

# Historical Archive Ontologies – Requirements, Modeling and Visualization

Katifori Akrivi, Torou Elena, Vassilakis Costas, Lepouras Georgios, Halatsis Constantin, Daradimos Elias  
[vivi@di.uoa.gr](mailto:vivi@di.uoa.gr), [etorou@di.uoa.gr](mailto:etorou@di.uoa.gr), [costas@uop.gr](mailto:costas@uop.gr), [gl@uop.gr](mailto:gl@uop.gr), [halatsis@di.uoa.gr](mailto:halatsis@di.uoa.gr), [drid@mm.di.uoa.gr](mailto:drid@mm.di.uoa.gr)

**Abstract—** Most ontology development methodologies and tools for ontology management deal with ontology snapshots, i.e. they model and manage only the most recent version of ontologies, which is inadequate for contexts where the history of the ontology is of interest, such as historical archives. This work presents a set of requirements for the modeling and visualization of a temporal ontology used as a tool for the representation of historical information. In accordance to these requirements, a visualization plug-in was designed and implemented, featuring a set of tools that enable users to efficiently examine ontology temporal characteristics such as class and instance evolution along the timeline.

**Index Terms—** ontology, time, visualization method, entity timeline

## I. INTRODUCTION

The recent progress in the area of digital libraries and the semantic web has lead to new ways of digitizing, organizing and presenting library material, enhanced with the incorporation of semantics. To this end an ontology can be a very useful tool, which presents an overview of the domain related to a specific area of interest and may be used for browsing and query refinement. Ontologies model concepts and relationships in a high level of abstraction, providing rich semantics for humans to work with and the required formalism for computers to perform mechanical processing and reasoning.

An ontology may be also used for organizing knowledge related to a specific organization or domain. The representational model it offers may be used for knowledge classification and indexing, which will be subsequently exploited for information retrieval, either for educational or research purposes. Especially in the context of a Historical Archive, an ontology may be used for recording historical information extracted from the archive documents. In such an environment an ontology is of particular importance both for browsing and querying, since the historical archive material may either be unavailable in electronic format or available as scanned images, hindering thus the use of other computer-assisted information retrieval methods such as full text search.

However in such a context a very important factor is the

concept of time. In this case, the material that the ontology refers to covers a large time span, which can extend to some decades or even centuries. This fact results in classes, instances and/or relationships between them that change with the passage of time. In such cases, the ontology should be able to reflect the evolution of the real-world, providing facilities for designating the time instants (e.g. Jan 31, 2000 19:37) or periods (e.g. [Jan 31, 2000-Feb 17, 2000]) [7] for which each represented real-world state is valid.

Most ontology development methodologies and tools for ontology management deal with ontology snapshots, i.e. they model and manage only the most recent version of ontologies, which is inadequate for contexts where previous versions of the ontology are required to offer complete information. In this paper, we present a modeling approach for ontologies that are evolving over time, its implementation in the Protégé tool, and a visualization plug-in enabling users to efficiently explore the ontology evolution.

The rest of the paper is organized as follows: the next two sections present related work in the areas of temporal ontology modeling and visualization, and provide some useful definitions. Section “Historical Archive Ontology Requirements” introduces the requirements for temporal ontology modeling, while section “Proposed Approach” presents the proposed modeling and visualization for the temporal characteristics of the ontology; useful heuristics for investigating relations between entities are proposed as well. Finally, the last section concludes the paper and outlines future work.

## II. RELATED WORK

There are several recent works related to ontology versioning and evolution. The system PromptDiff [11] has been developed in the context of a collaborative environment for managing ontologies, in order to support ontology versioning. Given two versions of an ontology, it allows users to: (1) examine the changes between versions visually; (2) understand the potential effects of changes on applications; and (3) accept or reject changes. PromptViz [16] is a tool providing advanced visualization using Treemaps to help users understand the location, impact, type and extent of changes that have occurred between versions of an ontology.

In [9], a different approach for reconciling the different

ontology versions with each other is introduced. The presented framework, which aims to provide means for reasoning based on a complete versioning history, includes the generic notion of a *change bridge* for describing ontology resource changes, and a basic set of particular change bridge types that constitute the class hierarchy of a change bridge ontology. Ontology changes are represented as instances of the change bridge types relating concepts in successive ontology versions with each other. The change bridge ontology is represented using the Resource Description Framework (RDF). In [3] the changes in an ontology are handled as database operations and triggers, whereas in [15] the abstract data type *knowledge base* is introduced which contains a description logic representation and a basic set of operations to work on it. A formalization of operations that change the knowledge base is presented, in order to support the evolution of ontologies. The operations can be extended or changed to satisfy local needs.

Finally, [2] briefly presents requirements for visualizing changes to ontologies and [5] attempts a comparative evaluation of ontology editors concerning the subject of function supporting ontology evolution. A survey of techniques for visualizing differences is included in [16].

All the previously mentioned approaches propose solutions that deal with either ontology versioning or evolution from the ontology designer point of view, meaning that the evolution of the ontology is treated as an issue for the ontology creators in order to keep track of changes to the ontology. An ontology for a historical archive however, as it represents temporally evolving information, should provide means for modeling time information as well. Furthermore, systems incorporating version management mostly give the ability to modify an ontology and save different versions of it, but do not facilitate version time stamping, browsing and querying across multiple versions, or tracking of how the ontology evolved from one version to another. The approach proposed in this paper attempts to complement the existing versioning and evolution approaches by treating evolving ontologies as integral articles, allowing users to fully exploit the temporal dimension in both querying and browsing. The modeling constructs are complemented with visualization aids to further assist the users in their exploration and browsing tasks.

### III. DEFINITIONS

According to [5], an ontology is an explicit specification of a conceptualization. The term “conceptualization” is defined as an abstract, simplified view of the world that needs to be represented for some purpose. It contains the objects, concepts and other entities that are presumed to exist in some area of interest and the relations that hold them. The term “ontology” is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems what “exists” is exactly that which can be (and has

been) represented.

