Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 314 (2015) 83–100

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Software Architecture for Document Anonymization

Horacio Vico[1]

División Informática
Poder Judicial
Montevideo, Uruguay

Daniel Calegari[2]

Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Abstract

Organizations often have a dilemma in relation to their documents: ensure confidentiality of the data or publish the information contained in them, for transparency, scientific interest, or other reasons. In this context is that document anonymization arises, i.e. the replacement of sensitive data in such a way that preserves the confidentiality of the documents without altering their value or usefulness. There are proposals for (semi)automatic anonymization, but they are often domain-specific or they partially address the problem. In this paper we present a software architecture for supporting document anonymization, which is based on the representation of the problem as a domain and platform independent configurable business process. In addition, we analyze the technological alternatives for implementing the architecture and we present a functional prototype applied to the domain of legal documents.

Keywords: document anonymization, software architecture, business process

1 Introduction

Document management is the set of activities for the creation, reception, organization, storage, preservation, access and dissemination of documents within an organization [20]. Through the use of technology it is possible to improve management and maximize the value of the huge amount of information within those documents. In this context, two apparently conflicting interests arise: organizations must ensure confidentiality of the personal information they handle, but restricting access to such information is not a valid alternative. This happens whenever the

[1] Email: hvico@poderjudicial.gub.uy

[2] Email: dcalegar@fing.edu.uy

http://dx.doi.org/10.1016/j.entcs.2015.05.006
1571-0661/© 2015 The Authors. Published by Elsevier B.V.

organization needs to make available much of their information, either because of transparency or because the stored documents have a scientific, biomedical or legal value which has no relation with the personal information contained within them.

A solution to this problem is document anonymization, i.e. the total or partial replacement of personal data by eliminating any reference to their identity, without altering the value or usefulness of the original document [6]. Such anonymization can be manually performed by a user, being tedious, repetitive, error-prone and time consuming, not providing added value to the organization. This problem has driven research and development of techniques and methodologies for (semi)automatic anonymization.

The problem is not trivial given that many documents have no structured format to easily identify the sensitive information within them, thus it is necessary to combine different computational disciplines such as natural language processing, text mining, and machine learning to solve the problem. Particularly, from the point of view of the software architecture, the integration of different technological elements that can be used in an anonymization process represents a major challenge.

There are previous works proposing architectures to tackle with this problem [16,12,9], but they have some limitations: (a) they are domain-specific solutions making them hard to adapt to other contexts; (b) they partially resolve the problem without clearly defining the overall anonymization process; (c) they are not flexible enough to add new tools into the process; (d) the implementation of these solutions is not public and therefore it is not possible to experiment with them.

In this paper we present a software architecture that supports document anonymization overcoming the limitations of existing architectures. In particular, this architecture represents the problem as a domain and platform independent configurable business process [23]. This allows its implementation using a Business Process Management System (BPMS, [23]). Furthermore, we analyze different technological alternatives to implement a functional version of the reference architecture with freely available tools, thereby reducing licensing fees. Finally, we made a more qualitative assessment of the proposal by implementing a functional prototype applied to the domain of legal documents, in the context of the Judiciary of Uruguay.

The rest of this paper is structured as follows. In Section 2 we introduce the main aspects concerning document anonymization. In Section 3 we present the main architectural initiatives addressing this problem and we describe their common and special features. In Section 4 we describe our proposal and in Section 5 we present the development of a prototype of the reference architecture and its application on a case study. Finally, in Section 6 we present the main conclusions and some ideas for future work.

2 Anonymization of Documents

Very often, documents stored in an organization contain personal or sensitive information of citizens or legal persons, whose privacy must be guaranteed by the organization. In fact, there are regulatory and legislative basis that expose the dissemination of personal data to civil and criminal penalties. Particularly in Uruguay there is a law of personal data protection [5] regulating the responsibility for the protection of personal data held by citizens or legal persons. Among other things, the law determines what kind of information is public (e.g. name, surname, ID) and what kind information is defined as sensitive data (i.e. data revealing racial or ethnic origin, religious or moral convictions or information concerning health or sexual life) and therefore need the consent of their owner to be published.

