Knowledge Acquisition from Historical Documents for Preserving Transylvanian Cultural Heritage

Ioan SALOMIE, Mihaela DINSOREANU, Cristina POP, Sorin SUCIU
Computer Science Department, Technical University of Cluj-Napoca,
15 C. Daicoviciu str. RO-3400 Cluj-Napoca, Romania
ioan.salomie@cs.utcluj.ro

***Abstract*—This paper proposes a solution for preserving the cultural heritage by performing knowledge acquisition from historical documents. We developed a system that gathers knowledge by processing the content of historical documents to enable knowledge retrieval as response to ontologically-guided queries. Knowledge acquisition, one of the main workflows in our system, aims to semantically annotate the content of historical documents and to enrich the domain ontology through lexical annotation and knowledge extraction processes. We use two types of rules in knowledge extraction, one dealing with extracting the relevant information from the documents' content and another one for mapping the extracted information to ontology concepts and properties. Our work was validated on documents available in the Cluj County National Archives addressing the Transylvanian medieval history.**

***Index Terms*—knowledge acquisition, lexical annotation, knowledge retrieval, ontology, semantic annotation**

I. INTRODUCTION

Digitizing historical documents would greatly help historians and archivists to have an easy access to information and also to discover new knowledge through machine reasoning and learning. Digital content also helps in preserving the content of documents.

Research in fields like Natural Language Processing, Semantic Web and Information Extraction developed methods, techniques and technologies that enable the automatic processing of documents' content. In the historical domain, the task of document processing for information retrieval is laborious and time consuming mostly due to the documents' content heterogeneity.

Through our work we aim to create a system that facilitates the access to the documents in the National Archives by *(i)* creating a digital repository of semantically annotated historical documents, *(ii)* allowing machine reasoning and learning based on content data, and *(iii)* providing researchers and historians a means to obtain relevant results to their queries. We have accomplished these objectives by adopting Semantic Web techniques for knowledge capturing, representation and processing. We add a layer of machine-processable semantics over the content of historical documents by using a historical domain ontology containing concepts, instances and relations.

This paper presents our approach to the knowledge acquisition process. By analyzing together with historians a corpus of archival documents, we created a historical domain ontology and history-specialized rules for extracting and semantically annotating relevant information from the documents' content. Our approach was inspired by the OntoPop methodology [1] [2], introducing new processing steps. Our solution was tested on documents available in the Cluj County National Archives [12] regarding the medieval history of Transylvania.

In the following introductory sub-sections we briefly describe the corpus of raw documents and the architecture of our system. Next, in the main sections of this paper, we detail the knowledge acquisition layer.

A. Corpus Description

Our corpus is obtained by pre-processing a set of original archival documents (called ODoc documents) about historical facts concerning the medieval history of Transylvania. The historical evolution of medieval Transylvania determined the heterogeneity of the ODoc documents in the archives.

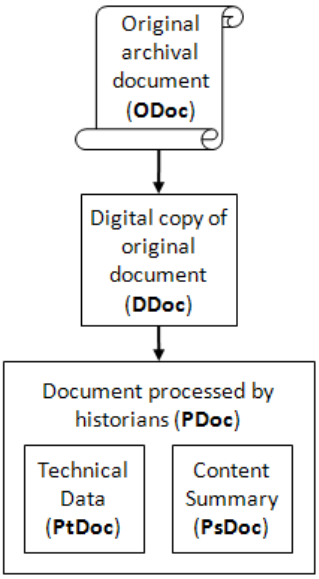

Figure 1. Historical document pre-processing steps.

Some important factors that contribute to the heterogeneity are: (i) the documents' language (Latin, Hungarian, German and Romanian), (ii) the institution that issued the document (different kinds of royal, local or

Digital Object Identifier 10.4316/AECE.2009.02003

religious authorities), (iii) whether the document is the original or a copy, (iv) whether the documents were printed or handwritten and (v) the writing embellishments that decorate the documents.

These characteristics lead to great difficulties in documents' automatic processing for information extraction and therefore we decided to use as input for our system the document summaries created and provided by archivists. Each ODoc document is digitized, thus creating a DDoc document. The DDoc document is then manually processed by the archivists as part of their professional task. As a result of this processing, a PDoc (standing for processed Document) document is generated. Each PDoc contains the technical data (PtDoc) and the summary (PsDoc) of the original document. Figure 1 presents the pre-processing steps that create the PtDoc and PsDoc starting from an ODoc.