Therefore, as defined in [12], an ontology is a formal explicit description of concepts, or classes in a domain of discourse. Properties -or slots- of each class describe various features and attributes of the class, and restrictions on slots (called facets or role descriptions) state conditions that must always hold to guarantee the semantic integrity of the ontology. Each slot has a type and could have a restricted number of allowed values. Allowed classes for slots of type Instance are often called a **range** of a slot. An ontology along with a set of individual instances of classes constitutes a knowledge base.

A more mathematical definition can be found in [1] and may be adapted to the terminology used by Protégé as follows:

An ontology is a triple  $O = (C, S, isa)$  where:

- 1)  $C = \{c_1, c_2, \dots, c_m\}$  is a set of classes, where each class  $c_i$  refers to a set of real world objects (class instances),
- 2)  $S = \{s_1, s_2, \dots, s_n\}$  is a set of slots, where each slot  $s_i$  could refer to:
  - a. a property of a class, which for example may be an Integer or String
  - b. a binary typed role, i.e. the representation of a relation between classes.
- 3)  $isa = \{isa_1, isa_2, \dots, isa_p\}$  is a set of inheritance relationships defined between classes. Inheritance relationships carry subset semantics and define a partial order over classes, organizing classes into one or more tree structures.

In order to accommodate the individual instances, this definition can be extended with a fourth element  $I = \{i_1, i_2, \dots, i_q\}$ , where every  $i_w$  is an instance of some class  $c_x \in C$ . Each instance  $i_w$  includes a concrete value for every slot  $s_y$  associated with  $c_x$  or its ancestors (as defined by the *isa* set).

Ontologies can be represented as directed graphs where nodes correspond to classes and instances and links to *roles* and *isa* relationships.

Ontologies may be enriched with axioms and production rules [4]. Axioms model sentences that are always true. They are included in an ontology for several purposes, such as constraining its information, verifying its correctness or deducing new information. Production rules follow the structure *If... Then...* and are used to express sets of actions and heuristics which can be represented independently from the way they will be used.

Regarding the accommodation of changes in the ontology, if a system maintains a record of all states of the knowledge base content, then history support [7] is provided. A system accommodates schema evolution if the schema can be changed without data loss, while schema versioning is provided if a record of all schemas is maintained and each (past or current) schema can be used for browsing and querying [17].

#### IV. HISTORICAL ARCHIVE ONTOLOGY REQUIREMENTS

As previously mentioned, an ontology is a structure inherently more complex than a hierarchy of concepts, as each class has its slots and it may be related through them with other classes or it may have more than one parents. A number of basic requirements can be identified for ontology management tools including fundamental ontology editing (adding, deleting and modifying classes, properties, relations and instances), browsing and querying mechanisms for locating information of interest to the users, and ontology visualization for more effective identification of patterns or relations in the ontology. Support for various ontology formats can be useful for interoperability purposes, whereas groupware editing facilities (version check-in and check-out, version reconciliation etc) can prove valuable in environments where more than one person are involved in the ontology development process.

The previously mentioned functions are more or less offered by most ontology editors. However, this is not the case with functionality related to the representation of the temporal characteristics of an ontology. Although there is a number of ontology editors like Protégé [14] that support versioning and ontology comparison, whereas other editors, like Kaon [8], include support for ontology evolution, they do not offer the functionality required for modeling and visualizing an ontology with temporal characteristics i.e. an ontology that contains the history of its individual classes and instances. The purpose of the present work is to provide a comprehensive model for modeling and visualizing class and instance evolution and integrate this model into an ontology management tool, formulating thus an environment for definition, querying, browsing, updating and visualizing snapshot and temporally-enriched ontologies alike.

##### A. *Ontology Evolution Operations*

In order to compile a set of requirements concerning the management of the time aspect of an ontology, besides the bibliographic resources employed [3, 7, 9, 10, 11, 13, 14, 16], structured interviews were conducted with the University of Athens Historical Archive personnel and users. The purpose of these interviews was to provide useful insight as to how users search or browse the material, and what their needs are regarding time issues. The results of these interviews are still being analyzed and have not been published yet, however some first insight as to how historians search the available material and how they conduct their research has aided the identification of the requirements presented in the following sections.

Furthermore, the query log of the historical archive (containing queries submitted by researchers to the historical archive personnel for retrieval of documents relevant to the researchers' interests) was surveyed, since it also constitutes a useful source of information as to what kind of queries an

ontology related to the archive material should support.

After investigating user needs, several conclusions were drawn concerning the requirements that a temporally enhanced ontology should satisfy. Firstly, the content of the knowledge base, i.e. the instances, may evolve along the timeline in the following ways:

- 1) New class instances may be created (e.g. a new University Department is founded).
- 2) Existing instances may cease to be valid (e.g. a Professor retires).
- 3) A single instance may also be split into two distinct instances (e.g. a department is split into two)
- 4) Two instances may be merged into a single one (e.g. two chairs are merged to form a department).
- 5) The value of instance slots may change. Besides changes of simple values (e.g. an academic's rank could be "Lecturer" for the period [Sep 2001-Mar 2004] and "Assistant Professor" for the period [Apr 2004-now]), this includes changes in relationships between instances (e.g. the value of the relationship "Chairman" between an instance of "Department" and an instance of "Faculty member"), as well as changes in multi-valued slots.

Complementary to changes on instances, changes on the structure (or schema) of the ontology may also occur, in order to reflect the changes to the represented universe of discourse with the passage of time. The following list summarizes the types of changes that may occur in an existing ontology, adapted from the change types presented in [10] after taking into account the results of structured interviews.

- 1) Creation of a new class C. The class is simply incorporated into the schema. For example, at some point the profession of "Calligrapher" was incorporated in the University Organization
- 2) Deletion of a class C. This happens when a class is no longer valid. At some point the "Calligrapher" class is no longer valid as this profession becomes obsolete. However, there have been cases in the university history that a class ceases to be valid at some point and reappears after a certain time period.
- 3) Modification of a class C. This case includes changing the name of the class or adding, removing or changing its slots. It may also mean moving the class to another part of the hierarchy. As a simple example, the "e-mail address" slot has to be added to the "Person" class, although it was not valid for this class until a few decades ago.
- 4) Merge classes  $C_1, C_2, \dots, C_n$  into  $C_m$ . e.g. the classes "Chair Professor" and "Chair Assistant" may be merged into a single class "Teaching and Research Staff".
- 5) Split a class  $C_m$  in several classes  $C_1, C_2, \dots, C_n$  e.g. the class "Postgraduate student" may be split into "MsC student", "Doctoral student" and "Postdoctoral student".

