Personalized access to multi-version XML documents in an eGovernment scenario (Extended Abstract)

Fabio Grandi[1], Federica Mandreoli[2], Riccardo Martoglia[2], Enrico Ronchetti[2],
Maria Rita Scalas[1], and Paolo Tiberio[2]

[1] Alma Mater Studiorum – Università di Bologna, Italy
{fgrandi,mrscalas}@deis.unibo.it
[2] Università di Modena e Reggio Emilia, Italy
{fmandreoli,rmartoglia,eronchetti,ptiberio}@unimo.it

Abstract. In this paper, we present some results of an ongoing research involving the design and implementation, in an eGovernment scenario, of a multi-version repository of norm texts supporting efficient and personalized access. In particular we defined a multi-version XML data model supporting both temporal versioning –essential in normative systems– and semantic versioning. Semantic versioning is based on the applicability of different norm parts to different classes of citizens and allows users to retrieve personalized norm versions only containing provisions which are applicable to their personal case. We describe the organization and present preliminary performance figures of a prototype system we developed.

1 Introduction

Nowadays we are witnessing a strong institutional push towards the implementation of eGovernment support services, aimed at a higher level of integration and involvement of the citizens in the Public Administration (PA) activities that concern them. In this framework, collections of norm texts and legal information presented to citizens are made available and are becoming popular on the internet. The offering of personalized versions is aimed at improving and optimizing the involvement of citizens in the eGovernance process. In existing systems, personalization is either absent (e.g. `www.normeinrete.it`) or predefined by human experts and hardwired in the repository structure (e.g. `www.italia.gov.it`), whereas flexible and on-demand personalization services are lacking.

In this challenging scenario takes its place the research activity entitled "Semantic web techniques for the management of digital identity and the access to norms", which we are carrying out as part of the PRIN national project "European Citizen in eGovernance: legal-philosophical, legal, computer science and economical aspects" [3]. One of the main objectives of such activity is the development of techniques allowing an effective and efficient access to multi-version norm repositories supporting temporal queries and personalization facilities. First of all, 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. For instance, it is crucial to reconstruct the consolidated version of a norm as produced by the application of all the modifications it underwent so far. Moreover, another kind of versioning plays an important role, because some norms or some of their parts have or acquire a limited applicability. For example, a given norm may contain some articles which are only applicable to particular classes of citizens (e.g. public employees). Hence, a citizen accessing the repository may be interested in finding a *personalized version* of the norm, that is a version only containing articles which are applicable to his/her personal case.

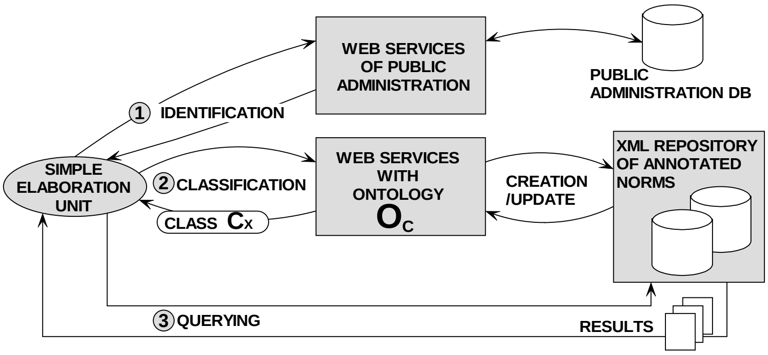


Fig. 1. The Complete Infrastructure

In this paper, we present the current achievements of our research activity concerning efficient and personalized access to multi-version XML document repositories. We defined an XML data model which combines semantic annotations with temporal versioning in order to provide a multi-versioning mechanism capturing limited applicability and supporting personalized access. We then describe and evaluate the performance of a prototype system we developed to support and test these features.

The paper is organized as follows: Sec. 2 describes the complete infrastructure involved in the research project. Sec. 3 investigates the aspects of selective access to multi-version documents, while Sec. 4 describes the implemented prototype and presents its preliminary performance evaluation. Finally, Sec. 5 concludes the paper.

