INTEGRATED SYSTEM FOR DEVELOPING SEMANTICALLY-ENHANCED ARCHIVE eCONTENT

Mihaela Dinşoreanu Mihaela.Dinsoreanu@cs.utcluj.ro

Ioan Salomie Ioan.Salomie@cs.utcluj.ro

Cristina Bianca Pop Cristina.Pop@cs.utcluj.ro

Technical University of Cluj-Napoca
Department of Computer Science
Cluj-Napoca, România

Rezumat: Lucrarea abordează problema procesării cunoaşterii din documentele istorice disponibile în arhive. Astfel, se propune o soluţie integrată care efectuează extragerea de informaţii şi dobândirea de cunoştinţe pe de o parte şi regăsirea de informaţii şi cunoştinţe pe de altă parte. Se prezintă o metodă care adaptează cadrul text2Onto pentru a extrage semiautomat informaţiile relevante din conţinutul documentelor prin adnotarea textului din punct de vedere lexical şi semantic. Adnotările semantice vor popula ontologia unui domeniu care este utilizat în regăsirea de informaţii şi cunoştinţe. De asemenea, se prezintă o metodă pentru interogare în baza digitală de cunoştinţe a documentelor istorice în limbaj natural. Metoda este argumentată cu sugestii şi explicaţii semantice corespunzătoare fiecărui cuvânt. Soluţia integrată a fost testată şi validată pe un set de documente referitoare la istoria Transilvaniei.

Cuvinte cheie: dobândirea de cunoştinţe, adnotări semantice, regăsirea cunoştinţelor, interogarea limbajului natural.

Abstract: This paper addresses the problem of knowledge processing from historical documents available in archives. Thus, we propose an integrated solution which performs information extraction and knowledge acquisition on one hand and information and knowledge retrieval on the other hand. We present a method that adapts the Text2Onto framework to semi-automatically extract relevant information from the documents content through lexical and semantic text annotation. The semantic annotations will further populate a domain ontology which is used in information and knowledge retrieval. We also present a method for querying the digital knowledge base of historical documents in the Romanian natural language. The method is augmented with suggestions and word meaning disambiguation. We tested and validated our integrated solution on a set of documents addressing the history of Transylvania.

Keywords: knowledge acquisition, semantic annotation, knowledge retrieval, natural language query.

1. Introduction and Related Work

Historical documents represent valuable artifacts as they contain information that defines the identity of a nation. However, it is difficult to access this information as usually historical documents are not available to the general public due to the risk of deterioration. Current solutions try to address this issue by digitizing historical documents. Even so, it is difficult to automatically process the information of historical documents as they are available in large amounts, are distributed in archives and digital libraries, are written in natural language and contain unstructured and heterogeneous information. In this context, the challenge is to improve the current content management systems to automatically extract and process the relevant information from natural language historical documents.

The main objective of this paper is to address the challenge of processing natural language-written Romanian historical documents through information extraction and knowledge acquisition on one hand, and information and knowledge retrieval on the other hand.

Our approach addresses information extraction and knowledge acquisition by (*i*) adopting semantic Web techniques, (*ii*) adding a layer of machine-understandable semantics over the content of raw documents and (*iii*) capturing the semantics in a domain knowledge base. For knowledge acquisition we have used, adapted and improved the Text2Onto framework [1]. Text2Onto is an ontology learning framework which enriches a domain ontology with the relevant information extracted from text documents. Text2Onto uses a pipeline of GATE [2] components to linguistically process and annotate texts. Based on the linguistic annotations, the text is then semantically annotated using a set of JAPE rules which identify common language patterns. The final processing step consists of applying a set of algorithms and combiners to extract concepts, instances and relations which are further used to enrich the domain ontology. We chose the Text2Onto framework mainly because the level of the abstractions defined in this framework is more suitable for our needs compared to other approaches (e.g. OntoPop [3], Ontea [4], SOBA [5]). We adapted for the Romanian language the part of speech tagger and morphological analyzer components in the GATE pipeline, generated a domain specific set of JAPE rules and developed a new version of the OWL Writer component of Text2Onto.