The technical data included in the PtDoc refers to the date of issue, archival fund, catalogue number or other metadata. In Figure 2 we present an example of a PDoc document featuring its PtDoc and PsDoc. The PtDoc contains a set of technical data (the document number is 235, the language in which the document was written is Latin and the edition in which the original document has appeared is "Zimmermaan-Werner 1982 –I, nr. 169"), while the PsDoc is the original document summary.

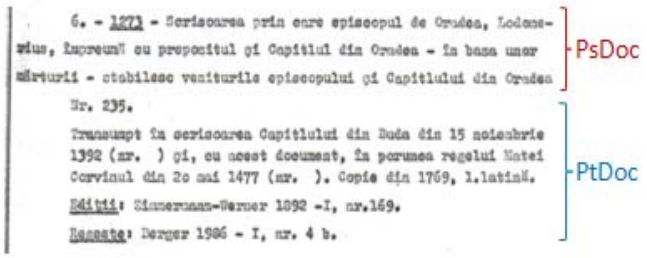

Figure 2. Example of a PDoc document.

B. *System Architecture*

An overview of our system is presented in Figure 3, outlining the conceptual layers and their associated resources and processes. The system is structured on three interacting processing layers: the raw data acquisition and representation layer, the knowledge acquisition layer and the knowledge processing and retrieval layer.

The Raw Data Acquisition and Representation Layer performs the task of collecting PDoc documents from external sources and of storing them in the Primary Database (PDB). PDoc documents can be acquired either from (external) databases, or by direct text input using the system's integrated user interface.

The relevant data from PDoc documents is identified and captured as knowledge by the Knowledge Acquisition Layer. The process of knowledge acquisition is divided in two main tasks: *(1)* document annotation and *(2)* ontology enrichment and population. PsDoc documents are lexically and semantically annotated based on the domain ontology and on a set of semantic rules. As PsDoc documents are processed, the domain ontology is populated with new instances and enriched with new concepts and relations. The Knowledge Acquisition Layer maintains a repository of Structured Data that stores XML files containing the lexical annotations associated to the documents. The Knowledge Server (KS) manages the domain ontology and the semantic annotations, stored as RDF files associated to the documents.

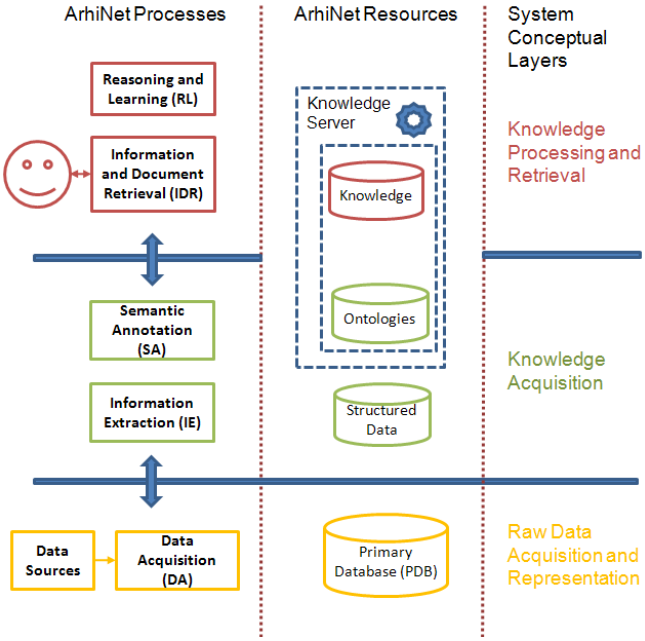

Figure 3. System Architecture.

The Knowledge Processing and Retrieval Layer retrieve the relevant information from the system's knowledge as response to ontologically-guided user queries. This layer also performs reasoning and learning in order to enhance the system knowledge.

The rest of the paper is organized as follows. Section II presents our approach to creating the core domain ontology necessary for our system. Section III presents knowledge acquisition from historical documents while Section IV introduces the Related Work. We end our paper with conclusions and future work proposals.