The definition of public or sensitive data is context-dependent and thus vary according to the domain of interest and country. For example, the Uruguayan law which protects personal data within medical records [5] states that the medical record is owned by the patient, it must be reserved and can only be accessed by the medical care and administrative personnel in a direct relationship with the patient, the patient itself or his family, and the Ministry of Public Health if necessary. However, HIPAA (Health Insurance Portability and Accountability Act) in the United States [3], that regulates similar aspects, defines that a hospital can use patient data for research without consent, if a process of depersonalization of the information is done.

Beyond the restrictions, depending on the nature of the business domain of an organization, their documental databases are a source of knowledge that can be used by other organizations or individuals, sometimes with a huge scientific, social or technical benefit, since documents have an intrinsic value that is independent of the confidential information contained therein. Examples include e-government, biomedical sciences (e.g. medical records management) and judicial areas. In response to this problem is that anonymization arises.

The anonymization is focused on the preservation of the confidentiality of personal data, without modifying the value or usefulness of the documents. An accepted definition of anonymity is in the Law 14/2007 of Spain [6] with respect to biomedical research. This law defines anonymization as the process by which it is no longer possible to establish by reasonable ways the link between data and the subject to which it is related. There are two levels of anonymization: irreversible and reversible. Irreversible anonymization involves removing any information that can identify an individual or organization without the possibility of recovering it later. Meanwhile, with reversible anonymization (also called depersonalization) sensitive information is cross-referenced with other information such that an authorized entity can re-identify the anonymized information.

The anonymization process must preserve the useful content of the document and maintain its semantic coherence. In this sense, it is desirable that when for example the name of a person is identified as sensitive information, the anonymization process must replace each reference to that person with the same generic term. Beyond that, the name could appear written in different ways, e.g. it is common to refer to a person initially with its full name and then only with its last name or initials. The anonymization process must combine these terms in a single entity (Named Entity [14]), in order to keep the consistency of the original document. It is also desirable to identify if the entity was a name, a geographic location, a date, etc., before the anonymization process begins. In this case the generic term must also refers to this fact in order to preserve even more the meaning of the document.

This process can be manually performed by a user, being tedious, repetitive, error-prone and time consuming, not providing added value to the organization. This happens in the Judiciary of Uruguay, context of special interest since the administration of justice often means exploring aspects of private individuals or organizations and this information is recorded in different documents (records, statements, etc.). The Judiciary of Uruguay has a database of legal documents which is openly accessible through the web [4]. However, the anonymization process of judicial sentences requires a user to read each statement contained in an unstructured document, to identify sensitive data and to manually replace them with generic identifiers keeping document's consistency. This process is expensive, so there is a need to assist it by using specialized technology.

3 Architectural Initiatives

There are some architectural initiatives defining a document anonymization process arising from academic work and available software tools.

An initiative focused on Spanish language documents is ANONIMYTEXT [16]. The authors propose a complete architecture for an anonymization software to be used with medical unstructured documents. They define a combination of techniques to achieve the identification of sensitive information. The proposed schema, which is shown in Figure 1, starts with a business expert (a medical doctor) that defines a list of sensitive concepts which can be identified within documents, such as names and addresses. This information is used to generate a dictionary of terms, taken as input of the second step which performs a semantic analysis of the text labeling the concepts of interest. Then, the sensitive information is detected using the labels and the configuration that is provide in accordance with the law. Then the document can be reviewed by a business expert who approves or rejects the text, also providing feedback to the system. Finally, the document is anonymized in a way that sensitive information is encrypted with a public key algorithm.

Another initiative is MOSTAS [12] referring to the identification of biomedical terms in unstructured documents in Spanish language. The proposed schema is illustrated in Figure 2. The system receives unstructured clinical information and a morpho-semantic analysis is performed using a general purpose Spanish dictionary. This analysis identifies general terms that have no value from the biomedical point of view. The words not recognized in this step (which may be named entities) are look into more specific dictionaries, e.g. biomedical acronyms and abbreviations.