In the case of information and knowledge retrieval, our approach uses the domain knowledge base for providing the most relevant results to ontology-guided natural language queries issued by archivists, historians or the general public. In the design of the method for information and knowledge retrieval we have inspired from Ginseng [6], Gino [7] and AquaLog [8]. Ginseng uses natural language to express queries based on an English-like grammatical structure augmented at runtime with elements dynamically obtained from a set of OWL ontologies. The static rules are actually general English constructs used to formulate sentences, while the dynamic rules are populated with matching instances from the loaded ontologies. Additionally, the grammar is used to provide a choice menu with ontology items matching the next grammar element to be introduced by the user based on the sentence structure. Compared to other natural language interfaces, Ginseng provides synonymy support. GINO extends Ginseng and provides ontology editing support through a natural language interface. AquaLog receives as input a natural language sentence which it translates into query triples that are further matched to ontology compatible triples sent to an inference engine to retrieve an answer. AquaLog also asks for disambiguation when a match between a query triple term and an ontology triple term cannot be found. Our querying approach combines a Ginseng-like grammatical structure with a disambiguation support similar to AquaLog. Unlike Ginseng which divides the grammatical rules into static and dynamic, our grammar is composed of a set of rules which contain both dynamic and static items. Similar to Ginseng and GINO we populate the dynamic items at runtime with entries from the ontology. Our grammar is designed for the Romanian language unlike previous approaches that have been designed for the English language.

The rest of the paper is structured as follows. Section II presents the archival domain model. Section III describes the proposed integrated system architecture and details its layers. Section IV contains a case study illustrating the system's functionality in the context of Romanian historical archives. The paper ends with conclusions and future work proposals.

2. The Archival Domain

The archival domain is modeled starting from a set of raw historical documents provided by the Cluj County National Archives [10]. These documents are hand written and contain many embellishments, making them hard to be automatically processed. Due to this difficulty, in our case studies we have used document summaries generated by the archivists. Within our model (see Figure 1), the central element is the document which belongs to the archival domain formally represented as domain knowledge by means of domain ontology (concepts and relations) and rules. Documents can be obtained from several data sources like external databases, Web sites or digitized manuscripts.

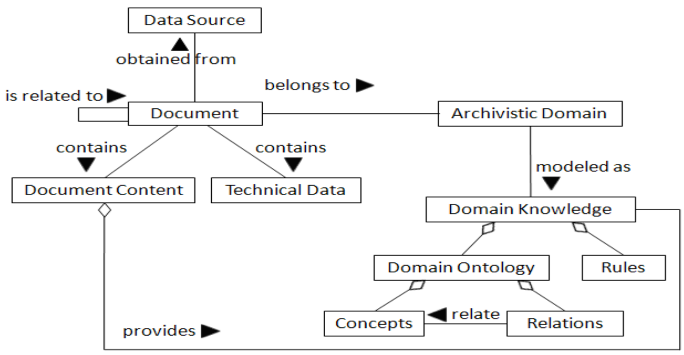
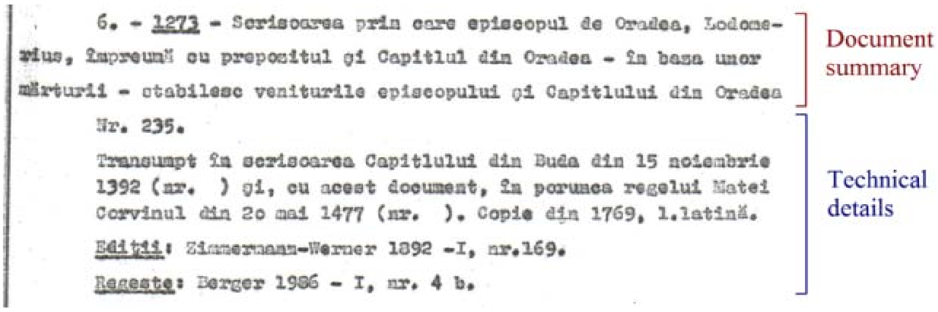