2 The complete infrastructure

In order to enhance the participation of the citizens to an eGovernance procedure of interest, their automatic and accurate positioning within the reference legal framework is needed. To solve this problem we employ Semantic Web techniques and introduce a *civic ontology*, which corresponds to a classification of citizens based on the distinctions introduced by subsequent norms which imply some limitation (total or partial) in their applicability. In the following, we refer to such norms as *founding acts*. Moreover, we define the citizen's *digital identity* as the total amount of information concerning him/her –necessary for the sake of classification with respect to the ontology– which is available online [12]. Such information must be retrievable in an automatic, secure and reliable way from the PA databases through suitable Web services (*identification services*). For instance, in order to see whether a citizen is married, a simple query concerning his/her marital status can be issued to registry databases. In this way, the classification of the citizen accessing the repository makes it possible to produce the most appropriate version of all and only norms which are applicable to his/her case.

Hence, the resulting complete infrastructure is composed by various components that have to communicate between each other to collect partial and final results (see Fig. 1). Firstly, in order to obtain a personalized access, a secure authentication is required for a citizen accessing the infrastructure. This is performed through a simple elaboration unit, also acting as user interface, which processes the citizen's requests and manages the results. Then, we can identify the following phases:

- the **identification phase** (step 1 in Fig. 1) consists of calls to identification services to reconstruct the digital identity of the authenticated user on-the-fly. In this phase the system collects pieces of information from all the involved PA web services and composes the identity of the citizen.

- the citizen **classification phase** (step 2 in Fig. 1) in which the classification service uses the collected digital identity to classify the citizen with respect to the civic ontology (O_C in Fig. 1), by means of an embedded reasoning service. In Fig. 1, the most specific class C_X has been assigned to the citizen;
- finally, in the **querying phase** (step 3 in Fig. 1) the citizen's query is executed on the multi-version XML repository, by accessing and reconstructing the appropriate version of all and only norms which are applicable to the class C_X. The querying phase will be deeply analyzed in the next Section.

In order to supply the desired services, the digital identity is modelled and represented within the system in a form such that it can be translated into the same language used for the ontology (e.g. a Description Logic [2]). In this way, during the classification procedure, the matching between the civic ontology classes and the citizen's digital identity can be reduced to a standard reasoning task (e.g. ontology entailment for the underlying Description Logic [7]).

Furthermore, the civic ontology used in step 2 and 3 requires to be created and constantly maintained: each time a new founding act is enforced, the execution of a **creation/update phase** is needed. Notice that this process (and also the introduction of semantic annotations into the multi-version XML documents) is a delicate task which needs advice by human experts and "official validation" of the outcomes and, thus, it can only partially be automated. However, computer tools and graphic environments (e.g. based on the Protégé platform [11]) could be provided to assist the human experts to perform this task. For the specification of the identification, classification and creation/update services, we plan to adopt a standard declarative formalism (e.g. based on XML/SOAP [13]). The study of the services and of the mechanisms necessary to their semi-automatic specification will be dealt with in future research work.

3 Personalized access to versions

Our research is currently focused on the querying phase described in Sec. 2. In particular, we defined efficient techniques for querying repositories storing legal documents supporting temporal and semantic versioning.

Temporal concerns are widespread in the eGovernment 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. In fact, whereas it is crucial to reconstruct the current (consolidated) version of a norm as it is the one that currently belongs to the regulations and must be enforced today, 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. Temporal versioning aspects are examined in Subsection 3.1. We then extend the temporal framework with semantic versioning in order to provide personalized access to norm texts, as described in Subsection 3.2. Semantic versioning also plays an important role, due to the limited applicability that norms or some of their parts have or acquire. Hence, it is crucial to maintain the mapping between each portion of a norm and the maximal class of citizens it applies to in order to support an effective personalization service. 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.

