An ontology and a collaborative knowledge base for history of computing

Stefano Ferilli
Department of Computer Science
University of Bari
Bari, Italy
stefano.ferilli@uniba.it

Domenico Redavid
R&D Department
Artificial Brain S.r.l.
Bari, Italy
redavid@abrain.it

Abstract

In recent years, the history of automatic computing devices and associated technologies has attracted more and more interest. The area is quite peculiar, very complex, very young, and involves rapid obsolescence. So, research practices and tools for handling it from a cultural heritage perspective are still immature. Still missing, and urgently needed, is an ontology that categorizes and systematizes the concepts of interest. Also, there is a pressing need to collect, preserve and make available precious knowledge that is at risk of going lost. It is currently scattered across many people spread all over the world.

As a contribution to this area of research, this paper proposes: an ontology for computing devices and their history, the first version of a knowledge graph for this field, and GraphBRAIN, a general-purpose tool to design and collaboratively populate knowledge graphs, that also provides advanced consultation and analysis tools. It may be used as an intermediate layer to provide services to end-user applications aimed at personalized fruition of cultural heritage, also in a touristic perspective.

1 Introduction

While Computer Science and Engineering, in the form we know them today, are quite young areas in the landscape of human knowledge, their incredibly rapid advances in the last decades, and the relevance computers gained in every aspect of our lives, recently raised significant interest in the study of the history of computing and in the preservation of knowledge and artifacts related to it. So, computing is now not just a means to support and foster all activities related to cultural heritage (which is widely known as *Digital Cultural Heritage*), but it is becoming the very object of study and fruition as cultural heritage, as well. Museums and private collectors started popping up all around the world, and some items were sold for values similar to those of works of art by famous artists (e.g., one of the surviving units of Apple I, the first computer developed by Apple, was sold for $ 375,000; one of the prototypes of Commodore 65 — a computer which was never released —, endowed with a — probably unique — expansion board, was sold for about $ 95,000; some of the first models of personal computers, produced in thousands units, are sold for several thousand dollars). Also, many events (conferences, seminars,

In: Carlo Meghini, Antonella Poggi (eds.): Proceedings of ODOCH 2019 – Open Data and Ontologies for Cultural Heritage – Rome, Italy, 3 June 2019, published at http://ceur-ws.org

shows) on 'vintage computing' have been organized, attracting scholars, researchers, amateurs, collectors or other kinds of enthusiasts. Italy is playing a leading role among the countries that are active in this field.

The ongoing excitement for this topic is leading to the discovery, retrieval and preservation of huge amounts of precious information, also including in the loop the people that by first hand contributed to the development and to the milestones of the field. So, this is an unprecedented opportunity, and one that probably will soon vanish and never happen again, to collect, safely store and sensibly organize all this wealth of information, so that it may be passed on to future generations for research, study or education purposes. Unfortunately, collecting such information is problematic, for two reasons. First, the field is so new that there is no established research and scholarship yet. Second, knowledge in this field is spread across many persons, each perhaps knowing just part of the story, or specialized only on some aspects of it. What is worse, technology in this field suffers from extremely rapid obsolescence, and thus new practitioners tend to ignore important technological information needed to understand and handle items of just a few years earlier. Leveraging the enthusiasm of practitioners in this field, a possible solution would be to provide a (set of) tool(s) allowing collaborative building and enrichment of a knowledge base covering all aspects of the history of computing. In a wikipedia-like perspective, the motivation to share knowledge would be the possibility of using also the information contributed by other people.

However, a collaborative approach in which many people, with different expertise, culture, background and perspective contribute small pieces that together make up the big picture, cannot reach successful results unless suitable knowledge representation schemes to organize the knowledge in this field are available and shared. Thus, a preliminary, fundamental need, is for identifying such schemes. Unfortunately, due to both the intrinsic complexity of the field, and to its peculiarities, the currently available resources, developed in the cultural heritage landscape, are patently inappropriate. Hence, there is a pressing need for the definition of a new, specific scheme, to be shared and reused by all the stakeholders involved in this area of interest.

The original contributions of this paper to tackle these problems are (1) the definition of a formal ontology for the history of computing, and (2) the first introduction of GraphBRAIN, an on-line tool to collaboratively design, build and maintain knowledge bases. The latter was used to define a first version of the former, to serialize it in Web Ontology Language (OWL), and to build a first version of the knowledge graph.

GraphBRAIN is a general-purpose knowledge base management system aimed at covering all stages and tasks in the lifecycle of a knowledge base, from knowledge acquisition, to knowledge organization, to knowledge exploitation. It provides the administrators of the knowledge base with a tool to build and maintain the ontologies that will act as schemes for the knowledge base, and to export them in standard Semantic Web formats. It uses the defined general and domain-specific ontologies to drive and support all other functionalities. Knowledge is stored in a graph structure, on which many querying, exploration and mining algorithms can be applied. The content of the knowledge base may be published as linked open data (LOD) [7]. Finally, in addition to the tools for building, maintaining, and enriching the knowledge base, GraphBRAIN also provides its users with a set of advanced tools for searching and browsing the information stored in the knowledge base, and a set of 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.

This paper is organized as follows. After discussing (some of) the peculiarities and complexities of the domain of automatic computing in the next section, the ontology for the history of computing is presented in Section 3. Then, Sections 4 and 5 briefly describe the features and interactive interface of GraphBRAIN, and its current content. Finally, after reporting some related work, Section 7 concludes the paper and outlines future work issues.