Figure 1. The archival domain model

The document content (see in Figure 2 an example) is expressed in natural language in an unstructured manner. In our case study, the document content actually represents a summary of the associated original document. Several documents may be related to one another by referring information about the same topics even if they are not containing the same lexical representations (e.g. names, events, etc.). The document also features a set of technical data, such as the date of issue, archival fund or catalogue number. In the case of the document shown in Figure 2, the technical data specifies the document number (“235”), the language in which the raw document was written (“Latin”) and the edition in which the original document has appeared (“Zimmermaan-Werner 1892 –I, nr.169”).

When searching in the archival documents it is important to identify all documents that are related to a specified topic. To enable information retrieval from all relevant documents, the domain knowledge is used to add a semantic mark-up level to the documents content.



6. - 1273 - Scrisoarea prin care episcopul de Oradea, Lodomerius, împreună cu prepozitul şi Capitlul din Oradea - în baza unor mărturii - stabilesc veniturile episcopului şi Capitlului din Oradea

Nr. 235.

Transumpt în scrisoarea Capitlului din Buda din 15 noiembrie 1392 (nr.) şi, cu acest document, în porunca regelui Matei Corvinul din 20 mai 1477 (nr.). Copie din 1769, l.latină.

Ediţii: Zimmermann-Werner 1892 -I, nr.169.

Regeste: Berger 1986 - I, nr. 4 b.

Figure 2. Example of a historical document with technical data and summary

The domain knowledge core (domain ontology and rules) is captured by processing and analyzing a large repository of archival documents, focusing on identifying their common concepts and relationships. Next, based on information extraction techniques applied on the raw documents, the domain knowledge is enriched through instance population.

3. The Integrated System

The integrated system has been designed as a management system for extracting and processing information, managing the resulting knowledge and querying it. For the system to achieve its goals, a set of distinct directions were identified: knowledge acquisition through information extraction and ontology enrichment, knowledge base management, reasoning on the domain repository, and guided natural language information retrieval. To address these issues we have organized the system’s conceptual architecture (see Figure 3) on the following three layers: the *Knowledge Acquisition layer*, the *Knowledge Processing layer* and the *Knowledge Retrieval layer*. Each layer supports user interaction and has an associated user type and workflow.

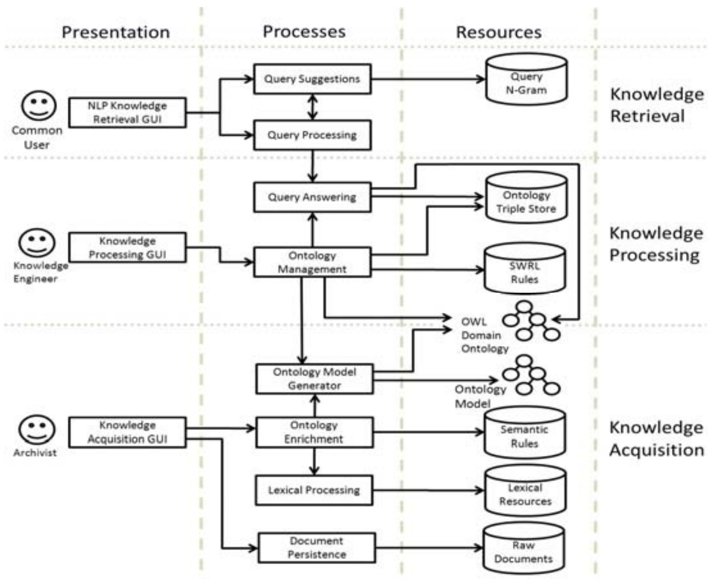


Figure 3.The conceptual architecture of the integrated system

The following sub-sections will detail each layer of the integrated system.

