Semantic Tools for Forensics: A Highly Adaptable Framework

Michael Spranger, Stefan Schildbach, Florian Heinke, Steffen Grunert and Dirk Labudde
University of Applied Sciences Mittweida
Bioinformatics Group
Department of MNI
Germany, 09691 Mittweida
Email: {*name.surname*}@hs-mittweida.de

***Abstract*—Textual information or data annotated with textual information (meta-information) are regular targets of securing or confiscating relevant material in the field of criminal proceedings. In general evaluation of relevant material is complex, especially the manual (re)search in the increasing amount of data as a result of cheaper storage capacity available nowadays therefore the identification of valid relations are enormously complex, error-prone and slow. In addition, the adherence to time limits and data privacy protection make searching even more difficult. The development of an (semi-)automatic high modular solution for exploration of this kind of data using capabilities of computer linguistic methods and technologies is presented in this work. From a scientific perspective, the biggest challenge is the automatic handling of fragmented or defective texts and hidden semantics. A domain-specific language has been defined using the model-driven approach of the Eclipse Modeling Framework for the purpose of developing forensic taxonomies and ontologies. Based on this, role-based editors have been developed to allow the definition of case-based ontologies and taxonomies and the results of manual annotation of texts. The next steps required for further development are going to include comparison of several back-end frameworks, e.g., for indexing, information extraction, querying and the providing of a graphical representation of relations as a knowledge map. Finally, the overall process needs to be optimized and automated.**

Keywords*—*forensic; ontology; taxonomy; querying; framework.

I. Introduction

The analysis of texts retrieved from a variety of sources, e.g., secured or confiscated storage devices, computers and social networks, as well as the extraction of information, are two of the main tasks in criminal proceedings for agents or other parties involved in forensic investigations. However, the heterogeneity of data and the fast changeover of communication forms and technologies make it difficult to develop one single tool covering all possibilities. In order to address this problem, a domain framework is presented in this paper applying computer linguistic methods and technologies on forensic texts.

In this context, the term *forensics* relates to all textual information which maybe used during the procedure of taking evidence in a particular criminal proceeding. In particular, it corresponds to the hidden information and relations between entities achieved through the exploration and application of computer linguistic processing of potential texts.

Generally, there are a variety of tasks which need to be addressed:

- Recognition of texts with a case-based criminalistic relevance
- Recognition of relations in these texts
- Uncovering of relationship networks
- Uncovering of planned activities
- Identification or tracking of destructive texts
- Identification or tracking of hidden semantics

In this context, the term *hidden semantics* is synonymous with one kind of linguistic steganography, whereas such texts are defined as ”...made to appear innocent in an open code.” [1]. Each of these tasks can be processed and solved by combining several highly specialized services that encapsulate a problem solver based on a specific text mining technology. This problem solver can be combined and recombined like a tool kit to achieve a polymorphous behaviour depending on the kind of texts and the particular question under investigation.

Basic structural concepts of an application framework suitable to deal with these problems are presented within this paper. The previous steps of development will be outlined in the following sections.

- Development of criminalistic ontologies
- Development of criminalistic corpora
- Development of the framework's architecture
- Implementation of a prototype for manual evaluation

Specific ontologies and taxonomies are not being introduced in this paper. Case-based specific ontology and taxonomy are currently under evaluation applying the generic ontology editor developed in this work and will be released soon together with basic structures.

II. Development of criminalistic ontologies

The term *ontology* is commonly understood as a formal and explicit specification of a common conceptualization. In particular, it defines common classified terms and symbols referred to a syntax and a network of associate relations [2] [3]. Developing ontologies for criminalistic purposes is a prior condition for annotating texts and raise questions in this particular domain. The term *taxonomy* as a subset of ontology is used for the classification of terms (concepts) in ontologies and documents. On the one hand, a criminalistic ontology is charactarised by its case-based polymorphic structure and on the other by special terms used in criminal proceedings.

A domain-specific language is necessary at the beginning to describe taxonomies and ontologies for the development of a criminalistic ontology. The domain ontologies considered need to be highly specialized by taking into account the individual nuances of the particular criminal proceeding and the legal requirements due to privacy protection. For these reasons, a vast ontology covering all areas of crime is not employable. Special case-based ontologies, in accordance to a suitable predefined ontology, are necessary and preferably developed by the person heading the investigation. Thus, it is important that the definition of the predefined ontology is easily and case specific adaptable.