2 Complexity of the Computing Domain

Computing-related items are clearly different, and stand apart from all other cultural heritage items, due to several peculiarities, including (but are surely not limited to) the following:

- devices (and sometimes components) involve, actually are symbiotic with, *software*, in such strict a relationship that either may be meaningless without the other (e.g., devices may require software to be able to do anything; part of the software is often hard-wired to the machine, in the form of ROM code; software is meaningless if suitable devices are not available to load and run it);
- *however*, software has very different features from hardware, and thus requires specialized concepts and relationships to be fully and properly described (e.g., source code, libraries, support);
- since devices, parts, and software are usually *mass-produced*, (sometimes complex) production-related information is to be handled (e.g., the 6502 microprocessor was produced by 3 different companies; the Amiga computer was produced by different companies across his lifespan; the C64 ALDI was produced only by

USA manufacturing facilities of Commodore), and this holds both for the overall device and for its single parts;

- *but* all instances of a given product, albeit "mass-produced", are not necessarily identical: while maintaining the same product code, they may have sometimes slight, but very important, external or internal differences in appearance or components, depending on the production site or period, on the marketing country, etc. (e.g., the Commodore PET was labeled CBM in France, or the Commodore VIC series was labeled VC in Germany, due to the acronyms being unpleasant words in those countries; computers sold in Europe were equipped with parts tailored for producing and using a PAL video signal, while computers sold in the USA were equipped with corresponding parts designed for a NTSC video signal);
- *and* sometimes, especially concerning very early objects or recent restorations or reproduction of old objects, devices, parts and objects are prototypes or handcrafted ("*homebrew*"), which requires completely different descriptions (e.g., concerning the manufacturing processes and materials);
- devices (and software) are *internally* composite and complex (made up of several parts), and their composition is an intrinsically relevant feature, both from a formal perspective (e.g., the schematics of a device are fundamental to understand its inner working, and maybe to appreciate its 'technical beauty' — say, the use of ingenious solutions to save board space or to reduce the number of parts) and from a practical one (e.g., the layout of parts on an Apple I board is extremely well-organized and balanced — it might have been different while using the same schematics);
- devices (and software) are also *externally* composite and complex (they involve secondary objects, such as manuals and packaging), and the external components are often as important as the main device (e.g., the box may feature art graphics by a well-known artist, or the manual may be fundamental to know how to use the device);
- even single parts of a device, albeit considered 'atomic' from a functional perspective and for cataloguing purposes, may be structurally compound (and possibly complex) objects, so it is not easy to capture and/or describe their features (e.g., what is the 'stuff' of which a transistor is made?);
- there may be different *versions* or 'flavors' of a device, part, software or even secondary objects (e.g., there are at least 10 different versions of the C64; early versions of the manual for the Commodore CP/M cartridge include schematics that were removed from later versions; early versions of the Simons' Basic report "Simon's" on the label, due to an error in spelling the author's surname);
- devices and software, while interesting by themseleves, sometimes are fully operational, or make full sense, only in combinations, which we may call *systems*; so, it is interesting to store knowledge about systems, as well (e.g., typical systems, systems purposely set up for specific purposes);
- even when not parts of a system, devices and software are not isolated objects; they may connect to each other, and there are very strict and sometimes subtle constraints for hardware and software *compatibility* (e.g., some software would not work with some versions of some parts in the devices; some peripherals cannot be used with some computer models — sometimes connectors are different, sometimes the problems are subtler);
- in addition to the hardware-software compatibility between computers and peripherals, there are also issues of compatibility among different models and among different versions of the same model (e.g., the Commodore VIC-20 is compatible with the C-64 only for BASIC programs, not for ASSEMBLY programs, due to the different memory maps; not all hardware and software for the C64 computer can be used with the newer C64c version released in the 90s, albeit they are generally considered as being equivalent machines);
- devices and systems may have different *configurations*, concerning both hardware and software (e.g., different amounts of storage, different expansion boards, different operating systems or applications, different peripherals); this requires an ability to express limits for the configurations (e.g., maximum RAM storage), or the configuration of specific items owned by a private, company or organization;
- devices (and systems) are for use, not just for show; as a consequence, true preservation must involve keeping them operational, which in turn requires to store knowledge about how to operate them, how to diagnose faults and how to fix them; while much of this knowledge is in books, manuals and data sheets, but as long as original parts are not available anymore, information about if, when and how different parts can replace original ones may be precious;
- in addition to this, repairing or personalizing functionality may have required sometimes so peculiar, ingenious, strange or anyhow noteworthy solutions that the tricks and restoring or personalization process are themselves of interest and should be recorded;
- both devices and their parts have (possibly complex) functionality, which should be described as well;

Table 1. Statistics on the content of the 'Computing' and 'General' ontologies

Ontology	Main Classes	Subclasses	Attributes	Relationships	Attributes
Computing	15	97	111	117	21
General	17	27	79	88	23

– there is no standard, nor precise categorization for devices (and sometimes for parts), which makes it difficult to establish a generally agreed taxonomy and to assign stable attributes to super- and sub-classes; some devices are so peculiar that considering a 'class' for them seems too much;
– additionally, some devices have multiple functionalities, which makes it hard to properly assign them to classes;
– the development of all kinds of objects, even seemingly simple ones, involves many contributors, at different levels and with different roles (e.g., chief architect, design team, authors of documentation, designers of external cases, etc.), and it would be unfair to record just the main person in charge of the development as the only author.

Some of these peculiarities are shared with other kinds of modern technological instruments, especially electronic ones, but computing-related devices take the complexity to the extreme. Given the lack of a specific knowledge representation scheme for this field, and the inappropriateness of existing schemes developed for other branches of knowledge, we set up to defining an ontology for this field.

Also, the ontology for computing devices must use entities coming from other, both general and domain-specific, ontologies. Indeed, in addition to representing knowledge about items, any knowledge base aimed at supporting research, education and, more in general, spread of awareness about a given topic, cannot ignore all the context-related information that provide background to the items, often explain and justify them, or connect them to the more general knowledge. E.g., one must also consider the need for representing people, companies, intellectual works, records and firsts, historical events, etc. Moreover, the raising interest in the history of computing, along with an increase in awareness about its relevance, also started a demand for fruition of this kind of cultural heritage items. Thus, also knowledge that is specific to cultural heritage and tourism should be connected to the strictly technical and historical part of the representation.

