundaries.

Addressing the ResearchSpace environment specifically, the archival ontology design and implementation (BoDi/RiC-O) demonstrates the platform compatibility beyond CIDOC CRM. The progressive loading patterns and transversal relationship system provide reusable strategies for hierarchical datasets where semantic connections extend beyond parent-child relationships. For archival LOD applications, the implementation demonstrates that file system metaphors can operate as navigation frameworks while exposing underlying semantic relationships, rendering visible the multiple contexts in which archival materials participate.

The current implementation requires further development. Technical extensions include multi-criteria search, alternative result visualizations, and simultaneous metadata comparison windows. Dataset development requires additional thematic access pathways leveraging horizontal connections, systematic contextual enrichment, and analog materials integration. Critically, formal usability evaluation with domain specialists through task-based protocols remains necessary to validate design assumptions, particularly regarding cognitive effectiveness of combining vertical and horizontal navigation and whether users can exploit semantic connections without explicit training in graph-based thinking.

Although this method was developed and validated within the ResearchSpace environment, its core principles are not platform-specific. Both the methodological approach (progressive loading, stateful

[16]The Semantic Tree Advanced module is currently being tested on the Giuseppe Albini Archive, a collection of a philologist and early twentieth-century politician, which presents different structural characteristics and materials. This ongoing experimentation serves to verify the system's flexibility across varying archival natures, provenance models, and temporal contexts, while also surfacing new feature desiderata emerging from different use cases.

exploration, and the integration of vertical hierarchy with transversal semantic traversal) and key implementation patterns (dynamic query generation, incremental retrieval, and relationship-driven expansion) can potentially be generalized to other graph-based archival interfaces and LOD ecosystems. To consolidate these results, however, performance measurements alone are insufficient: complementary usability evaluations—such as task-based studies with domain specialists assessing learnability, cognitive load, and the practical interpretability of horizontal relations—are still needed and have not yet been conducted.

The work provides empirical evidence that million-triple knowledge graphs can support user-friendly navigation when architectural choices address both performance bottlenecks and multi-dimensional nature of archival relationships. Results establish baseline measurements for progressive loading approaches and demonstrate that transversal navigation across semantic connections can be integrated within hierarchical interfaces without compromising performance. Further work is required to validate generalizability across different archival collections, ontological models, and user communities, and to assess long-term sustainability through extended adoption within the open-source ResearchSpace ecosystem.

Acknowledgments

This research was conducted with materials provided by the Associazione Valerio Evangelisti - Il Sol dell'Avvenire. The authors acknowledge their institutional support in granting access to the Valerio Evangelisti archive, without which this study would not have been feasible. Technical infrastructure for ResearchSpace was provided by the Department of Classical Philology and Italian Studies of the University of Bologna, whose dedicated hosting services enabled the digital platform supporting this investigation. Part of this work is included in the doctoral research project conducted by Lucia Giagnolini and forms a section of her PhD dissertation titled "Rappresentare il digitale d'autore: dal file system al knowledge graph", financed by Next Generation EU, Missione 4, Componente 2, Investimento 3.3 (D.M. 117/2023) CUP J33C22001820002.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Sonnet 4.5 in order to: rephrase, grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (2001) 34–43.

[2] G. Klyne, J. J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004. URL: http://www.w3.org/TR/rdf-concepts/.

[3] D. Allemang, J. Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, 2 ed., Morgan Kaufmann/Elsevier, 2011.

[4] M. Daquino, Linked open data native cataloguing and archival description, JLIS.it 12 (2021) 91–104. doi:10.4403/jlis.it-12703.

[5] H. Vargas, C. B. Aranda, A. Hogan, C. López, RDF explorer: A visual SPARQL query builder, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Proceedings, Part I, Springer, 2019, pp. 647–663.

[6] A.-C. N. Ngomo, L. Bühmann, C. Unger, J. Lehmann, D. Gerber, Sorry, I don't speak SPARQL: Translating SPARQL queries into natural language, in: D. Schwabe, V. A. F. Almeida, H. Glaser, R. Baeza-Yates, S. B. Moon (Eds.), 22nd International World Wide Web Conference, WWW '13, ACM, 2013, pp. 977–988.

[7] P. Bellini, P. Nesi, A. Venturi, Linked Open Graph: Browsing multiple SPARQL entry points to build your own LOD views, Journal of Visual Languages & Computing 25 (2014) 703–716.
[8] J. A. Raemy, R. Sanderson, Analysis of the Usability of Automatically Enriched Cultural Heritage Data, in: F. Moral-Andrés, E. Merino-Gómez, P. Reviriego (Eds.), Decoding Cultural Heritage, Springer, 2024, pp. 1–20. doi:10.1007/978-3-031-57675-1_4.
[9] V. A. Carriero, A. Gangemi, M. L. Mancinelli, A. G. Nuzzolese, V. Presutti, C. Veninata, ArCo: The Italian Cultural Heritage Knowledge Graph, in: The Semantic Web – ISWC 2019, Springer, 2019, pp. 36–52. doi:10.1007/978-3-030-30796-7_3.
[10] V. De Boer, J. Wielemaker, J. Van Gent, M. Hildebrand, A. Isaac, J. Van Ossenbruggen, G. Schreiber, Supporting Linked Data production for cultural heritage institutes: The Amsterdam Museum case study, in: The Semantic Web: Research and Applications, Springer, 2012, pp. 733–747. doi:10.1007/978-3-642-30284-8_56.
[11] L. Klic, Linked open images: Visual similarity for the semantic web, Semantic Web 14 (2022) 197–208.
[12] J. Langdon, Describing the Digital: The Archival Cataloguing of Born-Digital Personal Papers, Archives and Records 37 (2016) 37–52. doi:10.1080/23257962.2016.1139494.
[13] D. Oldman, D. Tanase, Reshaping the Knowledge Graph by Connecting Researchers, Data and Practices in ResearchSpace, in: D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee, E. Simperl (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, 2018, pp. 325–340.
[14] L. Galeazzo, R. Grillo, G. Spinaci, A Geospatial and Time-based Reconstruction of the Venetian Lagoon in a 3D Web Semantic Infrastructure, CEUR Workshop Proceedings 3643 (2024).
[15] P. Bonora, A. Pompilio, RePIM in LOD: Semantic technologies to manage, preserve, and disseminate knowledge about Italian secular music and lyric poetry from the 16th-17th centuries, Umanistica Digitale 6 (2022) 71–90.
[16] E. Andreose, Florentia Illustrata Knowledge Graph, Master's thesis, Università di Bologna, 2025. URL: https://amsacta.unibo.it/id/eprint/8236/.