The Eclipse Modeling Framework (EMF) [4] [5] has been chosen for the purposes of this work mainly because of its perfect integration into the Eclipse environment, but also for participating in the manifold advantages of the approach of a model-driven software development. To follow this paradigm, the next step required is the definition of an abstract syntax (meta-model) for describing such taxonomies and ontologies. The meta-model created that way is used for generating a concrete syntax, especially source code, that provides all model and utility classes required.

In the literature there are different approaches for representing semantics under discussion, with Topic Maps have been proven to be one of the most expressive. Topic Maps is an ISO-standardized technology for representation of knowledge and its connection to other relevant information. It enables multiple, concurrent, structurally unconstrained views on sets of information objects and is especially useful for filtering and structuring of unstructured texts [2] [6]. Therefore the Topic Maps standard has been chosen to be the starting point of the meta-model development. Since EMF already includes options for persistence as well as model searching and (strategic) traversing, only the necessary syntactical elements and paradigms from the ISO standard have been adopted. These syntactical elements provide a complete description of semantic relationships. Note, the specification given in this work takes into account the specifics of the domain with respect to slang, multilingualism and the underlying hidden semantics. The syntactical elements used for further development are defined below and examplified by Figure 1 attached.

Subject (Topic) *red circle* represents an abstract or concrete entity in the domain to be analyzed.

Instance (Topic) *yellow circle* is the concrete manifestation of a subject.

Descriptor (Topic) *orange circle* typifies any other syntactical elements; i.e. adds further details related.

Association *blue rectangle* is a relation between two topics, usually subject and instance.

Association Role specifies the roles of the topics in an association (optional).

Occurrence corresponds to the concrete manifestation of a topic in a resource, usually related to an Instance.

Topic Name is the name representation of topics (container).

Name Item denotes the name of a specific topic, associated to a Scope.

Facet names a class of attributes of a topic and can include several Facet Values.

Facet Value a particular attribute as distinct value, can be a topic or another Facet.

Scope defines semantic layers; e.g. causing system to focus by filtering particular syntactical elements.

Figure 1: Use case tax fraud - an application of Topic Maps derivative as developed under this work for modeling a criminalistic ontology.

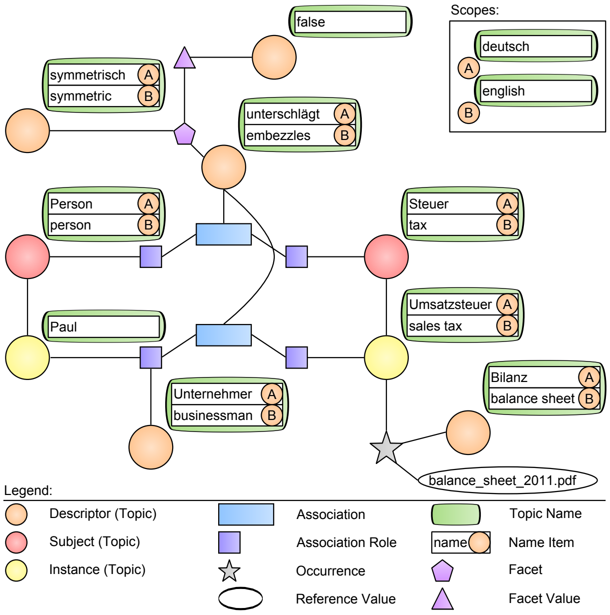

Fig. 1. Sample extract of the application "tax fraud" demonstrating a typical interaction of the different elements. The Subject *person* is described more specifically by adding the Instance *Paul*. The *person* called *Paul* then is related to the Subject *tax*, more specific the Instance *sales tax*. This by the Association, described in more detail by the Descriptor *embezzles*. *Pauls* role in this relationship is specified by the Descriptor of the Association Role *Businessman*. The Instance *sales tax* is creating an Occurence specified by the Descriptor *balance sheet* referring to the Reference Value *balance_sheet_2011.pdf* attached as evidence.

After developing the Domain Specific Language (DSL), the user interface of the ontology and taxonomy editor have been designed in this work.