3 An Ontology for Computing Devices and the History of Computing

Based on the motivations reported in the previous section, we developed an ontology for systematizing knowledge about computing devices, their history and, more in general, their background (e.g., events and people that were relevant in the development of the field). It currently includes 112 classes, 117 relationships and 132 attributes, some of which are domain-specific, while some others may be considered as borrowed from other (general or domain-specific) ontologies (see Table 1 for overall statistics). In the following we will review its main components, presenting them in an informal and intuitive style.

The top-level classes, and their immediate subclasses (if any), are the following (a short description is provided when not obvious).

– **Award**: any kind of recognition that can be awarded to, or record that can be marked by, persons, companies, devices, documents, or components. It has 3 subclasses:
  - **Education**: associated to (more or less formal) educational levels (e.g., B.Sc., M.Sc., PhD, etc., but also certifications, etc.).
  - **Prize**: awards formally granted (usually by some institution);
  - **Record**: the recognition of being the first or the best in doing something;
– **Collection**: any conceivable grouping of items. At the moment 4 specific kinds of groupings are considered, corresponding to subclasses:
  - **Persons** (e.g., families, teams, etc.).
  - **Devices** (e.g., families of electronic devices).
  - **Documents** (e.g., series, archives, etc.).
  - **Components** (e.g., families of electronic components).
– **Company**, currently used to represent both companies and institutions, corresponding to 2 subclasses of this class.
– **Component**: 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 11 subclasses, the definition of most of which should be intuitive:

  - **AuxiliaryBoard**: a PCB that is separate from the main board but still necessary to the proper functionality of a device (e.g., a PCB collecting RAM chips).
  - **Capacitor**
  - **CoProcessor**: a dedicated processor for specific kinds of tasks (e.g., graphics, sound, etc.).
  - **ExpansionBoard**: a PCB that can be added to a device to add non-fundamental functionality to it (e.g., an internal modem).
  - **EPROM**: an Erasable Programmable Read Only Memory chip.
  - **MainBoard**: the PCB representing the motherboard of a device (usually, the one on which the microprocessor is located).
  - **MicroProcessor**: a processor devoted to carrying out the main tasks in a computer.
  - **PROM**: a Programmable Read Only Memory chip.
  - **RAM**: a Random Access Memory chip.
  - **Resistor**
  - **ROM**: a Read Only Memory chip.
- **Configuration**: a relevant group of *Device*s, relevant because typical or determined in order to satisfy specific needs (e.g., a configuration of devices for desktop publishing);
- **Device**: a manufact having some kind of use at the human level of interaction. It starts a hierarchy of 54 subclasses, of which the following are immediate subclasses:
  - **Calculator** (5 subclasses);
  - **Computer** (5 subclasses);
  - **InputDevice** (8 subclasses);
  - **OutputDevice** (3 subclasses);
  - **InputOutputDevice** (3 subclasses);
  - **StorageDevice**: a device that provides storage functionality, either by itself or by handling some kind of external storage medium (10 subclasses);
  - **StorageMedium**: a storage support, that can store information but lacks the machinery to actually store and read it (6 subclasses);
  - **NonComputers**: devices that, while not being computers, have a strict relationship to computers, or have played a role in the history of computers and of their development (6 subclasses).
- **Document**, in its most general definition as "something that serves as evidence or proof". As such, it is not limited to printed documents (or documents that might in principle be printed, such as a PDF or word-processor file), but also includes audio-video recordings. It has currently 13 subclasses:
  - **Advertisement**, **AudioRecording**, **Book**, **Booklet**, **Card**, **Leaflet**, **Letter**, **Magazine**, **Manual**, **Movie**, **Picture**, **Postcard**, **Poster**
- **Event** (6 subclasses),
  - **Conference**: a meeting with mainly research or educational purposes.
  - **Fair**: a convention mainly oriented towards selling products and commerce.
  - **Show**: a convention mainly oriented towards showing new products.
  - **Concert**
  - **Lecture**
  - **Historical Event**: any significant event that should be recorded (e.g., the unveiling of a new product, the discovery or invention of a new technology, etc.).
- **IntellectualWork**: the original result of an intellectual effort, relevant for methodological or practical purposes (9 subclasses)
  - **Algorithm** (e.g., Quicksort);
  - **Approach** (e.g., Step-Wise Refinement for algorithm design);
  - **Invention** (e.g., the Microprocessor);
  - **ProgrammingLanguage**
  - **Subject** (e.g., Information Theory, started by Shannon, or Graph Theory, started by Euler);
  - **Technology**
  - **Theorem**
  - **TheoreticalModel** (e.g., Turing's machine);
  - **WorkOfArt** (e.g., a novel).
- **Item**: a specific, identifiable specimen of a (mass-produced) object. It has 5 subclasses, corresponding to the main classes to which the item belongs:

  - **Component**, **Device**, **Document**, **Software**, **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 27 subclasses, of which its direct subclasses are:
  - **Administrative**, **Building**, **Geographic**, **Mansion**
- **Software** (19 subclasses),
  - **Development**, **Educational**, **Embedded**, **OfficeAutomation**, **OperatingSystem**, **Videogame**
- **System**: a group of *Device*s that is functional only as a whole; it differs from a *Configuration* in that, in a *Configuration*, at least one of the *Device*s would be functional if taken alone.

Domain-specific classes are those under **Component**, **Configuration**, **Device**, **Package**, **Software** and **System**. Classes borrowed from the general ontology, also constructed as part of this work, are **Event**, **Person**, **Place**, **Collection**, and **Item**. In particular, the set of subclasses of **Collection** and **Item** is extended by defining additional domain-specific subclasses. Classes borrowed from other domain-specific ontologies are **Award**, **Company**, **Document**, **IntellectualWork**[1].

As regards the relationships, in addition to those specified by each partial ontology, other relationships were included to bridge the gap between classes belonging to different partial ontologies (e.g., **Document.concerns.Device**, **Device.wasIn.Place**). Moreover, the proposed ontology provides for relationships specifically designed to express the connections among computing devices and software (e.g., **Device.clones.Device**, **Software.compatibleWith.Software**) and to record useful information concerning restoration of devices (e.g., **Component.mayReplace.Component**). This is extremely important, because restoration of computing devices has peculiarities that cannot be expressed in existing ontologies designed for other kinds of cultural heritage.