### B. Historical Archive Ontology Visualization Tools Requirements

Regarding the Historical Archive users' needs for information retrieval and visualization, most commonly cited in both structured interviews and the user queries was found to be the need for extracting information relevant to the evolution of a class or instance in the university. The biography of a person or the history of a department were common queries posed by many researchers. Another point of interest to researchers is the set of events that took place in a specific date or period, for example "how many professors were elected during a specific year" or "which structural reforms took place in a designated period".

In order to gain further insight into which time-related ontology operations are required, an experiment was conducted [18]. One of its purposes was to identify what tools and elements could be added to an existing ontology editor in order to accommodate for modeling and presentation of historical data through the ontology. The experiment setup and results are briefly presented in the following section.

#### 1) An experiment for user requirement definition

For the experiment we used the context of the University of Athens Historical Archive. An ontology of the University of Athens was created with Time instants and Time Periods modeled as ontology classes [19]. This part of the ontology is shown in Figure 1. We used this approach because (a) Protégé does not provide data types for representing dates and periods and (b) we wanted to investigate if this representation of time would provide any added functionality to the visualization.

Validity time periods were added to classes and instances. Historical information concerning the evolution of classes or instances was presented to the user as comments. For example, for the department of Chemistry a user could read a comment like "It was established as the evolution of the Organic and Inorganic Chemistry Chairs". In the case of persons no comments as to their previous roles in the university were added. This was decided in order to simulate the fact that it is not always clear in the context of the historical archive if two references to a person with a specific name actually refer to the same person or two distinct persons with the same name. Our purpose was to see how the users would handle the queries requesting a person's biographical information when a reference to a person existed as an instance of more than one classes.

Most of the users that participated to the experiment were students of history-related departments, and researchers working in the Department of Informatics and Telecommunications of the University of Athens. All these users had some knowledge regarding the concepts of the "University" domain but a varying degree of computer expertise. The user group was composed of 5 men and 9 women. Eight of them are students or researchers of computer science related departments, while the remaining six are students or researchers that have at least once visited the

Athens University Historical Archive or another Archive for research purposes.

During the experiment the users were asked to complete a set of information retrieval tasks ranging from simple ones, such as finding the establishment date of a department, to complex ones such as retrieving information about the history of a department or a person's biography. Four different Protégé [14] ontology visualizations were used in order to also obtain comparative results as to the efficiency of these visualizations in information retrieval [18]. The users mostly used browsing for the information retrieval tasks, employing search only when the visualization permitted it. Task completion time was recorded, as well as whether the answer was erroneous or incomplete. User comments and actions during the experiment were also transcribed.

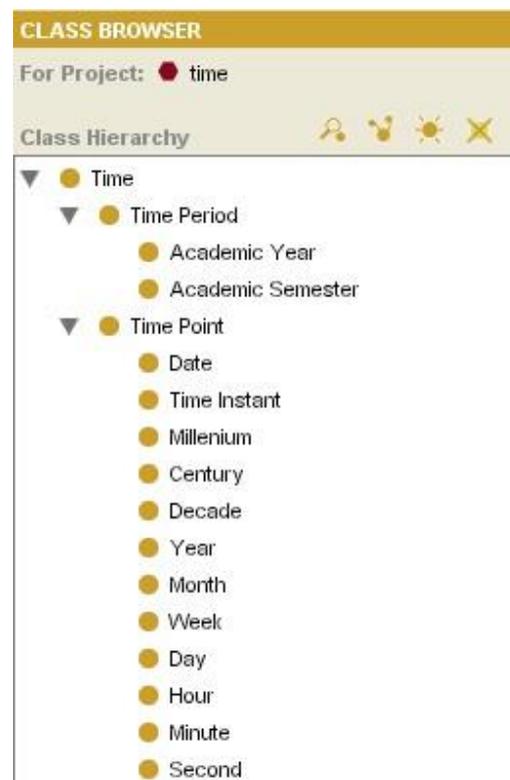


Fig. 1 The modeling of time in the historical archive ontology

The conclusions drawn by the analysis of user comments and actions are the following:

- 1) All users thought that it would be useful to have a direct connection to previous states of an instance -something like an entity history- among the instance information, in order to have direct access to this information. The implicit reference to the past state of an entity through the "comment" slot did not lead the majority of the users to further investigate the instances mentioned in the comment. As a result, the percentage of incomplete answers to questions of this type was more than 80%.
- 2) Many users expected to find useful information in the "Time" classes, such as events related to certain dates.
- 3) In the case of person biographies where no reference had

been made to previous roles of the person in the instance information, the way users faced the task falls in two general categories:

(a) Some users were satisfied by retrieving only one of the instances related to the person with the given name. For example, if they located the “Undergraduate Student” instance related to the person, they were satisfied with the result, ignoring the existing “Postgraduate Student” instance.

(b) Others located all the instances but did not seem to wonder if these instances represent in fact the same person. Only a small part of the users, which were the more experienced history researchers, seemed to wonder about that point and used the given information, such as the date of birth, to verify if the instances actually refer to the same person. Finally, two of the researchers employed a more organized search strategy, starting their search from the instances that should correspond to the first steps of a person in a university and moving up the hierarchy, for example checking first the “Student” classes and moving to “Researcher Personnel”, “Teaching Assistants” and “Professors”. One researcher even excluded whole sub-hierarchies of the ontology with observations of the type “If this person has been a Lecturer, s/he could not also be a Secretary”.

- 4) Browsing alone seemed insufficient in all the visualizations used. All the users commented on it and thought that it would be really useful to be able to combine an ontology visualization with an effective keyword search.
- 5) As a final point, it should be noted here the enthusiasm with which history researchers with little computer expertise viewed the representation of information with an ontology. They commented very positively on the organization and concentration of information and the ability to navigate easily from one part of the ontology to the other.