II. DOMAIN CORE ONTOLOGY

Ontologies stand at the basis of developing knowledge processing and retrieval systems, as they capture the knowledge about particular domains and provide support for semantic queries and reasoning. Therefore, one of the prerequisites in developing our system was to build a historical ontology from PsDoc documents available in the corpus.

Building the historical ontology is done through an iterative process which starts with designing a core of ontology concepts and relations. Starting from the core, the ontology is continuously enriched and populated as new documents are processed.

In order to design the core ontology we have studied the medieval history of Transylvania and analyzed a large set of corpus documents together with historians and archives experts to identify the common and relevant concepts and relations.

Most of the studied archival documents were official letters and this is why the sender (person or institution) of the document was considered to be an important ontology

concept. During the Middle Ages not everyone was entitled to issue letters (documents) and this is why the document senders were generalized as the concept of *Authority*.

We also considered as relevant the names of territories and places that appear in the documents. These names rarely refer to the place where the document originated, but they are generally tied to titles or disputed territories such as kingdom, principality, village or estate. Historians are interested in tracing all references of a territory to identify the persons who possessed it, whether the transfer of ownership was peaceful or not, or whether the territory was a donation or a disputed inheritance. We grouped territory types under the concept of *TerritorialDivision.*

Another noteworthy core ontology concept is Event which in the analyzed PsDoc documents is often related to complaints, donations, occupation or recognition. Depending on the way the documents are phrased, it is possible to identify various elements connected to an event: the involved parties, the reason that triggered the event or its date. These elements were represented as properties of the concept Event.

The Title (for example king or magister) of the persons mentioned in the PsDoc documents represents another important concept of the core ontology.

After several iterations and revisions, we concluded that the final structure of our medieval history core ontology is the one illustrated in Figure 4. We developed the domain ontology using Protégé [10].

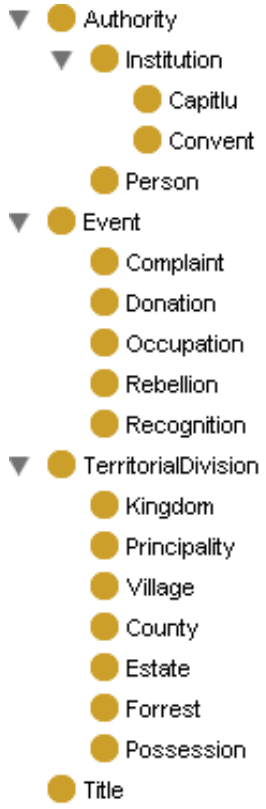

Figure 4. Our Domain Core Ontology.

III. KNOWLEDGE ACQUISITION FROM HISTORICAL DOCUMENTS

Information Extraction and Document Annotation is performed by the Knowledge Acquisition Layer. The objectives of this process are: *(1)* to extend the domain ontology by identifying and extracting relevant domain-specific information from the documents corpus and *(2)* to annotate the documents of the primary database with ontological concepts.

Each PDoc document in the PDB is processed by the Knowledge Acquisition Pipeline (see Figure 5). The pipeline is inspired by the OntoPop [1] [2] methodology introducing three new processing steps: *(i)* technical data extraction, *(ii)* synonyms population and *(iii)* homonym identification and representation.

The document processing pipeline, illustrated in Figure 5, is domain-independent and is intended to be applied to raw documents of any domain that preserve the same structure of the PDoc documents. In order to adapt the knowledge acquisition system to a new domain, we only need to change the domain-specific resources required at each processing step of the pipeline.

The domain-specific resources comprise the repository that stores the rules used to extract and represent the technical particularities of the documents, the pattern-matching rules used to perform lexical annotation, the domain ontology, the matching rules used to perform semantic annotation and the thesaurus and dictionary used for finding synonyms and homonyms.

