Multi-version Ontology-based Personalization of Clinical Guidelines

Fabio Grandi[a], Federica Mandreoli[b], Riccardo Martoglia[c]

[a] DISI, University of Bologna, Viale Risorgimento 2, Bologna, Italy (e-mail: fabio.grandi@unibo.it, phone: +39 051 20 93555).

[b] FIM, University of Modena and Reggio Emilia, Via Campi 213/b, Modena, Italy (e-mail: federica.mandreoli@unimo.it, phone +39 059 205 8321).

[c] FIM, University of Modena and Reggio Emilia, Via Campi 213/b, Modena, Italy (e-mail: riccardo.martoglia@unimo.it, phone +39 059 205 8322) - (Contact author).

Multi-version Ontology-based Personalization of Clinical Guidelines

Abstract— **When dealing with a specific patient case, a physician is often interested in retrieving a personalized version of a clinical guideline, that is a version tailored to his/her use needs. In our previous work, we proposed techniques to efficiently provide ontology-based personalized access to very large collections of multi-version clinical guidelines. In this paper, we address the problem of also dealing with a multi-version ontology used to support personalized access to clinical guidelines, which raises a number of issues that must be dealt with. Our approach allows the semantic indexing of guideline contents with respect to multi-version ontology classes and exploits the IS-A relationship among such classes for granting personalized access. Efficiency is ensured by a newly introduced annotation scheme for guidelines and solutions to cope with the evolution of ontology structure. The tests performed on a prototype implementation confirm the goodness of the approach.**

Keywords—**multi-version ontologies, multi-version clinical guidelines, personalized access, temporal indexing, semantic indexing, XML repositories.**

1. Introduction

The adoption of reference ontologies and their deployment for the personalization of multi-version resources has been recently proposed by several authors in the medical informatics domain (Grandi et al., 2012; Riano et al., 2012; Tu et al., 2011) (but also in other application fields, e.g., e-Government (Grandi et al., 2009)). In this work, as resources we consider *clinical guidelines* (Field & Lohr, 1990), that is "best practices" encoding and standardizing health care procedures, either in textual or in executable format, and their personalization with respect to an ontology of diseases, patients or available hospital facilities they are applicable to (Grandi et al., 2012). In practice, references to ontology classes are added to the computer encoding of resources (e.g., for which an XML (W3C Consortium – XML, 2015) format can conveniently be used) to introduce a sort of semantic indexing of contents representing their applicability, relevance or eligibility with respect to ontology classes. For instance, a given guideline (e.g., involving treatment of heart diseases) may contain different recommendations which are not uniformly applicable to the same classes of patients: one general therapy may be non applicable to persons who suffer from some metabolic disorders (e.g., diabetes mellitus) or chronic diseases (e.g., kidney failure) or present some addictions (e.g., cocaine); one first-choice drug may not be administered to patients who are already under treatment with possibly interacting drugs (e.g., anticoagulants), or show genetic or acquired hypersensitivity or intolerance to some substances (e.g., patients with enzymatic defects or documented allergies), and so on. Hence, when dealing with a specific patient care case, a physician may be interested in retrieving a *personalized version* of a clinical guideline, that is a version tailored to his/her use needs by means of all the available personalization coordinates involving the patient's health state, anamnesis and characteristics (e.g., genetic, demographic or preferential) and local settings (including available hospital resources, diagnostic facilities and physicians' skills). Therefore, the personalized version will only contain recommendations which are safely and effectively applicable by the user to the patient's specific case. To this purpose, we introduced in (Grandi et al., 2012; Grandi et al., 2009) a personalization query engine that, starting from a user-supplied list of ontology classes representing values of the semantic personalization coordinates, can exploit semantic indexing to retrieve the relevant contents only and produce a guideline version tailored to a specific use case. Notice that, coherently with ontology-based personalization solutions also proposed in other application fields (Callan et al., 2003; Riecken, 2000; Pretschner, 1998; Gauch et al., 2003; Middleton et al., 2004; Sieg et al., 2007; Cantador et al., 2008; Moreno et al., 2013), we use the term "personalized" as referred to the user of the computer system, that is the medical care provider who follows the guideline. This clarification is necessary not to make confusion with a more restrictive interpretation of "personalized" in use in the medical field, where the focus of personalization is the patient[1].

However, in a dynamic environment, the management of this kind of semantic versioning is interleaved with temporal aspects. The fast evolution of medical knowledge and the dynamics involved in clinical practice imply the coexistence of multiple temporal versions of the clinical guidelines stored in a repository, which are continually subject to amendments and modifications. Therefore, it is crucial to reconstruct the *consolidated version* of a guideline as produced by the application of all the modifications it underwent so far, that is the form in which it currently belongs to the state-of-the-art of clinical practice and, thus, must be applied to patients today. However, also past versions are still important, not only for historical reasons: for example, a

[1] http://www.fda.gov/scienceresearch/specialtopics/personalizedmedicine/

physician might be called upon to justify his/her actions for a given patient at a past time on the basis of the clinical guideline versions applicable to the pathology of patient and which were valid at that time.

Moreover, in a dynamic environment, the definition of domain ontologies themselves is also subject to modification as the medical knowledge, clinical environments and viable technologies evolve and, thus, also ontologies come out versioned as a consequence of updates periodically effected by domain experts and knowledge engineers or even standardization committees. As we will exemplify in Section 3, personalization of a guideline with respect to a past point in time must be effected by taking into account, in order to consider semantic indexing, the version of the reference ontology which was valid at the same time point. In other words, the selected guideline version and the ontology version used for personalization must be mutually *temporally consistent*. Since clinical guidelines have also been recently proposed to be used as evidence of the legal standard of care in medical malpractice litigation (Mello, 2001; Mackey & Liang, 2011), enforcement of temporal consistency is crucial to assess the responsibility of physicians having followed the guidelines in the past.

In this paper, we address the problem of dealing with multi-version ontologies in a framework for ontology-based personalized access to clinical guidelines in XML, extending the personalization approach proposed in (Grandi et al., 2012; Grandi et al., 2009). The approach allows the semantic indexing of guideline contents with respect to ontology classes and exploits the IS-A relationship among such classes for granting personalized access. Efficiency is ensured by a numbering scheme for ontology classes encoding the IS-A hierarchy and an XML-based engine that exploits such encoding for fast reconstruction of the requested guideline versions. When the reference ontologies are subject to change, the numbering scheme becomes a problem: if the same class belongs to two ontology versions, its codes are very likely different as long as the two ontology versions have a different IS-A structure. To overcome this drawback this paper introduces:

- a new annotation scheme for resources, based on ontology class identifiers which are independent of the encoding of the IS-A structure and an accessory data structure to bind the evolving IS-A encoding to time-invariant class identifiers;
- an efficient solution to cope with the evolution of the IS-A encoding when structural changes are applied to the ontology;
- a query processor that is able to efficiently solve XQuery-like requests with text, validity and applicability constraints when both guidelines and ontology are versioned.

The rest of the paper is organized as follows. Sec. 2 provides useful details of the framework for ontology-based personalization proposed in (Grandi et al., 2012; Grandi et al., 2009); Sec. 3 emphasizes the relevance of ontology versioning by means of a motivating example. Sec. 4 presents the approach for the representation and management of multi-version ontologies while Sec. 5 addresses the problem from a query processing point of view. Sec. 6 presents the main features of a prototyping solution we designed to assess the approach and the results we obtained on synthetic guidelines. Finally, related works and concluding remarks are addressed at Secs. 7 and 8, respectively.

2. A Framework for Ontology-based Personalization

The personalization method proposed in (Grandi et al., 2012; Grandi et al., 2009) is based on the adoption of reference domain ontologies and the introduction of semantic indexing of guideline contents with respect to ontology classes. Semantic indexing can then be used by personalization services to adapt generic resources to specific use cases, for example, to derive and enact individual and locally adapted care plans as proposed in (Grandi et al., 2012; Riano et al., 2012; Tu et al., 2011). Notice that, in this paper, we will adopt a single ontology (i.e., diseases) and a single time dimension (i.e., valid time (Jensen et al., 1998)) for the examples, in order to make them easier to follow. However, in the proposed framework, as stated and exemplified in (Grandi et al., 2012), multiple ontologies and multiple time dimensions can be supported as multiple semantic and temporal personalization dimensions, respectively. For instance, a reference ontology to be used for diseases can be derived from the ICD-10[2] international classification of diseases or from the SNOMED-CT[3] comprehensive healthcare terminologies.