3.1 Temporal Versioning

We first focused on the temporal aspects and on the effective and efficient management of time-varying norm texts. Our work on these aspects is based on our previous research experiences [4–6]. To this purpose, we developed a temporal XML data model which uses four time dimensions to correctly represent the evolution of norms in time and their resulting versioning. The considered dimensions are:

Validity time. It is the time the norm is in force. It has the same semantics of valid time as in temporal databases [8], since it represents the time the norm actually belongs to the regulations in the real world.

Efficacy time. It is the time the norm can be applied to concrete cases. While such cases do exist, the norm continues its efficacy even if no longer in force. It also has a semantics of valid time although it is *independent* from validity time.

Transaction time. It is the time the norm is stored in a computer system. It has the same semantics of transaction time as in temporal databases [8].

Publication time. It is the time of publication of the norm on the Official Journal. It has the same semantics as event time in temporal databases [9]. As a global and unchangeable norm property, it is not used as a versioning dimension.

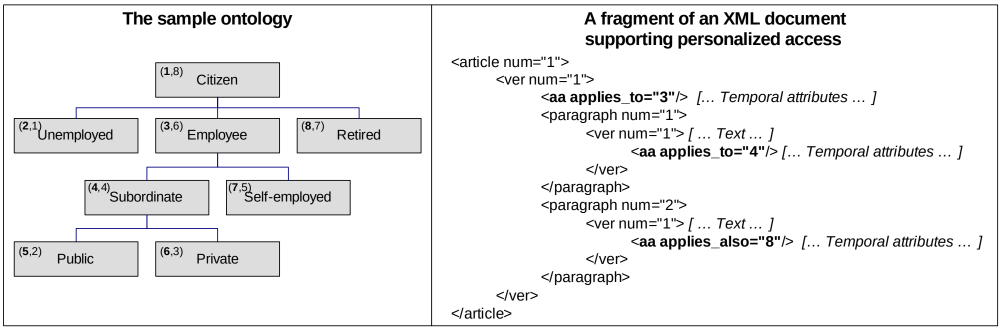
The data model was defined via an XML Schema, where the structure of norms is defined by means of a contents-section-article-paragraph hierarchy and multiple content versions can be defined at each level of the hierarchy. Each version is characterized by timestamp attributes defining its temporal pertinence with respect to each of the validity, efficacy and transaction time dimensions.

Legal text repositories are usually managed by traditional information retrieval systems where users are allowed to access their contents by means of keyword-based queries expressing the subjects they are interested in. We extended such a framework by offering users the possibility of expressing temporal specifications for the reconstruction of a consistent version of the retrieved normative acts (consolidated act). With respect to our first implementation of the temporal model, which is described in [4, 5], we deeply redesigned the overall system architecture, the document storage scheme and the query processing methods in order to improve efficiency. The redesign also took into account the new problems arising from the extension to support semantic versioning. The new system organization will be described in Section 4, whereas a detailed comparison between the two architectures can be found in [10].

3.2 Semantic Versioning

The temporal multi-version model described above has then been enhanced to include a semantic versioning mechanism to provide personalized access, that is retrieval of all and only norm provisions that are applicable to a given citizen according to his/her digital identity. Hence, the semantic versioning dimension encodes information about the applicability of different parts of a norm text to the relevant classes of the civic ontology defined in the infrastructure (O_C in Fig. 1). At the current stage of the research, semantic information is mapped onto a *tree-like* civic ontology, that is based on a taxonomy induced by IS-A relationships. The tree-like civic ontology is sufficient to satisfy basic application requirements as to applicability constraints and personalization services, though more advanced application requirements may need a more sophisticated ontology definition.