3.1 The Knowledge Acquisition Layer

The main objectives of the *Knowledge Acquisition layer* are the following: (1) to persist the raw documents in a document repository, (2) to extract the relevant information from the raw documents content, (3) to enrich the domain ontology with new concepts, instances and properties, and (4) to semantically annotate the raw documents with concepts from the system's domain ontology. In order to achieve these objectives we have adapted and integrated the Text2Onto framework [1] to our system such that it can deal with the Romanian language. Archivists are the only users that interact with the *Knowledge Acquisition layer* by submitting raw historical documents for processing. In what follows we detail the main activities of the *Knowledge Acquisition layer*:

1) *Document persistence*

Each raw document submitted to the *Knowledge Acquisition layer* is first persisted in a document repository which also stores the document's technical data.

2) *Lexical processing*

In this activity, a series of transformations are performed on the content of a historical document by submitting it to an adapted GATE pipeline [2]. In Table 1 we briefly describe the functionality of each transformation in the GATE pipeline as well as the adaptations we made to each of them.

Table 1. GATE Pypeline Description

Transformation	GATE functionality	Our adaptation
Tokenization	Divides the text into tokens: regular words, space and punctuation.	Identify words preceded by initials – for name specification (e.g. "A. Samsodi"). Identify words preceded by a hyphen – for verbs at past tense (e.g. "daruit-o" – English "donated")
Sentence splitting	Groups the tokens into sentences.	-
Part-of-speech (PSO) tagging	Assigns a part of speech to each word based on a lexicon which is manually populated.	Use the DEX database which contains all the possible inflexions for a large set of Romanian words.
Morphological analysis	Finds the stem of a word.	Use the DEX database which contains the roots for a large set of Romanian words.

3) *Semantic annotation*

In this activity, the language patterns specific for the Romanian historical domain are identified and annotated. The annotations represent ontology modeling primitives corresponding to ontology element types such as classes, instances, properties or subclass relations. These annotations are used by the Text2Onto algorithms to enrich the system's domain ontology. Semantic annotation is performed using a set of JAPE rules, each rule being composed of a (i) left hand side – contains regular expressions and annotations obtained in the Lexical Processing activity that are matched on text fragments, and a (ii) right hand side – contains a set of commands that are executed when the left hand side matches a text fragment.

4) *Ontology enrichment*

This activity applies the Text2Onto semantic processing algorithms on the semantic annotations identified in the semantic annotation activity to enrich the ontology model. The ontology model is created incrementally starting from an ontological core (contains concepts and properties most widely encountered in the processed Romanian historical documents). The ontology model is stored in memory as a Probabilistic Ontology Model (POM) [1] which is then translated in an OWL ontology representation by an adapted OWL writer component.

3.2 Knowledge Processing Layer

The *Knowledge Processing layer* provides support for ontology management through automatic reasoning, consistency checking, ontology classification, rule-based inference and ontology realization. Knowledge engineers are the only users that interact with the *Knowledge Processing layer* to manage the domain ontology obtained from the *Knowledge Acquisition layer*, by adding, removing or changing ontological concepts.

Due to the facts that SWRL rules can potentially generate many instances, rules implicit in the semantics of OWL generate many instances and reasoning tools generate auxiliary runtime data, we decided to (*i*) keep the Terminological Box elements (classes with subclasses and properties with their domain, range and sub-properties) in an OWL ontology and to (*ii*) keep the Assertional Box elements (class instances and property instances) in a relational database. This relational database is designed as a triple store containing a set of (*subject*, *object*, *property*) triples. As a result of this split storage we had to implement SWRL inference and queries processing.

In what follows we detail the activities performed in the *Knowledge Processing layer*:

1) *Ontology consistency checking and classification*

The main objectives of this activity are to ensure that the ontology does not contain inconsistent facts and that the ontology model contains the complete class and property hierarchies. These objectives are achieved by (i) checking the consistency of the ontology and classifying it each time the system's domain ontology is updated, (ii) processing only the Terminological Box elements and (iii) using the Pellet OWL-DL reasoner [9].

2) *SPARQL queries processing*

The aim of this activity is to provide information about Terminological and Assertional Box elements. This is achieved by (*i*) bounding the variables that refer to Terminological Box elements to their values using the OWL-DL reasoner, (*ii*) bounding the variables that refer to the Assertional Box elements by querying the relational database and (*iii*) consolidating the overall result after the execution of the previous steps.

