Computer Science and Information Systems 13(1):217–236 DOI: 10.2298/CSIS140808038S

Computer Aided Anonymization and Redaction of Judicial Documents

Goran Sladić[1], Stevan Gostojić[1], Branko Milosavljević[1], Zora Konjović[1] and Gordana Milosavljević[1]

[1] University of Novi Sad, Faculty of Technical Sciences, Trg D. Obradovića 6,
21000 Novi Sad, Serbia
{sladicg, gostojic, mbranko, ftn_zora, grist}@uns.ac.rs

Abstract. Public access to case law is a required prerequisite for the legal certainty and the rule of law. Nevertheless, according to the law, only authorized persons can access judgments in their non-anonymized and unredacted form. This paper proposes a computer aided method for anonymization and redaction of judgments, with an aim to improve efficiency of overall process. The anonymization and redaction procedure is based on the access control mechanism for XML documents. AKOMA NTOSO is chosen as an XML format in order to facilitate integration with other (legal) information systems, but the proposed method can be easily adapted to different document types and different XML formats. The method is verified by a prototype implementation which is validated by employees in a court of law.

Keywords: anonymization, redaction, RBAC, judicial documents, XML, AKOMA NTOSO.

1. Introduction

Since legal systems must be essentially public, that does not imply that all and any piece of information used in the judicial process must be published. At times, judges must have access to highly sensitive information that is not, and should not be made public. Some of this information may concern the parties; some may relate to non-parties such as witnesses, jurors, and victims; and some may relate to third persons not involved in any way in the legal proceeding. In addition to sensitive information concerning private individuals and businesses, a broad category of sensitive information involves the operation of government [39]. Judgments are published in law reports or on the internet in order for lawyers to get acquainted with the case law and to provide data for legal research.

Those judgments are anonymized in order to protect the privacy of individuals that participate in judicial proceedings and redacted in order to protect confidentiality of state or business secrets. In the case of the Republic of Serbia, they are anonymized and redacted by replacing or omitting text in accordance with the rules contained in [10]. For example, personal data of the parties in the proceedings are replaced with their initials while the evidence classified as a state or business secret is omitted by redaction. In the current practice, the anonymization and redaction are performed manually. Therefore, the process is time-consuming, expensive and error-prone (because employees can accidentally or intentionally omit to anonymize and redact some elements). If an editable version (e.g. Microsoft Word, LibreOffice) of a document is available, it is anonymized and redacted by manually replacing or omitting text using a text editor. On the other hand, if only an uneditable version (e.g. PDF, scanned image, hard copy) is available, the document is anonymized and redacted by manually retyping the existing document and applying anonymization and redaction rules.

Most commonly used electronic formats for the representation of judgments are LibreOffice Writer, Microsoft Word, and PDF documents. However, unstructured document formats have weaknesses compared to structured document formats since they are not machine-readable (e.g. the semantics of the text cannot be easily extracted). In contrast to legislative documents [24], there are not many structured standards for the representation of judicial documents. For example, the JuriX language [3] can be used to describe the content of judicial decisions (an order, a judgment, a court decision, etc.) as an XML document written according to the particular syntax. XML Schema Definition of Supreme Court Judgments [22] is an attempt to standardize the content of judgments across courts throughout Australia. Guidotti and Serrotti in [21] propose a DTD compatible with the structure proposed in the feasibility plan for the Norma In Rete project [16] to represent the Italian administrative high court decisions. Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies (AKOMA NTOSO) [37] recommends technical guidelines for developing and integrating parliamentary information systems throughout Africa. Although its primary focus is on the parliamentary information systems, one of the available document types specified by those guidelines are judgments. The OASIS Legal XML LegalDocML technical committee [28] works towards improvement of structured standards in the legal field starting from the results of the AKOMA NTOSO project.

In the position paper [36] we have proposed an idea that judgments, represented in the XML format, can be automatically redacted and anonymized using access control principles. This paper elaborates on the given idea and describes in detail how an XML access control solution can be adopted for judgments redaction and anonymization. Since majority of archival systems have access control features, we believe that anonymization and redaction can be implemented more easily and more efficiently by extending existing access control modules than as a separate module.

In the Role-based Access Control (RBAC) model, access to resources of a system is based on a role of a user in the system [15]. RBAC organizes individual users into roles according to their competency, authority, and responsibility within the enterprise and assigns permissions to roles according to their access rights. The core RBAC model consists of: users, roles, permissions and sessions, where permissions are composed of operations applied to objects (resources). In RBAC, roles are assigned to users, while permissions are assigned only to roles. A user’s interaction with the system, where a user activates a subset of the roles which she/he is assigned to, is called a session [15]. The main benefit of RBAC is the ease of administration of security policies and its scalability.

Results on XML access control reported in [4, 5, 6, 7, 8, 9, 13, 14] are based on the extension of RBAC in order to efficiently define and enforce access control policies for XML document collections. Major extensions that are introduced are granting and denying policies and access control enforcement on different granularity levels (e.g.

document definition, instance and fragment level). Moore in [27] considers the problem of deciding whether a fine-grained access control policy for XML documents updates allows a particular document to be constructed, which is related to the problem of determining whether a fine-grained access control policy preserves document schemas. Miklau and Suciu in [26] and Crampton in [11, 12] consider use of cryptographic methods for XML access control. In [30] access to a document and its parts can be defined based both on the current document content and on the history information that captures the operations performed on that document. Knowledge based formal approach to ensure the security of web-based XML documents is presented in [2].

eXtensible XML Role-Based Access Control Framework (XXACF) [31, 32, 33] enables access control definition and enforcement for XML documents. XXACF access control model is based on the RBAC model, extended to support different granularity levels, and they may be content dependent, thus facilitating efficient management of access control. It supports access control for following operations executed on XML documents: reading, searching, creating, updating and deleting a document/document fragment. The most notable improvements over the reviewed XML access control methods include: context-sensitive access control based on the hierarchical RBAC model, document-dependent definition of access control policies on different priority and granularity levels, and support for separate access control enforcement for different operations on documents and different ways of implementing the same operation.

To the best of our knowledge, existing RBAC models for XML documents do not handle the problem of anonymization and redaction. Since RBAC is a *de facto* standard for implementing access control in information systems, we have concluded that RBAC based solutions are most suitable for anonymization and redaction of documents. XXACF, beside support for granting and denying policies and fine-grained access control, also supports context-sensitive and content dependent permissions. Such features are necessary to anonymize and redact judicial documents according to the rules prescribed in [10]. Also, XXACF supports expressing anonymization and redaction rules in a declarative and brief form making it easy to use.

In order to design a solution that may be used in different judicial information systems we have chosen to use the standard AKOMA NTOSO XML-based format to represent structured textual documents. The presented solution is not dependent of any particular XML schema for judicial documents and may be used in different judicial environments. Nevertheless, usage of a well-known legal document format facilitates integration with other legal information systems. Its flexibility stems from the fact that access control rules may be defined for different XML schemas. If another XML format is to be used to represent judicial documents, only anonymization and redaction policies need to be redefined. There is no need to modify the system. The archived documents are the result of the judicial process described in detail by Gostojić et al. [18, 19].

The rest of this paper is structured as follows. The next section describes the AKOMA NTOSO document format and how AKOMA NTOSO can be used in our national legal system. The third section presents the extension of the eXtensible XML Role-Based Access Control Framework (XXACF) for anonymization and redaction. The application of XXACF for AKOMA NTOSO documents is given in Section 4. The fifth section presents the judicial archive with the focus on the anonymization and redaction of case law. In the conclusion, strengths and weaknesses of this approach are elaborated on, and directions of further research are given.