## 2) *Historical Archive Ontology Visualization Requirements*

For the efficient visualization of an ontology that accommodates history support, a set of tools that support the previously listed ontology evolution operations and information retrieval tasks should be developed. More specifically, the following requirements can be identified:

- 1) Direct identification of instances, properties and relationships that have evolved over time, as opposed to those that have retained a constant value.
- 2) For items that have evolved over time, presentation of the evolution timeline. A common query would be that of retrieving a person’s biography, i.e. if a certain person was ever a student in the University, when the person graduated, when the person became professor or employee in the university, etc. Or, it would be useful to know, in the case of the University of Athens Historical Archive Ontology, when the department of Informatics

was created and if it originated from another department.

- 3) For classes that have split or merged, it should be possible to identify their predecessors or successors.
- 4) It should be possible to extract a specific “point-in-time” of the ontology, i.e. creation of an ontology which contains the classes, relationships and instances that are valid for a designated time instance. The result of this extraction is a non-temporal ontology. This is similar to focusing on a specific time instant.
- 5) Facilities for extracting a specific “time slice” of the ontology, i.e. creation of an ontology which contains the classes, relationships and instances that are valid for a designated time period. The result of this extraction is a temporal ontology. This is similar to focusing on a specific time period. This extraction should be made either by explicitly defining a time period of interest or implicitly by choosing an instance or class the validity period of which will provide the time frame for the ontology extraction.
- 6) A holistic view of the timeline that the archive covers should be offered. The system should provide an overview of the time period covered by the ontology with the ability to zoom in and out and select specific sub-periods or time points in order to view the corresponding ontology.
- 7) Visualization of the “co-evolution” of different class instances or slots should be provided. A common method of analysis for historians is to correlate the evolution of different parameters (e.g. urban population and economic growth), to extract useful conclusions.
- 8) Presentation of events that took place on a specific date.

## V. PROPOSED APPROACH

In order to address the issues concerning the management of an historical archive ontology, there is a need for enriching classes and instances with temporal characteristics and creating a set of tools for presenting the temporal aspects of an historical archive ontology. The following sections summarize an approach to augmenting an ontology management system with functions related to time.

### A. *Historical Archive Ontology Modeling*

As previously mentioned, ontology schema evolution is accomplished if the schema is changed without loss of data. This means that all operations that transform the schema by addition, change or deletion of classes should be performed by making appropriate changes to the affected instances. In the case of a temporally enhanced ontology, where class evolution is of itself interesting for the user, the ontology model should provide the means for expressly representing each type of change, in order to ensure their availability to the user. To this end, a number of fields have been added into the ontology meta-schema, effectively being thus incorporated to all ontology elements, namely classes, instances and slots. The

new slots are described in the following paragraphs.

**Validity Period.** This is a slot containing one or more time periods, representing the periods when the specific class, instance or slot exists or is meaningful in the context of the universe of discourse.

**Next.** This slot is used when an ontology element is no longer valid and should be replaced by another element or elements. For example, if a student named “John Smith” graduates and starts to work as a researcher, the “Next” field in the old instance (of class “Student”) is set to point to the new instance (of class “Researcher”). The “Next” slot may also be assigned multiple values to allow for modeling the cases that the class, instance or slot was split in two or more; in such cases the slot value is a set of references to the classes, instances or slots that the current element (class, instance or slot, respectively) has been split into.

**Previous.** This slot is assigned a link to the ontology element (or elements) that the current element has replaced. For example, if a department was previously an interdisciplinary undergraduate study program, the “previous” slot of the “Department” is set to point to the “Interdisciplinary Undergraduate Study Program”. Or if the class, instance or slot was a result of merging two or more items, then this field is assigned with a set of references to the classes, instances or slots the current element has originated from.

In the context of a temporal ontology, the instances may change the values of some of their slots. For example, the salary of a secretary or the name of a university department may change at some point. This is also the case with set-typed slots, such as the members of a council that may be different in different time periods.

In order to record these changes in the instances, the concept of temporally enhanced slot has been introduced. This data type accepts as value a set of pairs of the following form:

<slot-value<sub>1</sub>>, <time-period<sub>1</sub>>

<slot-value<sub>2</sub>>, <time-period<sub>2</sub>>

...

<slot-value<sub>n</sub>>, <time-period<sub>n</sub>>

where <slot-value> is the value of the slot, either one of the simple data types such as String, Integer or Float or an instance of a class within the ontology, while <time-period> is the valid time for this value, which has a starting point and an ending point. There is a constraint defined so as the starting point cannot be greater than the ending point. Temporal slots are subdivided to single-valued temporal slots and multi-valued temporal slots.

**Single-valued temporal slots** may have only one value at a given time period – equivalently, the time periods associated with different slot values may not overlap. For example, the name slot of a university department may be the following:

Department of Informatics [16/06/1989 – 30/6/2002]

Department of Informatics and Telecommunications

[1/7/2002 – today]

The absence of overlapping is enforced by an appropriate constraint.

**Multi-valued temporal slots** may have multiple values for a single time point. For example, the members of a council may be defined as follows:

John Black [12/3/2000 - 2/4/2001]

Mary Peterson [5/5/2000 – 5/6/2002]

Sheila White [3/12/2000 – 1/6/2002]

As seen from this example there are periods where more than one person was a member of the council.

For accommodating the evolution of the knowledge base contents, two design options have been investigated. The first was to modify the ontology meta-schema in order to include the following elements:

- 1) Classes for representing time instants and time periods at different granularities (e.g. year, month, second) and calendars (e.g. Gregorian calendar, academic calendar etc), as shown in Figure 1.
- 2) A template for single-valued temporal slots. Effectively, this template maps to a multi-valued attribute with pairs of the form (slot-value, time-period), plus a constraint for non-overlapping of time periods.
- 3) A template for multi-valued temporal slots. Effectively, this template maps to a multi-valued slot with pairs of the form (slot-value, time-period), without any non-overlapping constraint.