As regards attributes, each class or relationship may have its own attributes, and inherits those of its superclasses (if any). E.g., subclass **Microprocessor** of class **Component** has its own attributes (*speed*, *bits*, etc.) in addition tho those of **Component** (e.g., *name*, *technology*, etc.).

The development of the above conceptualization of the history of computing field also brought some side-products and opportunities. In addition to providing us with an ontology that allows to carry out formal reasoning tasks on the collected knowledge, it may also be interpreted as a data schema for hosting the knowledge and consulting it efficiently in a DataBase Management System. Moreover, it may be a starting point for the definition of cataloguing standards for cultural heritage material related to the history of computing. Indeed, existing standards for cultural heritage, even those developed for technological and scientific stuff (e.g., [12]), are totally unable to express the complexity and subtleties of this specific field.

4 GraphBRAIN

GraphBRAIN[2] is a general-purpose system for the development, management and (personalized) fruition of a knowledge base. The underlying data management structure is a graph database. More specifically, Neo4j [11] was used as a DBMS. Nodes and arcs may have associated attribute-value maps; nodes (representing individuals) may be labelled with one or many labels (usually representing classes), while each arc (representing a relationship) may be labeled with one type only. No schema handling is provided for by Neo4j, meaning that the user is totally free to use any type and/or attribute name for any single node and arc. While ensuring great flexibility, this does not allow to associate a clear semantics to the graph items. For this reason, GraphBRAIN requires its users to work according to pre-specified data schemes, expressed in the form of ontologies. Thus, a characterizing feature of GraphBRAIN is its bringing to cooperation a database management system for efficiently handling, mining and browsing the individuals, with an ontology level that allows it to carry out formal reasoning and consistency or correctness checks on the individuals.

Using a suitable tool, GraphBRAIN administrators may create, build and maintain ontologies by specifying the types of entities and relationships to be considered, each with its attributes and associated datatypes. The universal class is implicit, so the user must start the ontology description from the top-level classes, which are automatically considered as disjoint by the system. Each top-level class may be the root of a hierarchy of subclasses, for which no assumption about disjointness is made. Several ontologies may be handled by GraphBRAIN;

[1] Additional links will be possible in the future, e.g. with other cultural heritage and tourism ontologies (e.g., concerning the class PointOfInterest, representing museums and other places where the cultural heritage material is on show).

[2] A demo of the system can be found at `http://193.204.187.73:8088/GraphBRAIN/`

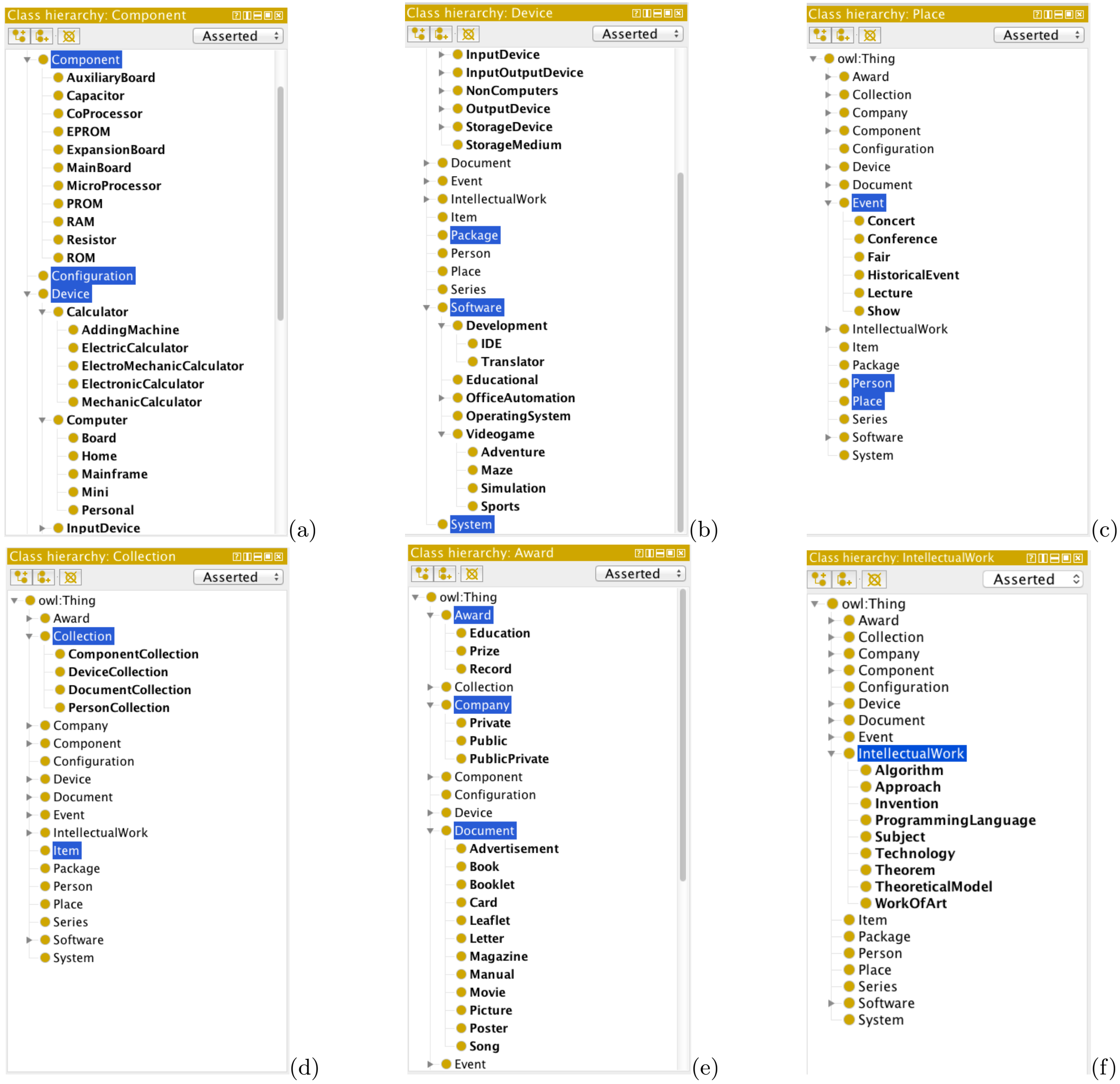

Fig. 1. OWL 'history of computing' ontology: Domain-specific (a) and b)), general (c) and d) and other domin specific (e) and f)) main classes.