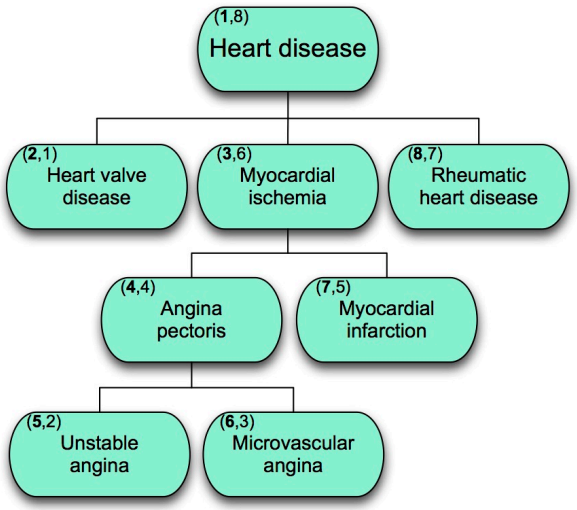
The main ontology feature which is relevant for our personalization approach is the hierarchy of classes (taxonomy) induced by the IS-A relationship. Hence, we do not consider properties or other features and also follow the simplified assumption made in (Grandi et al., 2012; Grandi et al., 2009) that the class hierarchy underlying the ontology is tree-shaped, that is each node in the class hierarchy (but the root) has a single parent. Owing to the tree structure, nodes can be assigned a preorder and a postorder code, corresponding to the sequence in which nodes are visited during a preorder or postorder traversal of the tree. Such codes can be used for efficiently characterizing the descendants of a node:

$$N \text{ is a descendant of } M \text{ iff } M.Pre < N.Pre \text{ and } N.Post < M.Post$$

with obvious meaning of the used dotted notation. For example, we can consider the sample ontology depicted in the left part of Fig.1, representing a portion of a taxonomy of diseases, where the (**preorder**, postorder) code pairs can be found in the top left of the depicted ontology classes.

The right part of Fig. 1 represents a portion of the XML encoding of a multi-version clinical guideline with embedded semantic indexing to be used for personalization (Grandi et al., 2012). It is made of an element “therapy” with two versions, the former (version 1) valid from T0 to T1 and applicable to class 3 (Myocardial ischemia) of the ontology to the left of Fig. 1 and the latter (version 2) valid from T1 on and applicable to class 4 (Angina pectoris) of the ontology. The special time value UC (Until Changed) is used to represent the To value of a right-unlimited time interval denoting a still valid version. The second version of element “therapy” contains a subelement “lifestyle”, which inherits the validity of its parent element (from T1 on) and extends the applicability inherited from its parent also to class 2 (Heart valve disease) of the same ontology (i.e., the applicability of this “lifestyle” element is class 2 or class 4). The only version (version 1) defined for “lifestyle” is necessary in order to redefine the inherited semantic pertinence (notice that the “pertinence” XML element is defined as a subelement of the “version” XML element).

[2] http://www.who.int/classifications/icd/en/
[3] http://www.ihtsdo.org/snomed-ct/




```
...
  <therapy>
  <version number="1">
    <pertinence>
      <valid from="T0" to="T1"/>
      <applies to="3"/>
    </pertinence>
       Contents of therapy–version 1
  </version>
  <version number="2">
    <pertinence>
      <valid from="T1" to="UC"/>
      <applies to="4"/>
    </pertinence>
       Contents of therapy–version 2
    <lyfestyle>
    <version number="1">
      <pertinence>
        <applies also="2"/>
      </pertinence>
       Contents of lifestyle–version 1
    </version>
    </lifestyle>
  </version>
  </therapy>
...
```

Fig. 1. A reference ontology and an XML guideline specification with temporal and semantic versioning

In our previous approach (Grandi et al., 2012; Grandi et al., 2009), as in Fig. 1, preorder codes were directly adopted as node identifiers to be used as a reference to ontology classes for semantic indexing of the guidelines which are the object of personalization. Throughout the rest of the paper, we will call this the *baseline solution*: preorder codes are, thus, embedded in the semantic markup of XML documents making up the guideline repository. When the reference ontologies are subject to change, this becomes a problem: if the same class belongs to two ontology versions, its preorder (and postorder) code is very likely different as long as the two ontology versions have a different structure. Hence, the ontology class identifiers are not time-invariant and the creation of a new ontology version also implies the creation of a new version of all the indexed guidelines, since all references to ontology classes embedded into a guideline have to be set to the new values of the class identifiers in the new guideline version in order to maintain semantic indexing. This may imply a huge amount of data to be updated even if a single class has been added to a reference ontology.

In this work, we will adopt a different approach, by introducing as proposed in (Grandi, 2013) time-invariant identifiers for reference to ontology classes and a separate management of preorder and postorder codes. In this way, the proposed encoding scheme implies an indirect reference from class identifiers used for semantic indexing of guidelines to preorder and postorder codes, which are used to support efficient query processing as described in Sec. 5.

3. Motivating example

In order to emphasize the relevance of ontology versioning in a personalization environment, we will introduce in this section a motivating example. To this end, let us consider the history of

treatment, as recommended by guidelines, of a patient suffering from a cardiovascular disease (CVD) for which dyslipidemia is taken as one of the major risk factors. In order to reduce the risk, patients with high cholesterol levels are subject to a lipid-lowering therapy. We further assume that the first treatment guideline was published in 1994 and recommended a target serum level for total cholesterol (TC) of 270 mg/dL. In 1998, we further assume a new guideline was published, defining the target levels as 193 mg/dL for TC and 116 mg/dL for low-density lipoprotein cholesterol (LDL-C). We finally assume that a new guideline published in 2003 reduced the goals to 175 mg/dL for TC and 100 mg/dL for LDL-C. With some simplifications, this is more or less what really happened in the evolution of European guidelines, as witnessed in (Erhardt & Gotto, 2006).

In an ontology-based personalization environment like the one we consider in this work, the evolution of medical knowledge underlying clinical practice implies changes to the reference domain ontologies. In our example, this leads to modifications to the definition of a "high-cholesterol cardio vascular disease" in an ontology of pathologies like that of Fig. 1. In particular, using DL-style ontology definitions, the evolution of the "HighCholesterolCVD" patient class definition consists of three versions: the first valid from 1994 to 1997 equivalent to:

$$\textbf{HighCholesterolCVD} = \textbf{CardioVascularDisease} \sqcap \exists \textbf{hasTClevel.over270}$$

the second valid from 1998 to 2002 equivalent to:

$$\textbf{HighCholesterolCVD} = \textbf{CardioVascularDisease} \sqcap \exists \textbf{hasTClevel.over193} \sqcap \exists \textbf{hasLDL-Clevel.over116}$$

and the third valid from 2003 equivalent to:

$$\textbf{HighCholesterolCVD} = \textbf{CardioVascularDisease} \sqcap \exists \textbf{hasTClevel.over175} \sqcap \exists \textbf{hasLDL-Clevel.over100}$$

In fact, by means of role value restrictions (in the form $\exists \mathbf{R.A}$), we stated that the high TC and LDL-C cholesterol levels are defined starting from suitably defined data types (i.e., "over270" used in the first version, "over193" and "over116" used in the second version, "over175" and "over100" used in the last one).

In order to retrieve a CVD treatment guideline personalized to the case of a specific patient, the patient has first to be classified by the physician using the system, possibly with the help of a suitable reasoning service, with respect to the ontology of diseases. To this end, his/her medical records are matched against the HighCholesterolCVD definition in order to check whether the patient is an instance of the class. Hence, in our framework, in December 2000, a CVD patient with TC and LDL-C serum levels of 180 and 120 mg/dL, respectively, had not to be considered for a lipid-lowering therapy since, according to the ontology valid in December 2000, would not be classified as a high-cholesterol patient (cholesterol levels were already under the therapeutic target). However, if the patient was prescribed then a lipid-lowering drug like cerivastatin by a zealous physician and died of fatal rhabdomyolysis as an adverse effect of the drug[4], the physician behavior might go under investigation (e.g. for an insurance controversy). If the physician's behavior has to be judged today, the prescription of a statin drug could even seem appropriate, as the patient had out-of-target TC and LDL-C levels according to the definitions in the current ontology. However, the correct temporal perspective must be applied, and the definitions in the

[4] Due to this potential adverse side effect, the drug was voluntarily withdrawn from the market by the producer in 2001.

ontology version valid in December 2000 must be used to evaluate the decision of the physician taken at that time. Hence, as he/she prescribed to the patient a potent drug like cerivastatin without an actual classification as a high-cholesterol patient (w.r.t. definitions in the available guidelines), then a wrong decision or at least an excess of zeal could be acknowledged.

