Exploring Cultural Knowledge Graphs with the TalkingGraphs Framework

Lia Draetta[1,*,†], Flaviana Corallo[1,†], Fabiana Vernero[1,†], Marco Antonio Stranisci[1,†] and Rossana Damiano[1,2,*,†]

[1]*Department of Computer Science, University of Turin*
[2]*Research Center "Digital Scholarship for the Humanities" (DISH), University of Turin*

Abstract
The increasing adoption of Knowledge Graphs (KGs) in domains such as cultural heritage has highlighted the need for intuitive tools that support the exploration of complex, interconnected data by non-technical users. This paper presents TalkingGraphs, a domain-independent framework for interactive KG exploration, designed according to human-centered principles. The framework combines a modular architecture with a configuration-driven approach, enabling the dynamic adaptation of the interface to different datasets without requiring code modifications. The system has been evaluated through a multi-method user study (n=10) involving two distinct domains, combining behavioral analysis, qualitative feedback, and standardized measures of cognitive load and technology acceptance. Results indicate high usability, low cognitive burden, and strong perceived ease of use, while also highlighting the importance of domain-specific adaptations for maximizing perceived usefulness.

Keywords
Cultural KGs, Cultural heritage interfaces, Heritage exploration tools

1. Introduction

The increasing adoption of Knowledge Graphs (KGs) across heterogeneous domains has highlighted the need for effective tools to explore, interpret, and communicate complex relational data. A KG can be broadly defined as a graph-based data structure designed to represent and accumulate knowledge about the real world, where nodes correspond to entities and edges encode the relationships between them. Unlike traditional relational databases, which rely on rigid schemas and tabular representations, KGs provide a flexible and semantically rich model in which the meaning of data is explicitly captured through interconnected structures.

This need is particularly strong in the cultural heritage domain, which requires intuitive visualization environments capable of supporting domain experts in navigating rich, interconnected knowledge without demanding advanced technical expertise. The importance of KGs for cultural data, in fact, lies not only in their expressive power but also in their openness and extensibility. Inherently designed to evolve over time, they are especially valuable in cultural domains, where knowledge is continuously expanded, refined, and reinterpreted through contributions from multiple sources [1]. Cultural data are inherently complex, multidisciplinary, and often distributed across institutions, making them difficult to access and interpret through traditional systems. Effective visualization and exploration tools become essential to unlock the full potential of these interconnected datasets, enhancing both research activities and public engagement [2, 3, 4, 5].

Despite the growing relevance of KGs, existing visualization solutions remain limited in their ability to combine interactivity, usability, and adaptability across different application contexts. This work presents the design and development of a domain-independent framework for building interactive KG

Workshop on Advanced Visual Interfaces for Cultural Heritage (AVICH 2026), June 08, 2026, co-located with the 18th International Conference on Advanced Visual Interfaces, Venice, Italy.
*Corresponding author.
†These authors contributed equally.
lia.draetta@unito.it (L. Draetta); flaviana.corallo@unito.it (F. Corallo); fabiana.vernero@unito.it (F. Vernero); marcontonio.stranisci@unito.it (M. A. Stranisci); rossana.damiano@unito.it (R. Damiano)

exploration platforms. The proposed approach, designed for cultural data and tested with real users, is motivated by a clear gap in the current landscape: the lack of flexible and easily integrable systems that enable intuitive, user-centered exploration of graph-based knowledge. To address this limitation, the framework emphasizes modularity and configurability, allowing the graphical interface to be fully customized through external configuration files, without requiring direct modifications to the source code.

This paper is structured as follows. After reviewing the related work in Section 2, we describe the framework design (Section 3) and architecture (Section 4). The evaluation procedure, materials and results are described in Section 5. Conclusion and future work end the paper.

2. Related Work

The representation and exploration of knowledge in the Semantic Web rely on standards defined by the World Wide Web Consortium (W3C)[1], among which the Resource Description Framework (RDF) provides the foundational data model. RDF represents information as subject–predicate–object triples, enabling flexible, machine-readable descriptions of entities and their relationships, and supporting the integration of heterogeneous and distributed data sources.

KG visualization has emerged as a multidisciplinary field at the intersection of data visualization, human-computer interaction, and semantic technologies. Due to the complexity and scale of graph-based data, visualization tools must go beyond static representations, offering interactive mechanisms for exploration, filtering, and contextualization. Existing approaches can be broadly categorized into tabular views, linked-node visualizations, and visual query composition interfaces.