There are other proposals that provide a partial solution to the problem, basically following the same approach as before, but with some differences. Free software HIDE [10] uses a labeling process for linking all sensitive information to an entity (age, address, name of a person, etc.) and lets you select a strategy for full or partial anonymization of some critical attributes. On the other hand, other natural

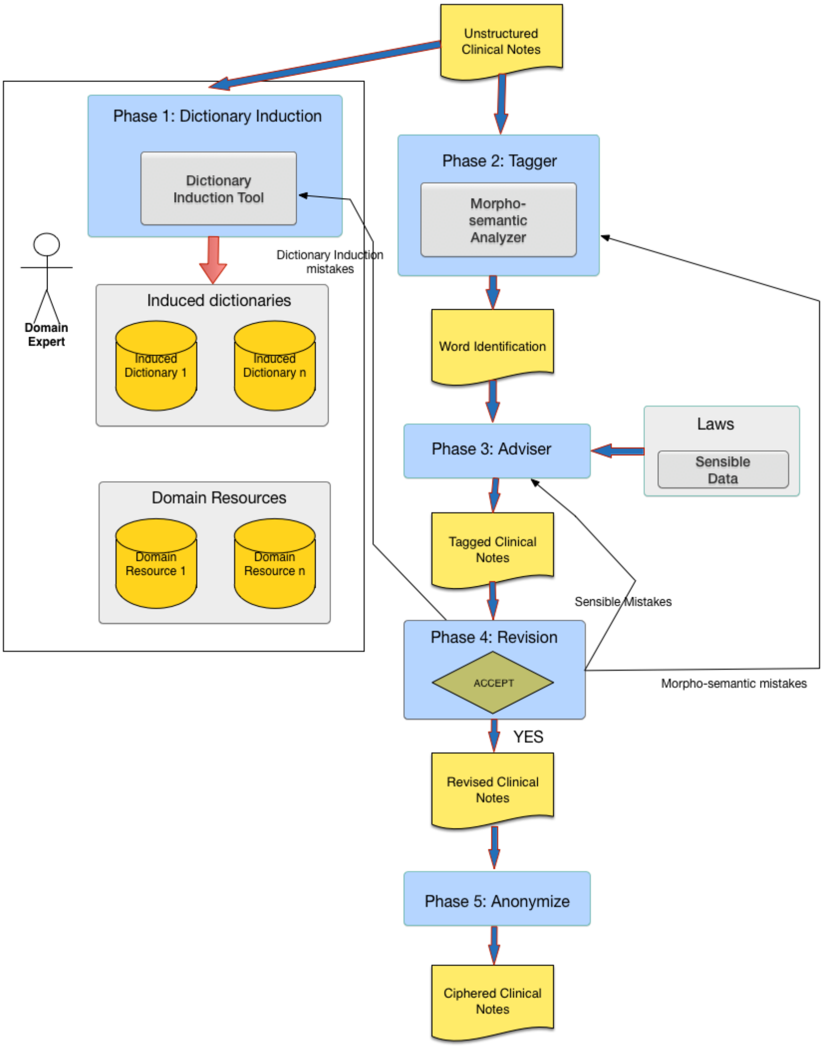

Fig. 1. ANONIMYTEXT's architecture

language processing tools can be used, such as a morphological tagger, which basically label the terms in a text by assigning a list of grammatical categories. In [8] a morphosyntactic tagger for Spanish language is presented that incorporates the use of heuristics to determine the nature of words processed by the morphological analyzer with higher accuracy. The author also proposes a heuristic of great interest for the anonymization process for the identification of proper names using a long list of names of people, towns, cities and countries. For the identification of the proper name, the heuristics consider the appearance of the word in any of these listings, the presence of a capital letter at the beginning of the word, the position of the word in the sentence, the fact of been recognized by the morphological analyzer or not, etc. An additional aspect, not specifically shown in any of the surveyed architectures is about the identification of co-references, which consists in applying a clustering technique to identified entities [18], e.g. group the full name of a person