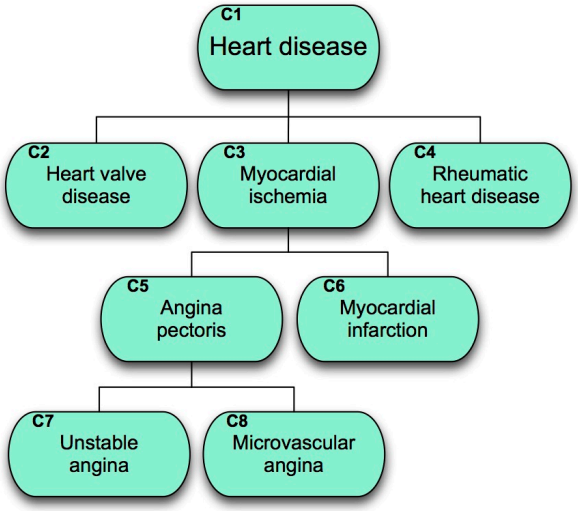
Therefore, the framework assumed in the previous Section must be extended to include representation and storage of ontologies in multi-version format and a query facility to extract a valid ontology version as a temporal snapshot from the multi-version repository. In brief, assuming ontologies are defined using the OWL language (W3C Consortium – OWL, 2015) or a sublanguage of it and represented as RDF/XML documents, "The Valid Ontology" approach consists of adding a custom XML markup to OWL documents in order to mark the boundaries of versioned portions and add timestamps to versions in a way similar to that applied to guideline documents (as exemplified in the right part of Fig. 1). The query engine already available in our framework can then be used to extract, as temporal snapshots (Jensen et al., 1998), individual ontology versions from their multi-version store.

Notice that the example presented above involved the evolution of an ontology class definition, which did not affect the ontology structure, only for the sake of providing a reasonably compact, significant and easy to understand example. In such a case, once the right ontology version has been selected and used for classification of the use case of interest, the personalization engine described in (Grandi et al., 2012) can be readily used for guideline personalization. But, in general, both class and property definitions can be versioned, also allowing IS-A hierarchies to arbitrarily evolve between versions. The real problem, that will be addressed in this work, is when the ontology evolution involves the class hierarchy, which is the foundation on which the whole building of our personalization method is based. Therefore, we will show in the next Section how the personalization framework must be upgraded in order to also cope with structural changes of an underlying reference ontology in an effective and efficient way.

4. Representation and Management of Multi-version Ontologies

In this Section, we present our new semantic indexing scheme, based on the solutions introduced in (Grandi, 2013), which is used to link resource portions to ontology classes in the presence of a multi-version ontology. It consists of the introduction of time-invariant ontology class identifiers and on the representation of the ontology class hierarchy structure by means of an accessory data structure, which we will call *code table*. For example, we can consider our sample ontology of Fig.1 with the class identifiers redefined as depicted in the left part of Fig.2, where the corresponding preorder and postorder codes of nodes can be found in the table to the right. The structure of the class hierarchy is completely defined by the information present in the table.

The new class identifiers are time-invariant: for instance, "Myocardial Infarction" remains class C6 in all the ontology versions it belongs to, regardless of its position in the evolving class hierarchy. A different code table is indeed associated to each ontology version to represent its structure. The time-invariant class identifiers can then be used for semantic indexing of guidelines instead of preorder codes as in the right part of Fig. 1.



Id	Pre	Post
C1	1	8
C2	2	1
C3	3	6
C4	8	7
C5	4	4
C6	7	5
C7	5	2
C8	6	3

Fig. 2. A reference ontology with time-invariant class identifiers and the code table representing its structure

In order to efficiently deal with the evolution of the code table when structural changes are applied to the ontology definition, three primitive operations have been defined in (Grandi, 2013) to perform the: (1) insertion of a leaf node; (2) insertion of an intermediate node; (3) deletion of a node. Any complex modification of the structure of an ontology class hierarchy can be effected via a suitable sequence of such modification primitives. Moreover, in (Grandi, 2013) it has also been shown how the whole evolution of a multi-version ontology can be stored in a single temporal table with schema **`TreeRelation(Id, Pre, Post, From, To)`**, from which the individual code tables associated to single versions can be extracted as snapshots by means of a temporal query. The semantics of the **`InsertUnder()`**, **`InsertOver()`** and **`DeleteNode()`** modification primitives have also been redefined in order to directly work on the **`TreeRelation`** temporal table in an efficient way. Full details on their definition can be found in (Grandi, 2013). In the Performance Evaluation section, for reference, we will also evaluate the ontology update costs paid in case such primitives are not used and, thus, the code table must be recomputed from scratch after the creation of each new ontology version.

For instance, assume our ontology created at time T0 is composed of a single version as depicted in Fig. 2 (the **`TreeRelation`** initially contains the same tuples as the code table in the right part of the figure, each augmented with timestamps From=T0 and To=UC). Assume further that, at time T2, we want to create a new "ST elevation myocardial infarction" class as a subclass of C6 and evidence, among the guideline portions formerly applicable to the "Myocardial infarction" class, the ones which are more specifically applicable to the new "ST elevation myocardial infarction" class. The selection of such guideline portions, whose annotations have to be updated as shown in the following, can be largely automated (e.g., based on the presence of "ST elevation" or "STEMI" keywords) but, for fine tuning, requires validation by human experts, as for the annotation of the first created guideline version.

The new ontology version can be created via the execution of the following operation:

```
InsertUnder(C6, T2)
```

whose effects on the **`TreeRelation`** temporal table are shown in Fig. 3. The new class has been assigned by the **`InsertUnder`** procedure a C9 time-invariant identifier, which must be used in the definition of the "ST elevation myocardial infarction" OWL class and in the markup of semantically indexed guidelines. Notice how, in the **`TreeRelation`** of Fig. 3, the nodes C1, C3, C4 and C6 are represented through two tuples each, representing their versions belonging to the two ontology versions, respectively (e.g., the first version of C1 with preorder 1, postorder 8 and validity [T0,T2) belongs to the first ontology version, whereas the second version of C1 with postorder changed to 9 and validity [T2,UC) belongs to the second ontology version). Nodes represented through a single tuple (e.g., C2) have a single version with validity [T0,UC) shared by both ontology versions.

Id	Pre	Post	From	To
C1	1	8	T0	T2
C2	2	1	T0	UC
C3	3	6	T0	T2
C4	8	7	T0	T2
C5	4	4	T0	UC
C6	7	5	T0	T2
C7	5	2	T0	UC
C8	6	3	T0	UC
C1	1	9	T2	UC
C3	3	7	T2	UC
C4	9	8	T2	UC
C6	7	6	T2	UC
C9	8	5	T2	UC

Fig. 3. The **`TreeRelation`** temporal table after the creation of the new ontology class

Therefore, the ontology version valid at a given time T can be retrieved by means of a classical temporal snapshot query over the temporal relation in Fig. 3

```
SELECT Id,Pre,Post FROM TreeRelation
WHERE From<=T AND T<To
```

We can notice that the retrieved snapshot coincides with the code table of the first ontology version shown in Fig. 1 if $T0 \leq T < T2$, whereas the outcome would be the code table of the second ontology version (with "ST elevation myocardial Infarction" class identified by C9) if $T \geq T2$.

As far as the annotated guidelines are concerned, the existing guideline portions which were marked as applicable to C6 but which are more specifically applicable to the new "ST elevation infarction" class must be updated. For example, with reference to the guideline portion in the right part of Fig. 1, assume this is the case of version 2 of the "therapy" element, as its contents have been reclassified as applicable to the new class C9. Then the guideline encoding must be updated as shown in Fig. 4. In this way, the updated annotations in the figure say that the (unchanged) contents are still applicable to class C6 in the first ontology version valid in [T0,T2) but applicable to class C9 in the second ontology version valid from T2 on. The existing guideline portions applicable to C6 but not specifically to the "ST elevation infarction" class, are left untouched indeed. Notice that, after such an update, by inheritance of the pertinence, the "lifestyle" subelement of Fig. 1 remains applicable to C6 or C2 in [T1,T2) but becomes applicable to C9 or C2 in [T2,UC).