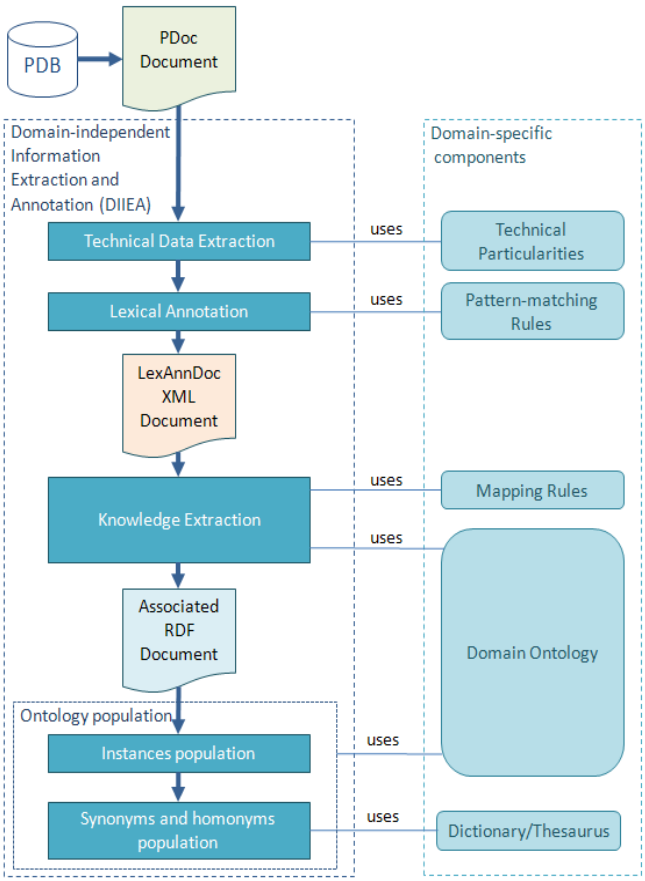

Figure 5. Knowledge acquisition pipeline.

In the following subsections, we describe the knowledge acquisition process that applies to each PDoc document, by detailing and exemplifying each pipeline processing step.

A. *Technical Data Extraction*

This step is used to obtain and process the relevant

technical data from the PtDoc part of the PDoc document. The technical data is domain specific and provides information about the type and content of the document. Usually, the technical data is created and managed by a domain expert. Our system uses the technical data for reasoning and validation purposes.

B. Lexical annotation

This step aims at extracting and lexical annotating the relevant information from the PsDoc part of the PDoc document. For lexical annotation we used the GATE [7] tool by adapting its resources to the particularities of the historical period considered. Some of these resources consist of pattern-matching rules which define relationships between the lexical elements and their annotations. The pattern-matching rules were created by a linguist after a thorough analysis of the historical documents and are represented as JAPE (Java Annotation Patterns Engine) grammars [7].

A JAPE grammar comprises a set of phases, each phase containing pattern/action rules. The rule's left-hand side includes the patterns to be matched while the right-hand side includes a set of actions that will be performed in case the left-hand-side pattern is matched.

Figure 6 illustrates an example of a JAPE rule created for the historical domain, which searches for instances of the child-parent relationship. The CandidateKinship_XSonOfY rule searches phrasal patterns of the form "X son of Y". The rule's left-hand-side contains the macros PERSON_COLLECTION and PERSON_COMPLEX. PERSON_COLLECTION identifies a sequence of TempPerson elements separated by a comma or by the conjunction "si" (in English: "and"). PERSON_COMPLEX identifies a sequence composed of *(i)* a TempPerson element, *(ii)* a comma or a "si" conjunction, *(iii)* a space and *(iv)* a TitleComplex element or a simple Title element. We decided to use macros because they can be reused in other rules. The rule CandidateKinship_XSonOfY searches for a sequence starting with *(i)* a PERSON_COLLECTION, *(ii)* a PERSON_COMPLEX or a TempPerson element, *(iii)* a comma, *(iv)* a space, *(v)* a kinship_relations element (for example "son"), *(vi)* a space, *(vii)* the conjunction "lui" (in English: "of") and *(viii)* finally a TempPerson element. For example, in the case of the text in Figure 7a, the rule CandidateKinship_XSonOfY matches on "Mihail si Nicolae, fii lui Albert de Juk", which in English reads "Mihail and Nicolae, the sons of Albert of Juk".

Besides the JAPE grammars, the lexical annotation process applied to the historical domain requires specific gazetteer lists which contain sets of Romanian names for persons, objects, cities or other things. GATE offers such gazetteers specialized for the Romanian language but we had to enrich them with information specific to the addressed historical periods such as events, kinship relations, titles, estates, etc.