Figure 2 presents an example of a temporal slot modeled in the ontology meta-schema. The “Name Evolution” slot presents changes to the name of the department from the date of its establishment whereas the “Chairman” slot depicts two consecutive department chairmen.

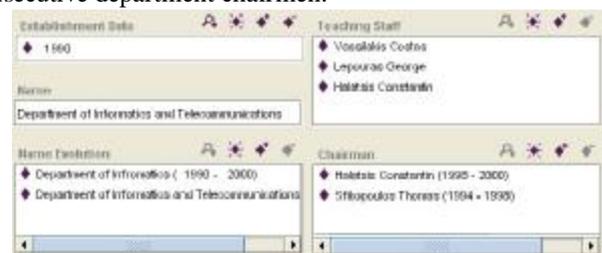


Fig. 2. Examples of Temporal Slots for the “Department of Informatics and Telecommunications” instance.

The second option was to implement these elements as build-in data types in Protégé. This has led to the creation of t-Protégé [20], which is in fact Protégé version 3.1, enhanced with temporal features, namely:

- 1) Data types expressing time quantities – more specifically *dates* (individual points in time) and *periods* (anchored segments of the time axis). The precision with which the time point may be specified varies from year-level to the level of a second. A period is entered as a pair of dates, enclosed in square brackets ([]) and separated with a comma. The dates may be expressed in different granularities, but it is required that the starting date should be less or equal to the end date.

- 2) Data types for storing histories of properties of different types (strings, integers, floats, booleans and instances [i.e. relationships]). These data types implement the notion of temporal slots as Protégé data types. These data types are used for both single-valued and multi-valued slots, by appropriately setting the “Multiple” option in Protégé’s slot definition screen.

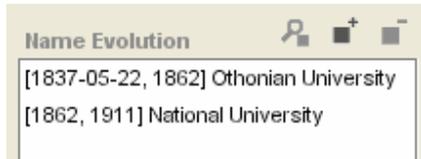


Fig. 3 The temporal slot “name evolution” modeled in t-Protégé

### B. Historical Archive Ontology Visualization

In order to facilitate the exploration of the temporal ontology and the exploitation of the meta-data information concerning time in the context of the archive ontology, there is a need for an appropriate visualization tool. This tool could allow the user to navigate in the ontology or select specific entities in order to view their course in time. Protégé [14] was selected as the ontology management tool the functionality of which would be augmented. This is due to a number of advantages it provides related to extensibility. It is developed using Java and its source code is available with detailed documentation for creating plug-ins.

The designed plug-in, named “Entity Timeline”, implements this functionality. It consists of five parts: the Class Browser, the Instance Panel, the Selected Entities Panel, the Keyword Search Window, and the Visualization window.

**Class Browser.** At the left part of the plug-in window there is an explorer-like view of the ontology, namely the one provided by the Protégé **Class Browser**. However, this view has been augmented in order to represent temporal characteristics. Classes with temporal attributes are designated by a small clock next to their description. Classes that have evolved (through slot addition or deletion, splitting or merging into other classes) are designated by a moving clock next to their description, as illustrated in Fig. 4. In this ontology, the “Personnel”, “Faculty”, “Department” and “Section” classes contain temporal slots allowing *their instances* to evolve, while the “Faculty”, “Department” and “Section” classes have *evolved themselves*, in the context of reforms within the university. It is possible also for a class to have evolved but to not contain temporal slots.



Fig. 4 Class browser with temporal feature indications

**Instance Panel.** This is a window where all the instances of a selected class are presented. Instances that have temporal slots are denoted with a small clock next to their name, while instances that have evolved along the time axis (any slot has changed its value) are designated by a small moving clock next to their instance icon. The user may tune this indication to appear if any of some selected slots have evolved. When the user selects the portion of the time axis s/he wants to focus on, instances may disappear (when their lifeline ceases to intersect with the displayed portion) or appear (if their lifeline did not formerly intersect with the selected time axis portion, but now does). The user may tune the visualization to grey-out instances that “do not exist” in the selected time axis portion, rather than totally hide them. Note that in Figure 5 all instances belong to the same class, thus they all marked with “static” clocks, indicating that they have temporal slots; in cases however that instances of different classes are displayed, “static” clocks will appear as needed.

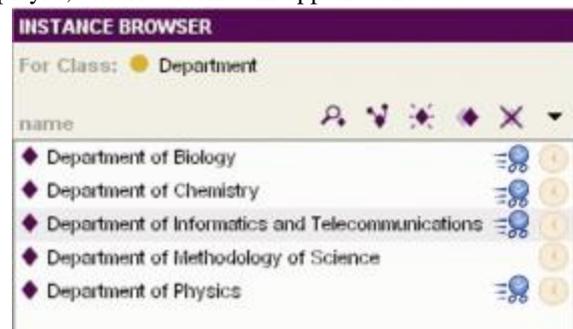


Fig. 5 The Instance View with clocks next to the instances that have changed during the selected time period.

**Selected Entities Panel.** This is a window placed at the left upper part of the Timeline plug-in tab where the user may place the instances or class s/he wishes to inspect with the Entity Timeline visualization. The panel has a toolbar which allows the user to make various configurations concerning the evolution indication beside the classes or instances or the disappearance or graying-out of instances or classes that are not valid at a specific time period.

At the right part of the plug-in window there is a tabbed pane with two tabs, the Search tab and the Visualization tab.

**Keyword Search.** At the right part of the plug-in window there is a keyword search tool (Figure 6), enabling the user to locate instances or classes. The results are presented in a tabular fashion, listing for each matching item the class name, instance name (if appropriate), matched slot(s) and matched value(s). The user may click on a result item to view it in more detail. Users may also constrain the results to a specific time period, which may be defined with the use of the time axis.

In some cases it may be interesting for the researcher to view available information for a specific date or period. This may be accomplished by providing a date in the search tool. Thus the user may select a date and view, for example, if departments were established, professors were elected or books published at that specific date.