2. AKOMA NTOSO

AKOMA NTOSO is the set of principles for electronic parliamentary services in a pan-African context [37]. It has several goals: to define common data exchange standard between parliaments, to specify a basic document model that can be used to build information system and to define simple citation mechanism. The document model aims to provide a long-term storage of and access to parliamentary, legislative and judicial documents that allows search, interpretation and visualization of documents [29]. EU Parliament uses AKOMA NTOSO for modeling amendments, amendment list, bills, proposals, consolidated version of those documents. Brazilian Senate uses a customized version of AKOMA NTOSO for the document management. Senate of Italy uses AKOMA NTOSO for publishing bills in open data. Library Congress of Chile uses AKOMA NTOSO for managing debates, bills and acts. Uruguay Parliament plans to use AKOMA NTOSO for modeling the whole law-making process of the bills. Federal Chancellery of Switzerland is going to use AKOMA NTOSO for the publication of bills and acts. European Commission plans to adopt AKOMA NTOSO for the document management [37]. Recently, the OASIS's LegalDocML technical committee has started work on structured standards in the legal field that are based on AKOMA NTOSO [28].

AKOMA NTOSO separates content of the document, its presentation and its metadata and it uses XML design patterns such as hierarchy, container, block element, inline element, marker, etc. [38] to decrease number of elements needed to represent a legal document. It can be used to represent judgments made by any type of court (supreme court, high court, constitutional court, etc.), of any level (first order, appeal, etc.), of any nature (civil, penal, administrative, etc.) and in any legal tradition (common and civil law).

We decided to use AKOMA NTOSO because it entered the OASIS standardization process [28] which aims to provide interoperability between different legal information systems and reusability of software based on it. Another important reason is that AKOMA NTOSO separates content, presentation and metadata and uses XML design patterns. AKOMA NTOSO has been used as is, without modifications, since it supports our national judgments drafting guidelines and anonymization rules.

Arbitrary judgments were used as a case study of applying access control requirements for XML documents. Those judgments were made by a first order magistrate court in the Republic of Serbia, although the same method, without loss of generality, can be applied to other types of judgments (and XML documents). It should be noted that archived judgments are the result of business processes which are not the focus of the paper. The business processes that implement judicial proceeding, the access control policies to those processes, and the implementation of those policies are described in detail in [18, 19].

Since existing format was used, the representation of a judgment was straightforward. Firstly, document was identified at different FRBR (Functional Requirements for Bibliographic Records) [17] levels as an URL according to AKOMA NTOSO guidelines. Then, the important metadata were identified and serialized into the metadata section of the document. At the end, the document structure was marked up using standardized set of elements.

The relative URI of the judgment at the work level is */rs/judgment/psns/2012-01-01/3-7293-11*, where *rs* is the two-letter country code according to the ISO 3166-1 standard, *judgment* is the type of the document, *psns* is the designation of the emanating actor (Magistrate Court in Novi Sad), *2012-01-01* is the judgment creation date according to the ISO 8601 standard and *3-7293-11* is a disambiguating number of the judgment. The relative URI of the judgment at the expression level is */rs/judgment/psns/2012-01-01/3-7293-11/srp*, where *srp* is the three-letter language code in which the expression is drafted according to ISO 639-2. The relative URI of the judgment at the manifestation level is */rs/judgment/psns/2012-01-01/3-7293-11/srp/main.xml*, where *xml* is a unique three letter acronym of the data format (for the XML manifestation).

Metadata are organized into several groups: identification, publication, analysis and references. The metadata belonging to the identification group identifies documents at different FRBR levels (Listing 1). The *FRBRthis* element contains the URI of the specific document's component, *FRBRuri* contains the URI of the whole document, *FRBRdate* contains a relevant date of the document and the *FRBRauthor* element contains a relevant author of the document at the particular FRBR level.

```
<identification source="#bungeni">
 <FRBRWork>
  <FRBRthis value="rs.judgment.psns.2012-01-01.3-7293-
                  11.main" />
  <FRBRuri value="rs.judgment.psns.2012-01-01.3-7293-11" />
  <FRBRdate date="2012-01-11" name="Hearing" />
  <FRBRauthor href="#IvanaIvanovic" as="#Author" />
 </FRBRWork>
 <FRBRExpression>
  <FRBRthis value="rs.judgment.psns.2012-01-01.3-7293-
                  11.srp.main"/>
  <FRBRuri value="rs.judgment.psns.2012-01-01.3-7293-11.srp"
                                                         />
  <FRBRdate date="2012-01-18" name="Delivery" />
  <FRBRauthor href="#Bungeni" as="#Editor" />
 </FRBRExpression>
<FRBRManifestation>
  <FRBRthis value="rs.judgment.psns.2012-01-01.7293-
                  11.srp.main.xml”/>
  <FRBRuri value="rs.judgment.psns.2012-01-01.7293-
                 11.srp.xml"/>
  <FRBRdate date="2012-01-18" name="XMLConversion"/>
  <FRBRauthor href="#Bungeni" as="#Editor"/>
 </FRBRManifestation>
</identification>
```

Listing 1. Identification metadata

The metadata belonging to the publication group contains details about the publication of the paper-based document (Listing 2).

```
<publication date="2012-01-18" name="MagistrateCourtGazette"
    showAs="Magistrate Court Gazette" />
```

Listing 2. Publication metadata

The metadata belonging to the analysis group contains the description of judicial arguments of the judgment (Listing 3).

```
<analysis source="#bungeni">
  <judicial>
    <result type="approve" />
  </judicial>
</analysis>
```

Listing 3. Analysis metadata

The metadata belonging to the references group (see Listing 4) contains the list of locations (*TLCLocation*), organizations (*TLCOrganization*), persons (*TLCPerson*), roles (*TLCRole*), etc. referenced from the document and relevant to understanding its content.

```
<references source="#bungeni">
 <TLCLocation id="NS" href="/ontology/location/novi.sad"
              showAs="Novi Sad"/>
 <TLCOrganization id="MUP"
                  href="/ontology/organization/rs.gov.mup"
   showAs=" Ministry of Interior of the Republic of Serbia"/>
 <TLCPerson id=" MilanaMilanovic"
            href="/ontology/person/party/rs.milana.
                  milanovic.1987-05-06"
            showAs=" Milana Milanović"/>
 <TLCRole id="Defendant" href="/ontology/role/rs.defendant"
          showAs="Defendant"/>
</references>
```

Listing 4. References metadata

Each judgment document consists of the header section, the body section and the conclusion section. The body section contains an *introduction* (the summary of the case), *decision* (the decision of the judge) and *motivation* (the argumentation of the judges). Fragments of the document, which are interesting from the perspective of anonymization and redaction, are described in the rest of this section.

The introduction section summarizes the case (Listing 5). The element *party* is of particular importance since it contains the name of the party that took part in the proceedings (the defendant in this particular case), and a link to the concept in the ontology that specifies it. The content of this element as well as the content of its *refersTo* attribute needs to be anonymized.