3) *Rule-based inference*

The main objectives of this activity are to reason about the information contained in the system's domain ontology and to produce new knowledge. These objectives are achieved by (*i*) using DL-safe SWRL rules containing variables that only refer to explicitly named Assertional Box elements, (*ii*) implementing each SWRL rule as a database stored procedure (each rule variable corresponds to a database record) and (*iii*) adopting a forward chaining strategy to generate at each database update all the possible inferences and to store the results back into the database.

4) *Realization*

Within this activity, the most specific class or property of a given instance is found so that the same Assertional Box assertions are generated as any OWL-DL reasoner, but performing all the work in the database. This objective is achieved by (i) taking into account implicit rules from the OWL standard that link instances to classes or properties, and (ii) defining the implicit rules for realization from the OWL standard as explicit SWRL rules.

3.3 The Knowledge Retrieval Layer

The main objective of this layer is to enable the general public to find the relevant historical documents and information by querying the system in natural language. This objective is achieved by (*i*) creating a natural language interface which offers guided querying through ranked suggestions and word disambiguation, (*ii*) composing queries with the aid of a support grammar and (*iii*) adding semantic relevance to query suggestions by taking into account the relations between ontology concepts present in the query.

In order to enable natural language guided knowledge retrieval we have defined an EBNF-like grammar which adheres to the syntax of the Romanian language. The grammar templates contain static and dynamic concepts. The static concepts include prepositions, conjunctions, sentence markers or domain specific words that help the user underlining the semantic meaning of the sentence. Dynamic concepts include nouns, verbs, verb phrases or time variables taken from the domain ontology.

In what follows we detail the main activities of this layer:

1) *Query suggesting*

Suggestions are presented to the user as soon as he/she begins typing the query. Ontology suggestions are requested through SPARQL queries and they are ranked according to an N-gram model. In our approach we employ two query suggestions methods: a grammar-only suggestions method and an incremental suggestion method.

The grammar-only suggestions method retrieves all ontology constructs matching each dynamic query item requested by the grammar. It does not try to semantically interpret the queries and it only knows the grammar constraints imposed to provide domain specific interrogations.

The incremental suggestion method adds semantic meaning to partially built queries by exploring the relationships between the concepts already written in the query. Thus, it retrieves ontology concepts matching each dynamic query item requested by the grammar which are in a valid relationship with already selected concepts.

2) *Suggestions ranking*

This activity relies on an N-gram model which represents a probabilistic method for predicting the Nth word from N-word sequence based on the previous N - 1 words. Search engines use such N-gram models for suggestion ranking to provide suggestions which have proved to be desired by users.

3) *Word meaning disambiguation*

There are situations in which the user may introduce unknown words – words that cannot be parsed in accordance to the grammar or matched to an ontology concept. In addition, spelling mistakes might be made or concepts that do not fit any grammar template might be introduced. To address these issues we provide two disambiguation methods, one involving query reparsing and word-to-concept association, and another one involving explicit selection of a possible meaning from provided dialog boxes.

4. Case Study

As the aim of our integrated system is to develop and process semantically enhanced archival eContent from documents about the history of Transylvania we have tested and validated it on a set of 100 historical documents provided by the Cluj County National Archives [10]. To exemplify how the integrated system works we are going to trace the execution of its layers using the document content illustrated in Figure 4.

Ladislau, regele Ungariei daruieste alesului Nicolae de Ocna Sibiului, pentru merite in luptele cu turcii mosiile Albesti (Feyeregyhaz) din comitatul Alba, partile de mosie din Jewedich din comitatul Tarnava si Langadar din comitatul Tarnava, foste ale lui Mihail de Nades.

a) Original Romanian text

Ladislau, the king of Hungary, donates to the chosen Nicolae from Ocna Sibiului, for his faithful military service in the battles with the Turkish army, the Albesti (Feyeregyhaz) estates from the Alba county, the estate parts from Jewedich from the Tarnava county and Langadar from the Tarnava county, which previously belonged to Mihail of Nades.

b) English translation

Figura 4. Document content example

A. Knowledge Acquisition

The text is first input into the GATE pipeline which tags each word with a set of annotations.