General-purpose tools aim to support diverse domains and data models. Protégé, for instance, is a widely used ontology engineering environment that provides visualization plugins such as OWLViz and VOWL, although its primary focus remains ontology editing rather than interactive exploration. Other systems, such as Aemoo, enable exploratory search by integrating multiple data sources (e.g., DBpedia and Wikipedia), but are limited by dependency on specific endpoints [2]. Similarly, ARCA [4] supports visual semantic search and incremental graph exploration through components such as Ontodia, though it still requires familiarity with underlying data structures.

In parallel, domain-specific tools have been developed to address particular application needs. Cytoscape [6], for example, supports the analysis of molecular interaction networks in the biomedical domain, while platforms such as ResearchSpace [3] facilitate the exploration of complex cultural heritage data through tailored interaction paradigms. Although these systems improve usability within their domains, they often lack flexibility and reusability across contexts. A similar limitation is observed in the SKATEBOARD interface [5], developed for the World Literature Knowledge Graph [7], whose design remains tightly coupled to its specific domain.

Overall, existing solutions highlight a trade-off between generality and domain specificity: general-purpose tools offer flexibility but limited usability, while domain-specific systems provide tailored functionality at the expense of interoperability. In this context, the framework proposed in this work aims to bridge this gap by combining domain independence, configurability, and user-centered interactive visualization.

3. Talkingraph Design

3.1. Requirements

The design of the proposed framework is informed by a human-centered requirements analysis that prioritizes usability, cognitive accessibility, and interactive exploration of KGs. The full set of requirements is reported in Figure 3. Core functionalities (RF-01–RF-10) support the full exploration workflow, including configurable access to SPARQL endpoints, interactive visualization of entities as nodes, and

[1]https://www.w3.org/TR/owl2-overview/

Group	ID	Description
Functional	RF-01	SPARQL endpoint configuration via external file
	RF-02	Querying via entity-type filters
	RF-03	Entity visualization as interactive nodes
	RF-04	Exploration of relationships between entities
	RF-05	Visualization of relationships as labeled edges
	RF-06	Entity information panel
	RF-07	Screenshot export of the board
	RF-08	Board reset functionality
	RF-09	Export of entities and triples in JSON
	RF-10	Domain-independent configuration
Usability	RNF-01	No need for SPARQL/RDF knowledge
	RNF-02	Minimal interaction steps
	RNF-03	Structured and readable information panel
	RNF-04	Continuous system feedback
Performance	RNF-05	Scalability in node visualization
	RNF-06	Asynchronous query execution
	RNF-07	Smooth rendering of graph elements
Maintainability & Portability	RNF-08	Modular architecture
	RNF-09	External configuration of domain
	RNF-10	Web-based deployment
	RNF-11	Well-structured and documented code

Table 1
Summary of functional and non-functional requirements

incremental expansion of relationships. These features are designed to enable exploratory sense-making through progressive disclosure, allowing users to navigate the graph step by step while avoiding visual overload. From an HCI perspective, usability requirements (RNF-01–RNF-04) play a central role: the interface abstracts underlying semantic technologies, enabling domain experts to interact with the system without prior knowledge of RDF or SPARQL, while ensuring clear feedback, reduced interaction effort, and structured presentation of information. Performance requirements (RNF-05–RNF-07) further contribute to the user experience by guaranteeing responsiveness, smooth interaction, and scalability when dealing with large graph structures. Maintainability and portability constraints (RNF-08–RNF-11) support a flexible and adaptable interaction design through modular architecture and external configuration, enabling consistent user experiences across different domains.

3.2. Interface Design

Based on the analysis presented above the key visualization features selected for the development of the framework are aimed at supporting cognitive efficiency and interactive sense-making.

- Node expansion (progressive disclosure): users can explore the graph by incrementally expanding a node and revealing only its immediate connections. This interaction strategy reduces visual clutter and supports focused exploration.
- Contextual node information (on-demand details): selecting a node displays its associated information in dedicated interface panels, typically with “Show/Hide” controls. This enables users to access detailed data only when needed, reducing cognitive load during analysis.
- Color-based visual encoding: differentiated color schemes are used to represent entity types, supporting rapid visual discrimination and facilitating the interpretation of structural relationships within the graph, while minimizing cognitive effort.

The user interface is designed as a modular and interactive environment that supports the exploration of KGs through coordinated visual and interaction components. It is organized into three main areas (Fig. 1): a **results panel**, a central **visualization space**, and a **contextual information panel**, complemented by a **control bar** for query and system actions.