In the baseline solution where preorder codes are used as class identifiers, update of resources after an ontology changes is mandatory to preserve the basic functionalities of the personalization approach, as the semantic indexing would be irremediably lost without updating what links the resource portions to the ontology classes. On the contrary, in the solution based on time-invariant class identifiers proposed in this paper, update of resources after an ontology change is aimed at improving the precision of the semantic indexing mechanism, by making the personalization approach benefit of the full available ontology. Hence, in some cases, it would not be strictly necessary to update the resources immediately after the ontology changes (e.g., updates could be effected in a *lazy* fashion). For instance, the existing portions which would be more specifically applicable to "ST elevation infarction" could still be retrieved as applicable to "Myocardial infarction", until the update is effected. However, in the performance comparison between our proposed solution and the baseline solution in Sec. 5, we will take into account the costs of such resource updates, since sooner or later they have to be done for a finer tuning of the semantic indexing mechanism. Notice also that such updates involve a small fraction of the available resources, whereas even the whole resource repository might have to be often updated after an ontology change in the baseline solution.

In case the effected ontology modification is **`InsertOver(Cy,T)`** and Cx was the parent of Cy before T, then the existing guideline portions whose semantic annotations must be changed are the ones that were applicable to Cx but not to Cy and that result applicable to the newly created class, which is child of Cx and parent of Cy in the new ontology version. In case the modification is **`DeleteNode(Cy,T)`** and Cx was the parent of Cy before T, then the existing guideline portions whose semantic annotations must be changed are the ones that were applicable to the deleted class Cy and which remain applicable to Cx in the new ontology version.

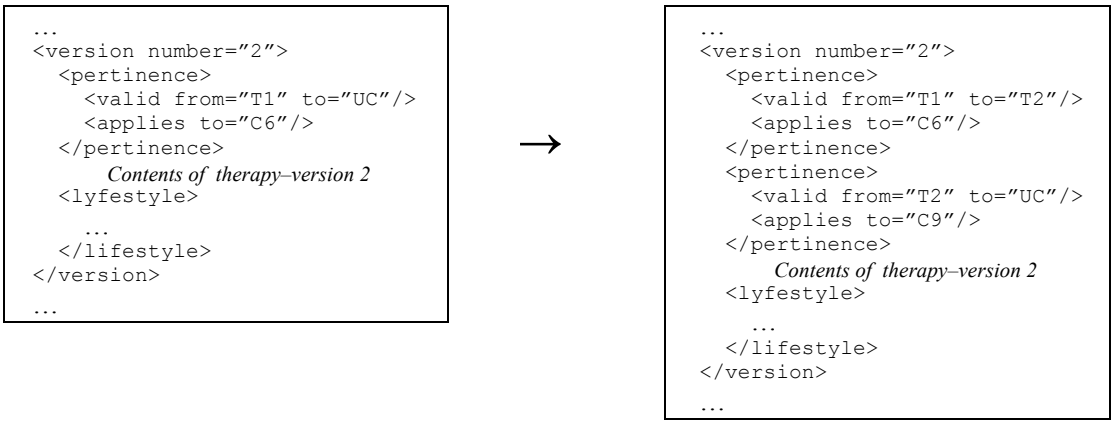


Fig. 4. Guideline contents update before (to the left) and after (to the right) the creation of a subclass C9 of C6 at time T1. The update is required since we assume the contents are more specifically applicable to the new class C9.

5. Personalization Query Processing with Multi-version Ontologies

The semantic indexing of resources which links their contents to reference ontologies is designed to support personalization queries (Grandi et al., 2012; Grandi et al., 2009). To this

purpose, and in order to show how query processing works in the presence of multi-version ontology, we consider the XQuery-like (W3C Consortium – XQuery, 2015) query template which follows:

```
FOR $x IN document("resources.xml")
WHERE TEXT_CONSTRAINT($x,CC)
  AND VALID($x,T) AND APPLICABLE($x,Cx:depth)
RETURN $x
```

which is a simplified form of the template introduced in (Grandi et al., 2012; Grandi et al., 2009) and which is embedded in our prototype system. The function `TEXT_CONSTRAINT()` applies textual constraints to the contents of the resources to be retrieved. Textual constraints can include both structural and lexical constraints, being expressible as an XPath expression (Berglund et al., 2011) which can be used for matching keywords within the resource structured contents. The function `VALID()` extracts a temporal snapshot of the resources by selecting the content versions valid at time "T". Finally, the function `APPLICABLE()` effects a semantic slicing of the resources by selecting the content versions which are applicable to instances of ontology class "Cx" and of its ancestors up to "depth" levels (in (Grandi et al., 2012; Grandi et al., 2009), the expression "Cx:depth" is called *navigational pattern*, with respect to the reference ontology). For example, the following query:

```
FOR $x IN document("guidelines.xml")
WHERE TEXT_CONSTRAINT($x, "//therapy//title/text(), anticoagulant")
  AND VALID($x, "2000-12-01")
  AND APPLICABLE($x, "Myocardial infarction":1)
RETURN $x
```

asks the system to retrieve the guideline portions which contain the word "anticoagulant" in the title of some contents of the section "therapy" and that are applicable to the "Myocardial infarction" class or to its parent class in the ontology, using a temporal perspective as of December 1, 2000 (i.e., the version either of the guidelines and of the reference ontology to be used is the one valid at T="2000-12-01").

In the presence of multi-version ontologies, the first step in query processing is the determination of the ontology classes denoted by the navigational pattern "Cx:depth" and of the preorder and postorder codes of such classes. This information can be retrieved by first deriving the ontology version valid at time T from the TreeRelation temporal table. Then, table rows CX and CY must be retrieved in the snapshot ontology, where CX corresponds to the class whose identifier is "Cx" and CY corresponds to the ancestor of the class CX that can be reached in "depth" steps starting from CX (to this purpose, simple SQL statements as shown in (Grandi, 2013) can be used).

CX and CY data are then used, in the second query processing step, to select the qualifying resource contents through their preorder and postorder codes. In particular, a resource version qualifies if its semantic pertinence implies the navigational pattern (Grandi et al., 2012; Tu et al., 2011; Grandi et al., 2009). Thanks to the properties of the preorder/postorder encoding, this notion of implication translates into verifying whether at least one of the ontology classes which make up the semantic pertinence of the resource is contained in the rectangular region defined in the preorder/postorder plane by the navigational pattern (Grandi et al., 2012; Grandi et al., 2009). Such rectangular region, in which all and only the nodes in the inheritance path from CY to CX fall, can be determined as the Cartesian product [CY.Pre,CX.Pre] x [CX.Post,CY.Post] (i.e., the lower right corner of the rectangle is CX, whereas the upper left corner is CY). Owing to the fact

that preorder and postorder codes associated to the same classes can be different in different ontology versions, we might have a different containment relationship to be checked for each ontology version.

For example, let us consider the multi-version ontology stored in our sample TreeRelation displayed in Fig. 3 and the query navigational pattern “Myocardial infarction:1”. Depending on the time “T” of interest, the CX and CY values are as summarized in the Table which follows:

	CX			CY		
Time	Id	Pre	Post	Id	Pre	Post
[T0,T2)	C6	7	5	C3	3	6
[T2,UC)	C6	7	6	C3	3	7

Table 1. Evaluation of the navigational pattern “Myocardial infarction:1”in different versions of the ontology of Fig. 3.

Let us further consider the guideline portion in Fig. 1, after the update shown in Fig. 4. The element therapy(v1) is applicable to class C3 in [T0,T1); the element therapy(v2) is applicable to class C6 in [T1,T2) and to class C9 in [T2,UC); the element therapy(v2)/lifestyle(v1) is applicable to class C6 or class C2 in [T1,T2) and to class C9 or class C2 in [T2,UC). The relative positioning of such resource pertinences with respect to the regions individuated by the navigational pattern “Myocardial infarction:1” in the preorder/postorder plane for different time values is displayed in Fig. 5.