For instance, the left part of Fig. 2 depicts a simple civic ontology built from a small corpus of norms ruling the status of citizens with respect to their work position. The



```
<article num="1">
    <ver num="1">
        <aa applies_to="3"/> [... Temporal attributes ... ]
        <paragraph num="1">
            <ver num="1"> [ ... Text ... ]
                <aa applies_to="4"/> [... Temporal attributes ... ]
            </ver>
        </paragraph>
        <paragraph num="2">
            <ver num="1"> [ ... Text ... ]
                <aa applies_also="8"/> [... Temporal attributes ... ]
            </ver>
        </paragraph>
    </ver>
</article>
```

Fig. 2. An example of civic ontology, where each class has a name and is associated to a (pre,post) pair, and a fragment of a XML norm containing applicability annotations.

right part shows a fragment of a multi-version XML norm text supporting personalized access with respect to this ontology. As we currently manage tree-like ontologies, this allows us to exploit the pre-order and post-order properties of trees in order to enumerate the nodes and check ancestor-descendant relationships between the classes. These codes are represented in the upper left part of the ontology classes in the Figure, in the form: (pre-order,post-order). For example, the class "Employee" has pre-order "3", which is also its identifier, whereas its post order is "6". The article in the XML fragment on the right-hand-side of Fig. 2 is composed of two paragraphs and contains applicability annotations (tag *aa*).

Notice that applicability is inherited by descendant nodes unless locally redefined. Hence, by means of redefinitions we can also introduce, for each part of a document, complex applicability properties including extensions or restrictions with respect to ancestors. For instance, the whole article in the Figure is applicable to civic class "3" (tag *applies_to*) and by default to all its descendants. However, its first paragraph is applicable to class "4", which is a restriction, whereas the second one is applicable to class "8" (tag *applies_also*), which is an extension. The reconstruction of pertinent versions of the norm based on its applicability annotations is very important in an e-Government scenario. The representation of extensions and restrictions gives rise to high expressiveness and flexibility in such a context.

3.3 Accessing the right version for personalization

The queries that can be submitted to the system can contain four types of constraints: temporal, structural, textual and applicability. Such constraints are completely orthogonal and allow users to perform very specific searches in the XML norm repository. Let us focus first on the applicability constraint. Consider again the ontology and norm fragment in Fig. 2 and let John Smith be a "self-employed" citizen (i.e. belonging to class "7") retrieving the norm: hence, the sample article in the Figure will be selected as pertinent, but only the second paragraph will be actually presented as applicable. Furthermore, the applicability constraint can be combined with the other three ones in order to fully support a multi-dimensional retrieval. For instance, John Smith could be interested in all the norms ...

– which contain paragraphs (*structural constraint*) dealing with health care (*textual constraint*), ...
– which were valid and in effect between 2002 and 2004 (*temporal constraint*), ...
– which are applicable to his personal case (*applicability constraint*).

Such a query can be issued to our system using the standard XQuery FLWR syntax as follows:

```
FOR    $a IN norm
WHERE  textConstr ($a//paragraph//text(), 'health AND care')
AND    tempConstr ('vTime OVERLAPS PERIOD('2002-01-01','2004-12-31')')
AND    tempConstr ('eTime OVERLAPS PERIOD('2002-01-01','2004-12-31')')
AND    applConstr ('class_7')
RETURN $a
```

where `textConstr`, `tempConstr`, and `applConstr` are suitable functions allowing the specification of the textual, temporal and applicability constraints, respectively (the structural constraint is implicit in the XPath expressions used in the XQuery statement). Notice that the temporal constraints can involve all the four available time dimensions (publication, validity, efficacy and transaction), allowing high flexibility in satisfying the information needs of users in the eGovernment scenario. In particular, by means of validity and efficacy time constraints, a user is able to extract consolidated current versions from the multi-version repository, or to access past versions of particular norm texts, all consistently reconstructed by the system on the basis of the user's requirements and personalized on the basis of his/her identity.