Within the lexical annotation flow, the document's content is passed along with the gazetteer lists through a pipeline of JAPE grammars. The ANNIE (A Nearly-New Information Extraction System) [7] information extraction system part of GATE uses the JAPE grammars to extract and structure the relevant information. The identified lexical elements (lexical annotation tags) are structured as a hierarchy and stored in an XML file (LexAnnDoc).

```
Phase: Kinship_XSonOfY
Input: Lookup Token SpaceToken TempPerson Title
                                        TitleComplex
Options: control = appelt

Macro: PERSON_COLLECTION
(
  ({TempPerson}
    ({Token.kind == punctuation, Token.string == ","}
      | (SPACE {Token.string == "si"}))?
    SPACE)+
  {TempPerson}
)

Macro: PERSON_COMPLEX
(
  {TempPerson}
  ({Token.kind == punctuation, Token.string == ","}
    | {Token.string == "si"})?
  SPACE
  ({TitleComplex} | {Title})
)

Rule: CandidateKinship_XSonOfY
(
   ((PERSON_COLLECTION) |
     PERSON_COMPLEX |
     {TempPerson})
   ({Token.kind == punctuation, Token.string == ","})?
  SPACE
  {Lookup.majorType == kinship_relations}
  SPACE
  ({Token.string == "lui"}
    SPACE)?
  {TempPerson}
):kinship -->
     :kinship.Kinship_XSonOfY =
          {kind = "Kinship_XSonOfY"}
```

Figure 6. Example of a JAPE rule.

In Figure 7a we present the PsDoc part of a PDoc document used as input for the lexical annotation process. In English, this summary reads: "Carol Robert, the king of Hungary, donates to Mihail and Nicolae, the sons of Albert of Juk, the Palostelek domain and the Imbuz (Omboz) forest, in the Dabaca County, for their faithful military services carried out together with the magistrate Stefan, against Moise, a rebel against the crown". The result of lexical annotation, performed on this text is illustrated in the XML file presented in Figure 7b. All the XML files (LexAnnDocs) resulted from the lexical annotation of the PDoc documents are stored in the Structured Data repository (see Figure 3).

Lexical annotation is a prerequisite for the semantic annotation of the documents that is presented in the next section.

Carol Robert, regele Ungariei, doneaza lui Mihail si Nicolae, fiii lui Albert de Juk mosia Palostelek si padurea Imbuz (Omboz), din comitatul Dabaca, pentru serviciile de arme credincioase aduse alaturi de magistrul Stefan, impotriva lui Moise, razvratit impotriva coroanei.

a)

```
- <paragraph>
  - <EventDonationComplex>
    - <PersonComplex>
        <Person>Carol Robert</Person>
      - <TitleComplex>
          <Title>regele</Title>
          <Location>Ungariei</Location>
        </TitleComplex>
      </PersonComplex>
      <EventDonation>doneaza</EventDonation>
    - <Kinship_XSonOfY>
      - <PersonCollection>
          <Person>Mihail</Person>
          <Person>Nicolae</Person>
        </PersonCollection>
        <Person>Albert de Juk</Person>
      </Kinship_XSonOfY>
    </EventDonationComplex>
  - <LocationTypeCollection>
      <LocationType_Estate>Palostelek</LocationType_Estate>
      <LocationType_Forest>Imbuz</LocationType_Forest>
    </LocationTypeCollection>
    <LocationType_County>Dabaca</LocationType_County>
    <Highlight>serviciile de arme credincioase</Highlight>
  - <PersonComplex>
      <Title>magistrul</Title>
      <Person>Stefan</Person>
    </PersonComplex>
  - <Antagonist>
      <Person>Moise</Person>
    </Antagonist>
    <EventRebellion>razvratit</EventRebellion>
  </paragraph>
```

b)

Figure 7. Lexical annotation results.

C. *Knowledge Extraction*

Knowledge extraction aims at semantically annotation using ontological concepts and ontology population by using mapping rules. A mapping rule defines *(i)* a way to associate domain ontology concepts to the atomic elements of the lexical annotation tags stored in the XML file LexAnnDoc and *(ii)* a series of actions that need to be executed in order to populate the domain ontology with the instances identified in the XML file LexAnnDoc.

