An AI-based Approach and Platform for the Preservation and Exploitation of Knowledge on the History of Computing (short paper)

Stefano Ferilli [1] Liudmyla Matviichuk [2] and Carla Petrocelli [3]

[1] *Università di Bari – DIB, Via E. Orabona 4, Bari, 70125, Italia*
[2] *Department of Informatics and Computing Tools, T. Shevchenko National University "Chernihiv Collegium", 53, Hetmana Polubotka Str., Chernihiv, Ukraine*
[3] *Università di Bari – DIRIUM, Piazza Umberto I 1, Bari, 70121, Italia*

Abstract

There is an urgent need for preserving and making available the knowledge related to the history of computing, for research and education purposes. This is a peculiar kind of Cultural Heritage, since it tightly mixes hardware, software, documental and even immaterial heritage. The interlinks among these items and their context is fundamental to properly understand them and their role. Advanced AI techniques can support this vision and open unprecedented opportunities to the researchers, practitioners and hobbyists. We are pursuing these objectives in a project based the GraphBRAIN platform for Knowledge Graphs management.

Keywords

Knowledge Graphs, History of Computing, Knowledge Representation and Reasoning

1. Introduction & Motivations

The word 'computing' refers to the science and technology of information processing and the industry dedicated to these topics. Everything that revolves around the modern 'computer' is an artifact, just as an archaeological find. The computer embodies in its own technical nature the same characteristics of change and the same speed of technological innovations that have led to its realization: in addition to its intrinsic design idea, it becomes a historical source as a 'cultural object'. From this connection of the technical object with its own context comes a first declination of the relationship between history and computer, the 'history of computer science', understood as a study of the evolution of computing machines and the automatisms for data processing and the operations performed on them. On the other hand, from the perspective of human history, technology has very quickly permeated, and contributed to the evolution of, our way of life; however, technological incarnations, especially 'modern' digital ones (hardware, software, applications), have such a short life cycle that their rapid obsolescence can cause loss of their knowledge. The generation of inventors/ pioneers of computer artifacts is gradually fading away, causing treasures of know-how to sink into oblivion. It is therefore urgent to react and create a heritage in which computer science is presented in its entirety and not necessarily linked to other sciences (mathematics, physics, chemistry), to serve as a permanent reference and a core of resources to learn, to understand, to see and to wonder, and to witnesses the importance it has in our society. Its roots must be known, and proper tools must be provided to understand it and the keys to interpret it.

Usually, catalogs related to the history of computing are just lists of items belonging to a given collection as if they were built based on inventories. In addition, they mainly focus on hardware,

[1] 1st Italian Workshop on Artificial Intelligence for Cultural Heritage (AI4CH22), co-located with the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022). 28 November 2022, Udine, Italy.
EMAIL: stefano.ferilli@uniba.it (A. 1); matviychuk2012@gmail.com (A. 2); carla.petrocelli@uniba.it (A. 3)
ORCID: 0000-0003-1118-0601 (A. 1); 0000-0002-2046-6153 (A. 2); 0000-0002-6009-3806(A. 3)

which is visible and tangible, neglecting the invisible and volatile heritage of software and technical documents, that are essential both by themselves and for understanding the hardware. Even should the cataloging form include the most detailed and comprehensive information that characterizes the object, it would be just a digitized mirror copy of the old paper archives[2]. In such a setting, the use of the most cutting-edge technology involved in the procedures of cataloging an object does not expand the knowledge of the object itself, it just makes it easier to consult the catalog.

The heritage also remains strongly localized: the information, however correct, rich, and exhaustive, can be consulted only limited to the database it belongs to. This thwarts the possibility of having a wider knowledge around the described object, based on relationships that could for example explain if there are similar pieces produced by another company, designed by another research group, or related to other objects, due to similar characteristics. This can be overcome only through a transition to a unified database, in which all the information relating to the object belonging to the historical heritage is collected, and from which all this information can be derived.