At this stage of the work, the real development of a criminalistic classification and ontology has been initiated. The basis for comprehending forensic data and its relationship to case-based information has been achieved in cooperation with the local criminal investigation department. In this way, a set of metadata could be established entitled to be as close to reality as possible.

III. Development of criminalistic corpora

An extensive corpus is needed for the evaluation of the implemented functionalities and development of powerful algorithms in order to detect more semantic details, especially in fragmented and defective texts and for detecting hidden semantics. Building the extensive corpus required using original data from prior preliminary investigations is not suitable because of legal requirements of data privacy protection. This data is exclusively available during the current proceedings.

An alternative method is the exploration of significant characteristics of forensic texts and generating corpora in an artificial way where it is possible to take a completely artificial creation of text into consideration. This can be realized in two ways. The first is character level based, which causes the text to be alienated by non-words and unsuitable, but proper names [7]. The second way, superior from our point of view, is based on morphemes. While the occurrences of non-words can be eliminated, the target language, in the current case German, as a non-agglutinative language, raises problems among this method in shaping and bending words [8]. In summary, the basic problem with both approaches is the possible semantic interruption of text units.

A further method is to generate texts by modifying existing sources. In this case, the internet holds numerous potential domain-specific corpora. Analyzing significant websites, ebooks or expert forums are just a few options for generating suitable texts.

Concluding, the Internet-based concept is more valuable for the project presented here. Therefore, a method for transforming texts is necessary. Common approaches, like lookup based exchanges of words (via free dictionaries), adapting typo errors (missing, wrong or twisted letters) and manipulating the orthography of words, are suitable in this case.

IV. The framework's architecture

Especially due to its platform independency, *Java* has been used for the development. The high modularity is ensured by employing the *Eclipse RCP* as a basis. Its *OSGi* [9] implementation *Equinox* allows to construct service-oriented architectures (SOA) within the *Java Virtual Machine*. The framework conceptually consists of three main modules (see Figure 2):

- **Ontology Machine** it includes all functionalities for developing criminalistic taxonomies and ontologies.
- **Indexing Machine** it includes functions and methods for extraction and annotation of forensic data.
- **Querying Machine** it includes the functions for searching and visualizing semantic coherences.

The framework is developed using the OSGi paradigm by participating in its progressive concepts of service oriented architectures, like loosely coupling, reusability, composability and statelessness. Each service encapsulates a single computer linguistic method or technology. In this way, it is ensured that new functionalities based on actual insights of research can be added without adapting the framework's architecture. A qualitative scheme of the service landscape is depicted in Figure 3. The core of the framework is split in three service-tiers:

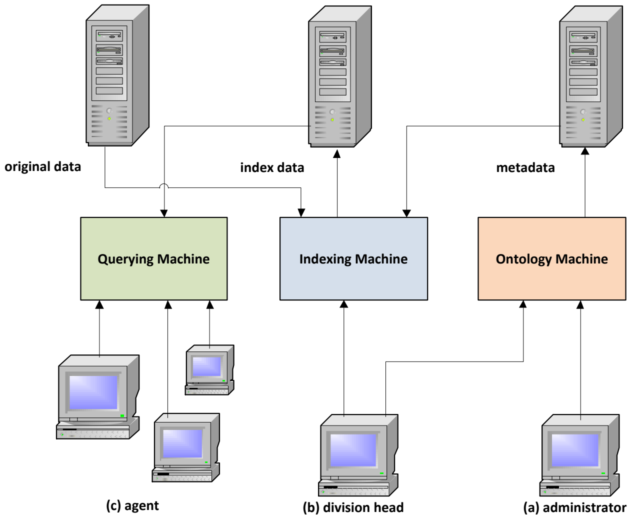

Fig. 2. A black-box view on the new multiple-role framework. (a) The *administrator* defines at least one taxonomy, in order to enable the classification of texts using the *Ontology Machine*. This data will be stored at the *metadata* server. (b) The person heading the investigation (*division head*) defines a case-based ontology using the same machine. In addition he/she can annotate the *original data* using the *Indexing Machine*. Whereas this machine combines original data and *metadata* and transforms it to index data. (c) The *agent* can access the system using the *Querying Machine*, which only has access to reading the *index data*.