The result of the knowledge extraction process is a RDF file that stores in a hierarchical structure the semantic annotations associated to the PDoc raw document. All the generated RDF files are stored in the Knowledge repository inside the Knowledge Server (see Figure 3).

In order ease the parsing process of the Knowledge Extraction module, the mapping rules are created based on the template presented in Figure 8. Each mapping rule is defined in one XML document.

The root element (<rule>) groups all information regarding the rule. The <sem_tag> element identifies the root of a sub-tree in the LexAnnDoc. The values of the <context> element represent the first level nodes of the sub-tree and thus differentiate between the possible structures of the same <sem_tag>.

If the rule <context> matches a sub-tree in the processed LexAnnDoc, a set of actions (grouped by the <actions> element) are performed. An action (marked by the <action> element) either *(i)* adds a new instance to a concept in the ontology and marks it in the RDF file or *(ii)* defines an object property between two instances. For both cases, the action has the same structure and is described by its type (<atype>), action class (<aclass>) and action object (<aobject>).

```
<rule id="">
  <sem_tag></sem_tag>
  <context>
    <child_tag></child_tag>
    …
  </context>
  <actions>
    <action>
      <atype></atype>
      <aclass></aclass>
      <aobject child="" descendentOf="" index="">
                                  </aobject>
    </action>
    …
  </actions>
</rule>
```

Figure 8. The structure of a mapping rule.

For adding a new instance, the content of the <atype> element should be “addInstance”. The content of the <aclass> element represents the ontology concept that will be instantiated. The <aobject> element uniquely identifies in the addressed sub-tree of the LexAnnDoc the new instance to be added to the ontology.

```
<rule id="rulePersonComlex_v4">
  <sem_tag>PersonComplex</sem_tag>
  <context>
    <child_tag>Title</child_tag>
    <child_tag>Person</child_tag>
  </context>
  <actions>
    <action>
      <atype>addInstance</atype>
      <aclass>Person</aclass>
      <aobject child="yes">Person</aobject>
    </action>
    <action>
      <atype>addInstance</atype>
      <aclass>Title</aclass>
      <aobject child="yes">Title</aobject>
    </action>
    <action>
      <atype>hasTitle</atype>
      <aclass>Person</aclass>
      <aobject>Title</aobject>
    </action>
  </actions>
</rule>
```

Figure 9. Mapping rule example.

In case the action instantiates a property, the <atype> element content is the ontology property that will be defined. The domain and range instances of the property are

identified in the addressed sub-tree of the LexAnnDoc by the <aclass> and <aobject> elements.

An example of a mapping rule is presented in Figure 9. For the LexAnnDoc of Figure 7b, this rule maps on the <PersonComplex> sub-tree which contains the children "<Title>magistru</Title>" and "<Person>Stefan</Person>". As a result, "magistru" will be added as an instance of the "Title" class and "Stefan" will be added as an instance of the "Person" class. Also, between these two new instances, the "hasTitle" property will be defined, having "Stefan" as domain and "magistru" as range.

Table 1 illustrates other examples of mappings between lexical elements and semantic concepts. In the table we can see that for the lexical tag "Person", the "addInstance(Person)" action is performed on the domain ontology. For a composed lexical tag, like "PersonComplex", which defines as the context a sequence of "Person", "Title" (honorific distinction associated to the person) and "Location", the performed actions should add to the ontology the instances of Person, Title and Location followed by defining their corresponding properties, i.e. to associate the title to the person (hasTitle(Person,Title)) and to associate the territory to the title (hasCorrespondingTerritory(Title,Location)).

TABLE I. LEXICAL TO SEMANTIC MAPPINGS

Lexical Tag	Context	Action
Person	-	addInstance (Person)
PersonComplex	Person Title	addInstance (Person) addInstance (Title) hasTitle (Person, Title)
PersonComplex	Person Title Location	addInstance (Person) addInstance (Title) addInstance (Location) hasTitle (Person, Title) hasCorrespondingTerritory (Title, Location)
PersonCollection	Person Person	addInstance (Person) addInstance (Person)
Kinship_XSonOfY	Person Person	addInstance (Person) addInstance (Person) hasFather (Person, Person)

In Figure 10 we present a fragment of the RDF file resulting from the semantic annotation of the document in Figure 7a.

```
<rdf:RDF ... >
  ...
  <rdf:Description ... >
    <ontClass:Event>razvratit</ontClass:Event>
    <ontClass:Title>magistru</ontClass:Title>
    <ontClass:Event>doneaza</ontClass:Event>
    <ontClass:Title>regele</ontClass:Title>
    <ontClass:Person>Carol Robert</ontClass:Person>
    <ontClass:Kingdom>Ungariei</ontClass:Kingdom>
    <ontClass:Person>Stefan</ontClass:Person>
    <ontClass:Person>Moise</ontClass:Person>
  </rdf:Description>
</rdf:RDF>
```

Figure 10. Fragment of a semantic annotation RDF file.