The decision section (Listing 6) gives a detailed overview of the judicial decision. This section is the most important one with regards to anonymization, because most of the personal data is contained in it. Apart from the content of the party element, contents of the elements person, date, location and inline could also be used to identify a defendant. Therefore, those elements and their attributes need to be anonymized as well.

```
<introduction>
  <p>IN THE NAME OF THE PEOPLE</p>
  <p>Magistrate Court in Novi Sad, judge
  <judge id="jud1" refersTo="#IvanaIvanovic">Ivana Ivanović
  </judge>, in compliance with <ref id="ref1"
  href="/rs/act/2005/101#art85-cla3"> article 85 item 3 of
  Petty Offense Law («Official Gazette of the Republic of
  Serbia» No. 101/05, 116/08 and 111/09)</ref> in the
  judicial proceedings against defendant <party id="p1"
  refersTo="#MilanaMilanovic" as="#Defendant"> Milana
  Milanović </party> from Novi Sad, because of violation of
  the <ref id="ref2" href="/rs/act/1992/51#art15-cla1">article
  15 item 1 of Public Order Law («Official Gazette of the
  Republic of Serbia» No. 51/92, 53/93, 67/93, 48/94 и
  101/05)</ref> on the <date date="2012-01-11">January 11th,
  2012</date> has made the following</p>
  <p>J U D G M E N T</p>
</introduction>
```

Listing 5. Introduction section

```
<decision>
 <p>DEFENDANT: <party id="par1" refersTo="#MilanaMilanovic"
   as="#Defendant">Milana Milanović</party> the dauther of
   <person id="per1" refersTo="#AleksandarMilanovic"
   as="#Father">Aleksandar</person>, born on <date date="1987-
   05-06" refersTo="">May, 6th 1987.</date> in <location
   id="loc1" refersTo="#ODZ">Odžaci</location>, personal
   number <inline name="personal-id">0605987815106</inline>,
   with the residence in <location id="loc2" refersTo="#NS">
   Novi Sad, Fruškogorska 11</location>,</p>
 <p>I S   F O U N D   G U I L T Y:</p>
 <p>Because on <date date="2010-01-28T01:55">January, 29th
  2010 around 02:45 AM</date> in <location id="loc3"
  refersTo="#NS"> Novi Sad, Fruškogorska 11, in the apartment
  number 119</location>, she disturbed public order by loudly
  playing music on a musical device. The sound was heard
  outside apartment and thus the defendant disturbed the
  public order and peace of the surrounding residents and
  doing so the defendant has broken article 15, item 1 of
  Public Order Law.</p>
</decision>
```

Listing 6. Decision section

3. XXACF Extension for Anonymization and Redaction

The process of accessing a judgment is presented in Figure 1. When a user wants to view a judgment she/he sends a request to the system using a web browser. Upon receiving the request, the system reads the requested judgment from the database and anonymization and redaction rules (access control permission) assigned to the user. If the user has rights to access the whole judgment (according to the loaded rules) it will not be anonymized. Otherwise the judgment will be anonymized. The unanonymized or anonymized judgment will be returned to the user.

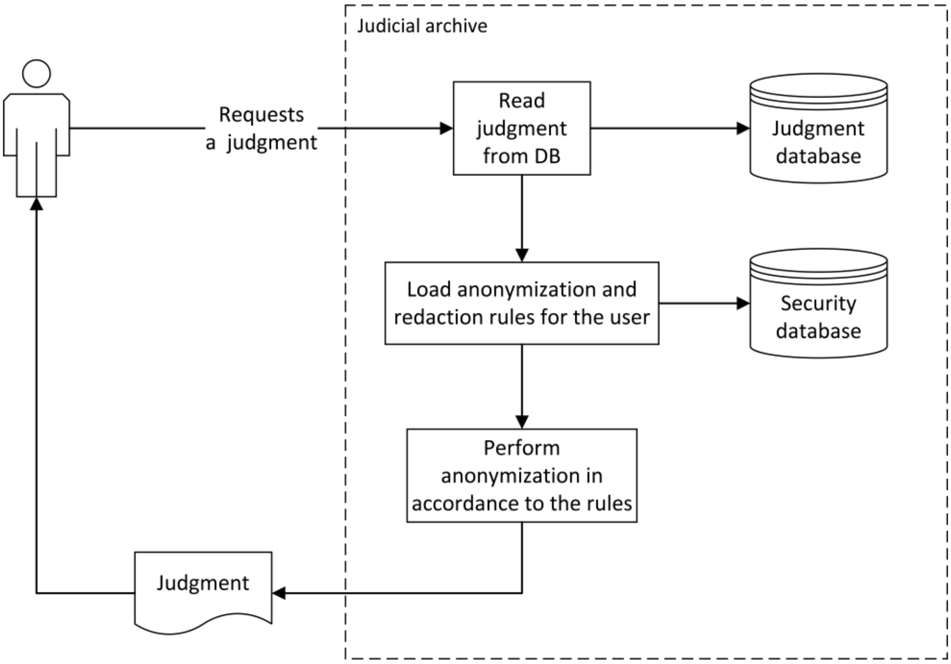

Judicial archive
Requests a judgment
Read judgment from DB
Judgment database
Load anonymization and redaction rules for the user
Security database
Perform anonymization in accordance to the rules
Judgment

Fig. 1. Accessing a judgment

The XXACF extension for anonymization and redaction of XML documents is presented in this section. Also, the overview of the core XXACF's entities is given in order to understand how anonymization and redaction can be implemented using XXACF.

The diagram in Figure 2 shows main classes and their relations. The *User* class models users in the system, the role is represented by the *Role* class, and the *Permission* class defines permissions. The *childRoles-parentRole* relation connects roles into the hierarchy, while the associations between *User*, *Role* and *Permission* establish appropriate assignments. The *Operation* class models the operation for which permission is defined. *AnonymizeReadOperation* performs reading with enforcing anonymization and redaction rules. When parts of document should be anonymized or redacted it is necessary to define anonymization/redaction rules. The *AnonymizeRule* class is used to define those rules. It defines how replacement of the data (in the XML document) matched by the pattern attribute will be performed. Specializations of the *Anonymizer* class implement those anonymization rules. *AnonymizeSearchOperation* actually represents extension of *AnonymizeReadOperation* to prevent a user from searching non-anonymized and unredacted data. If a user searches for data, she/he may get a certain number of hits which include searching non-anonymized and unredacted data. Those situations could compromise enforcement of anonymization and redaction rules. The purpose of *AnonymizeSearchOperation* is to prevent those cases.

The *Resource* class is used to represent resources for which permissions are defined. The permission can be defined for a document schema (the class *DocumentSchema*) or a document instance (the class *DocumentInstance*) identified by its unique identifier (the *id* attribute of the *Resource* class). If permissions are defined for a document schema they are applied to all document instances of that schema. On the other hand, permissions are applied only to an instance if they are defined at the instance level. The XPath expressions are used to define permission for fragments of documents or schemas. It is also possible to define content dependent permissions by XPath expressions which contain condition. The *DocumentInstance's* or *DocumentSchema's* attribute *fragment* denotes the fragment of a document instance or schema for which the permission is defined. By using document fragment it is possible to define anonymization and redaction rules for specific parts of documents.