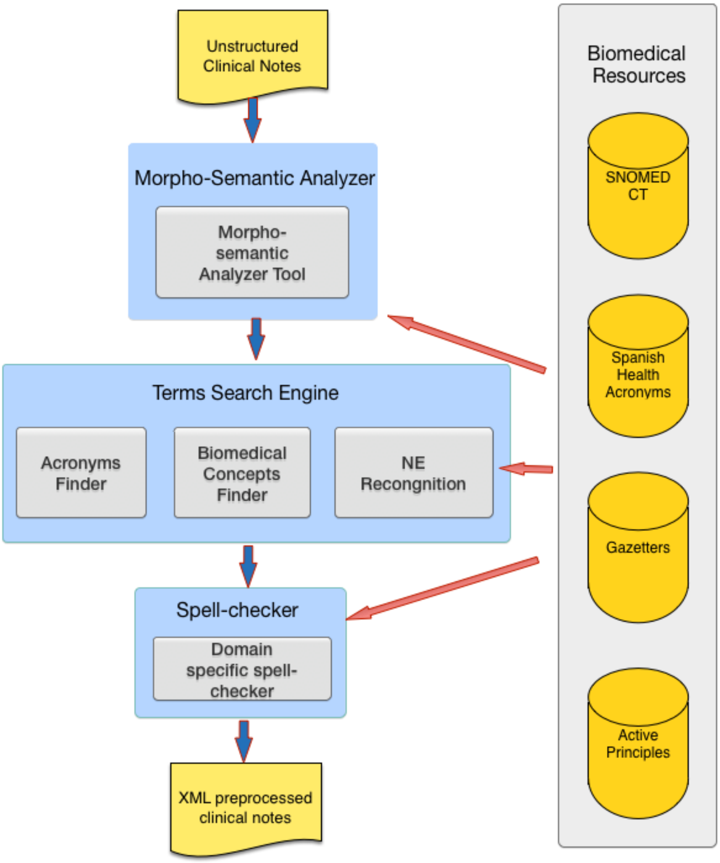

Fig. 2. MOSTAS's architecture

with only his name or initials.

From the analysis of the previous architectures, we identify a set of common and specific components which can be considered in our reference architecture. These components are summarized in Table 1. Beyond these aspects, the existing architectures present some limitations: (a) they are domain-specific solutions making them hard to adapt to other contexts; (b) they partially resolve the problem without clearly defining the overall anonymization process; (c) they are not flexible enough to add new tools into the process; (d) the implementation of these solutions is not public and therefore it is not possible to experiment with them.

In every proposal, the main component is the Named Entity Recognition (NER, [14]) that recognizes and labels entities of interest in the text. There are several NER tools [21,15,17], with more or less levels of accuracy in their results. Some of them also provide additional features such as the classification of entities. It has sense to think about an architectural in which it is possible to adapt and interchange these NER tools. In addition, every proposal identifies a component that performs the final processing of the text (Anonymizer). Specializations of this module must also be considered, as the anonymization process may be reversible or irreversible,

Table 1
Components of anonymization architectures

Component	ANONIMYTEXT [16]	MOSTAS [12]	HIDE [10]	Tagger [8]
NER	Yes	Yes	Yes	Yes
Corrector	No	Yes	No	No
Heuristics	Yes	Yes	No	Yes
NE Clustering	No	No	No	No
Reviewer	Yes	No	Yes	Yes
Anonymizer	Yes	Yes	Yes	No

partial or total, according to the requirements of each application domain.

Some of the approaches propose the concept of feedback after processing entities. One of them (MOSTAS) proposes a post-processing of the results using a spell checker when some elements were not identified. It would be desirable in a reference architecture that these potential post-processing steps have similar input and output interfaces, so that individual components may be added and removed as if they are links in a chain. Likewise, it is desirable to apply heuristics or group entities. Another approach is to allow an expert to identify the type of errors made by the NER component using some user interface and provide feedback to improve training of the NER component.

4 Reference Architecture

At a higher abstraction level, we define a generic architecture (which can be fully accessed in [22]) for any kind of anonymization system following the proposal of Rozanski and Woods [19]. We have documented the views of interest to this generic system, and the main aspects are explained in the following sections. In particular, we describe the following views:

- Functional View: describing in detail the steps required to perform the anonymization process, the software components within the system, their responsibilities, interfaces and interactions with the rest of the system.
- Information View: describing the main data structure for representing the unstructured text to be anonymized and the results of the anonymization process, as well as the information flow through the system.
- Development View: describing implementation aspects of the proposed architecture, such as extensibility mechanisms to add and combine tools used during the anonymization process, and existing technological alternatives.