D. Ontology Population

The new instances and relations identified during knowledge extraction are added to the domain ontology. We use a dictionary of synonyms to populate the domain ontology with all the synonyms of a newly found instance. The OWL-Lite ontology representation allows synonym definition through the "sameAs" property, which specifies that an instance X is equivalent with another instance Y. In the case of homonyms we defined a distance function that takes as arguments *(i)* the document technical data, *(ii)* attributes and relations of the ontology-stored instances and *(iii)* the current instance that is being verified for the homonymous relationship. In case the computed function value exceeds a certain threshold we consider that the two instances are identical, otherwise they are homonyms.

```
Person(?s) ^ Person(?f) ^ hasSon(?f, ?s)
                              -> hasFather (?s, ?f)
Person(?p) ^ Person(?b) ^ hasBrother(?p, ?b)
                              -> hasBrother(?b, ?p)
```

Figure 11. Example of SWRL logical inference rule.

The implemented ontology management activities aim to infer new relations and properties as a result of ontology modification due to previous population processes and to preserve ontology consistency. The inferring of the new properties uses the Jess rule engine [8]. For example, in a processed document we have identified the new instance "Mihail" and its associated property "hasFather" having the range "Albert de Juc". After populating the ontology with this information, the Jess rule engine infers the inverse property "hasSon" with the domain "Albert de Juc" and the range "Mihail". In order to allow this kind of logical inferences, the Jess rule engine requires SWRL [9] rules to be defined on the domain ontology. An example of ontology associated SWRL rules is illustrated in Figure 11. The first rule defines the "hasSon" relation between two persons as the inverse of the "hasFather" relation between the same persons. The second rule shows that the "hasBrother" relation between two persons is symmetrical.

Figure 12 illustrates the system's ontology after several ontology population iterations.

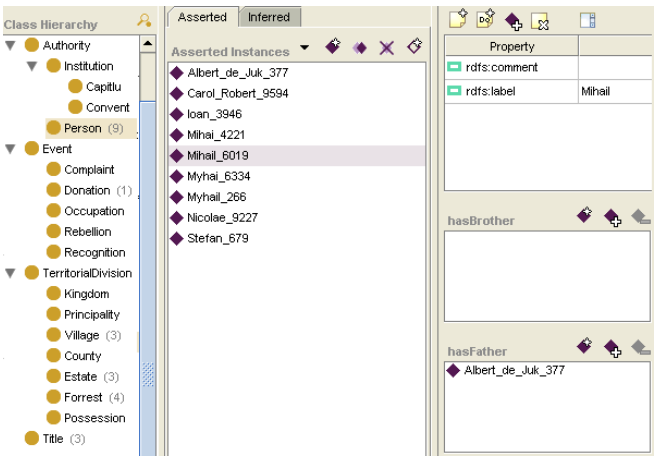

Figure 12. Domain ontology population.

IV. RELATED WORK

The OntoPop methodology [1] [2] provides a solution for the semantic annotation of documents. The solution mainly relies on knowledge acquisition rules which map the results obtained by text mining tools to formal representations such as RDF or OWL. One of the main advantages of OntoPop is the independence of each processing module which provides the methodology with flexibility features. However, the methodology has certain limitations because it does not address the processing of synonyms on one hand and the processing of multiple instances with the same lexical representation on the other hand. Our solution, presented in this paper, addresses the identified limitations by using the document technical data and reasoning for dealing with synonym and homonym instances in ontology population.

In [3], authors present SOBA, a system specialized on retrieving and annotating relevant football web pages. The Heart-of-Gold (HoG) architecture [6] is used as basis for lexical annotation and information extraction. The lexically annotated documents are processed and a football knowledge base is generated by mapping the annotated entities and events to ontological classes and properties [3]. SOBA processes domain structured documents, in contrast to our solution which considers documents with unstructured content.