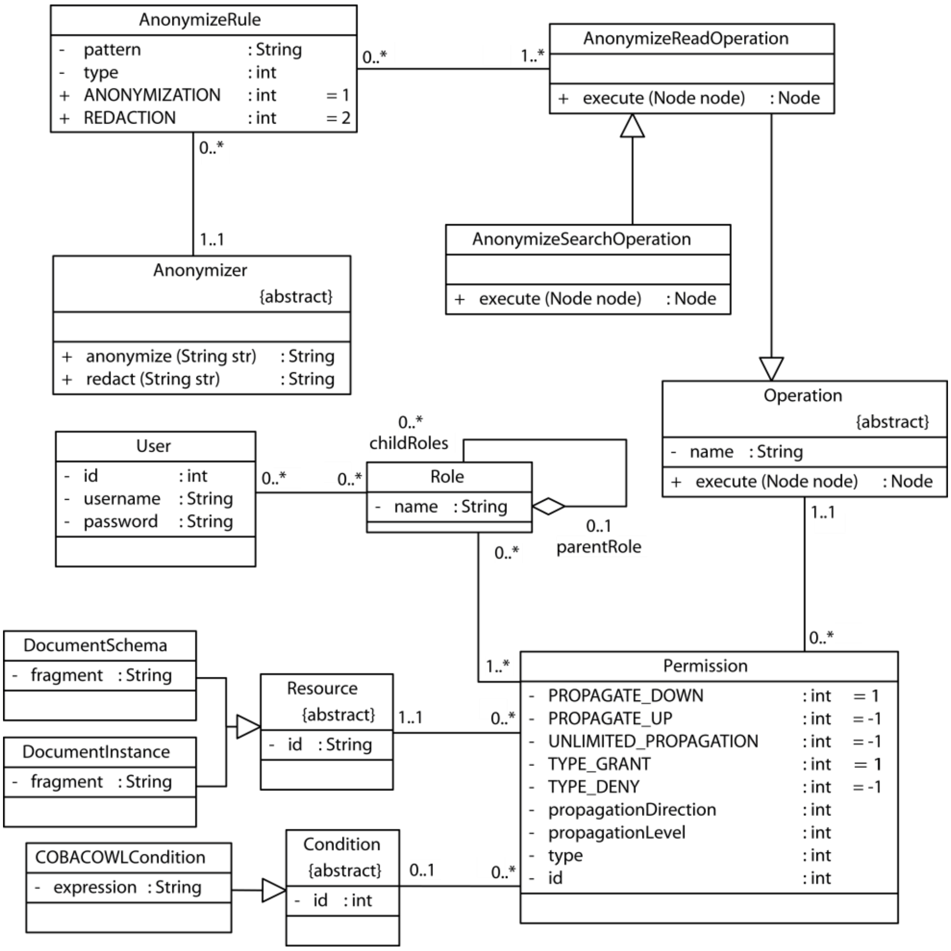


Fig. 2. Class diagram of the XXACF's anonymization and redaction extension

Although the standard RBAC model identifies granting policies only, the literature (see Section 1) identifies the need for denying policies to achieve more efficient XML access control. The permission type (granting or denying) of a policy is specified with by the *type* attribute of the *Permission* class. To avoid explicit permission definition for each entity, propagation of permissions is enabled, starting from the point specified by *Resource* down or up the hierarchy. The permission's propagation direction is identified by the *propagationDirection* attribute. The maximum number of XML hierarchy levels where the propagation is performed (propagation level), is defined by the *propagationLevel* attribute.

XXACF supports the context-sensitive access control that may depend on multiple context factors. It can support context-sensitive access control through the *Condition* class. If the condition assigned to the permission is satisfied, the permission will be applied, otherwise, it will not be the case. The *COBACOWLCondition* class is used for representing context condition using appropriate ontology based on the ontology defined for the COBAC model (context-sensitive access control model for business processes) [18, 19, 34, 35, 40]. Those conditions are used to specify different anonymization and redaction policies for users with the same user's role depending if they are participants of a particular judicial proceedings. For example, a user with the judge role can see her/his judgments in non-anonymized and unredacted form, but judgments of all other judges can see only in anonymized and redacted form.

Documents are anonymized and redacted when a user accesses them. The anonymization and redaction enforcement comprises of the following major steps:

1. Loading of user properties and roles – when the system accepts the request, it loads user's properties from the database and assigns roles to the user.
2. Selection of the applicable permissions – in this step the system finds anonymization and redaction policies which will be applied.
3. Marking document nodes – this is a process of applying permissions determined in the previous step to the nodes of DOM (Document Object Model) representation of XML document selected by those permissions.
4. Conflict resolution – it might be necessary to resolve conflicts since both policy types can be applied to the same nodes. As a result of this phase, only policies of one type (granting or denying) will be applied on the particular node.
5. Execution of the requested operation – the operation execution depends on the operation's type. The *read* operation retrieves only parts of the document that are allowed to be read by the user. If an anonymization and redaction rule is defined for some segment of the document, it will be removed or replaced. The *search* operation is performed in the same way as the read operation and it also performs anonymization and redaction rules when counting and displaying search results.

4. Anonymization and Redaction Policies for AKOMA NTOSO Documents

According to the rules prescribed in [10] it is necessary to anonymize: first and last name of a natural person, name of legal person, address of their residence, date and place of birth, personal identification number, taxpayer identification number, id card number, passport number, driving license number, license plates or other personal documents' number, e-mail and web address. Information that is classified as a state or a business secret and information that jeopardizes the privacy of the involved parties must be redacted.

XXACF has been used to define anonymization and redaction policies and to enforce anonymization and redaction of judgments represented in the AKOMA NTOSO format. The resources which are anonymized and redacted are whole judicial documents and their fragments. All permissions are applied to all documents or their fragments. In the current version of the system there is no requirement for defining anonymization and redaction rules for a particular document. All resources in the following permissions are identified by XPath expressions which select certain elements in the AKOMA NTOSO documents. For example: XPath expression */akomaNtoso/judgmentBody/decision* select the decision element (part) of the judgment.

Two types of users have been identified according to the functions they perform and permissions they have for viewing judgments: *Participant* (judge, clerk, accused and claimant) and *Public*. All users can search archived documents and view search results according to their access control rights. The other roles in the archive system (e.g. archiver and security administrator) are not described in this paper since they are not relevant for anonymization and redaction use case.

The *Participant* role can perform read and search of the whole document (without being anonymized) only if she/he participated in a judicial process that resulted in the particular judgment document (Table 1). The condition in the table verifies whether the current user was a judge who made the judgment or one of the parties. The function *currentUser()*, used in the condition, returns the identifier of the logged user, and the *xpath()* function returns values selected by the given XPath expression in the current document. The XPath expression *//judge/@refersTo* selects the *referesTo* attribute of all *judge* elements, while the XPath expression *//party/@refersTo* selects the *referesTo* attribute of the all *party* elements in the document. The attribute *referesTo* of the *party* element contains the identifier of the corresponding user of the organization (see Section 2). Thus, the subcondition *currentUser() in xpath(//judge/@ refersTo)* checks if the current user identifier corresponds to the identifier of at least one judge. The second subcondition, *currentUser() in xpath(//party/@ refersTo)*, checks if the current user identifier corresponds to the identifier of at least one party.

Table 1. Search and read permissions for *Participant*

Role	Participant
Operations	read, search
Propagation	*direction*: down, *level*: unlimited
Type	grant
Resources	/akomaNtoso
Condition	currentUser() in xpath(//judge/@ refersTo) or currentUser() in xpath(//party/@ refersTo)
Anonymize rules	*None*

The *Public* role can perform read and search of anonymized document (Table 2). The anonymization and redaction rules in Table 2 anonymize/redact data by calling specified anonymization/redaction function on the fragment of the document selected by the corresponding pattern. The anonymization and redaction rules are implemented as defined in [10]. The function *name_to_initials()* converts person's name to initials, the *location_to_initials()* function transforms location to initials and the *replace()* function replaces selected fragment with the given text (in this particular case the selected fragment is replaced with the ellipsis). The *party* element references the proper *TLCPerson* element (the attribute *refersTo* of *party* has same value as *id* of *TLCPerson*). Since the values of those attributes usually correspond to personal name, it is necessary to anonymize them. It is also necessary to preserve the referential integrity after anonymization. Therefore, the values of those attributes are replaced with their HMAC (Hash-based Message Authentication Code) value [23]. The similar case is applied to elements *person* and *location*. The content of the *date* and the *inline* (which contain personal identifier) elements has to be replaced with the ellipsis.