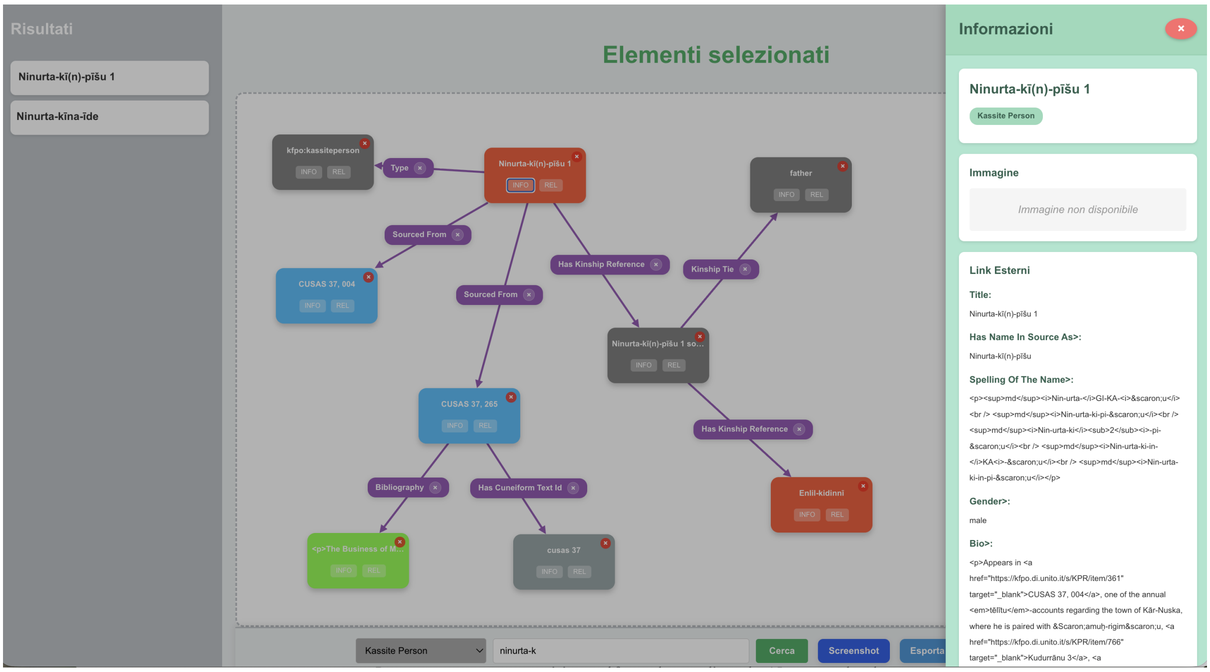

Figure 1: The TalkingGraphs interface (displayed here in Italian). The results panel, on the left, shows the list of the search results; the expanded graph portion, with nodes of different colors according to the entity type, is visible in the central visualization area; the control bar is located below the visualization area; the contextual information panel showing information about the selected entity is located in the right column.

The *results panel*, located in the left column of the interface, enables users to search and filter entities through a combination of textual input and category selection, supporting efficient navigation across large datasets. The selected entities are then displayed in the central *visualization area*, located in the central area of the interface, where they are represented as nodes connected by labeled edges that encode semantic relationships. This area implements a node-link paradigm with interactive features such as drag-and-drop manipulation and incremental expansion of connections, allowing users to progressively construct subgraphs of interest. The design adopts a color-coding strategy to visually distinguish entity types, facilitating rapid pattern recognition and reducing cognitive effort. Interaction is centered on direct manipulation: users can select nodes, trigger the exploration of related entities, and dynamically extend the graph through stepwise navigation. This incremental approach embodies the principle of progressive disclosure, preventing information overload while maintaining access to the richness of the underlying data. The *contextual information panel*, located in the right column of the interface, provides detailed metadata about selected entities, including properties, external links, and, when available, associated media resources, thereby enriching the exploratory process with additional layers of information. This panel is not visible at the start of the session and appears in the right column when the user requests detailed information about a selected entity. The *control bar*, visible in the bottom panel of the interface, offers auxiliary functionalities such as search execution, screenshot capture, data export, and workspace reset, supporting both analytical and documentation tasks.

From an HCI perspective, the interface combines continuous feedback, intuitive interaction patterns, and clear visual affordances to support a user-centered exploration paradigm, making complex KGs accessible and interpretable for both expert and non-expert users.

