2009 Third International Conference on Advances in Semantic Processing

The Valid Ontology: a Simple OWL Temporal Versioning Framework

Fabio Grandi and Maria Rita Scalas
Dipartimento di Elettronica Informatica e Sistemistica
Alma Mater Studiorum – Università di Bologna
Bologna, Italy
e-mail: {fabio.grandi, mariarita.scalas}@unibo.it

***Abstract*—In this paper, we consider the problem of managing multiple temporal versions of an ontology. For example, in the legal and e-Government domains, temporal versions of an ontology are a natural consequence of the dynamics involved in normative systems. To this purpose, we introduce "The Valid Ontology" approach, by adapting to OWL-encoded ontologies a versioning scheme we proposed in the past for generic Web resources. In particular, we propose to use a single temporal XML document to represent and store a multi-version ontology and use a temporal XML query processor to efficiently extract valid OWL ontologies from the XML document as temporal snapshots. The result is an efficient ontology temporal versioning solution, relying on standard XML technology.**

Keywords-ontology; temporal versions; timestamp; OWL; XML

I. Introduction

In previous work, we studied the design and implementation of Web information systems for e-Government applications [8,10,11]. In particular, we proposed the adoption of temporal database and Semantic Web techniques to provide personalized access to multi-version resources and services provided by the Public Administration. The offering of personalized versions is aimed at improving and optimizing the involvement of citizens in the e-Governance process. In particular, we considered the selective access to norm texts and documents made available on Web repositories in XML format [25], where personalization is supported by means of an underlying ontology [8]. However, as explained in details in the section which follows, temporal versioning of the ontology is required to provide the correct temporal perspective to personalization in such a context, and has not been considered before. In order to fill this gap, we present in this paper "The Valid Ontology", a simple solution to the ontology temporal versioning problem, relying on standard XML technology. In particular, it is based the adoption of a single temporal XML document [5] to represent a multi-version ontology, from which single ontology versions can be extracted as snapshots via a suitable preprocessing with the help of a temporal XML query engine.

The paper is organized as follows. Section II is devoted to the description of the application scenario, which provides motivation and background to this work. In Section III, we present "The Valid Ontology", our simple approach to represent and manage multi-version OWL ontologies [23]. Related work will be briefly discussed in Section IV, whereas conclusions and future work directions can be finally found in Section V.

II. Motivaton

In the e-Government framework, the fast dynamics involved in normative systems implies the coexistence of multiple temporal versions of the norm texts stored in a repository, since laws are continually subject to amendments and modifications. In fact, it is crucial to reconstruct the consolidated version of a norm as produced by the application of all the modifications it underwent so far, that is the form in which it currently belongs to the regulations and must be enforced today. However, also past versions are still important, not only for historical reasons: for example, if a Court has to pass judgment today on some fact committed in the past, the version of norms which must be applied to the case is the one that was in force then. In other words, temporal concerns are widespread in the e-Government domain and a legal information system should be able to retrieve or reconstruct on demand any version of a given document to meet common application requirements. Hence, personalization in such a context is based on the user's temporal perspective.

Furthermore, another kind of versioning plays an important role in an e-Government scenario, because some documents or some of their parts have or acquire a limited applicability. For example, a given norm (e.g., defining tax treatment) may contain some articles which are applicable to different classes of citizens: one article is applicable to unemployed persons, one article to self-employed persons, one article to public servants only and so on. Hence, a citizen accessing a retrieval service may be interested in finding a tailored version of the norm, that is a version only containing articles which are applicable to his/her personal case. Hence, personalization in such a context is based on limited applicability to the citizen's case and semantic versioning is required to the document repository. Finally, notice that temporal and limited applicability aspects though orthogonal may also interplay in the production and management of versions. For instance, a new norm might state a modification to a preexisting norm, where the modified norm becomes applicable to a limited category of citizens only (e.g., retired persons), whereas the rest of the citizens remain subject to the unmodified norm.

In such an application context, personalization is based on a special kind of ontology-based user profiling, where

978-0-7695-3833-4/09 $26.00 © 2009 IEEE
DOI 10.1109/SEMAPRO.2009.12

citizens are classified according to their position before the law in order to find out the exact resources which apply to their individual case [11]. To this purpose, we introduced a *civic ontology*, which corresponds to a classification of citizens based on the distinctions introduced by successive norms (*founding acts*) that imply some limitation, total or partial, in their applicability [8]. For instance, Fig. 1 depicts a portion of a civic ontology built from a small corpus of norms ruling the status of citizens with respect to their work position. Hence, the XML encoding of norm documents can be extended with semantic annotations linking the text portions with the ontology classes they are applicable to and which can be used by a suitable query engine to produce a personalized version of the document.