Ontea [4] [5] represents a semi-automatic annotation and information retrieval technique which relies on the use of regular expression patterns, lemmatization methods and specialized indexing mechanisms. Annotations are stored in a knowledge base. Ontea proved to be suitable for processing Germanic and Slavic languages but it was not tested for Latin languages. Our approach presented in this paper is tailored to scale up to process multilingual documents, including Latin languages.

V. CONCLUSIONS AND FUTURE WORK PROPOSALS

In this paper we present a knowledge acquisition approach to generating and semantically enhancing archival eContent. We implemented this approach in a system that enables knowledge retrieval operations triggered by semantically enhanced queries. Within our system, knowledge acquisition is a four-stage process that aims to *(i)* extract the technical data found in the archival documents, *(ii)* lexically annotate the relevant information extracted from the content of documents, *(iii)* map the identified lexical elements to ontology concepts through knowledge acquisition rules and *(iv)* to populate the domain ontology. The results of the knowledge acquisition process consist of RDF files of semantic annotations attached to the archival documents and of an enriched domain ontology, altogether providing the necessary resources for knowledge retrieval.

Archival documents dating from the medieval times of Transylvania are not written only in Romanian, but also in German and Hungarian. Although the system architecture provides support for dealing with multilingual terms, the pattern-matching and mapping rules specific to the considered languages need to be developed as a subject of future work. This way, the system may allow queries expressed in a particular language to retrieve results from documents written in other languages.

REFERENCES

[1] F. Amardeilh: “Web Sémantique et Informatique Linguistique: propositions méthodologiques et réalisation d’une plateforme logicielle”. These de Doctorat, Universite Paris X-Nanterrere, 2007.

[2] F. Amardeilh: “OntoPop or how to annotate documents and populate ontologies from texts”. ESWC’06 Proceedings of the Workshop on Mastering the Gap: From Information Extraction to Semantic Representation, 2006.

[3] P. Buitelaar, P. Cimiano, S. Racioppa, M. Siegel: “Ontology-based Information Extraction with SOBA”. Proceedings of the International Conference on Language Resources and Evaluation, pp. 2321-2324, 2006.

[4] M. Laclavik, M. Ciglan, M. Seleng, S. Krajei: “Ontea: Semi-automatic Pattern based Text Annotation empowered with Information Retrieval Methods”. In Tools for acquisition, organisation and presenting of information and knowledge: proceedings in Informatics and Information Technologies, ISBN 978-80-227-2716-7, part 2, pp. 119-129, Kosice, Vydavatelstvo STU, Bratislava, 2006.

[5] M. Laclavik, M. Seleng, M. Babik: “OnTeA: Semi-automatic Ontology based Text Annotation Method”. In Tools for Acquisition, Organisation and Presenting of Information and Knowledge, ISBN 80-227-2468-8, , pp. 49-63, Vydavatelstvo STU, Bratislava, 2006.

[6] U. Schäfer: “Integrating Deep and Shallow Natural Language Processing Components – Representations and Hybrid Architectures”. Saarbrücken Dissertations in Computational Linguistics and Language Te, DFKI GmbH and Computational Linguistics Department, Saarland University, Saarbrücken, Germany, 2007.

[7] V. Tablan, D. Maynard, K. Bontcheva, H. Cunningham: “Gate – An Application Developer’s Guide”. Available online: http://gate.ac.uk/, 2004.

[8] Jess the Rule Engine for the Java Platform, Version 7.1. Available online: http://www.jessrules.com/jess/docs/Jess71.pdf, 2008.

[9] I. Horrocks et al.: “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”. Available online: http://www.w3.org/Submission/SWRL/, 2004.

[10] M. Horridge, et al: “A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools”. Available online: http://owl.cs.manchester.ac.uk/-tutorials/protegeowltutorial/

[11] O. Matei, Ontology-based knowledge organization for the radiograph images segmentation Advances in Electrical and Computer Engineering, vol. 8, no. 1, 2008, pp. 56-61.

[12] CCNA, Cluj County National Archives, Available online: http://www.clujnapoca.ro/arhivelenationale