4. System Architecture

The proposed framework adopts a modular, domain-independent architecture designed to support the interactive exploration of KGs through a clear separation of concerns. A key element of the framework is the use of external configuration files to ensure domain independence and flexibility.

4.1. Configuring Talking Graphs

The system relies on a structured YAML configuration[2] that defines all domain-specific aspects, including the SPARQL endpoint, entity types, visual properties, and query-related parameters. This configuration is loaded and validated in the backend using FastAPI and Pydantic libraries [3], which ensure type safety and consistency of the parameters, while SPARQLWrapper [4] is employed to dynamically construct and execute queries against the configured endpoint. This approach decouples the application logic from the data model, allowing the same platform to be reused across heterogeneous domains without modifying the source code.

In particular, entity types are described through configurable attributes such as labels, colors, ontology classes, and associated properties to be displayed in the interface. Additionally, the configuration file specifies namespace prefixes and query parameters, enabling the dynamic construction of SPARQL queries tailored to the target dataset. From an HCI perspective, this mechanism supports adaptability while preserving consistency in the interaction model: users interact with a stable interface, while the underlying domain representation can evolve transparently. This design choice not only improves maintainability and portability but also enables rapid prototyping and integration of new datasets, making the framework suitable for exploratory and interdisciplinary scenarios.

4.2. Modular Design

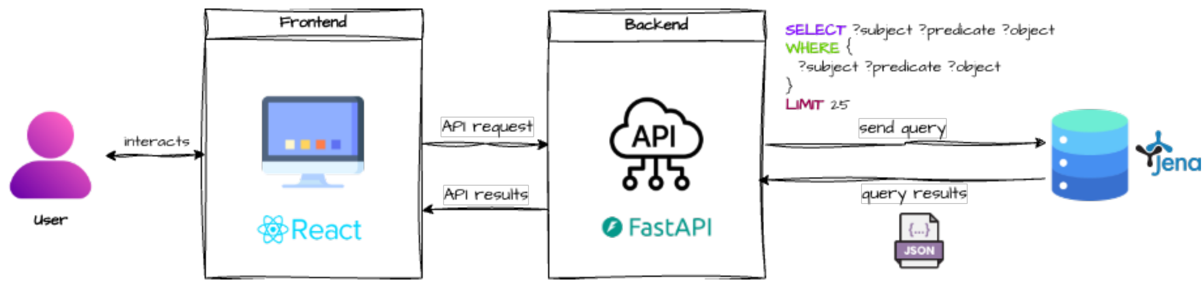

Figure 2: Schematic representation of the data flow in TalkingGraphs

The system follows a client–server paradigm structured into three main layers: *Frontend*, *Backend*, and *Data layer*, which interact through well-defined interfaces to ensure flexibility, scalability, and maintainability. From a data flow perspective (see figure 2), user interactions in the frontend (e.g., searching for an entity or expanding a node) trigger API requests to the backend. The backend translates these requests into SPARQL queries, sends them to the triplestore, and processes the returned results. The formatted data are then sent back to the frontend, where they are rendered as interactive graph elements. This asynchronous pipeline ensures that the system remains responsive even when handling complex queries.

At the presentation layer, the **Frontend** is implemented as a web-based interface using *React*[5], responsible for managing user interaction and visualizing the graph. The interface communicates with the backend exclusively through RESTful API calls, ensuring that all data access and processing logic is abstracted away from the user. This design supports a responsive and dynamic user experience, where visual updates are triggered by asynchronous data retrieval and state changes.

The application logic is handled by the **Backend**, developed in Python using FastAPI. This layer acts as an intermediary between the frontend and the KG, encapsulating the complexity of query construction, data validation, and response formatting. A key feature of the backend is the dynamic generation of SPARQL queries based on user input and configuration parameters, as illustrated in

[2]https://yaml.org/
[3]FastAPI https://fastapi.tiangolo.com/; Pydantic https://docs.pydantic.dev/
[4]https://github.com/RDFLib/sparqlwrapper
[5]https://react.dev

Section 4.1. Queries are constructed using templates that incorporate prefixes, entity types, and search criteria defined in an external configuration file. The backend executes these queries through the SPARQLWrapper library and processes the results into a simplified JSON format suitable for frontend consumption; data validation and serialization are managed through Pydantic models.