From the computational technologies that have revolutionized the field of archival sciences, we expect a radical change that enables a more effective management of cultural heritage: sharing information (which is not simply a union of catalogs) implies a transformation of the represented object as part of a system of knowledge around it, that enriches it and connects it to the information coming from its content and context. For instance, the object 'punched card', whose localized cataloging can provide the simple information that it is an element used in textile machines from 1801[3], if extended with relations about its other uses over time, can be connected to mechanical musical instruments[4], to mechanical calculating machines[5], to the tabulators used in 1890 for the US census [8], and even to the first electronic computers and programming languages. This transforms punched cards from simple paper elements into media, used both to store information and to transmit data, but also into a tool to switch to a computer 'the code' which allows a transformation of the data in the expected results. In our example the object 'punched card' is connected to objects of other types: a craftsman (Jacquard), a document (the user guide for the operation of the mechanical piano), a scientist (Charles Babbage), a company (IBM producing tabulators), an electronic computer (ENIAC), programming languages, etc. These paths make explicit the roles played by the protagonists in the evolutionary stages of the object itself, but also outline the historical steps that led to the change of certain paradigms in the history of computing. This new perspective of sharing and inter-relating data contributes to the growth of 'knowledge' related an object, so as to describe it in all its complexity, and radically changes the approach to querying the object being cataloged, but also the storytelling associated with it.

This requires a change of paradigm with respect to the past. We believe it is necessary to:

1. Deconstruct the traditional record-based approach and move to a description in which all the entities involved in a description 'live' with their own dignity and can be related to each other, rather than being just field values (author, title, etc.) in the records.
2. Widen the scope of the description from a fixed set of fields describing each item to a larger and more variable set, including aspects that were so far neglected by the research and practice, such as physical, content, context, and even usage aspects.
3. Enable advanced support provided by AI tools to help the different kinds of users in carrying out their activities and accomplishing their tasks, in a personalized and pro-active way.

We also believe that this requires different solutions than those currently proposed in the literature, that may boost the effectiveness of data management so as to support the needs of different kind of users, providing them new possibilities for data exploitation.

This paper describes our project aimed at proposing a vision for long-term preservation and advanced exploitation of knowledge on material and immaterial cultural heritage, specifically in the field of the history of computing, and methodological and technological solutions to implement it.

---

[2] Even considering the national standards for the management of technological and scientific archives (e.g., see [6]), the impression is that these tools are used with the sole purpose of making data usable without making explicit the complex network of knowledge they preserve.

[3] Joseph Marie Jacquard made his loom 'programmable', making it possible to create the pattern of a fabric on the basis of a pattern stored on a support (list of punched cards) that can be replaced and always precisely reproduced [4]. [5]

[4] Under a sort of piano keyboard there was the punched board, and under the latter, several strings of the musical instrument. A mechanism dragged the punched board under elements that allowed the percussion of the strings only at points where the board had holes.

[5] One thinks of Charles Babbage's *Analytical Engine*, well described in the 1842 article by Ada Lovelace where, in addition, the first program to make the machine calculate Bernoulli's progression numbers is presented [3].

Compared to previous work, here we propose an expanded, 'holistic' data schema, and a more systematized list of kinds of automated reasoning to be applied to the available knowledge.

2. Related Works

The main projects undertaken in the past in the directions we envisaged, have tried to overcome the reported limitations, without however fully responding to the needs set out above. This is the case of the French 'collaborative and participatory' project for the realization of the Musée de l'Informatique et du Numérique (MINF), launched following an agreement between academic, socio-economic and associative structures, signed at the symposium *Vers un Musée de l'Informatique et de la Société Numérique en France*, held in 2012 [22]. On this occasion, it also emerged the urgency of keeping track of tools related to computer sciences of which, given their specificity in terms of identification and speed of obsolescence, there is a risk of losing knowledge In order to define an identifying historical heritage, the project was conceived as a network of physical spaces for the preservation, dissemination and promotion of IT tools. These spaces, distributed in different *Lieux* on the French territory, are however always linked to temporary or permanent exhibitions of museums located on the national territory, which are therefore not shared outside the country's borders. More recent projects concern digital archives that deal with the cataloging of artifacts of artistic, historical, and cultural value and have integrated their material using experiential feedback, the result of interactions on social media platforms. Among these, the one with European relevance is SPICE (Social Participation, Cohesion, and Inclusion through Cultural Engagement), which aims to produce, collect, interpret and archive the proposals, reactions and responses of users interacting with these heritages, with the objective of capturing citizens' calls to rethink the nature of the computational infrastructures that support data management [1,4].