For example, Figure 5 illustrates that as a result of the tokenization process, the token "*Ladislau*" was annotated with the type "kind" and the value is "word". The following annotation is "length" with value 8. The next annotation is "orth" and its value is "upperInitial", indicating that it is written with capital initial letter and the last annotation is "string" with the value "*Ladislau*", representing the concrete word. Similarly all the words from the text are annotated with their corresponding kind, length, orth and string.

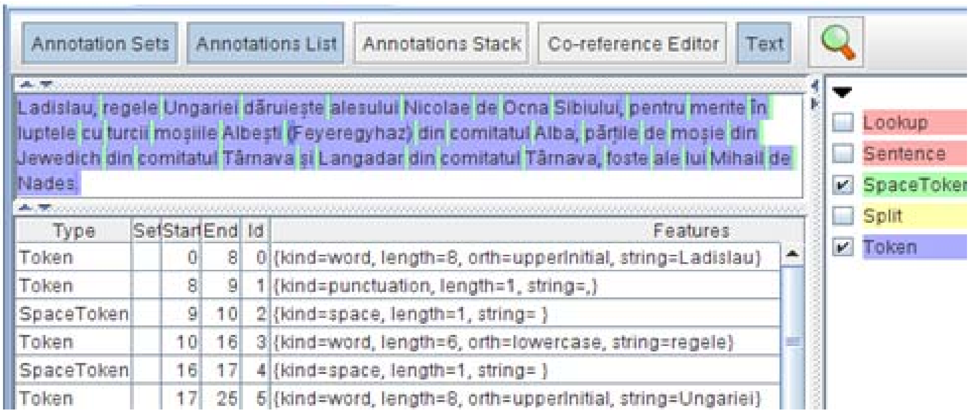


Figura 5. GATE Tokeniser annotations (fragment)

In Figure 6 a small part of the POSTagger results can be seen. For example, for "*mihail*" the determined part-of speech is NNSPN (Proper Noun Phrase in Nominative/Accusative) and for "*luptă*" (engl. "*battle*") is NNSCN (Noun Phrase in Nominative/ Accusative).

```
sibiului NNSPG
mihail NNSPN
pentru PP
merit NNSCN
de PP
în PP
luptă NNSCN
cu PP
```

Figura 6. POSTagger Annotations (fragment)

Figure 7 presents a section of the Morphological Analyzer results, in which it can be seen that the determined stem for the word “comitatul” (engl. “the county”) was “*comitat*”.

```
comitatul Base Word: comitat
din Base Word: din
, Base Word: ,
alba Base Word: alba
de Base Word: de
părţile Base Word: parte
din Base Word: din
moşie Base Word: moşie
din Base Word: din
jewedich Base Word: jewedich
târnava Base Word: târnava
```

Figura 7. Results of the Morphological Analyzer (fragment)

For the document in Figure 4, a set of JAPE rules and macros are activated. In Figure 8 we present one example of such an activated JAPE rule which matches on complex structures to extract a relation between instances (one subject, and multiple objects). The name of the relation is formed based on the identified verb. Table II shows the resulted annotations derived from the matching on the rule in Figure 8.

```
Rule: TransitiveVerbPhrase2
(
    (ProperNounPhrase1):subject1
    (SimilarInstance)?
    (ProperNounPhraseFrom)?
    (
        (NounProperPhrasesAlternativesNonGenitive)
        |
        (NounPhraseDescriptionNonGenitive)
        |
        (NounPhraseDescription2NonGenitive)
    )?
    ({Token.string == ","})?
    ({Token.category == VBPT}):verb1

    (   {Token.string == "şi"}
        (NounPhraseComplete)?
        (ProperNounPhrase1):object4
        (SimilarInstance)?
        (ProperNounPhraseFrom)?
    )?
):transitive -->
:transitive.TransitiveVerbPhrase = { rule = "TransitiveVerbPhrase2" },
:verb1.Verb = { rule = "TransitiveVerbPhrase2" },
:subject1.Subject = { rule = "TransitiveVerbPhrase2" },
:object1.Object = { rule = "TransitiveVerbPhrase2" },
:object2.Object = { rule = "TransitiveVerbPhrase2" },
:object3.Object = { rule = "TransitiveVerbPhrase2" },
:object4.Object = { rule = "TransitiveVerbPhrase2" }
```