The **Data layer** consists of an RDF triplestore, such as Apache Jena Fuseki[6], which hosts the KG and exposes a SPARQL endpoint. The triplestore is responsible for storing, querying, and reasoning over structured semantic data represented as RDF triples. By isolating this layer from the frontend, the architecture ensures that different datasets or endpoints can be integrated without affecting the user interface or application logic. Thanks to the use of an external YAML configuration file to store the SPARQL endpoint, query parameters, and namespace prefixes, the same system can be deployed across multiple domains by simply adapting the configuration, thus supporting reuse and rapid prototyping

5. Evaluation

5.1. Test domains

The evaluation was conducted on two different cultural domains, namely, the study of ancient societies and the exploration of todays' scientific literature, to capture a diversity of data and methods, as detailed in the following.

Kassite Prosopography (KP): A prosopographical dataset derived from cuneiform documents and encoded as a KG, representing the network of socio-economic relationships within the Kassite kingdoms of the Babylonian region (14th–13th centuries BCE) [8]. This domain contains 541 entities of type Person connected to each other by relations of various type (administrative, commercial, and family relations), sourced from 1879 cuneiform documents described by 150 bibliographic entries.

REO: A KG representing the academic research ecosystem, capturing relationships between university personnel (e.g., professors, PhD students, and researchers), their scholarly outputs, associated projects, and public engagement activities. The current implementation relies on a test subset derived from IRIS data from the Department of Computer Science at the University of Turin. The REO KG is composed of a total of 69,645 triples and includes entities from the following main classes: Person, Organization, Project, Work, Public Engagement Initiatives, and Topics.

5.2. Methodology

The evaluation of the proposed framework was designed to assess usability, effectiveness, and user acceptance through a multi-method approach combining behavioral observation, qualitative feedback, and standardized quantitative measures. The study involved 10 participants, recruited through personal connections (convenience sampling) and distributed across the two application domains: REO (Academic research involving 7 users, all affiliated with university administrative offices across different departments, who regularly work with research data and tools for data visualization and summarization) and KP (prosopographical data, 3 users: a graduate student, a post-doc researcher, and a senior researcher), allowing the evaluation to capture both domain-independent properties and context-specific differences.

The *evaluation protocol* was structured in three progressive phases. First, exploratory tasks allowed users to freely interact with the interface, enabling the identification of spontaneous navigation strategies and potential usability issues. Second, scenario-based tasks required participants to achieve realistic information goals without explicit instructions, simulating real-world usage conditions. Typical tasks included searching for a specific person and their relations with other persons in the prosopographic domain or looking for a researcher's role and involvement in projects in the REO domain. Finally, structured tasks with predefined success criteria were used to measure effectiveness, efficiency, and error rates.

[6]https://jena.apache.org/documentation/fuseki2

Task	REO		KP	
	Completion	Time	Completion	Time
1	100%	9s	100%	10s
2	100%	10s	100%	9s
3	100%	30s	100%	30s
4	100%	46s	100%	47s
5	57%	53s	100%	58s
6	57%	62s	100%	65s
7	100%	9s	100%	10s
8	100%	5s	100%	4s
9	100%	5s	100%	4s
10	100%	5s	100%	4s

Table 2
Task completion and time

Dimension	Mean score
Mental demand	6.6
Physical demand	2.5
Temporal demand	4.2
Performance	16.4
Effort	6.8
Frustration	2.4

Table 3
NASA-TLX results

To investigate user experience in more depth, we also relied on further data sources: (i) behavioral data collected during task execution, (ii) qualitative insights gathered through think-aloud protocols and open-ended questions, and (iii) quantitative measures obtained through NASA-TLX to estimate cognitive load [9] and TAM to investigate technology acceptance [10].

5.3. Usability, Performance, and User Experience

The behavioral results indicate a highly positive usability profile of the system (Table 2). Almonst all participants successfully completed all assigned tasks without requiring external assistance, demonstrating a high degree of learnability and autonomy even at first use. Task completion times were consistently low, typically ranging from a few seconds to about one minute depending on complexity, confirming the efficiency of the interaction model and the clarity of the navigation structure. A minor exception was observed in tasks related to keyword-based search and result exploration, where a subset of participants adopted alternative search strategies (e.g., expanding search results). However, these deviations did not prevent successful task completion and were attributed more to ambiguity in task formulation than to interface design issues.

Qualitative feedback collected through open-ended questions further supports these findings. Participants consistently described the system as intuitive and accessible, even without prior experience with KGs. The progressive, relation-driven navigation model —based on incremental expansion of graph structures— was particularly appreciated for enabling exploratory reasoning and supporting understanding of semantic relationships.