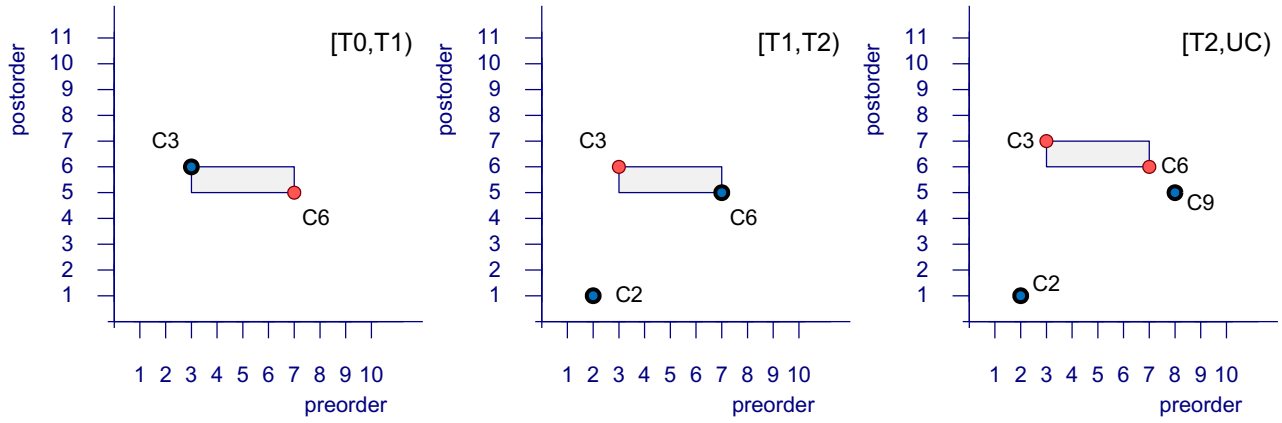


Fig. 5. Query processing in the preorder/postorder plane. Placement of candidate guideline elements is shown by blue circles with a solid border.

Hence, at any time T∈[T0,T1), the only valid element therapy(v1) qualifies since its semantic pertinence is class C3 which has coordinates (3,6)) and is contained in the region individuated by “Myocardial infarction:1” (that is [3,5] x [7,6]). At any time T∈[T1,T2), the valid elements are therapy(v2), which qualifies since its semantic pertinence is class C6 (which has coordinates (7,5)) and is contained in the region individuated by “Myocardial infarction:1” (which is [3,5] x [7,6]), and its subelement therapy/lifestyle(v1), which also qualifies as it inherits the applicability class C6 from its parent (whereas its other applicability class C2 with coordinates (2,1) lays outside of the region). At any time T∈[T2,UC), there are no contents in the considered guideline

portion which can qualify: the navigational pattern “Myocardial infarction:1” translates into the [3,6] x [8,7] region and the valid elements are therapy(v2), with applicability classes C9 with coordinates (8,5), and therapy(v2)/lifestyle(v1), which inherits applicability to class C9 and is also applicable to class C2 with coordinates (2,1), both laying outside of the region.

Therefore, considering the guideline portion in the right part of Fig. 1, a query with navigational pattern “Myocardial infarction:1” returns the first version of the element “therapy” if the temporal selection condition involves a time $T \in [T0,T1)$, retrieves the second version of the element “therapy” (inclusive of the subelement “lifestyle”) if the temporal selection condition involves a time $T \in [T1,T2)$ and returns an empty result if the temporal selection involves a time $T \geq T2$.

As outlined in (Grandi et al., 2012, Sec. 3.6), in order to obtain a fully fledged efficient and scalable personalization engine, the selection of resource contents based on the semantic indexing described above can be combined with the holistic technology described in (Grandi et al., 2012) and relying on the holistic temporal slicing techniques presented in (Mandreoli et al., 2006). In a few words, the holistic technology is based on a four-level architecture on which stack-based algorithms can be executed for efficient path and twig matching in querying an XML file. Full details on the implemented techniques for efficient personalization query processing can be found in (Grandi et al., 2012).

Finally, we can observe that the query template considered in this section can be easily extended to support other temporal selection operators (e.g., to test overlap or containment of intervals) and to retrieve data valid over temporal intervals (i.e., also belonging to more than one temporal version of the resource) like in the more general formulation presented in (Grandi et al., 2012). Furthermore, also the applicability constraint can be extended to the general form presented in (Grandi et al., 2012), where combinations with “AND” and “OR” logical operators of several navigational patterns in positive or negated form can be specified (in order to qualify for a negated navigational pattern, a resource must have its representative point outside the region defined by the navigational pattern in the plane). Also applicability constraints involving multiple reference ontologies in the same query can be specified and processed as shown in (Grandi et al., 2012).

6. Implementation Notes and Performance Evaluation

The techniques presented in the previous sections have been implemented in a complete prototype of the personalization engine. The engine supports:

- multi-version ontology management and update, exploiting the insertion/deletion algorithms described in Section 4 for the modification of the ontology structure;
- multi-version XML document (i.e. guideline) collection management, exploiting indirect referencing to the ontology as described in Section 4;
- query processing capabilities on both ontology (so to retrieve the identifier of the relevant classes) and document collection, in order to fully support the guideline personalization process described in Section 5.

While the ontology structure evolution is directly managed in relational tables as described in the previous sections, for XML document management we exploit an “XML-native” architecture relying on a Multi-version XML Query Processor we designed on purpose, which is able to

manage the XML data repository and to support all the query facilities in a single component. The engine stores and reconstructs on-the-fly the XML guideline versions; personalized access is supported by means of a fast handling of the applicability constraints. Moreover, additional temporal, textual and/or structural constraints are supported. Note that, differently from typical stratum approaches (Grandi et al., 2009), our native implementation stores multi-version guidelines as a collection of ad-hoc tuples, each representing one of its multi-version parts. In this way, we do not need to retrieve whole XML documents and build space-consuming structures such as DOM trees to process a query (Mandreoli et al., 2006).

In order to evaluate the performance of our engine in a variety of scenarios, we performed a number of exploratory tests, involving both (a) multi-version ontology and document collection updating (Section 6.2) and (b) multi-version ontology and document collection querying (Section 6.3). For reference, both kinds of tests will include relevant comparisons with notable baselines.

6.1 Experimental Setting

We performed the tests on multi-version ontologies of various sizes: ontologies O1 (approximately containing 130 classes), O2 (600 classes) and O3 (1200 classes).

Our reference guideline collection, D1, contains 10000 XML guidelines; furthermore, we will also test the querying scalability w.r.t. the number of documents with the additional document sets D2 and D3, containing 20000 and 40000 XML guidelines, respectively. The total size of the resulting collections is 250MB (D1), 500MB (D2) and 1GB (D3).

The ontologies and guidelines are synthetically generated; the synthetic guidelines have a structure conforming to the GEM guideline schema presented in (Shiffman et al., 2000). As far as the multi-version aspects are concerned, we applied our multi-version encoding scheme to the ontologies and guidelines, using three temporal dimensions, by means of a configurable XML generator (Mandreoli et al., 2006). On average, each document contains 50÷60 nodes, with a depth level of 10, and 15÷20 of these nodes own 2 or 3 time-stamped versions.

All the experiments have been effected on an Intel Core i7 2.3Ghz OS X workstation, equipped with 8 GB RAM and a 256 GB SSD hard drive.

6.2 Multi-version Ontology and Document Collection Update

First of all, we tested the performance of our engine in performing updates (i.e. insertions and deletions of classes) to the available ontology classes and, in case, to propagate such modifications to the document collection. Three ontology update scenarios, U1, U2 and U3, were considered, where several randomly generated updates were successively applied to an initial ontology:

- Scenario U1: starting ontology of 100 classes, modified with 50 update operations (resulting in 5 versions);
- Scenario U2: starting ontology of 500 classes, modified with 250 update operations (resulting in 25 versions);
- Scenario U3: starting ontology of 1000 classes, modified with 500 update operations (resulting in 50 versions).

The scenarios produced the previously introduced ontologies O1, O2 and O3 that have been exploited for the querying tests.