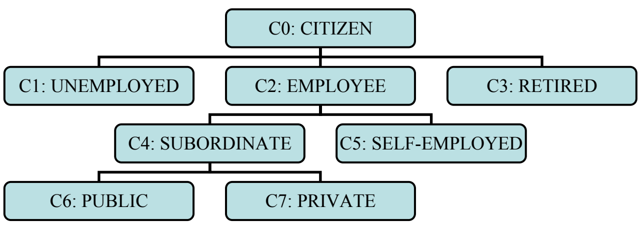

Figure 1. A sample portion of a civic ontology.

However, in our previous work we did not take into account the fact that also the civic ontology definition itself is subject to modifications due to the dynamics of the normative systems, because also the ontology founding acts possibly undergo amendments and modifications. Therefore, exact positioning of a citizen in the framework of the regulations valid at a given time in the past should be effected, for consistency reason, by classifying the citizen with reference to the version of the civic ontology which was valid at the same time point. In other words, the right temporal perspective must be given also to the citizen classification process. For example, assume that, in a given state, a law in force from January 1, 1990 has changed the definition of "adult" from a 21-year old person to an 18-year old person. Since adults have a different status before the law with respect to juveniles (e.g., concerning the possibility to legally own a gun, have sex, drink alcohol, drive a car, etc.), their behavior might be evaluated right or wrong according to their legal age. Moreover, being an adult or not could be considered either an extenuating or an aggravating circumstance when considering an illegal behavior. Hence, a Court which has to pass judgment today on a crime committed in the past, not only must apply the version of norms applicable to the crime which were in force then, but also classify the citizen with respect to the version of the civic ontology which was valid then in order to find out the right norms which are properly applicable to the case. For example, it is important to ascertain whether an accused person, who was 20-year old at the time of the crime, was actually already an adult or not. If the crime has been committed before 1990, the answer is negative, according to the ontology version valid at the time of the crime, which must be used for a correct classification.

Therefore, the framework assumed in our previous works must be extended to include representation and storage of ontologies in multi-version format and a query facility to extract a valid ontology as a temporal snapshot from the multi-version repository.

III. "The Valid Ontology" Approach

In order to introduce multi-version temporal ontologies, we propose to apply a scheme quite similar to "The Valid Web" approach we introduced in [7] for generic World Wide Web resources (e.g., HTML pages). We assume ontologies are defined using the OWL language [23] or a sublanguage of it (which was a requirement in the e-Government domain) and represented as RDF/XML documents. In particular, we propose to add a custom XML markup to OWL documents which allows us to mark the boundaries of versioned portions and add timestamps to versions.

Let us work on our example of adult age change introduced on January 1, 1990. We assume the "adult age" before and after the change correspond to the data types over20 and over17, respectively, defined as restrictions of the positive integer domain in an XSD Schema [26] as shown in Fig. 2.

```
<xsd:scheme>
  <xsd:simpleType name=”over20”>
    <xsd:restriction base=”xsd:positiveInteger”>
      <xsd:minInclusive value=”21” />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:scheme>

<xsd:scheme>
  <xsd:simpleType name=”over17”>
    <xsd:restriction base=”xsd:positiveInteger”>
      <xsd:minInclusive value=”18” />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:scheme>
```

Figure 2. Example of "over17" and "over20" XSD data type definitions.

Hence, the "Adult" class definition in the multi-version civic ontology can be defined as shown in Fig. 3, that is as consisting of two versions, the former equivalent to:

Adult = Person $\sqcap$ $\exists$age.over20 (valid from 1900 to 1989)

and the latter equivalent to:

Adult = Person $\sqcap$ $\exists$age.over17 (valid from 1990 on).