On the technological side, data networking is known in AI for being the core of *knowledge*. So, we call for a step up from the Data Base (DB) perspective to the Knowledge Base (KB) – more specifically, Knowledge Graph (KG) – one. The research on KGs carried out in Knowledge Representation (KR) proposed solutions for representing and storing knowledge that have departed from the mainstream solutions for DBs. The established representation standard for formal ontologies is the Ontology Web Language (OWL), and the associated data storage technology, triplestores, adopts the RDF graph model, based on triples (*Subject*, *Predicate*, *Object*) of atomic (Uniform Resource Identifiers – URIs – or literal) values. In the DB community, significant success has been obtained by a new graph-based NoSQL technology, based on the Labeled Property Graphs (LPG) model, that allows to associate sets of attribute-value pairs and labels to nodes and arcs. Thus, the LPG model is more expressive than (and incompatible with) the RDF one. We believe that data representation and storage must rely on state-of-the-art DB technology, in order to ensure optimization and efficiency in data storage and handling, and that the research in KR may provide solutions for effectiveness in data usage. So, we propose to adopt LPG-based DBs for data storage and basic handling, and formal ontologies as data schemas. Some works tried to investigate this combination, but they mostly focus on applying OWL solutions to LPGs, at the cost of not fully exploiting the power of LPGs ([16-19]) or of proposing non-standard extensions of RDF [20,21]. We call for an LPG-centric approach that can fully exploit the features of this model. A solution for this is the GraphBRAIN framework [14], and associated tools for schema and instance handling [15].

3. Technological Platform

For our project we adopted GraphBRAIN, a general-purpose KB management system aimed at covering all stages and tasks in the lifecycle of a KB, from knowledge acquisition, to knowledge organization, to knowledge exploitation. It brings to cooperation an graph DBMS for efficiently handling, mining and browsing the individuals, with an ontology level that defines the DB schema. As in relational DBs, and differently from standard KGs, the schema is kept separate from the data. This allows to superimpose different ontologies/schemas on one graph, representing different views on the same data. Some classes and relationships may appear in different ontologies, possibly with different attributes, in order to reflect different perspectives on them. This allows cross-fertilization among, and

knowledge reuse across, different domains: individuals of shared classes act as bridges, allowing the users of a domain to reach information coming from other domains. The ontologies are built and maintained by GraphBRAIN's administrators, while instances are fed into the KB collaboratively by the users, or by automatic knowledge extraction from documents and other kinds of resources (e.g., the Internet). The functionalities of GraphBRAIN are exposed as services, and external applications can use GraphBRAIN through an API ensuring that all accesses to the DB and operations on its content are compliant with the data schema.

The data are stored in Neo4j [10], that implements the LPG model: nodes represent entity instances and arcs represent instances of binary relationships, whose type is specified by their labels, and whose attributes are specified by their properties. Neo4j is schema-less; in GraphBRAIN the ontologies allow to associate a clear semantics to the graph items, and enable high-level reasoning on the available knowledge. They express what the DB can store and how it is structured, so that only data that are compliant to the ontologies may be added to the graph. They drive and support all functionalities: KB creation and enrichment; advanced tools for searching and browsing the KB; automated reasoning, mining, analysis and knowledge extraction tools that may be used interactively by end users or provided as services to other systems for obtaining selective and personalized access to the stored knowledge. While the ontologies are described in a proprietary XML format purposely designed for the LPG model, GraphBRAIN can also import OWL ontologies and/or individuals, export its KGs to OWL, so as to allow application of existing Semantic Web tools on them, or publish KB content as linked open data (LOD) [13], to make it interoperable with other resources.

GraphBRAIN can manage attachments for each instance. In this way it also acts as an archive, whose content is indirectly organized according to formal ontologies, and thus may foster interoperability with other systems. Finally, users may add comments, or approve/disapprove, each entity or relationship instance, and even each single attribute value thereof. Using the comments, the users may also provide suggestions to improve and extend the ontology. Through the approval/disapproval mechanism, the system may establish a trust mechanism for the users that supports ‘distributed’ quality assurance on the content of the KB. Users are encouraged to provide high-quality knowledge, because using a combination of their number of contributions and trust they are assigned ‘credits’ that they may spend in using advanced features provided by GraphBRAIN. Interactions of users are tracked in order to build models of their preferences to be used for personalization purposes.