**Main Visualization.** This window visualizes the evolution of entities along the timeline.

At the upper part a **Timeline** is visible. The user may select the time period of interest through the Timeline and view the ontology that corresponds to this period. Only classes and instances that are valid in the selected period are displayed;

the rest are normally invisible, but the user may configure the plug-in to display them as grayed-out. Several tools are available for representing the historic information modeled within the ontology.

**Entity Timeline.** The user may select one or more classes or instances and view their course over the selected period. This is accomplished with the Entity Timeline visualization. This visualization is somewhat similar to the Lifelines [13] visualization used to represent information related to a specific person along a time axis but it has been adapted to the special characteristics of an ontology.

Instance	Class	Matched Slot	Matched Value
◆ Department of Informatics and Telecommunications	● Department	name	Department of Informatics a...
	● Department	:NAME	Department

Fig. 6 The Search Tab in the Protégé Entity Timeline plug-in. A part of the results for the keyword “Department” constrained in the period 1900 to 2003 are presented.

The plug-in firstly determines the selected entity’s predecessors and successors by following the “previous” and “next” links iteratively, until the timeline ends are reached. Once the entities involved in the evolution are determined, they may be visualized using one of the two following options:

(a) A color-coded table (Fig. 7). The selected entity is presented with its previous ones on the left and the next ones on the right. Note that in this example, the chosen entity is the last in the chain, thus no successors appear on its right. If an entity has split, then the table cell corresponding to it will

span across all rows containing its successors in the next columns (as in the first two columns of Fig. 7), while a similar technique is employed for entities that have merged into a single one (as in the last two columns of Fig. 7).

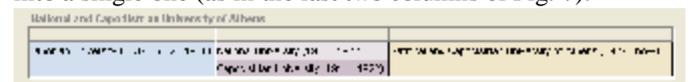


Fig. 7 The Evolution Timeline of the National and Capodistrian University of Athens as represented with the table visualization

(b) A graph representation (Figure 8). The evolution of an entity is represented by a graph where the entities involved in the evolution are the graph nodes. Nodes are connected

through directed links, with direction from the previous to the next node.

In both these cases the user may opt to view the entity validity period and/or Class name along with the Entity

name. The entities whose valid time does not intersect with the selected range are omitted or grayed-out, according to the option the user has selected in the configuration panel.

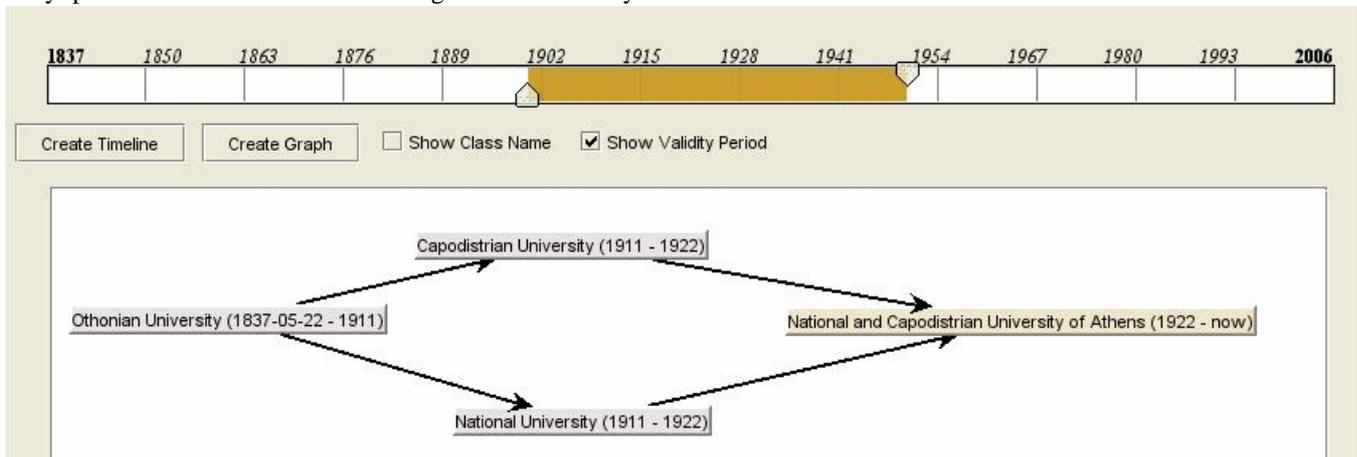


Fig. 8 The Evolution Timeline of the National and Capodistrian University of Athens along with the Timeline slider for the selection of time periods of interest. The “University” Instance validity periods are represented along with the Instance name, as the “Show Validity Period” is checked.

An important issue here is that of the history of instances. In the case of instances that the “Previous” and “Next” slots have been filled in by the ontology designer it is easy to extract the entity timeline information. There are however cases that this information has not been recorded either because it is not known or because it has not been added by the ontology designer. It would be useful in this case to provide semi-automated tools that would aid the researcher in locating this information. These tools are based on heuristics, and are described in more detail in section “Heuristics for Assisting Historical Research”.

**Ontology Snapshot or Time Slice Extraction.** This is accomplished automatically when the user selects a time point or period by displaying only the instances and classes that are valid in the specific time frame. A shortcut that has proven to be useful in the context of the Historical Archive was to provide the user the ability to select an entity and extract the ontology snapshot corresponding to the selected entity’s valid time. This shortcut effectively relieves the researcher from the burden of noting down the validity time and then setting the timeline markers (see Fig. 8) accordingly. For example, one could select as focal instance the “Othonian University” instance of the “University” class, which is in fact the University which later evolved to the “National and Capodistrian University of Athens”. When making this selection the user will be able to view the structure of the university along with its relevant instances.

### C. Example of the Entity Timeline Visualization Usage

This section presents some examples of the usage of the Timeline plug-in to retrieve information in the context of the university ontology. The first example (Fig. 9) refers to the Entity Timeline tool and the next to the presentation of events

that took place at a specific date.



Fig. 9 The Entity Timeline of the Physiognostic Department in Protégé

The user wishes to investigate the history of the Physiognostic Department, a department that is now obsolete. The researcher firstly locates the department and adds it to the “Selected Entities” panel. This may be accomplished by browsing the ontology or through the Keyword Search.