In "The Valid Ontology" approach, both class and property definitions can be versioned and timestamped, also allowing IS-A hierarchies to arbitrarily evolve between versions. However, we presented here a versioning example involving XSD Schema types (actually not part of the OWL standard) only for the sake of providing a reasonably compact, significant and easy to understand example.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf= …… >
<owl:Ontology rdf:about="" />
…
<owl:Class rdf:ID="Adult">

  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#age" />

      <version num="1">
      <timeStamp from="1900-01-01"
                 to="1989-12-31" />
        <owl:someValuesFrom rdf:resource=
              "myNamespace/xsdExample#over20" />
      </version>

      <version num="2">
      <timeStamp from="1990-01-01"
                 to="9999-99-99" />
        <owl:someValuesFrom rdf:resource=
              "myNamespace/xsdExample#over17" />
      </version>

    </owl:Restriction>
  </owl:intersectionOf>

</owl:Class>

…
</rdf:RDF>
```

Figure 3. Fragment of a versioned ontology definition.

Notice that the resulting temporal XML [5] document does not represent a legal OWL (or RDF) ontology definition, as new tags "version" and "timeStamp" are not legal OWL elements and, thus, they would not be recognized by any OWL processor. However, our proposed multi-version format allows us to represent multiple ontology versions in a compact temporal XML form. Then, when a single ontology version (e.g., the one valid at a given date) is required, a temporal XML processor can be used to extract the desired snapshot. In particular, the XML temporal processor we developed in our previous work [10] can very efficiently be used to this purpose. For instance, in order to extract the version valid on March 12, 1990, the query shown in the Fig. 4 which follows (expressed in XQuery [27] syntax) can be issued.

```
FOR $a IN document("ontocivic_versioned.xml")
WHERE tempConstr( "from<='1990-03-12'
                  and to>='1990-03-12'" )
RETURN $a
```

Figure 4. Temporal query to reconstruct the ontology snapshot valid on 1990, March 12.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf= …… >
<owl:Ontology rdf:about="" />
…
<owl:Class rdf:ID="Adult">

  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#age" />

        <owl:someValuesFrom rdf:resource=
              "myNamespace/xsdExample#over17" />

    </owl:Restriction>
  </owl:intersectionOf>

</owl:Class>

…
</rdf:RDF>
```

Figure 5. The ontology version retrieved by the query in Fig. 4.

The query results would be as shown in Fig. 5, and can be interpreted as a regular OWL ontology definition by any OWL-compliant tool like a reasoner.

However, one of the main advantages of "The Valid Ontology" approach is that it can be deployed with standard XML technology, as it was for the original "The Valid Web" approach. In our personalization framework, we could use –at no additional cost– the available XML query engine used for retrieving norm documents to also extract temporal snapshots of the ontology from the multi-version repository. If the ontology definition is not very large indeed, off-the-shelf XML tools can also be used to extract snapshots. For example, even an XSL style sheet quite similar to the one proposed in "The Valid Web" approach can be used to extract temporal snapshots from the multi-version XML ontology. Else, a commercial XQuery engine can be used as well, as it was in our first implementation of the e-Government personalization framework based on a "stratum" approach [11]. Although we clearly showed that the "stratum" approach is outperformed in almost any query setting by the "native" XML processor (based on partitioned storage and temporal-aware holistic twig join algorithms [19]), performance is not actually a matter of concern when the multi-version ontology is not very large. Notice that preliminary investigations showed that, in this application domain, the real problem is the management of the versioned extant XML norm documents, whose size exceeds by orders of magnitude the size of the civic ontology in RDF/XML format. Hence the query engine designed to search the norm repository can also be used for the ontology version management with an almost negligible overhead.

In this paper, for the sake of simplicity, we only consider a single temporal dimension (i.e. valid time [5,17]) for versioning, although other temporal dimensions could be easily added. For instance, in the e-Government scenario, also *efficacy*, *publication* and *transaction time* dimensions are meaningful and could be taken into account [9]. The advantage of a compact representation of multiple versions in a single XML document as part of "The Valid Ontology" approach is even magnified by the introduction of additional time dimensions, with respect to separate storage of individual versions, where the number of versions explodes due to changes along different time dimensions and unmodified ontology parts are duplicated along consecutive versions.

IV. RELATED WORK

Several authors in the Semantic Web field previously considered ontology evolution and versioning problems [1,2,4,6,13,14,15,18,20,21]. Some works focused on modeling, implementing and detecting changes in the framework of an ontology management system [6,18,20,21]. Other works stressed the consistency problem in the presence of ontology evolution [4,13]. Some authors used ontology versioning to support different viewpoints or contexts over the same extant data through different ontology versions [1,2,15].