A Web application was developed to allow users interaction with the KGs. It provides form-based interfaces (automatically generated from the ontology specification) for feeding or querying instances of entities and relationships in the KB[6], and a graph view where a selected portion of the instances can be graphically displayed and subsequently explored, expanded or shrinked, and the details of instances can be shown. This is useful to browse the available knowledge without a pre-defined goal in mind, but letting the data themselves drive the search. This also enables serendipity in information retrieval, since the users may find unexpected information that is relevant to their information needs. The displayed portion of the graph can be selected based on the result of a specific user query or automatically as a connected neighborhood of the most relevant nodes or, if a user model is available, based on statistics collected about his previous interaction with the system, the starting nodes may be those more related to his interests, preferences, aims, background, etc. The possibility of translating selected portions of the graph into natural language is also envisioned.

GraphBRAIN can export its KGs (ontologies and instances) to several different formats, enabling several kinds of automated reasoning, including:

- Associative reasoning (finding indirect connections between items, extracting personalized and relevant portions of the graph, etc.), carried out by the graph DB manipulation language and libraries;
- Ontological reasoning (inheritance, consistency, etc.), carried out by OWL reasoners;
- Logical multi-strategy (deduction, abduction, abstraction, induction, argumentation, probabilistic inference, analogy) reasoning carried out by a Logic Programming-based inference engine;

[6] A demo Web Application is available at http://193.204.187.73:8088/GraphBRAIN/

- Analytical reasoning (clustering items, spotting anomalous or exceptional situations, identifying regularities, assessing node relevance or centrality, predicting links, etc.)

Some of the underlying algorithms are reused from the literature; others have been extended or purposely developed. Specific AI research is being carried out to develop an integrated framework in which all these kinds of reasoning can be tightly combined, not just exploited separately.

Figure 1 shows the form-based and graph-browsing sections in the Web application.

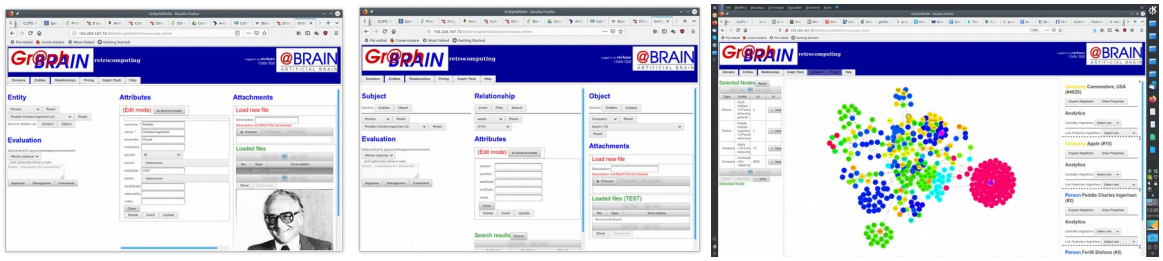

Figure 1: GraphBRAIN Web application interface

4. Data Schema / Ontology for the History of Computing

Following the work in [23], we propose here an approach to knowledge representation that considers and brings to cooperation many different aspects of the cultural heritage items:

- Formal, including the metadata used in traditional records;
- Physical, including materials, processing and mechanics;
- Content, of various kind: textual (if applicable), visual, logical, conceptual (interested in the meaning conveyed by the items);
- Context, adding information that is external to the cultural heritage items proper, but that may be useful or relevant to properly understand it;
- Lifecycle, including process and usage data, useful for personalization purposes.

We call it a *holistic* description approach. The classes in our ontology include:

- **Award**: any kind of recognition to persons, companies, devices, documents, or components (including educational attainments, prizes and records);
- **Collection**: any conceivable grouping of items (e.g., groups of persons, series of documents, families of devices or components);
- **Organization**, including companies and institutions;
- **Part**: a part, useful or needed to build a Device but not providing a high-level (i.e., perceivable or meaningful for a final user) functionality on its own. It has many subclasses, including electronic or electric or mechanical components, boards, etc.;
- **Configuration**: a group of Devices, relevant because typical or determined in order to satisfy specific needs (e.g., a configuration of devices for desktop publishing);
- **Device**: an artifact having some kind of use at the human level of interaction. Among its many subclasses, the most relevant concern computers, calculators, peripherals, etc.;
- **Document**, including printable, audio, video and multimedia types, each with a corresponding hierarchy of subclasses;
- **Event** (conferences, fairs, shows, lectures, etc.);
- **IntellectualWork**: the original result of an intellectual effort, relevant for methodological or practical purposes (including algorithms, approaches, inventions, programming languages, disciplines, technologies, theorems, theoretical models, etc.);
- **Item**: a specific, identifiable specimen of a (mass-produced) object (e.g., a device or component or document or software or system);
- **Package**: a specific packaging of a Device (or of a set of devices sold together);
- **Person**: reporting personal data about persons;
- **Place** It is the root of a hierarchy currently made up of several subclasses, describing geographical, administrative, and other kinds of places;

- **Software** (with a hierarchy of subclasses, such as Development, Educational, Embedded, OfficeAutomation, OperatingSystem, Videogame);
- **System**: a group of Devices that is functional only as a whole (different from a Configuration, where at least one of the Devices would be functional if taken alone).

Moreover, classes **Category** and **Word** allow, respectively, to conceptually or lexically tag all other items, and to connect them semantically, since they are interlinked in the KB.

Sample relevant relationships include:

- Document.concerns.{Concept,Component,Device,Document,Person,Software,...}
- Device.wasIn.{Event,Place}
- Device.clones.Device, Component.clones.Component, Software.clones.Software
- Software.compatibleWith.Software, Device.compatibleWith.Device
- Software.requires.Software
- {Item,Component,Device,Document,Person,Software}.belongsTo.Collection
- {Person,Organization}.owns.Device
- Person.developed.{Component, Device, Document, IntellectualWork}
- Component.mayReplace.Component
- Person.interactedWith.{Device,Person,System}
- Word.describes.{Concept,Component ,Device,Document,Person,Software,...}

The resulting graph will allow indirect, non-trivial connections between the represented items. E.g., it might allow to discover that a person who patented a component was at the same show as an employee of a company using that component in a device, which might explain why that company used that component. Other examples of opportunities provided by this conceptualization include the possibility of recording anecdotes told by the original players of the computer revolution, or technical information that can be precious to restore items or to run obsolete software, which cannot be expressed in existing ontologies designed for other kinds of cultural heritage.

Table 1
Current content of the Knowledge Base on the history of computing

	Data points (e)	Instances (e)	Attribute values (e)	Data points (r)	Instances (r)	Attribute values (r)	Total
Computing	8565	1699	6866	3747	2080	1667	12312
Overall	2424578	336617	2087961	538118	496679	41439	2962696

Table 1 reports statistics on the current content of our KB, by type of information. The history of computing section, built collaboratively, includes 1699 entity instances and 2080 relationship instances, described by 6866 and 1667 attribute values, respectively. The rest, which is the vast majority, consists of context information (concepts, words, places, etc.) added partly automatically and partly collaboratively. This includes the WordNet lexical ontology [12], the standard part of the Dewey Decimal Classification (DDC) system [11], the ACM Computing Classification System (CCS) [24], and the IEEE thesaurus and taxonomy [25].

5. Conclusions and Future Work

We stressed the urgent need for preserving and making available the knowledge related to the history of computing, for research and education purposes. This is a peculiar kind of Cultural Heritage, posing several challenges since it tightly mixes hardware, software, archival/bibliographic and even immaterial items. Storing the interrelationships among the items and between items and their context is fundamental to properly understand them. KRR techniques from AI can support this vision and open unprecedented opportunities to the researchers, educators, practitioners and hobbyists. We started a preservation project based on the GraphBRAIN platform for Knowledge Graphs management. In this paper we described its setting and the functions currently provided. Ongoing and future work aims at expanding the KB and the set of AI-based functions provided in the platform.

6. References

[1] E. Daga, L. Asprino, et al., Integrating Citizen Experiences in Cultural Heritage Archives: Requirements, State of the Art, and Challenges in Journal on Computing and Cultural Heritage, vol. 15. n. 1, 2022, pp. 1-35.

[2] L. Heide, Shaping a technology: American punched card systems 1880-1914 in IEEE Annals of the History of Computing, vol. 19, no. 4, 1997, pp. 28-41, 1997.