Table 2. Anonymized search and read permissions for *Public*

Role	Public	
Operations	read, search	
Propagation	*direction*: down, *level*: unlimited	
Type	grant	
Resources	/akomaNtoso	
Condition	None	
Anonymize rules	pattern	anonymize/redact function
	//party/text()	name_to_initials()
	//person/text()	name_to_initials()
	//TLCPerson/@showAs	name_to_initials()
	//location/text()	location_to_initials()
	//TLCLocation/@showAs	location_to_initials()
	//date	replace('...')
	//inline[@name=”personal-id”]	replace('...')
	//party/@ refersTo	hmac()
	// person/@ refersTo	hmac()
	//TLCPerson/@id	hmac()
	//location/@ refersTo	hmac()
	//TLCLocation /@id	hmac()

The *href* attribute of the *TLCPerson* and *TLCLocation* elements refers to the proper information about person/location in the metadata database. Therefore, it is necessary to remove that reference in order to completely anonymize document. The permissions in Table 3 deny access to those attributes of the *Public* role.

Since the permissions in Table 2 grant access and the permissions in Table 3 deny it, the attribute *href* of *TLCPerson/TLCLocation* will be assigned to both granting and denying permissions. According to the conflict resolution principle "more specific object takes precedence" (see Section 3) the final permission for this attribute will deny access. The denying permissions in Table 3 are more specific than the granting permissions in Table 2, because they are defined for more specific entity (the permissions in Table 2 are defined for the whole document, while the permissions in Table 3 are defined for the specific attribute), and therefore they are selected as the final permissions.

Table 3. Deny search and read permissions for *Public*

Role	Public
Operations	read, search
Propagation	*direction*: down, *level*: unlimited
Type	deny
Resources	//TLCPerson/@href, //TLCLocation/@href
Condition	*None*
Anonymize rules	*None*

5. Judicial archive

This section describes the prototype of the judicial archival system with the focus on the anonymization and redaction of case law. This system is a part of a larger information system which is used to retrieve and browse legal norms and legislation [**20**]. This system can be used by lawyers to access judgments (case law) that are made by applying retrieved legal norms (contained in a piece of legislation being browsed). Therefore, this system provides retrieval of judgments of interest.

The global architecture of the prototype system for archiving judicial documents is presented in Figure 3. This system is designed as a typical multi-tier application. The whole server-side of the system is implemented using the Java open source technologies. Users can search archive and read search results through the web interface. The archived documents are stored in the native XML database (eXist), the documents' metadata are archived in the RDF store (Joseki), while anonymization and redaction policies are kept in the relational database (MySQL). The archive uses the XSLT processor (Xalan) for converting XML documents to HTML, the XML DB API library for accessing the native XML database, the Java persistence API (JPA) for accessing the relational database (Hibernate) and the RDF store (Joseki). The anonymization and redaction is performed by the XXACF implementation. Apache Tomcat is used as the application server.

If the user has permissions to access whole (non-anonymized and unredacted) judgment, it will be displayed in the non-anonymized form (Figure 4). The elements of the judgment that must be anonymized and redacted are shown in bold.

On the other hand, if the user does not have permissions to access the whole (non-anonymized and unredacted) judgment, it will be displayed in the anonymized and redacted form (Figure 5). The name of the defendant and the name of the location are replaced with the initials, while the date and the personal number are replaced with the ellipsis. The anonymized and redacted elements are marked with ellipse. According to [10], judgments can be anonymized by replacing text with dummy text (e.g. initials or ellipsis) and redacted by omitting text if the quantity of text is significant (if parts of the judgment are classified as a state or business secret, etc.). For example, personal data of the parties are replaced with initials while the evidence that is classified as a state or business secret is omitted by redaction.

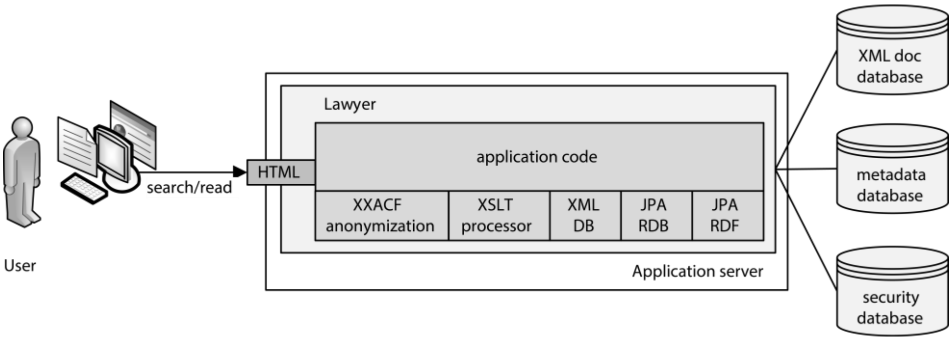
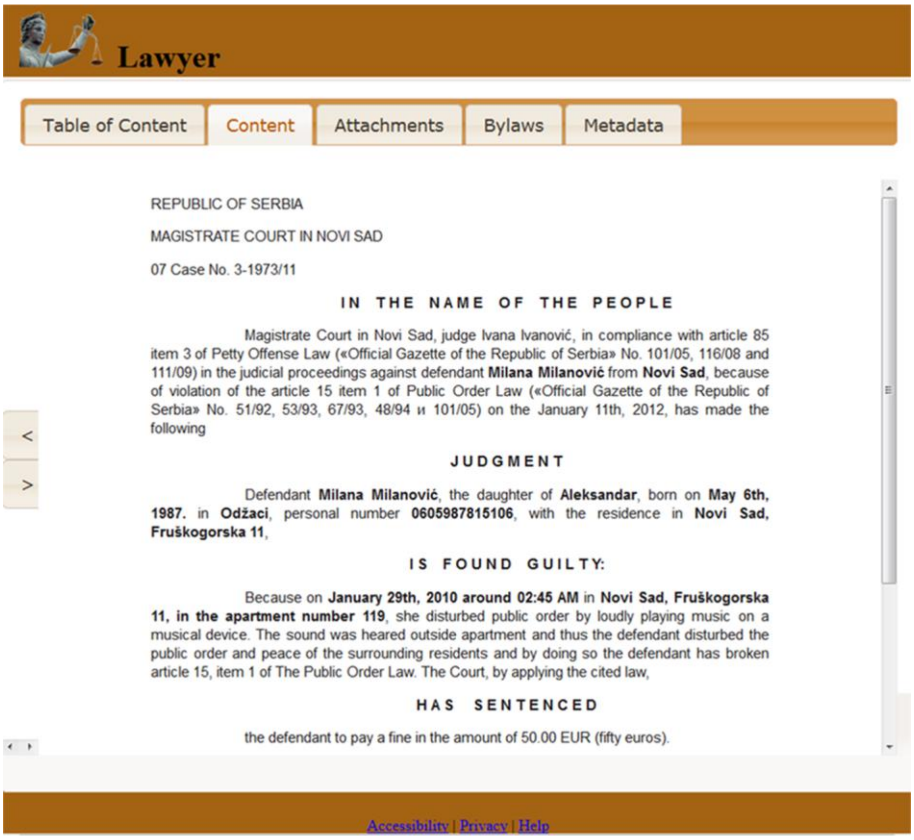