From the analysis of the existing architectures and their strengths and aspects to improve, we define the set of functional and non-functional requirements presented

Table 2
Functional and non-functional requirements for the reference architecture

FR1	The system must be able to process unstructured documents in plain text format.
FR2	The system must be able to use different software tools for text processing interchangeably. It must be possible to configure the tool to use at each stage.
FR3	The system must allow a domain expert to validate the anonymized output and approve or reject the resulting document providing feedback to the system.
FR4	It must be possible to configure the level of anonymization: total (all attributes that identify individuals or organizations), or partial (a subset of those attributes).
FR5	The system must allow to configure the kind of anonymization: irreversible (completely removing sensitive information), or reversible (sensitive information is replaced with a ciphered reference).
FR6	The system must allow the definition of rules and patterns (heuristics) by an expert user to provide domain specific characteristics of documents.
FR7	The system must allow to interact with external sources, via adapters, to provide information about domain specific terms (dictionaries, thesauri, newsletters, acronyms, etc.)
NFR1	Adaptability: to support FR2.
NFR2	Configurability: to support FR2 and FR5.
NFR3	Interoperability: to support FR7.
NFR4	Extensibility: to support FR2 y NFR1 without big changes.
NFR5	Auditing: every user interaction with the system must be recorded.
NFR6	Security: non-anonymized documents must not be accessed by unauthorized users.

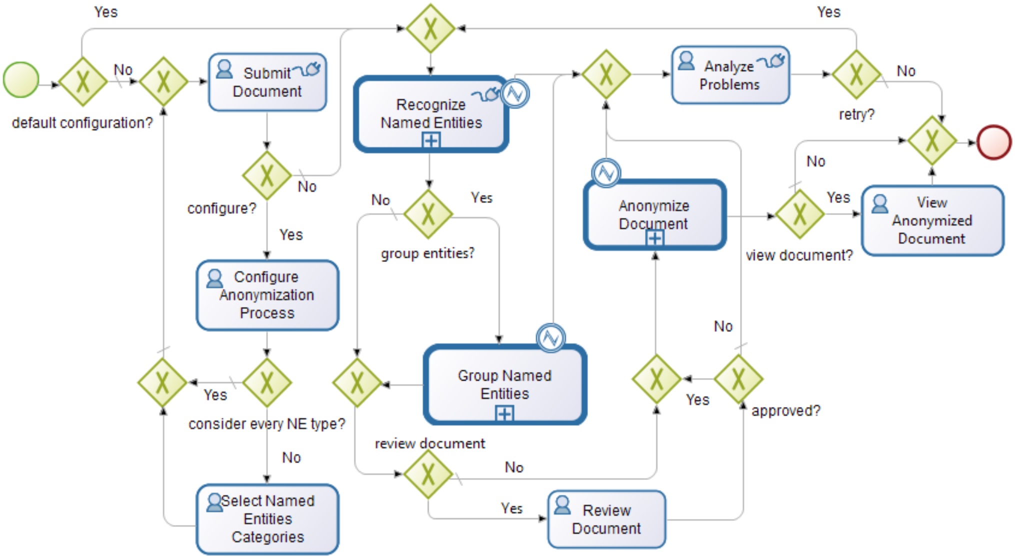

Fig. 3. Anonymization process

in Table 2. These requirements were taken as drivers for the design and description of the reference architecture.

4.1 Functional View

As the only human actor, the anonymization system interacts with a domain expert, which will be the responsible of provide feedback as well as approve or reject the processed documents. Although the goal is that the system can be a semi-automatic anonymization service that can also be used by other systems in the organization, there may be end users who want to select documents and see them anonymized.