4 Implementation and performance evaluation

The temporal and personalization query features have been implemented in a prototype system, which represents a complete redesign and extension of a previous system described in [4, 5]. The new architecture is based on an "XML-native" approach, as it is composed of a Multi-version XML Query Processor designed on purpose, which is able to manage the XML data repository and to support all the temporal, structural, textual and applicability query facilities in a single component. The prototype is implemented in Java JDK 1.5 and exploits ad-hoc data structures (relying on embedded "light" DBMS libraries) and algorithms which allow users to store and reconstruct on-the-fly the XML norm texts satisfying the four types of constraints. Such a component stores the XML norms not as entire documents but by converting them into a collection of ad-hoc temporal tuples, representing each of its multi-version parts (i.e. paragraphs, articles, and so on); these data structures are then exploited to efficiently perform structural join algorithms [1] we specifically devised and tuned for the temporal/semantic multi-version context. Textual constraints are handled by means of an inverted index. The improvement with respect to our first temporal prototype are manifold: the system accesses and retrieves only the strictly necessary data by querying ad-hoc and temporally-enhanced structures without accessing whole documents; hence, there is no need to build space-consuming structures such as DOM trees to process a query and only the parts which satisfy the query constraints are used for the reconstruction of the results. Furthermore, the new architecture also provides support to personalized access by handling the applicability constraints. Owing to the properties of the adopted pre- and post-order encoding of the civic classes, the system is able to very efficiently deal with applicability constraints during query processing by means of simple comparisons involving such encodings and semantic annotations.

As a consequence, we expected a high overall query processing efficiency together with low memory requirements. In order to evaluate the performance of our system, we built a specific query benchmark and conducted a number of exploratory experiments to test its behavior under different workloads. The experiments have been effected on a Pentium 4 2.5Ghz Windows XP Professional workstation, equipped with 512MB RAM and a RAID0 cluster of 2 80GB EIDE disks with NT file system (NTFS). We performed the tests on three XML document sets of increasing size: collection C1 (5,000 XML norm text documents), C2 (10,000 documents) and C3 (20,000 documents). In this paper, due to space requirements, we will present in detail the results obtained on the collection C1, then we will briefly describe the scalability performance shown on the other two collections. The total size of the collections is 120MB, 240MB, and 480MB, respectively. In all collections the documents were synthetically generated by means of an ad-hoc XML generator we developed, which is able to produce different documents compliant to our multi-version model. For each collection, the average, minimum and maximum document size is 24KB, 2KB and 125KB, respectively.

Query	**Constraints**			**Selectivity**	**Performance**
	Tm	**St**	**Tx**		**(msec)**
Q1 (Q1-A)	-	✓	✓	0.6% (0.23%)	1046 (1095)
Q2 (Q2-A)	-	✓	✓	4.02% (1.65%)	2970 (3004)
Q3 (Q3-A)	✓	✓	-	2.9% (1.3%)	6523 (6760)
Q4 (Q4-A)	✓	✓	✓	0.68% (0.31%)	1015 (1020)
Q5 (Q5-A)	✓	✓	✓	1.46% (0.77%)	2550 (2602)

Table 1. Features of the test queries and query execution time (time in msecs, collection C1)

Experiments were conducted by submitting queries of five different types (Q1-Q5). Table 1 presents the features of the test queries and the query execution time for each of them. All the queries require structural support (St constraint); types Q1 and Q2 also involve textual search by keywords (Tx constraint), with different selectivities; type Q3 contains temporal conditions (Tm constraint) on three time dimensions: transaction, valid and publication time; types Q4 and Q5 mix the previous ones since they involve both keywords and temporal conditions. For each query type, we also present a personalized access variant involving an additional applicability constraint, denoted as Qx-A in Table 1 (performance figures in parentheses).

Let us first focus on the "standard" queries. Our approach shows a good efficiency in every context, providing a short response time (including query analysis, retrieval of the qualifying norm parts and reconstruction of the result) of approximately one or two seconds for most of the queries. Notice that the selectivity of the query predicates does not impair performances, even when large amounts of documents containing some (typically small) relevant portions have to be retrieved, as it happens for queries Q2 and Q3. Our new system is able to deliver a fast and reliable performance in all cases, since it practically avoids the retrieval of useless document parts. Furthermore, consider that, for the same reasons, the main memory requirements of the Multi-version XML Query Processor are very small, less than 5% with respect to "DOM-based" approaches such as the one adopted in [5, 4]. Notice that this property is also very promising towards future extensions to cope with concurrent multi-user query processing.