a) Persistence: In addition to index and metadata server the persistence-tier includes the original data server. It keeps sensitive and evidentiary data strictly separated from other parts of the system. Its interface permits read-only access for the system. The index server provides access to the processed and annotated documents in their intermediate form. The metadata server manages the ontology and taxonomy data in addition to user accounts. In contrast to the original data server, the interfaces of index and metadata server provide full access to the system.

b) Logic: Four low-level services compose the core of the logic-tier. The extracting service is mainly responsible for extracting text from numerous data types, such as .doc, .pdf, .jpg, etc. In addition, several filters for morphological analysis can be applied. The document provider service transforms the extracted data into the document-based intermediate form. It collaborates as service composition with the extracting service, therefore service consumers only need to utilize this service. The main task of the index service is to provide CRUD-operations (acronym for Create, Read, Update, Delete) for accessing index data. It will be used for annotating and querying the document's index by the high-level services of

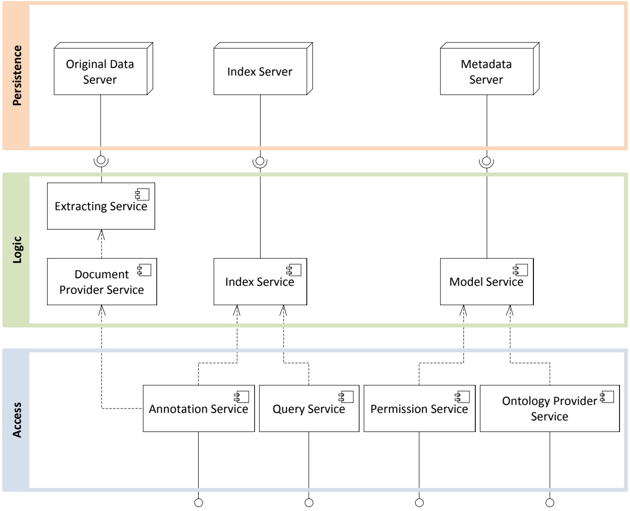

Fig. 3. Architecture overview. The framework's service landscape is divided into three tiers. Each machine (see Fig. 2) can use services from the *access-tier* directly. The *logic-tier* is providing atomic services and service compositions to solve a single problem (The figure shows only a few services exemplified). Accessing these services is only possible by utilizing services of the lower-level tier. The *persistence-tier* is responsible for keeping data and contains the three servers described in Fig. 2. It is only accessible by services of the *logic-tier*.

the access-tier (see c). In the same way, the model service is providing CRUD-operations for accessing metadata. This service is being used by higher-level services working with ontologies and user permissions.

c) Access: The access-tier contains the high-level services for using the low-level services from outside of the core. Subsequently the data is bound to the user interface. The function of high-level services is similar to the facade pattern [10]. The annotation service takes the documents from the document provider service and enriches them with additional user-specified data or data derived by other automatic information extraction services. The index service is used for transforming the data into the document-based intermediate form and pushes them to the index server. The query service fetches index data via the index service from this server, satisfying various filter criteria. The ontology provider service has to perform two tasks. On the one hand, it controls the collaborative access to the ontology model. On the other hand, it provides CRUD-operations on this on a higher level than the model service. Finally, the permission service controls the access permissions of each user to the well-defined data types (see I). Because the user data model is developed in a model-driven way analogue to the data model of ontologies and taxonomies this service collaborates with the low-level model-service. Thus, the same infrastructure as the ontology provider service can be used.

Especially the *logic-tier* is designed to include new functionality, since its services have an open architecture for extending their capabilities. For example, they provide interfaces for adding further services, such as text extraction methods, machine learning algorithms, etc.

V. Conclusion

The development of a high-modular framework for applying methods of natural language processing on forensic data is discussed In this work. Its service-oriented architecture is particularly suitable to include new functionality based on actual insights of research. In this way new knowledge will become available for the points of interest in shorter times.

The main task of the new framework is to support the criminal proceedings in evaluation of forensic data. The concept discussed in this paper is schematically summarized and illustrated in Figure 4. As elucidated, the structure mentioned gives the advantage that accessing and working with the framework is reliably ensured by using the few high-level services exclusively. In contrast, the service-compositions on the lower level can be as complex as needed and can be adapted at any time to achieve improved problem solving.