Ontology update (time in secs)						
	With insert/delete algorithms (Algs)				Without (Baseline)	
	avg	tot	min	max	avg	tot
Update Scenario U1	0.026	1.281	0.010	0.046	0.106	5.294
Update Scenario U2	0.135	33.834	0.047	0.322	0.446	111.612
Update Scenario U3	0.266	133.069	0.099	0.571	0.891	445.324
Document collection update (time in secs)						
	With indirect reference (IndRef)				Without (Baseline)	
	avg	tot	min	max	avg	tot
Update Scenario U1	0.160	7.983	0.103	0.215	17.554	877.693
Update Scenario U2	0.145	36.374	0.095	0.208	109.597	27399.223
Update Scenario U3	0.136	67.754	0.093	0.212	215.980	107990.052

Fig. 6. Update time for multi-version ontology (top) and guideline document collection (bottom)

The upper left part of Figure 6 shows the average, total, minimum and maximum update time required to perform the updates. Thanks to our optimized insert/delete algorithms ("Algs" in figure), each modification is performed in a fraction of seconds, also in the larger scenarios. Please also note that the document collection update time is equally low (lower left part of figure). The main reason is that, thanks to the use of indirect referencing ("IndRef" in figure) in the guideline document collection, only a small fraction of collection updates, i.e. only those that refine the annotations referring to newly inserted ontology classes and those due to class deletions, is actually required.

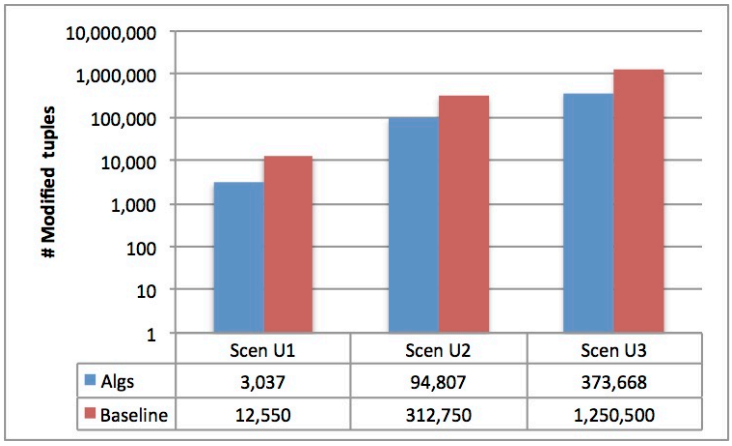


Fig. 7. Number of modified ontology tuples in update scenarios U1, U2, U3

The advantages of our solution is even more evident by comparing update time figures with the following baseline:

- for updating the ontology, an approach (upper right part of figure) that does not exploit our insert/delete primitives, thus requiring recomputation of the code table from scratch after the creation of each new ontology version. The performance slowdown is very noticeable (nearly 4x slower). Further, Figure 7 illustrates the comparison in terms of

number of tuples that need to be modified in the database: note the logarithmic scale and the large gap, especially for larger ontologies, making this an unfeasible solution;

- for updating the document collection, an approach (bottom right in figure) that does not exploit indirect reference, i.e. where preorder codes are directly embedded in the semantic markup of XML documents. In this case, each update to the collection D1 turns out to be a very demanding task, taking more than 3.5 minutes on average in the larger scenario.

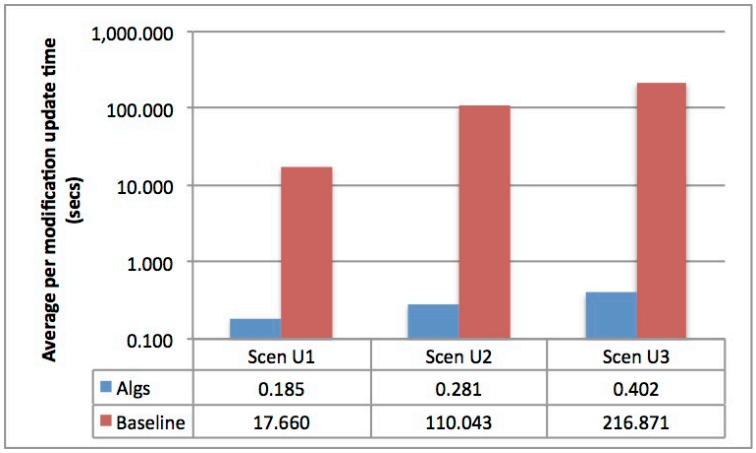


Fig. 8. Average per-modification ontology update time in update scenarios U1, U2, U3

Finally, Figure 8 graphically summarizes total (ontology and document collection, when needed) average per-modification update time, showing that exploiting the proposed update primitives together with indirect reference allows us to manage ontology modifications in almost real time (i.e. fraction of seconds per modification instead of several minutes), an undisputed advantage for practical use of the engine.

6.3 Multi-version Ontology and Document Collection Querying

In order to give a complete picture of the engine performances, we will now investigate its efficiency and scalability in querying the multi-version ontology and guideline collections.

Experiments were conducted by submitting a selection of 100 representative queries and by measuring the engine response time. The queries were designed so to stress the different system capabilities, therefore they contain structural constraints, textual search by keywords, temporal conditions and applicability constraints. Since the system answered all our queries with very uniform performances, we will show summarized processing time for all of them.

Ontology querying (time in secs)			
	avg	min	max
Ontology O1	0.001	0.001	0.001
Ontology O2	0.008	0.007	0.011
Ontology O3	0.017	0.001	0.027
Document collection querying (time in secs)			
	avg	min	max
Collection D1	0.529	0.213	1.320
Collection D2	0.989	0.398	2.420
Collection D3	1.583	0.614	3.870

Fig. 9. Ontology and document collection querying time

Figure 9 shows average, minimum and maximum querying times for both ontology (upper part, ontologies O1, O2 and O3) and document collection (lower part, guidelines sets D1, D2 and D3). As we can see, the time required for querying a multi-version ontology, also in the larger scenarios, is quite negligible (some milliseconds), thanks to the management approach proposed in the paper.

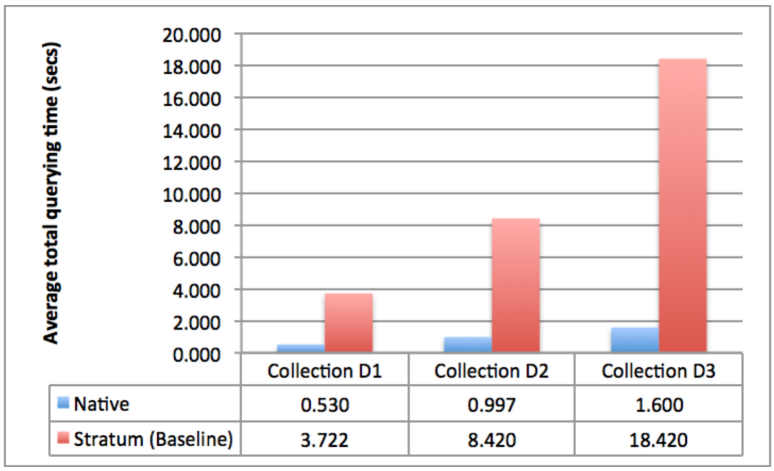


Fig. 10. Average total querying time for guideline collections D1, D2 and D3

As to guideline querying, our approach shows a good efficiency in every context, providing a short response time (including query analysis, retrieval of the qualifying guideline parts and reconstruction of the result) of less than one or two seconds on average. Therefore, total engine querying time (ontology and guidelines) are confirmed to be in this order of magnitude: Figure 10 makes those times explicit by showing the case of querying the largest ontology (O3) together with the different document sets. The figures of our system are denoted as "Native" in the figure, since they exploit our "XML-native" engine implementation.

Figure 10 is also representative of the good scalability of the system. For instance, the queries executed in 0.53 seconds on the 10,000 documents of collection D1, and took 1.6 seconds on D3, which is four times larger than D1, with a scalability proportion of approximately 0.75:1. This is an important achievement: as also the comparison in figure shows, non-optimized solutions such as our past "stratum" implementation on top of a DBMS (Grandi et al., 2009) are not only significantly (more than 10 times) slower, but also do not benefit from larger document sets and

scale with a 1:1 proportion. This is due to the post-processing phase necessary to deal with the temporal and applicability aspects, which is forced to work on one document at a time.

7. Related Work

Generally speaking, personalization can be defined as the ways in which information and services can be tailored to match the unique and specific needs of an individual or a community (Callan et al., 2003). It entails the understanding of the needs of individuals and efficiently and knowledgeably address their needs in a given context (Riecken, 2000). Pretschner (1998) presents a thorough discussion of ontology-based personalization techniques and systems. More recent proposals of ontology-based personalization solutions, also in various application fields, include (Gauch et al., 2003; Middleton et al., 2004; Sieg et al., 2007; Cantador et al., 2008; Moreno et al., 2013). When temporal resources are accessed, the necessity of also versioning the ontology used for personalization has been identified (for applications.in the legal domain) in (Grandi & Scalas, 2009).