In our approach, the existence of multiple temporal versions is a requirement of the application field and not a feature of the ontology authoring environment or management system to be kept under control. Moreover, the only consistency which has to be enforced in an e-Government personalization framework is bound to the temporal perspective. As a matter of fact, the temporal perspective has to be the same in the choice of the ontology version and of the data resources indexed by the ontology classes, which must be valid at the same time (*synchronous management* [3]). In particular, prospective and retrospective use of ontologies [18] is not allowed. Other consistency problems generally considered for ontology evolution, like consistency between different versions of the same ontology, are not relevant in this context: in the legal domain, successive ontology versions can actually be mutually inconsistent as changes are a consequence of authority decisions taken by human beings, the lawmakers, for whom consistency and even rationality is not usually a priority. Inconsistency between successive versions can arbitrarily and deliberately be introduced, possibly reflecting a change of mind or a different political view on some state of affairs, or simply being a unwelcome side-effect. The temporal perspective, which requires to retrieve the temporal version of the ontology and of the related resources valid at a time point in the past, can be considered as a special kind of viewpoint/context.

On the other hand, temporal versioning of ontologies has also been explicitly considered by other authors [12,16]. However, although such proposals provide a solid semantic foundation and introduce languages for temporal reasoning on multi-version ontologies, they do not seem suitable to efficiently support single version extraction as required in an ontology-based personalization environment like the one we devised for e-Government applications.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented "The Valid Ontology", a simple framework to represent and store multiple temporal versions of an ontology in a compact temporal XML format and efficiently extract ontology snapshots from the multi-version XML document via a temporal XML processor. The proposed solution completes a personalization platform for e-Government applications we previously designed to support ontology-based personalized access to e-Government resources, where personalization is based on a user-defined temporal perspective and applicability constraints embedded in semantic annotations [11]. Temporal versioning of the underlying ontology is necessary to ensure consistency between the temporal perspective and the evolution of applicability constraints.

The main advantage of "The Valid Ontology" approach is that it can be deployed with standard XML technology. In our personalization framework, we could use −at no additional cost− the available XML query engine also to extract temporal snapshots of the ontology form the multi-version repository. In our future work, we will also consider management of temporal versions of large ontologies built using triple-store technology [22], which is suitable to implement scalable architectures for semantics based information systems. RDF triple representation can be easily extended with timestamps in order to capture temporal versioning. Another option we plan to consider is the introduction of version timestamps through the annotation feature in OWL 2 [24] ontology specifications.

Future work will also be devoted to a more complete assessment and performance evaluation of "The Valid Ontology" approach in a real e-Government scenario. Multi-version ontologies (built from a corpus of Italian norms concerning education) are currently under development to this purpose. Other application fields where "The Valid Ontology" approach could be used or adapted (e.g., management of Clinical Guidelines [10]) will also be explored.

REFERENCES

[1] A. Arara and D. Benslimane, "Towards formal ontologies requirements with multiple perspectives," Proc. Intl' Conf. on Flexible Query Answering Systems (FQAS 2004), Springer-Verlag, . Jun. 2004, pp. 150-160.

[2] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini and H. Stuckenschmidt, "Contextualizing ontologies," J. Web Semantics, Vol. 1, No. 4, Oct. 2004, pp. 325-343.

[3] C. De Castro, F. Grandi and M.R. Scalas, "Schema versioning for multitemporal relational databases," Information Systems, Vol. 22, No. 5, 1997, pp. 249-290.

[4] P. De Leenheer, "Revising and managing multiple ontology versions in a possible worlds setting," Proc. OTM Workshops, Springer-Verlag, Oct. 2004, pp. 798-809.

[5] C.E. Dyreson and F. Grandi, "Temporal XML," in Encyclopedia of Database Systems, L. Liu and M.T. Öszu, Eds. Heidelberg: Springer-Verlag, in press.

[6] J. Eder and C. Koncilia, "Modelling changes in ontologies," Proc. OTM Workshops, Springer-Verlag, Oct. 2004, pp. 662-673.

[7] F. Grandi, F. Mandreoli, "The Valid Web: an XML/XSL infrastructure for temporal management of Web documents," Proc.

Intl’ Conf. on Advances in Information Systems (ADVIS 2000), Springer-Verlag, Oct. 2000, pp. 294-303.