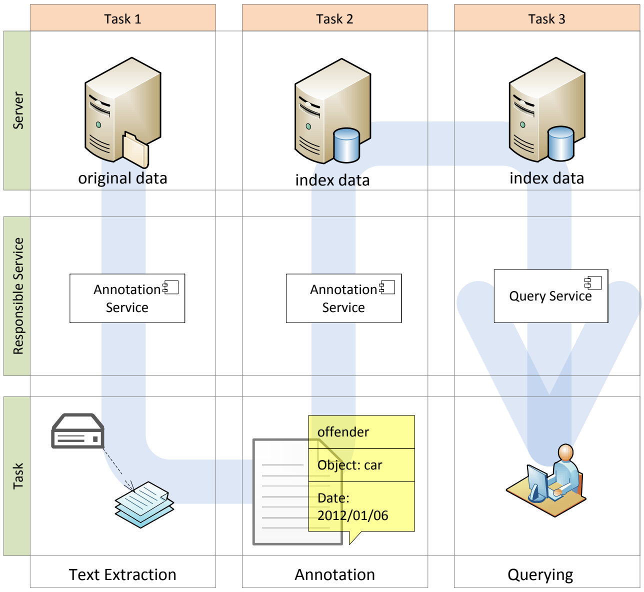

Fig. 4. Task matrix. Task 1) Texts can be extracted from the original data by using the high-level *Annotation Service*. Task 2) Subsequently, extracted texts can be annotated (semi-)automatically and indexed by the same service. To ensure proper processing of this task, an ontology and taxonomy have to be created before using the *Ontology Provider Service* (schematically depicted in Figure 3). Task 3) At this point, each agent can access the indexed data and create knowledge maps using the high-level *Query Service*.

Currently, the first prototype for manual annotation and development of criminalistic taxonomies and ontologies is evaluated in practice. In the next steps, the development of powerful algorithms for automation is emphasized. Especially, ways to extract information from defective texts and hidden semantics will be evaluated and revised.

Acknowledgment

The authors would like to thank the members of the criminal investigation department Chemnitz/Erzgebirge (Germany). We acknowledge funding by ”Europäischer Sozialfonds” (ESF), the Free State of Saxony and the University of Applied Sciences Mittweida.

REFERENCES

[1] Friedrich L. Bauer, *Decrypted Secrets - Methods and maxims of Cryptology*, 1st ed. Berlin, Heidelberg, Germany: Springer, 1997.

[2] Andreas Dengel, *Semantische Technologien*, 1st ed. Heidelberg, Germany: Spektrum Akademischer Verlag, 2012.

[3] Thomas R. Gruber, *Toward Principles for the Design of Ontologies Used for Knowledge Sharing.* In Nicola Guarino and Roberto Poli (Eds), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993.

[4] The Eclipse Foundation 2012, *Eclipse Modeling Framework Project (EMF)*, viewed 03 August 2012, http://www.eclipse.org/emf.

[5] Dave Steinberg, Frank Budinsky, Marcelo Paternostro and Ed Merks, *EMF Eclipse Modeling Framework*, 3rd ed. Boston : Addison-Wesley, 2009.

[6] JTC 1/SC 34/WG 3, *ISO/IEC 13250 - Topic Maps, Information Technology, Document Description and Processing Languages*, 2nd ed. 2002.

[7] Ilya Sutskever, James Martens, and Geoffrey Hinton, *Generating Text with Recurrent Neural Networks*, In Proceedings of the International Conference on Machine Learning (ICML), pp. 1017-1024, 2011.

[8] Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne Jekat, Ralf Klabunde and Hagen Langer, *Computerlinguistik und Sprachtechnologie - Eine Einführung*, 3rd ed. Heidelberg, Germany: Spektrum Akademischer Verlag, 2010.

[9] OSGi^{TM}Alliance 2012, *Technology*, viewed 03 August 2012, http://www.osgi.org/Technology/HomePage.

[10] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software.*, 1st ed. Amsterdam : Addison-Wesley Longman, 1994.

[11] Jeff McAffer, Paul VanderLei and Simon Archer, *OSGi and Equinox - Creating Highly Modular Java Systems*, 1st ed. Boston : Addison-Wesley, 2010.