Figura 8. TransitiveVerbPhrase2 JAPE rule

Table 2. TransitiveVerbPhrase2 Rule matching results

Annotations	Transitive Verb Phrase 2
Subject1	"*Ladislau*"
Verb1	*"dăruieşte"* (engl. donates)
Object1	*"Nicolae de Ocna Sibiului"*
Object2	*"Albeşti"*
Object3	*"Jewedich"*
Object4	*"Langadar"*

Using the annotations obtained as a result of applying the JAPE rules, Text2Onto algorithms populate POM with the identified instances, concepts, instance-of relations, subclass-of relations, similarity and general relations (see Figure 9).

```
POMRelation=[rege( ladislau, ungaria ), dăruieşte( ladislau, nicolae de ocna sibiului )
POMInstanceOfRelation=[instance-of( ladislau, rege ), instance-of( alba, comitat ),
POMInstance=[feyeregyhaz, ungaria, mihail de nadeş, târnava, ladislau, jewedich,
POMSubclassOfRelation=[subclass-of( instituţie, autoritate ), subclass-of( domn, titlu )
POMConcept=[persoană, instituţie, magistru, locaţie, moşie, şcoală, biserică, curator,
POMSimilarityRelation=[similar-to( albeşti, feyeregyhaz )]}
```

Figure 9. Probabilistic Ontology Model (fragment)

The determined modeling primitives are subsequently translated in OWL DL (see Figure 10).

```
<a persoană rdf:ID="ladislau">
    <owlx:Label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ladislau</owlx:Label>
    <a:are_Document rdf:resource="#Top_1.txt"/>
    <a:dăruieşte rdf:resource="#albeşti"/>
    <a:dăruieşte rdf:resource="#jewedich"/>
    <a:dăruieşte rdf:resource="#langadar"/>
    <a:este_rege_in rdf:resource="#ungaria"/>
</a:persoană>

<a:ţară rdf:ID="ungaria">
    <owlx:Label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ungaria</owlx:Label>
    <a:are_Document rdf:resource="#Top_1.txt"/>
</a:ţară>
```

Figure 10. The resulted ontology (fragment)

B. Knowledge Processing

The ontological elements extracted from the text in Figure 4 trigger the execution of the SWRL inference process part of the Knowledge Processing layer. As a result, the SWRL rules illustrated in Figure 11 are run and assert that "Nicolae de Ocna Sibiului" owned (Romanian "detinutDe") the "Albesti", "Langadar" and "Jewedich" estates and their property deeds (Romanian "actproprietate").

$$doneaza(?x,?y) \wedge donatLui(?y,?z) \wedge actproprietate(?d) \wedge \\ certifica(?d,?y) \Rightarrow detinutDe(?z,?d) \wedge detinutDe(?z,?y)$$

Figure 11. Example of SWRL rules

C. Knowledge Retrieval

In Figure 12 we illustrate the step-by-step construction of a natural language-guided query.