some classes and relationships may appear in different ontologies, but different ontologies may define different attributes for the shared classes and relationships, in order to reflect different perspectives on them. In particular, in addition to various domain-specific ontologies, GraphBRAIN provides a top-level ontology defining very general and highly reusable concepts and relationships (e.g., **Person**, **Place**; **Person.wasIn.Place**). This top-level ontology plays a crucial role to interconnect the domain-specific ontologies, ensuring an overall connected knowledge graph. Indeed, there is a single, shared graph underlying all the domains. Thanks to the classes shared across different domains, this allows the system to reuse knowledge across domains, and thus to reach a wider range of outcomes for satisfying the user information needs. So, if an individual is used by different ontologies, it acts as a bridge among those ontologies, allowing the users of a domain to obtain additional information coming from other domains.

The ontologies are saved in an internal format, used as a schema for the graph database. The tool may also export them into standard Semantic Web formats, to make them publicly available for reuse. Currently, it can serialize them to Ontology Web Language (OWL)[3] format, with namespace prefix **cpg**, so that it can be published and exploited for ensuring semantic access to the knowledge base and make it interoperable with

[3] http://www.w3c.org/owl

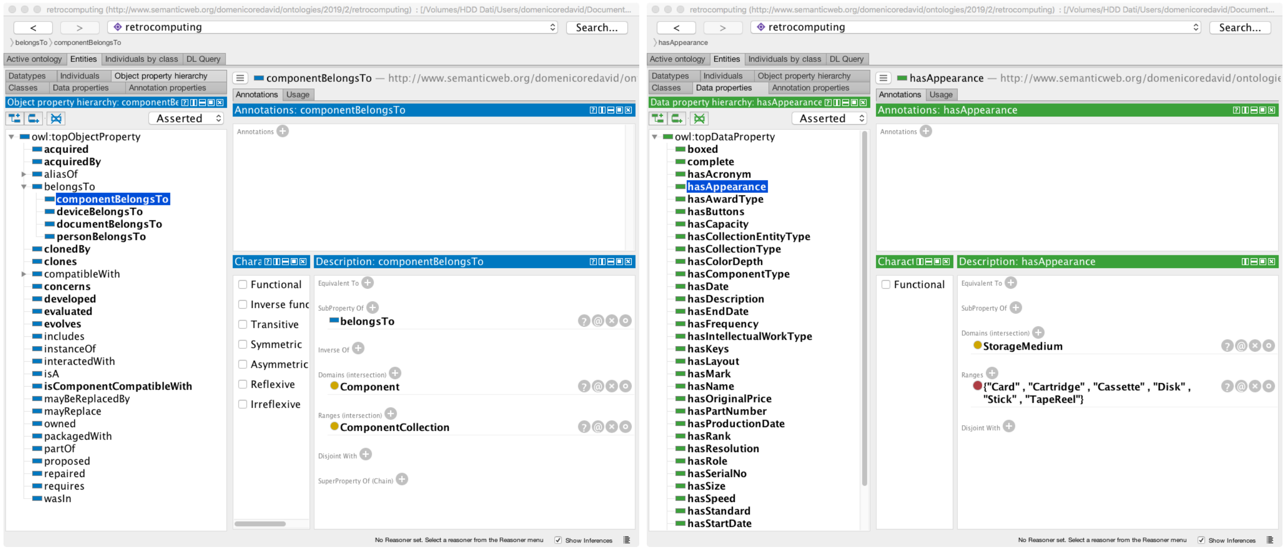

Fig. 2. OWL 'history of computing' ontology: Object and DataType properties.

other resources. Figure 1 shows the main OWL ontology classes and subclasses, object properties and datatype properties, of the history of computing ontology described in Section 3. The tool models the particular case of different collection types by declaring some specific OWL classes and sub-properties. For example, concept **Collection** is the range of object property **cpg:belongsTo**, whose domain can be any of the disjoint classes **cpg:Component**, **cpg:Device**, **cpg:Person**, and **cpg:Document**. The tool defined one sub-property of **cpg:belongsTo** for each of these domain classes (see Figure 2). In this way, instead of having a generic property (**cpg:Component** or **cpg:Device** or **cpg:Person** or **cpg:Document**) **cpg:belongsTo cpg:Collection**, one may assert instances of **cpg:Component cpg:componentBelongsTo cpg:ComponentCollection** or **cpg:Device cpg:deviceBelongsTo cpg:DeviceCollection** or **cpg:Person cpg:personBelongsTo cpg:PersonCollection** or **cpg:Document cpg:documentBelongsTo cpg:DocumentCollection**.

After setting up the ontologies, information is fed into the knowledge base by interaction with users or by automatic knowledge extraction from documents and other kinds of resources (e.g., the Internet). The interactive interface consists of two form-based tabs, one for entities and one for relationships, allowing the user to insert/update/remove instances. The forms are automatically generated by the system from the internal format specification of the ontologies. For this reason, albeit GraphBRAIN may handle several ontologies, each specifying a different domain, the form-based interface for data management and querying requires the user to select one of the available domains in order to load the corresponding scheme/ontology to be used.

Additional functionality is also provided. First, users may manage (add, show, delete) attachments for each instance. In this way GraphBRAIN goes beyond knowledge management tools, becoming a full-fledged digital library, whose content is indirectly organized according to formal ontologies, and thus may foster interoperability with other systems. Second, users may add comments, or approve/disapprove, each entity or relationship instance, and even each single attribute value thereof. This can be used to ensure some kind of 'distributed' quality assurance on the content of the knowledge base, and to establish a trust mechanism for the users. Using the comments, the users may also provide useful suggestions to improve and extend the ontologies[4]. Also, users are encouraged to provide high-quality knowledge, because using a combination of their number of contributions and trust they are assigned 'points' that they may spend in using advanced features provided by GraphBRAIN.