Fig. 3. Architecture of the judicial archive prototype



REPUBLIC OF SERBIA

MAGISTRATE COURT IN NOVI SAD

07 Case No. 3-1973/11

IN THE NAME OF THE PEOPLE

Magistrate Court in Novi Sad, judge Ivana Ivanović, in compliance with article 85 item 3 of Petty Offense Law («Official Gazette of the Republic of Serbia» No. 101/05, 116/08 and 111/09) in the judicial proceedings against defendant **Milana Milanović** from **Novi Sad**, because of violation of the article 15 item 1 of Public Order Law («Official Gazette of the Republic of Serbia» No. 51/92, 53/93, 67/93, 48/94 и 101/05) on the January 11th, 2012, has made the following

JUDGMENT

Defendant **Milana Milanović**, the daughter of **Aleksandar**, born on **May 6th, 1987.** in **Odžaci**, personal number **0605987815106**, with the residence in **Novi Sad, Fruškogorska 11**,

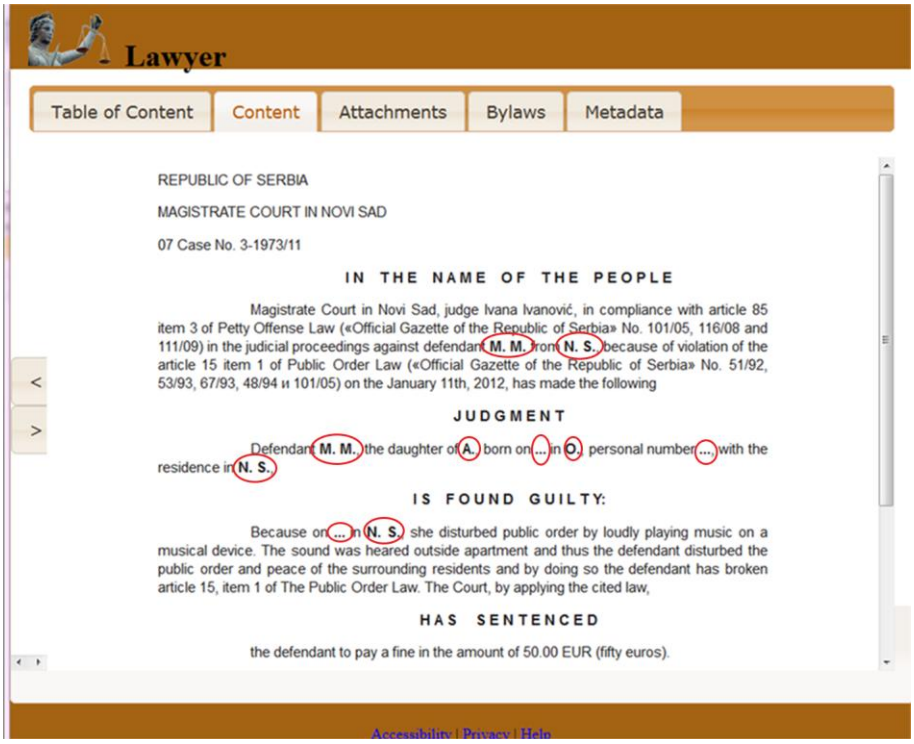
IS FOUND GUILTY:

Because on **January 29th, 2010 around 02:45 AM** in **Novi Sad, Fruškogorska 11, in the apartment number 119**, she disturbed public order by loudly playing music on a musical device. The sound was heared outside apartment and thus the defendant disturbed the public order and peace of the surrounding residents and by doing so the defendant has broken article 15, item 1 of The Public Order Law. The Court, by applying the cited law,

HAS SENTENCED

the defendant to pay a fine in the amount of 50.00 EUR (fifty euros).

Fig. 4. Non-anonymized and unredacted judgment



REPUBLIC OF SERBIA

MAGISTRATE COURT IN NOVI SAD

07 Case No. 3-1973/11

IN THE NAME OF THE PEOPLE

Magistrate Court in Novi Sad, judge Ivana Ivanović, in compliance with article 85 item 3 of Petty Offense Law («Official Gazette of the Republic of Serbia» No. 101/05, 116/08 and 111/09) in the judicial proceedings against defendant M. M. from N. S. because of violation of the article 15 item 1 of Public Order Law («Official Gazette of the Republic of Serbia» No. 51/92, 53/93, 67/93, 48/94 и 101/05) on the January 11th, 2012, has made the following

JUDGMENT

Defendant M. M. the daughter of A. born on ... in O. personal number ... with the residence in N. S.

IS FOUND GUILTY:

Because on ... in N. S. she disturbed public order by loudly playing music on a musical device. The sound was heared outside apartment and thus the defendant disturbed the public order and peace of the surrounding residents and by doing so the defendant has broken article 15, item 1 of The Public Order Law. The Court, by applying the cited law,

HAS SENTENCED

the defendant to pay a fine in the amount of 50.00 EUR (fifty euros).

Fig. 5. Anonymized and redacted judgment

The prototype of this system was tested by employees of Magistrate Court in Novi Sad, the Republic of Serbia. The goal of this pilot project has been to create judgments' archive that supports automatic anonymization of judgments according to the specific anonymization rules. The testing corpus contained 374 judgments which were marked up by hand. The result of evaluation showed that AKOMA NTOSO is suitable format for the testing corpus; anonymization and redaction rules can be completely represented by using XXACF and that automatic anonymization can be successfully applied to documents in the AKOMA NTOSO format.

The performance of the system was tested on the same corpus. We analyzed the time needed for executing loading a judgment from the database, anonymizing a judgment, and presenting a judgment to the user. The tests presented in this section were executed on a computer with i7 processor and 8GB of RAM, running Linux OS, Java 7 and Apache Tomcat 7 as a runtime environment. In the experimental setup, the server and the client are deployed on the same computer. On average, a judgment has 624 nodes, of which 194 nodes need to be anonymized or redacted (52 nodes converted to initials, 77 nodes hashed, 12 nodes replaced with ellipsis, and 53 nodes redacted).

The average time of loading a judgment from the database, anonymizing a judgment, and presenting a judgment to the user phases executed by *Public* and *Participant* roles is presented in Figure 6. Same data, along with standard deviation, is shown in Table 4.

The time needed to execute the first and the last phase is independent of the role. If a user with the *Public* role accesses a judgment, the anonymization phase is an order of magnitude longer than the rest of the phases. The duration of all phases is in the same order of magnitude if a judgment is accessed by a user with the *Participant* role. The anonymization phase for the role *Public* is longer than the same phase for the role *Participant* because it requires execution of anonymization and redaction methods (conversion to initials, hashing, replacing with ellipsis, and redaction) which take considerable time.