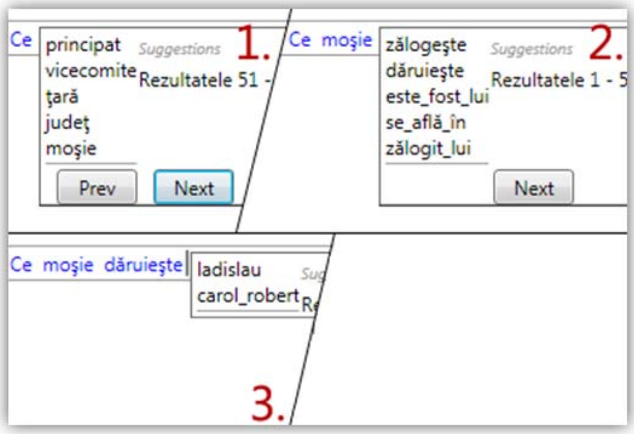


Figure 12. Step-by-step construction of a natural langiage-guided query

In the first step, after writing the static word 'Ce' – one of the two partially filled SPARQL queries will be: SELECT ?X WHERE {?X ?Property ?Z . ?X rdf:type owl:Thing } as 'Ce' is a conjunction used with objects. In the second step, after selecting the word 'moşie' corresponding to the dynamic grammar item ObjectInstance the query will be: SELECT ?Property WHERE {?X ?Property ?Z . ?Z rdf:type o:moşie}. In the third step, after selecting the word 'dăruieşte' corresponding to the dynamic grammar item Verb the query will be: SELECT ?X WHERE {?X o:dăruieşte ?Z . ?Z rdf:type o: moşie}. Finally, after selecting 'ladislau' – the user ends the query and requests execution which returns all estates which were gifted by Ladislau.

5. Conclusions and Future Work Proposals

This paper presents our integrated approach for information extraction, knowledge acquisition and information and knowledge retrieval from Romanian historical documents. Information extraction and knowledge acquisition have been approached by adapting and modifying the Text2Onto framework and its associated resources, due to the grammatical constructions specific to the Romanian language and historical documents. Our approach also addresses the problem of knowledge management by implementing functionalities such as automatic reasoning, consistency checking, ontology classification, rule-based inference and ontology realization. In addition, we provide support for natural language-based interrogations formulated by the general public in their search for documents and information.

As future work, we intend to write new JAPE rules suitable for identifying other grammatical structures. In addition, we intend to improve the query answering performance by using several optimization mechanisms and also to automate the creation of SWRL rules by means of ontology data mining techniques.

REFERENCES

1. **CIMIANO, P.; VÖLKER, J.:** Text2Onto–A Framework for ontology Learning and Data-Driven Change Discovery. Proc. of the 10th International Conference on Applications of Natural Language to Information Systems, Vol. 3513 of LNCS, 2005, pp. 227-238.

2. **CUNNINGHAM, H. et al.:** Developing Language Processing Components with GATE. Available online at: http://gate.ac.uk/.

3. **AMARDEILH, F.:** OntoPop or how to annotate documents and populate ontologies from texts. Proceedings of the ESWC 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation, 2006.

4. **LACLAVIK, M.; CIGLAN, M.; SELENG, M.; KRAJEI, S.:** Ontea: Semi-automatic Pattern based Text Annotation empowered with Information Retrieval Methods. Tools for acquisition, organisation and presenting of information and knowledge: Proceedings in Informatics and Information Technologies, Kosice, Vydavatelstvo STU, Bratislava, part 2, (2007), ISBN 978-80-227-2716-7, 2007, pp. 119-129.

5. **BUITELAAR, P.; CIMIANO, P.; RACIOPPA, S.; SIEGEL, M.:** Ontology-based Information Extraction with SOBA. Proceedings of the International Conference on Language Resources and Evaluation, 2006, pp. 2321-2324.

6. **BERNSTEIN, A.; KAUFMANN, E.; KAISER, C.; KIEFER, C.:** Ginseng: A Guided Input Natural Language Search Engine for Querying Ontologies. Jena User Conference, 2006.

7. **BERNSTEIN, A.; KAUFMANN, E.:** GINO - A Guided Input Natural Language Ontology Editor. Proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, Georgia, 2006, pp. 144-157.

8. **LOPEZ, V.; MOTTA, E.; UREN, V.; PASIN, M.:** AquaLog: An Ontology-driven Question Answering System for Organizational Semantic Intranets. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5(2), 2007, pp. 72-105.

9. The Pellet OWL-DL Reasoner. Available: http://clarkparsia.com/pellet.

10. Cluj County National Archives, http://www.clujnapoca.ro/arhivelenationale/.