As far as the medical domain is concerned, several data models and computer tools have been proposed to manage versioning issues and to support adaptable clinical guidelines, including (Fridsma et al., 2006; Owens & Nease, 1997; Shahar et al., 1998; Sanders, 1998; Sanders et al., 2001; Boxwala et al., 2002; Peleg & Kantor, 2003; Scott-Wright et al., 2004; Terenziani et al., 2004, 2005; Groot et al., 2008; Kaiser & Miksch, 2009; Peleg et al., 2013). In particular, (Riano et al., 2012; Tu et al., 2011; Shahar et al., 2004; Zheng et al., 2014) are the proposals more strictly related to our work, which consider the adoption of semantic annotations to support versioning and indexing, and the use of ontologies to support selective querying or adaptation of clinical guidelines (or, more in general, biomedical data). Riaño et al. (2012) propose an approach based on two personalization processes: the former is used to adapt the contents of a medical ontology to a given concrete patient using information available in health-care records; then the latter uses the personalized patient ontology to derive an individual intervention plan in the form of a personalized computer-interpretable guideline. In (Tu et al., 2011), the authors propose a practical method for transforming free-text eligibility criteria specified in clinical trials into computable criteria to support some personalized access (for the "searching for studies enrolling specific patient populations" use case). The DeGeL environment (Shahar et al., 2004) allows assisted conversion of textual guidelines into a computer-interpretable format, semantic annotation (via the Uruz markup tool) and indexing (via the IndexiGuide semantic classifier) of the guidelines to support selective access to the annotated guideline repository (via the Vaidurya context-sensitive search and retrieval tool). The DBOntoLink system (Zheng et al., 2014) allows users to express ontology-based specifications to query semantically annotated biomedical data stored in a traditional database via a Semantic Adapter module. Efficiency is guaranteed by the underlying DBMS capabilities and management of caching for ontology operations.

However, all such proposals rely on the query functionalities of a standard relational or XML DBMS for the management of large guideline collections, while adaptation is effected through main-memory processing of the retrieved clinical guidelines. In particular, none of them considered the maintenance of temporal versions of the guidelines, and took into account the proliferation of versions inherent to the management of multiple versioning dimensions. In our previous research (Grandi et al., 2009), we showed how such an approach, in the presence of temporal and semantic versioning, becomes definitely inadequate when the number of resources and versions per resource increases since, in order to select the relevant parts and consistently

assemble the personalized versions, one document at a time must be processed and a large amount of main memory is required, which strongly limits scalability and concurrency. In (Grandi et al., 2012), we addressed the efficiency problems in the context of ontology-based access to multi-version clinical guidelines with the introduction of a personalization query engine exploiting technologies developed on purpose. However, by extending (Grandi et al., 2012) with the support of the ontology versioning, the present paper is, to the best of our knowledge, the first to combine effectiveness and efficiency concerns with the support of a full temporal perspective, in providing personalized access to a large collection of multi-version resources.

The literature presents a large body of works dealing with ontology evolution and management of multiple ontology versions, as witnessed by the annotated bibliography published in (Grandi, 2012). Notice that, in this work, we do not deal with ontology evolution in a strict sense but, rather, with evolution of the class IS-A hierarchy extracted from the ontology, which is necessary to support the personalized access to data resources. No other features of the underlying ontologies, including individuals and properties, are of interest for the (XML) data management problem considered here. When the evolution of a fully fledged ontology is required, the proposed approach can be straightforwardly merged with one of the frameworks described in (Zablith et al., 2013). In this case, all modifications involving the class hierarchy can be collected, that is saved as a byproduct of the ontology evolution process or even extracted a posteriori by means of some change detection tool (Grandi, 2012, Sec. 3.4; Zablith et al., 2013, Sec. 7.1.2), to be applied to our accessory data structure, as it has been described in this work.

In order to efficiently manage the evolution of the ontology class hierarchy, in this paper and in its precursor (Gauch et al., 2003), we proposed to use a temporal relation. Several papers dealt with efficient storage of ontologies in a relational database, including the SOR (Lu et al., 2007), Minerva (Heymans et al., 2008) and ROSM (Zhou & Yongkang, 2011) approaches. All these approaches can be adapted to store multi-version ontologies by making temporal the involved relations; they also provide relations for the storage and management of ontology instances, whereas the ontologies used for personalization as considered in this work (e.g., derived from the SNOMED-CT terminologies) are not equipped with instances; the only table, out of the fourteen in the schema of (Zhou & Yongkang, 2011) and more than three dozens used in (Lu et al., 2007; Heymans et al., 2008), which is relevant for the personalization method considered in this work is the one storing the "SubClassOf" information. The TreeRelation considered in this paper substitutes (or complements) the information stored in a "SubClassOf" relation made temporal.

8. Conclusions

In this paper, we extend the framework for ontology-based personalized access to clinical guidelines introduced in (Grandi et al., 2012) in the direction of multi-version ontologies. The proposal aims at limiting computational and storage costs when the ontologies used for personalized access undergo modifications and there is the need of maintaining all the ontology versions for personalized access. It involves the introduction of a novel numbering scheme and efficient technologies for ontology evolution and query processing. Efficiency and scalability of the approach has been evaluated by means of experiments conducted on a prototype of the personalization engine.

In our future work, we will consider the managing of graph-like ontologies. In this way, we will support multiple inheritances among ontology classes, thus including a wide range of healthcare ontologies (Zhang et al., 2004).

Further work will also include the assessment of our proposal in a concrete working environment, with a repository of real clinical guidelines and real users evaluating the clinical utility of the proposed guideline multidimensional versioning and personalization methods, which is a fundamental step in developing a tool aimed at improving and optimizing healthcare activities.

Acknowledgement

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

Berglund, A., Boag, S., Chamberlin, D., Fernández, M.F., Kay, M., Robie, J., et al. (2011). XML Path Language (XPath) 2.0 (Second Edition), W3C Recommendation. Retrieved September 1, 2014 from http://www.w3.org/TR/xpath20/.

Boxwala, A.A., Zeng, Q., Tate, D., Greenes, R.A. & Fairchild, D.G. (2002). Applying Axiomatic Design Methodology to Create Guidelines that are Locally Adaptable. In Proc. *AMIA*, San Antonio, TX, p. 980.

Callan, J., Smeaton, A., Beaulieu, M., Borlund, P., Brusilovsky, P., Chalmers, M. et al. (2003). Personalization and recommender systems in digital libraries, in: *Joint NSF-EU DELOS Working Group Report*, ERCIM, Sophia Antipolis, France. Retrieved September 1, 2014 from http://www.ercim.org/publication/ws-proceedings/Delos-NSF/Personalisation.pdf.

Cantador, I., Bellogín, A. & Castells, P. (2008). A Multilayer Ontology-based Hybrid Recommendation Model, *AI Communications*; 21 (2), pp. 203–210.

Erhardt, L.R. & Gotto, A. (2006). The evolution of European guidelines: Changing the management of cholesterol levels, *Atherosclerosis*; 185 (1), pp. 12-20.

Field, M.J. & Lohr, K.N. (Eds.). (1990). *Clinical Practice Guidelines: Directions for a New Program*, National Academy Press, Washington, DC.

Fridsma, D.B., Gennari, J.H. & Musen, M.A. (1996). Making Generic Guidelines Site-specific. In Proc. *AMIA*, Washington, DC, pp. 597–601.

Gauch, S., Chaffee, J. & Pretschner, A. (2003). Ontology-Based User Profiles for Search and Browsing, *Web Intelligence and Agent Systems*; 1 (3-4), pp. 219–234.

Grandi, F. (2012). Introducing an Annotated Bibliography on Temporal and Evolution Aspects in the Semantic Web, *ACM SIGMOD Record* 2012; 41 (4), pp. 18–21.

Grandi, F. (2013). Dynamic Multi-version Ontology-based Personalization, Proc. *Second International Workshop on Querying Graph Structured Data (GraphQ)*, pp. 224–232, Genoa, Italy, ACM Press, New York.

Grandi, F. & Scalas, M.R. (2009). The Valid Ontology: a Simple OWL Temporal Versioning Framework, Proc. *SEMAPRO - Intl' Conf. on Advances in Semantic Processing*, Sliema, Malta, pp. 98-102, IEEE Computer Society Press.