The same form-based interfaces can be used to query the knowledge base for instances of entities and relationships. The retrieved instances may be graphically displayed in another tab, as nodes and arcs in the graph. This allows the user to continue his search in a less structured way, by directly browsing the graph (by expanding or compressing node neighbors). This is useful to explore the available knowledge without a pre-defined goal in

[4] This is especially useful in domains such as the history of computing, where, as discussed in Section 2, the knowledge to be handled is so complex that, even defining an initial scheme to the best of one's capabilities, it is likely that sooner or later it will turn out to be insufficient or utterly unsuitable to grasp some part of the domain, in which case practitioners might suggest how to extend or adapt the problematic parts of the scheme.

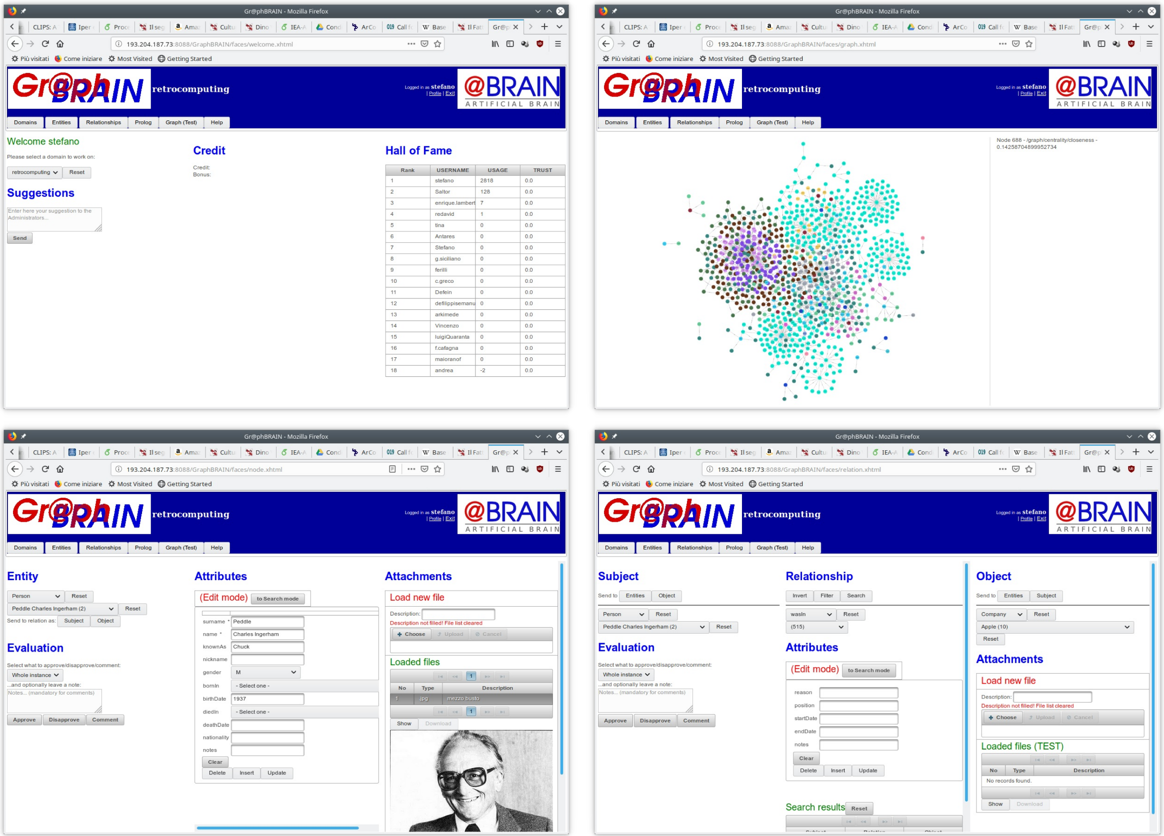

Fig. 3. GraphBRAIN interface for managing and consulting the knowledge base.

mind, but letting the data themselves drive the search. Thus, serendipity in information retrieval is supported, and the users may find unexpected information that is relevant to their information needs.

The on-line end-user interface of GraphBRAIN is shown in Fig. 3. The top-left screenshot shows the selection of a domain, while the top-right screenshot shows an overview of a portion of the overall graph. The bottom-right screenshot shows the interface for modifying and consulting the entities in the knowledge base, while the bottom-left screenshot shows the interface for modifying and consulting the relationships.

Moreover, several analysis, mining and information extraction functionalities are provided, such as:

- assess relevance of nodes and arcs in the graph, and extract the most relevant ones;
- extract a portion of the graph that is relevant to some specified starting points (nodes and/or arcs);
- extract frequent patterns and associated sub-graphs;
- predict possible links between nodes.

Some of the underlying algorithms are reused from the literature; others have been purposely extended to improve their ability to return personalized outcomes that may better satisfy the user's information needs. This would ensure that each user obtains tailored information, which is another novelty introduced by GraphBRAIN. For instance, since the graph is too large to be entirely displayed, when opening the graph tab, the neighborhood (computed by a modified version of the Spreading Activation procedure) of the most relevant nodes (based on PageRank, betweenness and harmonic centrality, etc.) is shown. 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. Of course, the displayed portion of the graph may also be the result of a specific user query.