The time needed to answer the personalized access versions of the Q1–Q5 queries is approximately 0.5-1% more than for the original versions. Moreover, since the applicability annotations of each part of an XML document are stored as simple integers, the size of the tuples with applicability annotations is practically unchanged (only a 3-4% storage space overhead is required with respect to documents without semantic versioning), even with quite complex annotations involving several applicability extensions and restrictions.

Finally, we only post here a comment about the performance of our current prototype in querying the other two collections C2 and C3 and, therefore, concerning the scalability of the system. We ran the same queries of the previous tests on the larger collections and saw that the computing time always grew sub-linearly with the number of documents. For instance, query Q1 executed on the 10,000 documents of collection C2 (which is as double as C1) took 1,366 msec (i.e. the system was only 30% slower);

similarly, on the 20,000 documents of collection C3, the average response time was 1,741 msec (i.e. the system was less than 30% slower than with C2). Also with the other queries the measured trend was the same, thus showing the good scalability of the system in every type of query context.

5 Conclusions

In this paper, we presented the current results of an ongoing research activity we are carrying out in the context of a national research project in order to support efficient and personalized access to multi-version XML document repositories in an eGovernment scenario. We defined a data model supporting both temporal versioning and personalized access, build a prototype system implementing the data model and evaluated the its performance through some exploratory experiments. In the future, we will strengthen the proposed approach, in particular by considering more advanced application requirements leading to a more sophisticated (e.g. graph-based) ontology definition, and by completing the required technological infrastructure with the specification and implementation of the remaining auxiliary services described in Section 2.

References

1. S. Al-Khalifa, H.V. Jagadish, J. M. Patel, Y. Wu, N. Koudas, and D. Srivastava. Structural joins: A primitive for efficient XML query pattern matching. In *Proc. of 18th ICDE Conf.*, pages 141–154, San Jose, CA, 2002.
2. F. Baader, I. Horrocks, and U. Sattler. Description Logics for the Semantic Web. *Künstliche Intelligenz*, 16(4):57–59, 2002.
3. The "Semantic web techniques for the management of digital identity and the access to norms" PRIN Project Home Page. http://www.cirsfid.unibo.it/eGov03/.
4. F. Grandi, F. Mandreoli, and P. Tiberio. Temporal modelling and management of normative documents in XML format. *Data & Knowledge Engineering*, 47, 2005 (in press).
5. F. Grandi, F. Mandreoli, P. Tiberio, and M. Bergonzini. A temporal data model and management system for normative texts in XML format. In *Proc. of 15th WIDM Conf.*, pages 29–36, New Orleans, LA, 2003.
6. F. Grandi, F. Mandreoli, P. Tiberio, and M. Bergonzini. A temporal data model and system architecture for the management of normative texts. In *Proc. of the 11th SEBD Conf.*, pages 169–178, Cetraro, Italy, 2003.
7. Ian Horrocks and Peter F. Patel-Schneider. Reducing owl entailment to Description Logic satisfiability. In *Proc. of ISWC 2003*, pages 17–29, Sanibel Island, FL, 2003.
8. C. S. Jensen, C. E. Dyreson, and (Eds.) et al. The Consensus Glossary of Temporal Database Concepts - February 1998 Version. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases — Research and Practice*, pages 367–405. Springer-Verlag, 1998. LNCS No. 1399.
9. Seung-Kyum Kim and Sharma Chakravarthy. Modeling time: Adequacy of three distinct time concepts for temporal data. In *Proc. of 12th ER Conf.*, pages 475–491, Arlington, TX, 1993.
10. F. Mandreoli, R. Martoglia, F. Grandi, and M. R. Scalas. Efficient management of multiversion XML documents foe e-Government applications. *submitted for publication*, 2005.
11. OWL plugin for Protégé. http://protege.stanford.edu/plugins/owl/.
12. S. Rodotà. Introduction to the "one world, one privacy" session. In *Proc. of 23rd Data Protection Commissioners Conf.*, Paris, France, http://www.paris-conference-2001.org/eng/contribution/rodota_contrib.pdf, 2001.
13. Web services activity. W3C Consortium, http://www.w3.org/2000/xp/Group/.