Grandi, F., Mandreoli, F. & Martoglia, R. (2009). Issues in Personalized Access to Multi-version XML Documents, in: Pardede E (Ed.). *Open and Novel Issues in XML Database Applications*, Information Science Reference Series, IGI Global, Hershey, pp. 199–230.

Grandi, F., Mandreoli, F. & Martoglia, R. (2012). "Efficient management of multi-version clinical guidelines, *Journal of Biomedical Informatics*; 45(6), pp. 1120–1136.

Grandi, F., Mandreoli, F., Martoglia, R., Ronchetti, E., Scalas, M.R. & Tiberio, P. (2009). Ontology-Based Personalization of E-Government Services, in *Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies*, Germanakos P and Mourlas C, Eds. Information Science Reference Series. IGI Global, Hershey, PA, pp. 167–187.

Groot, P., Hommerson, A. & Lucas, P. (2008) Adaptation of Clinical Practice Guidelines. In Ten Teije A, Miksch S, and Lucas P (Eds.). Computer-based Medical Guidelines and Protocols: A Primer and Current Trends. *Studies in Health Technology and Informatics 139*. Amsterdam, The Netherlands: IOS Press, pp. 121–139.

Heymans, S., Ma, L., Anicic, D., Ma, Z., Steinmetz, N., Pan, Y. et al. (2008). Ontology Reasoning with Large Data Repositories, in: M. Hepp, P. De Leenheer, A. de Moor, Y. Sure (Eds.), *Ontology Management*, CHE 7, Springer Verlag, Berlin, pp. 89–128.

Jensen, C.S., Dyreson, C.E., Böhlen, M.H., Clifford, J., Elmasri, R., Gadia, S.K. et al. (1998). The Consensus Glossary of Temporal Database Concepts - February 1998 Version, In *Temporal Databases, Research and Practice*, Etzion O, Jajodia S and Sripada S, Eds. LNCS, Vol. 1399. Springer Verlag, Heidelberg, Germany, pp. 367–405.

Kaiser, K. & Miksch, S. (2009). Versioning Computer-interpretable Guidelines: Semi-automatic Modeling of 'Living Guidelines' Using an Information Extraction Metod. *Artificial Intelligence in Medicine*; 46, pp. 55–66.

Lu, J., Ma, L., Zhang, L., Brunner, J.S., Wang, C. & Pan, Y. (2007). A Practical System for Ontology Storage, Reasoning and Search, in: *33rd Int. Conf. on Very Large Data Bases (VLDB)*, ACM Press, Vienna, Austria, pp. 1402–1405.

Mackey, T.K. & Liang, B.A. (2011). The Role of Practice Guidelines in Medical Malpractice Litigation. *Virtual Mentor* 13, 1, 36–41. Retrieved April 1, 2015 from http://virtualmentor.ama-assn.org/2011/01/hlaw1-1101.html.

Mandreoli, F., Martoglia, R. and Ronchetti, E. (2006). Supporting Temporal Slicing in XML Databases. In Proc. *EDBT*, Munich, Germany, 2006, pp. 295–312.

Mello, M.M. (2001). Of Swords and Shields: The Role of Clinical Practice Guidelines in Medical Malpractice Litigation, *University of Pennsylvania Law Review*; 149 (3), pp. 645–710.

Middleton, S.E., Shadbolt, N. & De Roure, D.C. (2004). Ontological user profiling in recommender systems, *ACM Transactions on Information Systems*; 22 (1), pp. 54–88.

Moreno, A., Valls, A., Isern, D., Marina, L. & Borràs, J. (2013). SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities, *Engineering Applications of Artificial Intelligence*; 26 (1), pp. 633–651.

Owens, D.K. & Jr Nease, R.F. (1997). A normative analytic framework for development of practice guidelines for specific clinical populations. *Medical Decision Making*; 17, pp. 409−426.

Peleg, M. & Kantor, R. (2003). Approaches for Guideline Versioning using GLIF. In Proc. *AMIA*, Washington, DC, pp. 509–513.

Peleg, M., Shahar, Y. & Quaglini, S. (2013). Making healthcare more accessible, better, faster, and cheaper: the MobiGuide Project. Eur. J. ePractice 20, pp. 5–20. <http://www.mobiguide-project.eu/images/ePractices_Making_eHealth.pdf >

Pretschner, A. (1998.) *Ontology based personalized search*. Unpublished master's thesis, University of Kansas, Lawrence, Kan.

Riaño, D., Real, F., López-Vallverdú, J. A. F. , Ercolani, S., Mecocci, P., Annicchiarico, R. et al. (2012). An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients, *Journal of Biomedical Informatics*; 45 (3), pp. 429–446.

Riecken, D. (2000). Personalized views of personalization. *Communications of the ACM*, 43 (8), pp. 27–28.

Sanders, G.D. (1998). *Automated creation of clinical-practice guidelines from decision models*. Ph.D. Thesis, Stanford Medical Informatics, Stanford University, CA.

Sanders, G.D., Jr Nease, R.F. & Owens, D.K. (2001). Publishing Web-based Guidelines using Interactive Decision Models. *Journal of Evaluation in Clinical Practice*; 7 (2), pp. 175–189.

Scott-Wright, A.O., Fischer, R.P., Denekamp, Y. & Boxwala, A.A. (2004). A Methodology for Modular representation of Guidelines. In Proc. *MEDINFO*, pp. 149−153

Shahar, Y., Miksch, S. & Johnson, P. (1998). The Asgaard Project: A Task-specific Framework for the Application and Critiquing of Time-oriented Clinical Guidelines. *Artificial Intelligence in Medicine*; 14, pp. 29–51.

Shahar, Y., Young, O., Shalom, E., Galperin, M., Mayaffit, A., Moskovitch, R. et al. (2004). A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools, *Journal of Biomedical Informatics*; 37(5), pp. 325-344.

Shiffman, R.N., Karras, B.T., Agrawal, A., Chen, R., Marenco, L. & Nath, S. (2000). GEM a Proposal for a More Comprehensive Guideline Document Model using XML. *Journal of the American Medical Informatics Association*; 7 (5), pp. 488–497.

Sieg, A., Mobasher, B. & Burke, R.D. (2007). Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search, *IEEE Intelligent Information Bulletin*; 8 (1), pp. 7–18.

Terenziani, P., Montani, S., Bottrighi, A., Molino, G. & Torchio, M. (2005). Clinical Guidelines Adaptation: Managing Authoring and Versioning Issues. In Proc. *AIME*, Aberdeen, Scotland, 2005, pp. 151–155.

Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G. & Correndo, G. (2004). A Context-adaptable Approach to Clinical Guidelines. In Proc. *MEDINFO*, San Franscisco, CA, pp. 169–173

Tu, S. W., Peleg, M., Carini, S., Bobak, M., Ross, J., Rubin, D. et al. (2011). A practical method for transforming free-text eligibility criteria into computable criteria, *Journal of Biomedical Informatics*; 44 (2), pp. 239–250.

W3C Consortium. (2015). Extensible Markup Language (XML) Home Page, Retrieved April 1, 2015 from http://www.w3.org/XML/.

W3C Consortium. (2015). Web Ontology Language (OWL) Home Page, Retrieved April 1, 2015 from http://www.w3.org/2004/OWL/

W3C Consortium. (2015). XML Query Language (XQuery) Home Page, Retrieved April 1, 2015 from http://www.w3.org/XML/Query/.

Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E. et al. (2013). Ontology Evolution: A Process Centric Survey, *Knowledge Engineering Review*; 30 (1), pp. 45–75.

Zhang, L., Perl, Y., Halper, M., Geller, J. & Cimino, J.J. (2004). An Enriched Unified Medical Language System Semantic Network with a Multiple Subsumption Hierarchy, *Journal of the American Medical Informatics Association*; 11 (3), pp. 195–206

Zheng, S., Wang, F. & Lu, J. (2014). Enabling Ontology Based Semantic Queries in Biomedical Database Systems, *International Journal of Semantic Computing*; 8 (1), pp. 67–83.

Zhou, Z. & Yongkang, X. (2011). A study on ontology storage based on relational database, in: *13Th IEEE Joint Int. Computer Science and Information Technology Conf. (JICSIT)*, IEEE Computer Society Press, Chongqing, China, pp. 1–5.