5 Current Content of GraphBRAIN

A prototype of GraphBRAIN was used to build ontologies and a knowledge graph, as part of a larger ongoing project [5] in which GraphBRAIN will act as the knowledge base management platform underlying an inte-

Table 2. Statistics on the current content of the GraphBRAIN knowledge base

	Classes			Relationships				
Ontology	Inst	Attr	A/C	Inst	Attr	A/R	R/C	A/C + R/C
tourism	93	389	4.18	160	54	0.34	1.72	5.90
computing	449	1 506	3.35	540	208	0.39	1.20	4.55
general	333 020	1 744 116	5.24	488 639	39 186	0.08	1.47	6.71
Total	333 562	1 746 011	5.23	489 339	39 448	0.08	1.47	6.70
Total knowledge items	2 079 573			528 787				

grated system, currently under development, aimed at supporting all stakeholders involved in touristic activities (tourists, entrepreneurs and institutions). Four ontologies are currently present in the system:

general including very general concepts and relationships that are expected to be present in almost all domains;
tourism concerning history, cultural heritage items, points of interest, logistics and services, etc.;
food especially concerning the perspective of typical dishes and beverages from specific touristic regions;
computing (the ontology presented in Section 3).

So, the part concerning the history of computing is included as a specific kind of cultural heritage, with the aim of integrating it with more traditional kinds of cultural heritage, both from a scholarly perspective and for fostering its fruition in a touristic perspective. E.g., a tourist interested in the history of computing, while in Bari, might be spotted the chance to visit the collection at the Department of Computer Science, in order to see a specimen of the Olivetti Programma 101 computer.

The available ontologies share some classes and relationships, which allow to relate knowledge items from different domains, extending in this way the available scope of search beyond the single perspectives. In particular, the *general* ontology acts as a kind of hub to inter-link the other ontologies, and allow specific information from one domain to be connected to specific information from other domains. Some concepts expressed by the general ontology are: **Category**, **Document**, **Person**, **Place**, **Word**. Class **Category** is aimed at hosting items from different taxonomies. Currently, it is filled with the concepts included in WordNet [10, 4] and with the subject categories included in the standard part of the Dewey Decimal Classification (DDC) system [2]. All relationships from WordNet (hyperonymy, several kinds of meronymy, etc.), plus other typical relationships among concepts, are included in the ontology to interlink the concepts. Note that the classes in these taxonomies are reified, becoming individuals in the knowledge graph. This allows to handle them within the graph, instead of formalizing thousands of classes in the ontology. Also, in this way the concepts may be linked to individuals of other classes (e.g., documents, persons, places) and used as tags to express information about them (e.g., 'Alan Turing' might be linked with 'Computer Science', 'World War II', etc.). Class **Word** is used both to express the synset definitions in WordNet, and to express linguistic information in the knowledge base. Words may be also used for lexically tagging other items, just like concepts may be used to semantically tag them.

The current content of GraphBRAIN is summarized in Table 2. For each ontology, the number of instances (*Inst*) and attributes (*Attr*), for both classes and relationships, is shown, along with the average number of attributes (A/C) and relationship instances (R/C) per class instance. Column $A/C + R/C$ reports the average amount of information (i.e., the sum of number of attributes and number of relationships) associated to each class instance. Obviously, the vast majority of knowledge items is in the *general* ontology, where WordNet and the DDC taxonomy were automatically loaded, in addition to other items manually entered by the users. Next comes the *computing* ontology, which was the main focus of our work and the first domain-specific ontology built in GraphBRAIN. Finally, the *tourism* ontology is the most recently added, while the *food* one was only defined but not published yet for users to enter individuals. There are less class instances than relationship instances, indicating a quite connected graph, which is important for interlinking the knowledge and providing the users with information based on graph browsing. The R/C parameter reveals that the tourism subgraph is the most connected, followed by the general subgraph and finally by the computing subgraph. As expected, the average number of attributes per instance is larger for class instances than for relationship instances. Indeed, relationships are by themselves information carriers. Comparing A/C and A/R, we see that the 'information density' is different between classes and relationships for the various domains. For classes, the richer information is in the general subgraph, followed by tourism and then by computing. For relationships, it is in the computing subgraph, followed by tourism and then by general, which is significantly poorer than the others.

While available according to the linked data perspective, the GraphBRAIN knowledge graph is not available in its entirety as Linked Open Data. Indeed, it is not directly accessible to the public. Access is available only through the querying and graph browsing facilities in the on-line interface, or through pre-defined tools exposed as services, that, based on their input parameters, return relevant portions of the graph serialized as RDF.

6 Related Work

Some related work concerns the development of ontologies, and associated knowledge graphs, for cultural heritage. Focusing on the Italian landscape, we mention the following initiatives, that we plan to connect to GraphBRAIN:

Cultural-ON (Cultural ONtology) [5] an ontology aimed at modeling the data regarding cultural institutes or sites, their contact points, all multimedia files which describe them, the agents that play a specific role in them, events that can take place in them, and any other information useful to the public in order to access them. It is aligned with external ontologies (FOAF, PROV, schema.org, Dublin Core, etc.) [9].

ArCo (Architecture of Knowledge) an ontology for, and a knowledge graph of, Italian Cultural Heritage[6]. It models many types of cultural properties (including technological heritage), for which it allows to capture details such as elements affixed on cultural properties, copies, forgeries and other works related to a cultural property, specific surveys, cadastral information, historical locations, the communication medium of intangible demo-ethno-anthropological heritage, etc. It currently reuses, and is aligned to, CIDOC-CRM, EDM, Cultural-ON, and OntoPiA. The resulting knowledge graph currently includes, and provides as LOD, 293 classes and 469 properties, and a dataset of 173M triples (data from 800.000 records of the General Catalogue of Cultural Heritage[7], a database of Italian cultural heritage entities).

Concerning the use of ontologies together with graph DBs, some works exist that analyze the possibilities for cooperation. In [3] the potential of applying graph DBMSs to an ontological context in order to create essentially an ontological tensor, e.g. the algebraic counterparts of the combinatorial multilayer graphs, is outlined, and its complexity is assessed. Interestingly, both representations were applied to an open dataset with persons and relationships extracted from the official biography of Steve Jobs and the 1999 film Pirates of Silicon Valley. [8] discusses technical issues that might limit the impact of symbolic Knowledge Representation on the Knowledge Graph area, and summarizes some developments towards addressing them in various logics.