[8] F. Grandi, F. Mandreoli, M.R. Scalas and P. Tiberio, "Management of the citizen's digital identity and access to multi-version norm texts on the semantic web", Proc. Intl' Symposium on Challenges in the Internet and Interdisciplinary (IPSI Pescara 2004), IPSI, Jul.-Aug. 2004.

[9] F. Grandi, F. Mandreoli, P. Tiberio, "Temporal modelling and management of normative documents in XML format", Data & Knowledge Engineering, Vol. 54, No. 3, Sept. 2005, pp. 327–354.

[10] F. Grandi, F. Mandreoli, R. Martoglia, “Issues in personalized access to multi-version XML documents,” in Open and Novel Issues in XML Database Applications, E. Pardede, Ed. Hershey, PA: IGI Global, 2009, pp. 199-230.

[11] F. Grandi, F. Mandreoli, R. Martoglia, E. Ronchetti, M.R. Scalas and P. Tiberio, “Ontology-based personalization of E-government services,” in Intelligent User Interfaces, C. Mourlas and P. Germanakos, Eds. Hershey, PA: IGI Global, 2009, pp. 167-203.

[12] C. Gutierrez, C. Hurtado and A. Vaisman, “Temporal RDF,” Proc. European Semantic Web Conference (ESWC 2005), Springer-Verlag, May-Jun. 2005, pp. 93-107.

[13] P. Haase and L. Stojanovic, “Consistent evolution of OWL ontologies,” Proc. European Semantic Web Conference (ESWC 2005), Springer-Verlag, May-Jun. 2005, pp. 182-197.

[14] P. Haase, Y. Sure and D. Vrandečić, “Ontology management and evolution - survey, methods and prototypes,” EU-IST IP Project Sekt, Deliverable D3.1.1, http://www.sekt-project.com/rd/deliverables/

[15] J. Heflin and Z. Pan, “A model theoretic semantics for ontology versioning,” Proc. Intl' Semantic Web Conf. (ISWC 2004), Springer-Verlag, Nov. 2004, pp. 62-76.

[16] Z. Huang and H. Stuckenschmidt, “Reasoning with multi-version ontologies: a temporal logic approach,” Proc. Intl' Semantic Web Conf. (ISWC 2005), Springer-Verlag, Nov. 2005, pp. 398-412.

[17] C.S. Jensen et al., “The Consensus Glossary of Temporal Database Concepts - February 1998 Version,” in Temporal Databases: Research and Practice, O. Etzion, S. Jajodia, S.M. Sripada Eds. Heidelberg: Springer-Verlag, 1998, pp. 367-405.

[18] M. Klein and D. Fensel, “Ontology versioning on the Semantic Web,” Proc. Semantic Web Working Symposium (SWWS 2001), Aug. 2001, pp. 75-91.

[19] F. Mandreoli, R. Martoglia and E. Ronchetti, “Supporting temporal slicing in XML databases,” Proc. Intl’ Conf. on Extending Database technology (EDBT 2006), Springer-Verlag, Mar. 2006, pp. 295-312.

[20] N.F. Noy and M.A. Musen, “Ontology versioning in an ontology management framework,” IEEE Intelligent Systems, Vol. 19, No. 4, Jul.-Aug. 2004, pp. 6-13.

[21] T. Redmond, M. Smith, N. Drummond and T. Tudorache, “Managing change: an ontology version control system,” Proc. Workshop on OWL: Experiences and Directions (OWLED 2008), CEUR-WS, Oct. 2008.

[22] K. Rohloff, M. Dean, I. Emmons, D. Ryder and J. Summer, “An evaluation of triple-store technologies for large data stores,” Proc. OTM Workshops, Springer-Verlag, Nov. 2007, pp. 1105-1114.

[23] OWL, The Web Ontology language Home Page. W3C Consortium, http://www.w3c.org/2004/OWL/

[24] B. Motik, P.F. Patel-Schneider and B. Parsia (eds.), “OWL 2 Web ontology language,” W3C Candidate Recommendation, Jun. 2009, http://www.w3.org/TR/owl2-syntax/

[25] XML, The eXtensible Markup Language Home Page. W3C Consortium, http://www.w3c.org/XML/

[26] XMLSchema, The XML Schema Home Page. W3C Consortium, http://www.w3c.org/XML/Schema/

[27] XQuery, The XML Query Home Page. W3C Consortium, http://www.w3c.org/XML/Query/