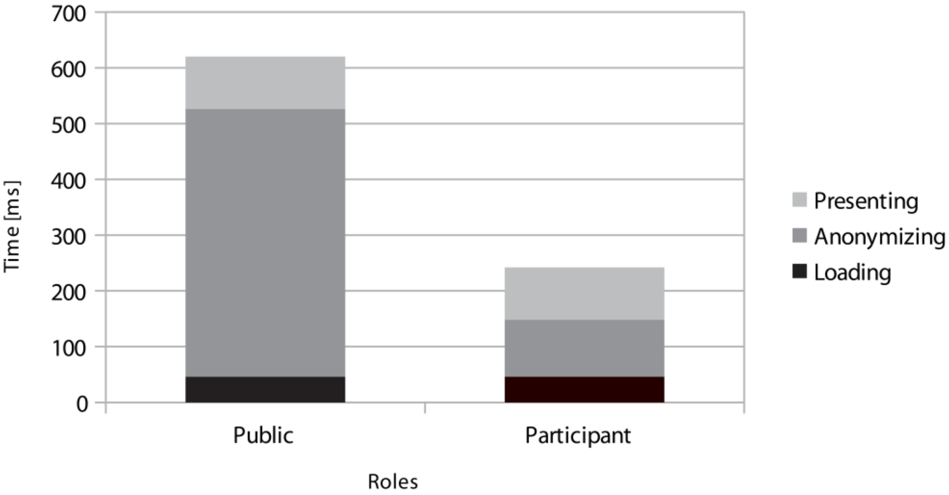


Fig. 6. Average time for accessing a judgment

Table 4. Access time average and standard deviation

Phase / Role	Public		Participant	
	Average [ms]	Std. Dev. [ms]	Average [ms]	Std. Dev. [ms]
Presenting	46	22	22	22
Anonymizing	480	102	81	28
Loading	94	12	94	12

6. Conclusion

This paper proposes a method for anonymization and redaction of judgments represented in the AKOMA NTOSO format. This method is based on the anonymization and redaction extension of the XXACF framework. The system's prototype has been evaluated in Magistrate Court in Novi Sad, the Republic of Serbia. The proposed method represents a successful application of the context-dependent role-based XML access control for anonymization and redaction of judicial documents.

When judicial proceedings are completed, final judgments are made and archived. As far as the described solution is concerned, both archived and non-archived judgments can be anonymized and redacted in the same manner. However, according to the rules of the court proceedings, only the final (and subsequently archived) judgments can be published. Judgments that are being drafted are not publicly accessible.

Preliminary tests of the prototype, performed by employees of the court, has shown that judgments are anonymized and redacted in accordance with anonymization and redaction rules prescribed in [10]. There have not been found any deviations from those rules. The presented real-life prototype implementation of the anonymization and redaction system presents the proof of the practical value of the proposed method. Methods similar to those that are used to anonymize and redact judgments can be used to anonymize archival records, as well as any other document type that requires anonymization and redaction.

The resulting system and its prototype implementation have yielded the following benefits:

(1) provides a more efficient work and labor saving; judgments anonymization and redaction has typically been a manual procedure. However, manual anonymization is labor-intensive and error-prone. Publishers of anonymized and redacted judgments would benefit from the proposed solution for providing automated anonymization and redaction process.

(2) public access to judgments; using limited resources available to courts, it is possible to publish a larger number of judgments with automated anonymization and redaction process than by manual anonymization and redaction methods.

(3) customizable for different anonymization and redaction rules; since anonymization and redaction rules are expressed declaratively, there is no need to change the implementation of the proposed system to support new anonymization and redaction rules.

(4) anonymization and redaction of judgments represented in different XML-based formats; The anonymization and redaction rules prescribed in [10] are independent of a judgment representation format. As stated in the introductory section, XXACF can be used for anonymization and redaction of judgments in different XML-based formats. The anonymization and redaction rules described in Section 4 can be customized to different XML-based judgments formats only by customizing corresponding XPath expressions or adding new permissions (the implemented XXACF's anonymization and redaction extension remains unchanged).

In cases where anonymization and redaction rules are not context-dependant, the system's performance can be optimized by creating anonymized documents offline and providing those documents to users.

However, the successful anonymization and redaction of documents is dependent of their proper markup. A potential drawback of the presented prototype is that it can anonymize and redact judgments incorrectly if data that needs to be anonymized and redacted is not marked up according to the XML format in use.

Since there are many legacy documents which are in PDF or DOC format, or even there is no electronic version of a document available, there is a need for automatic or semiautomatic conversion of those documents into AKOMA NTOSO format. According to the research presented in [1] and [25], scanning, OCR (Optical Character Recognition), NLP (Natural Language Processing) and text mining techniques can be used for this purpose. Anonymization of metadata (according to a particular schema or ontology) has not been dealt with and is one of the directions for further research. Those issues need to be addressed in order to deploy the system at scale.

References

1. Bacci, L., Spinosa, P., Marchetti, C., Battistoni, R., Florence, I., Senate, I., and Rome, I.: Automatic Mark-up of Legislative Documents and Its Application to Parallel Text Generation. In Proceedings the Workshop on Legal Ontologies and Artificial Intelligence Techniques/ Workshop on Semantic Processing of Legal Texts (LOAIT), Barcelona, Spain, 45-54. (2009)
2. Bai, Y.: Access Control for XML Document. In Proceedings of the 21st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Wroclaw, Poland, 621-630. (2008)
3. Belhissi, R., Moudam, Z., and Chenfour, N.: JuriX Framework for XML Modelling of Judicial Documents: A Support System for Checking the Regularity of Judgments. Engineering Science and Technology: An International Journal (ESTIJ), Vol. 1, No. 1, 51–57. (2011)
4. Bertino, E., Castano, S. and Ferrari, E.: Securing XML Documents with Author-X. IEEE Internet Computing, Vol. 5, No. 3, 21–31. (2001)
5. Bertino, E., and Ferrari, E.: Secure and Selective Dissemination of XML Documents. ACM Transactions on Information and System Security, Vol. 5, No. 3, 290–331. (2002)
6. Bertino, E., Carminati, B. and Ferrari, E.: Access Control for XML Documents and Data. Information Security Technical Report, Vol. 9, No. 3, 19–34. (2004)
7. Bhatti, R., Joshi, J., Bertino, E. and Ghafoo, A.: Access Control in Dynamic XML-based Web-services With X-RBAC. In Proceedings of the 1st International Conference on Web Services, Las Vegas, USA, 243-249. (2003)
8. Bhatti R., Bertino, E., Ghafoor, A. and Joshi, J.: XML-based Specification for Web Services Document Security. Computer, Vol. 37, No. 4, 41–49. (2004)
9. Botha A. R. and Eloff, J.: A Framework for Access Control in Workflow Environments. Information Management and Computer Security, Vol. 9, No. 3, 126–133. (2001)
10. Court of Appeals in Novi Sad: Rules for Data Anonymization in Judicial Decisions. Court of Appeals in Novi Sad, Serbia (in Serbian). (2011)
11. Crampton, J.: Applying Hierarchical and Role-based Access Control to XML Documents. In Proceedings of the 1st Workshop on Secure Web Service, Fairfax, Virginia, USA, 37-46. (2003)
12. Crampton, J.: Applying Hierarchical and Role-based Access Control to XML Documents. Computer Science and System Engineering, Vol. 21, No. 5, 352–338. (2006)
13. Damiani, E., Samarati, P., De Capitani di Vimercati, S., and Paraboschi, S.: Controlling Access to XML Documents. IEEE Internet Computing, Vol. 5, No. 6, 18–28. (2001)
14. Damiani, E., De Capitani di Vimercati, S., Paraboschi, S and Samarati, P.: A Fine-grained Access Control System for XML Documents. ACM Transactions on Information and System Security, Vol. 5, No. 2, 169–202. (2002)
15. Ferraiolo, F. D., Ravi Sandhu, R., Gavrila S., Richard, D. K. and Chandramouli, R.: Proposed NIST Standard for Role-based Access Control. ACM Transactions on Information and System Security, Vol. 4, No. 3, 224–274. (2001)
16. Francesconi, E.: The Norme in Rete Project: Standards and Tools for Italian Legislation. International Journal of Legal Information, Vol. 34, No. 2, 358–376. (2006)
17. International Federation of Library Associations and Institutions: Functional Requirements for Bibliographic Records. (2007). [Online]. Available: http://www.ifla.org/en/publications/functional-requirements-for-bibliographic-records (current May 2014)
18. Gostojić, S., Sladić, G., Milosavljević, B., Konjović, Z.: Context-sensitive Access Control Model for Government Services. Journal of Organizational Computing and Electronic Commerce, Vol. 22, No. 2, 184–213. (2012)