Then, in the Visualization Tab, the user displays the timeline of the department. The timeline is set to [1900, 1950] using the markers, and the “show validity period” option is set in order to display the validity period along with the Instance name.

In this case, the user may see that the department of our example was founded in 1943. It was created as a continuation of the three Chairs, the “Chair of Mineralogy and Petrology”, the “Chair of Paleontology and Geology” and the “Chair of Geography”. It was split in 1973 into two departments, namely the “Department of Geology” and the Department of Biology”. By double-clicking on these Instance names, the user may investigate the particular details of the Instances.

In the example of Figure 10, the user has chosen to view the related to the University events that took place in 1837. By typing the date in the “Search” text box, all classes and instances related to this date are tracked down. As it is displayed in the results area, in the year 1837 (which is the year the Othonian University was established), two faculties were established and a book was published.

### D. Heuristics for Assisting Historical Research

A commonly requested feature in research within historical

archives is a set of automatic or semi-automatic tools that would aid the researcher in locating instances that *probably* refer to the same entity -the same person for example- or are related to the evolution of an entity, e.g. a laboratory later becoming a museum. In order to meet these needs we propose a set of heuristics that are created to simulate to some extent the way the historian locates this information. This proposal is complemented with related visualizations, to further assist the user in viewing the entity evolutions and navigating across involved entities.

A prime candidate for locating instances corresponding to the same real-world entity (e.g. person, department etc) is the “Name” slot. The user may designate the top level class (or classes) under which search would be conducted, and the system traverses all instances to locate sets of instances having identical or *similar* [6] values for the name slot. *Temporal continuity* of instance sets, i.e. an instance within the set ceases to exist at the exact time point that another

instance within the set begins to exist, is also a good indication that the instances within a set actually refer to the same real-world entity.

Similarity of other slots may also be checked, with the user defining which slots are considered important; for instance if a search under the class “Person” is conducted, the system could be asked to check the date and place of birth, parent names and so forth. As another example, consider the case that a certain Geology “Laboratory” ceased to exist as such at a certain date, and some time later a Geology “Museum” was established at the same premises. These known facts can be combined to deduce that the laboratory was transformed into a museum, and may give incentive for further investigation to the researcher. So, for these cases similarity in the name of the instance and the fact that certain dates may be consecutive is in fact an indication of a relation of succession between the two instances.

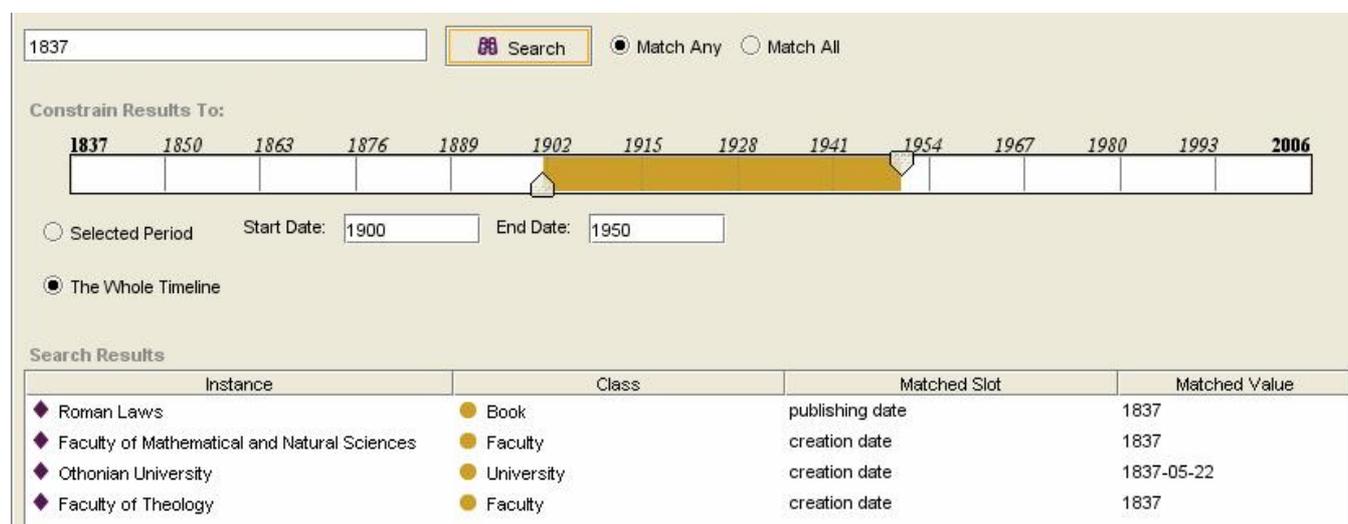


Fig. 10. Presentation of events that took place in the year 1837.

More elaborate criteria involving different slots and arbitrary value relationships could also be defined; for example, if the date that a “Student” named John Smith graduated from university is later than the date that a “Lecturer” John Smith was employed in a department, then there is little probability that the two instances refer in fact to the same person.

By applying the heuristics presented above, the system formulates sets of ontology items that *potentially* constitute the evolution of a single real-world entity. The relationships between potential predecessors and successors are recorded into slots introduced for this purpose, namely “possible next” and “possible previous” (similarly to the “next” and “previous” slots described in section “Historical Archive Ontology Modeling”). Each relationship recorded within these slots is tagged with a *confidence metric*, which is effectively a quantification of the similarity computed by the heuristic. Since these relationships are stored into concrete

slots, they can be visualized as any standard relationship as shown in Figure 11 (Protégé Jambalaya visualization).