However, differences emerged between the two domains. Users in the KP domain reported a high level of satisfaction and considered the system already suitable for their exploratory needs. In contrast, REO users expressed a more critical perspective: while recognizing the conceptual value of the framework, they highlighted the need for additional functionalities to fully integrate it into their everyday workflows.

5.4. Cognitive Load and Technology Acceptance

The analysis of NASA-TLX results (Table 3) provides further insight into the cognitive aspects of the interaction [9]. The overall workload profile indicates a low cognitive burden, characterized by low levels of physical demand and frustration, and moderate levels of mental effort consistent with first-time use of a new system. Importantly, the performance dimension achieved the highest score, indicating that users perceived themselves as highly successful in accomplishing the tasks, in line with the observed task completion rates.

The TAM results complement this picture by highlighting a strong perceived ease of use (mean ≈ 5.95/7), with low variability across participants, indicating consistent agreement on the accessibility of the system. Perceived usefulness, while still positive (mean ≈ 4.57/7), showed higher variability, reflecting differences in domain requirements and user expectations. In particular, some REO users expressed concerns about the system's ability to immediately improve their workflow compared to existing tools, whereas KP users reported a clearer alignment between system capabilities and their needs.

Taken together, these findings suggest that the framework achieves a strong baseline in usability and user acceptance, with particular strengths in ease of use and exploratory support. At the same time, the variability in perceived usefulness highlights the importance of domain-specific adaptation and further functional development to maximize adoption across different contexts.

6. Future Work and Conclusion

This paper presented TalkingGraphs, a domain-independent framework for the interactive exploration of cultural KGs, designed according to human-centered principles. The evaluation results highlight strong usability, low cognitive load, and effective support for exploratory tasks, confirming the suitability of the approach for both expert and non-expert users.

Future work will focus on extending the framework along two main directions. First, we plan to validate the system across additional cultural domains, in order to further assess its generality and adaptability. Second, we aim to incorporate new functionalities identified during the evaluation, such as path search and advanced result filtering, to better support complex analytical tasks and improve integration into domain-specific workflows. Overall, these developments will contribute to strengthening the framework's applicability and effectiveness as a flexible tool for the exploration of heterogeneous cultural KGs.

Declaration on Generative AI

The authors used AI tools in order to: 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] V. A. Carriero, A. Gangemi, M. L. Mancinelli, L. Marinucci, A. G. Nuzzolese, V. Presutti, C. Veninata, Arco: The italian cultural heritage knowledge graph, in: International semantic web conference, Springer, 2019, pp. 36–52.

[2] A. G. Nuzzolese, V. Presutti, A. Gangemi, A. Musetti, P. Ciancarini, Aemoo: exploring knowledge on the web, in: Proceedings of the 5th Annual ACM Web Science Conference, 2013, pp. 272–275.

[3] D. Oldman, D. Tanase, Reshaping the knowledge graph by connecting researchers, data and practices in researchspace, in: International Semantic Web Conference, Springer, 2018, pp. 325–340.

[4] E. Bernasconi, M. Ceriani, M. Mecella, T. Catarci, Design, realization, and user evaluation of the arca system for exploring a digital library, International Journal on Digital Libraries 24 (2023) 1–22.

[5] E. Bernasconi, D. Di Pierro, D. Redavid, S. Ferilli, Skateboard: semantic knowledge advanced tool for extraction, browsing, organisation, annotation, retrieval, and discovery, Applied Sciences 13 (2023) 11782.

[6] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks, Genome research 13 (2003) 2498–2504.

[7] M. A. Stranisci, E. Bernasconi, V. Patti, S. Ferilli, M. Ceriani, R. Damiano, The world literature knowledge graph, in: International Semantic Web Conference, Springer, 2023, pp. 435–452.

[8] R. Damiano, C. Mattutino, C. Finocchiaro, E. Devecchi, Querying prosopographic datasets: a case study on ancient near eastern persons in cuneiform sources, in: Proc. of the Workshop on

Advanced Visual Interfaces and Interactions in Cultural Heritage (AVI-CH 2024), volume 3798, CEUR-WS, 2024, pp. 1–9.
[9] S. G. Hart, L. E. Staveland, Development of nasa-tlx (task load index): Results of empirical and theoretical research, in: Advances in psychology, volume 52, Elsevier, 1988, pp. 139–183.
[10] F. D. Davis, A. Granić, N. Marangunić, The technology acceptance model: 30 years of TAM, Springer, 2024.