[3] L.F. Menabrea, Sketch of the Analytical Engine Invented by Charles Babbage, Esq., trans. Augusta Ada Byron King, Countess of Lovelace, in Scientific Memoirs, vol. 3, 1843, pp. 666-731.

[4] D. Otero, P. Martin-Rodilla, J. Parapar, Building Cultural Heritage Reference Collections from Social Media through Pooling Strategies: The Case of 2020's Tensions Over Race and Heritage. Journal on Computing and Cultural Heritage, vol. 15, n. 1, 2022, pp. 1-13

[5] C. Petrocelli. The Art of Weaving a Code: The Jacquard Loom, the Analytical Engine, and Women's Work. In E. Vavarella (ed.), *rs548049170_1_69869_TT*, pp. 110-117, Mousse Publishing, Milano, 2020.

[6] R. Rojas, U. Hashagen, The ENIAC: History, Operation and Reconstruction in VLSI in *The First Computers: History and Architectures*, MIT Press, 2002, pp.121-178

[7] J.E. Sammet, Programming languages: history and future. Communation of the ACM, 15(7): 601-610, 1972

[8] L.E. Truesdell, *The Development of Punch Card Tabulation in the Bureau of the Census*, U.S. Government Printing Office, 1965, pp. 35-50.

[9] F. Vannozzi. Catalogare il patrimonio scientifico e tecnologico: da sic a sts a pst, storia di un percorso (e di una collaborazione). In: Pratesi, G., Vannozzi, F. (eds.) *I valori del museo. Politiche di indirizzo e strategie di gestione*, pp. 98-101.

[10] I. Robinson, J. Webber, E. Eifrem. Graph Databases, 2nd edn. O'Reilly Media, 2015

[11] M. Dewey. A classification and subject index for cataloguing and arranging the books and pamphlets of a library. Amherst, Mass., 1876

[12] G.A. Miller. Wordnet: A lexical database for english. Communications of the ACM 38, 39-41, 1995

[13] T. Heath, C. Bizer. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers, 2011

[14] S. Ferilli. Integration Strategy and Tool between Formal Ontology and Graph Database Technology. Electronics, ISSN 2079-9292, 10:2616, 27 pp., MDPI, 2021

[15] S. Ferilli & D. Redavid. The GraphBRAIN System for Knowledge Graph Management and Advanced Fruition. In *Foundations of Intelligent Systems*, LNAI 12117, 308-317, Springer, 2020

[16] H. Chiba, R. Yamanaka, S. Matsumoto. G2GML: Graph to Graph Mapping Language for Bridging RDF and Property Graphs. In The Semantic Web – ISWC 2020, pp. 160–175, Springer, 2020

[17] https://protegeproject.github.io/owl2lpg (consulted on 14 October 2022).

[18] https://github.com/SciGraph/SciGraph/wiki/Neo4jMapping (consulted on 14 October 2022).

[19] https://github.com/VirtualFlyBrain/neo4j2owl (consulted on 14 October 2022).

[20] https://github.com/cmungall/owlstar (consulted on 14 October 2022).

[21] Hartig, O. Foundations to Query Labeled Property Graphs using SPARQL. In Joint Proceedings of the 1st International Workshop on Semantics for Transport and the 1st International Workshop on Approaches for Making Data Interoperable, Central Europe (CEUR) Workshop Proceedings vol. 2447, CEUR-WS.org, 2019.

[22] "#MINF_POUR UN MUSÉE DE L'INFORMATIQUE ET DU NUMÉRIQUE EN FRANCE" report, 2015. https://project.inria.fr/minf/files/2011/12/MINF_Rapport_HD.pdf (consulted on 14 October 2022)

[23] S. Ferilli, D. Redavid. An Ontology and a Collaborative Knowledge Base for History of Computing. In *Proceedings of the 1st International Workshop on Open Data and Ontologies for Cultural Heritage (ODOCH-2019)*, Central Europe (CEUR) Workshop Proceedings vol. 2375, 49-60, 2019

[24] https://dl.acm.org/ccs (consulted on 14 October 2022).
[25] https://www.ieee.org/publications/services/thesaurus-thank-you.html (consulted on 14 October 2022)