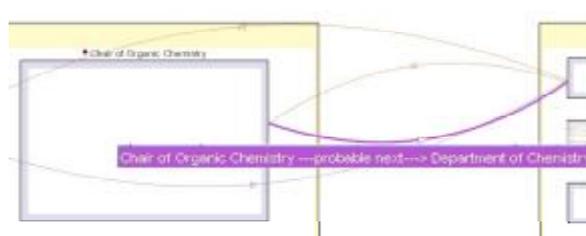


Fig. 11 Visualizing possible evolutions in the Protege visualization Jambalaya

The entity timeline plug-in may also be employed to perform the visualization (Fig. 12). Visualization is performed by means of a graph, which is constructed as in the case of Fig. 8, but this time the *possible next* and *possible previous* slots are used instead of the *next* and *previous* ones. Moreover, each arc is labeled with the confidence metric that corresponds to it. In the example shown here, the confidence

of the transitions between the Physiognostic department and its successors are high (80%) because (a) the entities are temporally continuous, (b) a portion of the name matches exactly (department) and (c) the rest of the name of the originating entity (“Physiognostic”) is a super-concept of the terms “Biology” and “Geology” that complement the successor entities’ names. The super-concept relationship is determined using Wordnet [21]; although the Wordnet database does not include the term “Physiognostic” it does include the terms “Natural Science” which has been defined as a synonym to “Physiognostic” in a domain-specific thesaurus that complements the ontology. The confidence for the transitions from the Chairs to the Physiognostic department appear lower since no portion of the names provides an exact match; the confidence on the arc emanating from the “Chair of Mineralogy and Petrology” in particular appears even lower because neither of the terms “Mineralogy” and “Petrology” is included in the Wordnet database (and no synonyms are provided), thus confidence is solely based on temporal continuity and class succession information (class “Department” is known to succeed class “Chair” in the University context).



Fig. 12. Visualizing possible evolutions in TimeLine

The incorporation of such heuristics in the Entity Timeline plug-in is currently at the design stage. An issue is to what extent will these heuristics be predefined or the user may define them as complex queries. Probably both approaches should be incorporated in order to provide support both to inexperienced users and to those that are able to use a more complex query mechanism.

## VI. CONCLUSIONS AND FUTURE WORK

This work presents the requirements, modeling and implementation of an prototype that contains a set of tools for the visualization of historical information stored in an ontology. It attempts to complement existing ontology versioning and class and instance evolution approaches with history support, allowing thus users to explore the ontology in the time dimension as well. It is currently being implemented as a Protégé plug-in. The current version does not contain the implementation of a query mechanism supporting heuristics and complex queries and still lacks full implementation and testing of all of its features. Furthermore, a thorough evaluation of the system needs to be carried out, in order to investigate its efficiency in answering complex, time-related queries in comparison with existing ontology browsers, and to verify the extent to which this tool is useful in the context of an historical archive.

## REFERENCES

- [1] B. Amann, I. Fundulaki, Integrating Ontologies and Thesauri to Build RDF Schemas, Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries, pp. 234 – 253, 1999
- [2] B. Blundell, S. Pettifer, Abstract: Graph Visualization to Aid Ontology Evolution in Protégé, 7th International Protégé Conference, 2004 <http://protege.stanford.edu/conference/2004/>
- [3] P. Ceravolo, A. Corallo, G. Elia and A. Zilli, Managing Ontology Evolution Via Relational Constraints, Eighth International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'04), 2004.
- [4] O. Corcho, A. Gómez – Pérez, A roadmap to ontology specification languages, 12<sup>th</sup> International Conference on Knowledge Acquisition, Modelling and Management (EKAW), 2000, Springer
- [5] T. R. Gruber, A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, Special issue: Current issues in knowledge modelling, Vol 5, Issue 2, pp. 199-220, 1993
- [6] V. Hatzivassiloglou, J. Klavans, and E. Eskin, Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.
- [7] C. S. Jensen, The Consensus Glossary of Temporal Database concepts, Technical Report, Department of Computer Science, Aalborg University, available at <http://www.cs.auc.dk/~csj/Glossary/>
- [8] KAON, <http://kaon.semanticweb.org/>
- [9] T. Kauppinen and E. Hyvönen, Bridging the Semantic Gap between Ontology Versions, Web Intelligence Symposium, Finnish AI Conference, 20 year Anniversary (SteP 2004), Heureka - the Finnish Science Centre, 2004
- [10] N. F. Noy, M. Klein, Ontology Evolution: Not the Same as Schema Evolution, Knowledge and Information Systems Vol. 6, Issue 4, 428-440, ACM.
- [11] N. F. Noy, S. Kunnatur, M. Klein, and M. A. Musen, Tracking Changes During Ontology Evolution, Third International Conference on the Semantic Web (ISWC-2004), 2004
- [12] N. F. Noy, D. L. McGuiness, Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001
- [13] C. Plaisant, B. Milash, A. Rose, S. Wido, and B. Schneiderman, LifeLines: Visualizing Personal Histories, In proc. of the ACM CHI conference, 1996
- [14] Protégé, <http://protege.stanford.edu/>
- [15] T. Sindt, Formal Operations for Ontology Evolution, Proceedings of the International Conference on Emerging Technologies, 2003
- [16] D. Steven, J. Perrin, PROMPT-Viz: Ontology Version Comparison Visualizations with Treemaps. Master Of Science Thesis in the Department of Computer Science, University of Victoria, 2004. Retrieved from <http://www.cs.uvic.ca/~chisel/thesis/Davi>
- [17] M. Völkel, C. F. Enquix, S. Ryszard Kruk, A. V. Zhdanova, R. Stevens, and Y. Sure, SemVersion - Versioning RDF and Ontologies KnowledgeWeb Deliverable D2.3.3.v1, Institute AIFB, University of Karlsruhe, 2005
- [18] A. Katifori, E. Torou, C. Halatsis, C. Vassilakis, G. Lepouras, A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results, IV 2006
- [19] E. Torou, A. Katifori, C. Vassilakis, G. Lepouras, C. Halatsis, Creating an Historical Archive Ontology: Guidelines and Evaluation, to appear in the Proceedings of the First IEEE International Conference on Digital Information Management (ICDIM 2006) December 06-08, 2006, Bangalore, India
- [20] C. Vassilakis, G. Lepouras, A. Katifori, t-Protégé – A Temporal Extension for Protégé, Technical Report TR-SSDBL-06-001, June 2006, Tripoli, Greece, available at <http://t-protege.uop.gr/>
- [21] C. Fellbaum (ed), WordNet An Electronic Lexical Database, MIT Press, 1988.