Several kinds of tools have been proposed in the literature for ontology development, each one with specific targets as regards the construction, editing, annotation and merging of ontologies [1]. Among them, the most popular and mature tool is protégé (`https://protege.stanford.edu`), based on the OWL-API, which is fully compliant with the OWL specifications by W3C (`http://owlcs.github.io/owlapi`). For this reason, GraphBRAIN adopted the same OWL-API for its ontology handling functionality, so that the generated ontologies are fully compliant with the standard and may be edited using protégé. We decided to develop a specific ontology definition and handling tool for several reasons. First, it had to be embedded into GraphBRAIN's interface, so that the administrators could seamlessly and collaboratively build and refine the ontologies. Second, while existing tools are mainly aimed at defining formal ontologies starting from an RDF knowledge base model, our perspective stemmed from the need to define a schema for the graph DB, and the translation in standard ontology format was a consequential objective. For this reason, the tool was developed so as to allow the users to comfortably define a schema to be used for building the knowledge base, on which the ontological perspective was added in order to enable OWL reasoning capabilities. It is also important to point out that there are various approaches to assessing the quality of tools for the construction of ontologies [6]. In particular, these can be a valid guide for the extension of our tool with advanced features.

7 Conclusions and Future Work

While most people are used to consider computers as a modern object, pervasive in our lives and thus of little cultural importance by itself, Computer Science and Engineering are starting to raise significant interest also from a cultural heritage perspective. Indeed, in spite of them being quite young disciplines in the landscape of human knowledge, their steady advancements and quick technological obsolescence cause even devices that are just a few decades old to become historic pieces. While, on one hand, this situation resulted in the current lack of ontological resources to describe computer devices along with their background and history, the peculiarities

[5] http://dati.beniculturali.it/lodview/cis/.html

[6] Release of the first stable version expected shortly on http://dati.beniculturali.it/
Currently available as an unstable resource at http://wit.istc.cnr.it/arco/index.php?lang=en

[7] http://www.iccd.beniculturali.it/it/per-consultare

and complexity of this branch of human knowledge makes all available resources, developed for other areas, unsuitable for it. Hence the need for a new, domain-specific ontology, which is the main objective of this paper.

Also, there is a current lack of formal and well-established resources reporting knowledge about the field. Instead, precious knowledge about this domain is scattered across many people spread all over the world. This also causes an urgent need to collect this information in a knowledge base, to make it available to all interested stakeholders (scholars, researchers, but also common people). As a contribution in this direction, this paper also proposes a knowledge graph on computing devices and their history.

As a third contribution, it also introduces GraphBRAIN, a general-purpose tool developed to design and populate knowledge graphs, and to allow collaborative enrichment thereof, in addition to advanced fruition, consultation and analysis tools, that may be used as an intermediate layer to provide services to end-user applications aimed at personalized fruition of cultural heritage, also in a touristic perspective.

There are several directions for ongoing and future work. On the ontological side, we are currently extending the number and content of ontologies in GraphBRAIN, and specifically we are refining the ontology on the history of computing, based on the feedback emerging from actual use of the system. Starting from the ontology, we will try and define a cataloguing standard for this kind of cultural heritage items, which is still missing but would be extremely important. Having the standard, a catalogue might be extracted automatically from the knowledge base. Concerning the knowledge base, we plan to contact pilot users and associations willing to contribute their knowledge about the history of computing and of computing devices. As to the platform, we are continuously improving the interface, also adding functionalities and features. The analysis and mining algorithms, in particular, will be extended and adapted for providing ever more advanced tools and services aimed at supporting researchers, scholars and other stakeholders in tailored fruition of the knowledge base.

References

1. Abburu, S., Babu, G.S.: Survey on ontology construction tools. International Journal of Scientific & Engineering Research **4**, 1748–1752 (2013)
2. Dewey, M.: A classification and subject index for cataloguing and arranging the books and pamphlets of a library. Amherst, Mass. (1876)
3. Drakopoulos, G., Kanavos, A., Mylonas, P., Sioutas, S., Tsolis, D.: Towards a framework for tensor ontologies over neo4j: Representations and operations. In: 8th International Conference on Information, Intelligence, Systems & Applications, IISA 2017, Larnaca, Cyprus, August 27-30, 2017. pp. 1–6. IEEE (2017)
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
5. Ferilli, S., De Carolis, B., Buono, P., Di Mauro, N., Angelastro, S., Redavid, D.: Una piattaforma intelligente per la gestione integrata del settore turistico. In: Primo Convegno Nazionale CINI sull'Intelligenza Artificiale – Workshop on AI for Cultural Heritage. p. 2 pages (2019), http://www.ital-ia.it/submission/163/paper, in Italian
6. Gherasim, T., Harzallah, M., Berio, G., Kuntz, P.: Methods and tools for automatic construction of ontologies from textual resources: A framework for comparison and its application. In: Guillet, F., Pinaud, B., Venturini, G., Zighed, D.A. (eds.) Advances in Knowledge Discovery and Management - Volume 3 [Best of EGC 2011, Brest, France]. Studies in Computational Intelligence, vol. 471, pp. 177–201. Springer (2011)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers (2011)
8. Krötzsch, M.: Ontologies for knowledge graphs? In: Artale, A., Glimm, B., Kontchakov, R. (eds.) Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017. CEUR Workshop Proceedings, vol. 1879. CEUR-WS.org (2017), http://ceur-ws.org/Vol-1879/invited2.pdf
9. Lodi, G., Asprino, L., Nuzzolese, A.G., Presutti, V., Gangemi, A., Recupero, D.R., Veninata, C., Orsini, A.: Semantic web for cultural heritage valorisation. In: Hai-Jew, S. (ed.) Data Analytics in Digital Humanities, pp. 3–37. Springer International Publishing, Cham (2017)
10. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM **38**, 39–41 (1995)
11. Robinson, I., Webber, J., Eifrem, E.: Graph Databases. O'Reilly Media, 2nd edn. (2015)
12. Vannozzi, F.: 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. Economia e management della cultura e della creatività, Franco Angeli Edizioni (2015), https://core.ac.uk/download/pdf/53672194.pdf, in Italian