19. Gostojić, S., Sladić, G., Milosavljević, B., and Konjović, Z.: Flexible Access Control for Judicial Processes. In Proceedings of the 6th International Conference on Methodologies, Technologies and Tools enabling e-Government, Belgrade, Serbia, USA, 44-54. (2012)
20. Gostojić, S., Milosavljević, B. and Konjović, Z.: Ontological Model of Legal Norms for Creating and Using Legislation. Computer Science and Information Systems, Vol. 10, No. 1, 151–171. (2013)
21. Guidotti, P and Serrotti, L.: Legal Drafting Systems for Judges. (2002). [Online]. Available: http://espejos.unesco.org.uy/simplac2002/Ponencias/Derecho/DER36.rtf (current May 2014)
22. Kirk, G. and Lazberger, J.: Proposed XML Schema Definition of Supreme Court Judgments. Supreme Court of Western Australia. (2006)
23. Krawczyk, H., Bellare, M., and Canetti, R.: HMAC: Keyed-Hashing for Message Authentication. The Internet Engineering Task Force (IETF). (1997)
24. Lupo, C., Vitali, F., Francesconi , E., Palmirani, M., Winkels, R., de Maat, E., Boer, A. and Mascellani, P.: General XML Format(s) for Legal Sources. Technical Report, University of Amsterdam. (2007)
25. Maat, de. E., Winkels, R., and van Engers, T.: Automated Detection of Reference Structures in Law. In Proceedings of the 19th Conference on Legal Knowledge and Information Systems, Paris, France, 41. (2006)
26. Miklau, G. and Suciu, D.: Controlling Access to Published Data Using Cryptography. In Proceedings of the 29th International Conference on Very Large Data Bases, Berlin, Germany, 898-909. (2003)
27. Moore, N.: Computational Complexity of the Problem of Tree Generation under Fine-grained Access Control Policies. Information and Computation, Vol. 209, No. 3, 548–567. (2011)
28. Organization for the Advancement of Structured Information Standards – OASIS: OASIS LegalDocumentML (LegalDocML) TC. (2012). [Online]. Available: http://www.oasis-open.org/committees/legaldocml (current May 2014)
29. Palmirani, M. and Vitali, F.: Akoma-Ntoso for Legal Documents. In: Sartor, G., Palmirani, M., Francesconi E., Biasiotti, M.A. (eds.): Legislative XML for the Semantic Web, Springer-Verlag, Berlin Heidelberg New York, 75–100. (2011)
30. Roder, P., Tafreschi, O. and Eckert, C.: History-based Access Control for XML Documents. In Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, Singapore, ACM, 386-388. (2007)
31. Sladić, G., Milosavljević, B., and Konjović, Z.: Extensible Access Control Model for XML Document Collections. In Proceedings of the 2nd International Conference on Security and Cryptography, Barcelona, Spain, ACM, 373-380. (2007)
32. Sladić, G., Milosavljević, B., Konjović, Z. and Vidaković, M.: Access Control Framework for XML Document Collections. Computer Science and Information Systems (ComSIS), Vol. 8, No. 3, 591–609. (2011)
33. Sladić, G., Milosavljević, B., Surla, D., and Konjović, Z.: Flexible Access Control Framework for MARC Records. The Electronic Library, Vol. 30, No. 5, 623–652. (2012)
34. Sladić, G., Milosavljević, B., and Konjović, Z.: Modeling Context for Access Control Systems. In Proceedings of the 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY 2012), Subotica, Serbia, IEEE, 373-380. (2012)
35. Sladić, G., Milosavljević, B., and Konjović, Z.: Context-sensitive Access Control Model for Business Processes. Computer Science and Information Systems (ComSIS), Vol. 10, No. 3, 939-972. (2013)
36. Sladić, G., Gostojić, S., Milosavljević, B., and Konjović, Z.: Automatic Anonymization of Judgments in Electronic Format. In Proceedings of the 3rd International Conference on Information Society Technology and Management (ICIST), Kopaonik, Serbia, pp. 206-211. (2013)

37. United Nations Department of Economic and Social Affairs – UNDESA: AKOMA NTOSO. (2000). [Online]. Availablehttp://www.akomantoso.org (current May 2014)
28. Vitali, F., Di Iorio, A. and Gubellini, D.: Design Patterns for Descriptive Document Substructures. In Proceedings of the 2005 Extreme Markup Languages Conference, Montréal, Quebec, Canada. (2005)
39. Winn, P.: Judicial Information Management in an Electronic Age: Old Standards, New Challenges. Federal Courts Law Review, Vol. 3, No. 2, 135–176. (2009)
40. Zarić, M., Segedinac, M., Sladić, G. and Konjović, Z: A Flexible System for Request Processing in Government Institutions. Acta Polytechnica Hungarica, Vol. 11, No. 6, 207–227. (2014)

Goran Sladić is holding the assistant professor position at the Faculty of Technical Sciences, Novi Sad, Serbia since 2011. Mr. Sladić received his Bachelor degree (2002), Master degree (2006) and and PhD degree (2011) all in Computer Science from the University of Novi Sad, Faculty of Technical Sciences. Since 2002 he is with the Faculty of Technical Science in Novi Sad. His research interests include information security, document management systems, XML technologies, context-aware computing and workflow systems.

Stevan Gostojić is assistant professor of applied computer science and informatics at Faculty of Technical Sciences, University of Novi Sad. He has a Ph.D. in electrical engineering and computer science from the same university. His research interests are legal informatics, e-government, and document management.

Branko Milosavljević is holding the full professor position at the Faculty of Technical Sciences, Novi Sad, Serbia since 2014. Mr. Milosavljević received his Bachelor degree (1997), Master degree (1999), and PhD degree (2003) all in Computer Science from the University of Novi Sad, Faculty of Technical Sciences. Since 1998 he is with the Faculty of Technical Science in Novi Sad.

Zora Konjović is holding the full professor position at the Faculty of Technical Sciences, Novi Sad, Serbia since 2003. Mrs. Konjović received her Bachelor degree in Mathematics from the University of Novi Sad, Faculty Science in 1973, Master degree (1985) and Ph. D. degree (1992) both in Robotics from the University of Novi Sad, Faculty of Technical Sciences. Since 1973 till 1980 she was with the Faculty of Science in Novi Sad, and since 1980 she is with the Faculty of Technical Sciences, University of Novi Sad.

Gordana Milosavljević is assistant professor at Faculty of Technical Sciences, University of Novi Sad. She teaches courses in Software Modeling, Business Information Systems and Model Driven Software Development. Her research interests focus on agile methodologies, model-driven development and enterprise information systems design.

Received: August 8, 2014; Accepted: May 20, 2015.