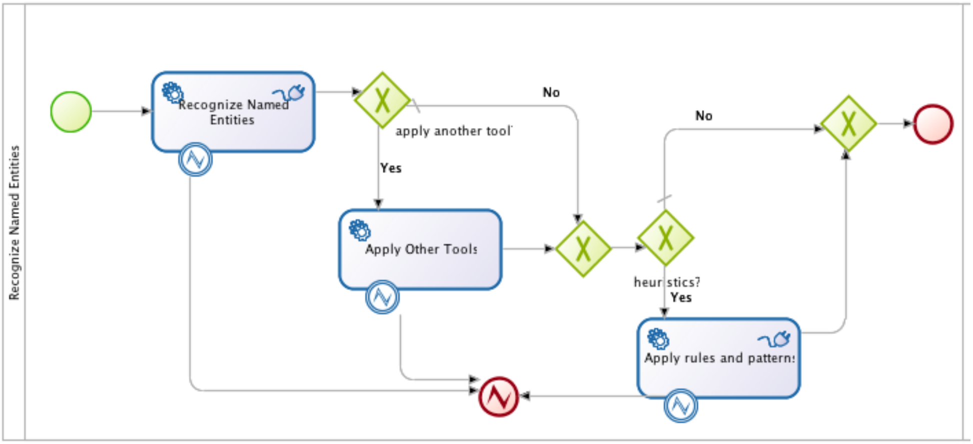

Fig. 4. “Recognize Named Entities” subprocess

This is done through the construction of a system over the anonymization one, as discussed in Section 5. Finally, the anonymization system has interfaces to other systems, typically document management systems through which it receives and stores the documents before and after the anonymization process is performed.

From a functional point of view, the anonymization process can be seen as a business process, i.e. a set of activities that are performed in coordination in an organizational and technical environment, to realize a business goal [23]. This is a configurable domain and platform independent process which can be described using BPMN notation [11] as shown in Figure 3. This notation simplifies the involvement of business experts who can visualize how the anonymization system works with a friendly notation. Furthermore, this description can be executed using a Business Process Management System (BPMS). As an added value, we gain access to basic security, auditing and interoperability infrastructure that is already provided by any BPMS, aspects that were identified as important requirements to our architecture.

The anonymization process can be automatically performed based on a default configuration (Configure Anonymization Process task) that only involves the selection of the document to be anonymized, or can be configure to allows revision by an expert. It also allows to select the categories of items to be anonymized (Select Named Entity Categories task). The automatic process is useful to provide real-time information to users outside the organization, for example through the web, but it requires high reliability levels in the results.

Once the document is selected (Submit Document task), the process recognizes the named entities of interest (Recognize Name Entities subprocess in Figure 4) using different NER tools, external tool as thesauri and newsletters, rules and patterns. The recognized entities are classified and grouped (Group Named Entities subprocess in Figure 5) using a predetermined algorithm. The goal is that the same entity can be homogeneously identified, e.g. “United Nations” and “UN”.

After the recognition and grouping is done, a domain expert can review the labeled document, approve or reject it (Review Document task) and provide feedback to the anonymization process based on the errors found. If the document

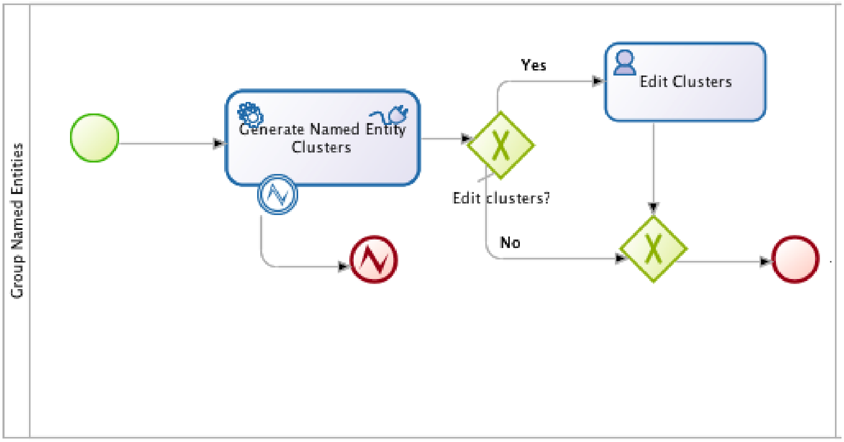

Fig. 5. “Group Named Entities” subprocess

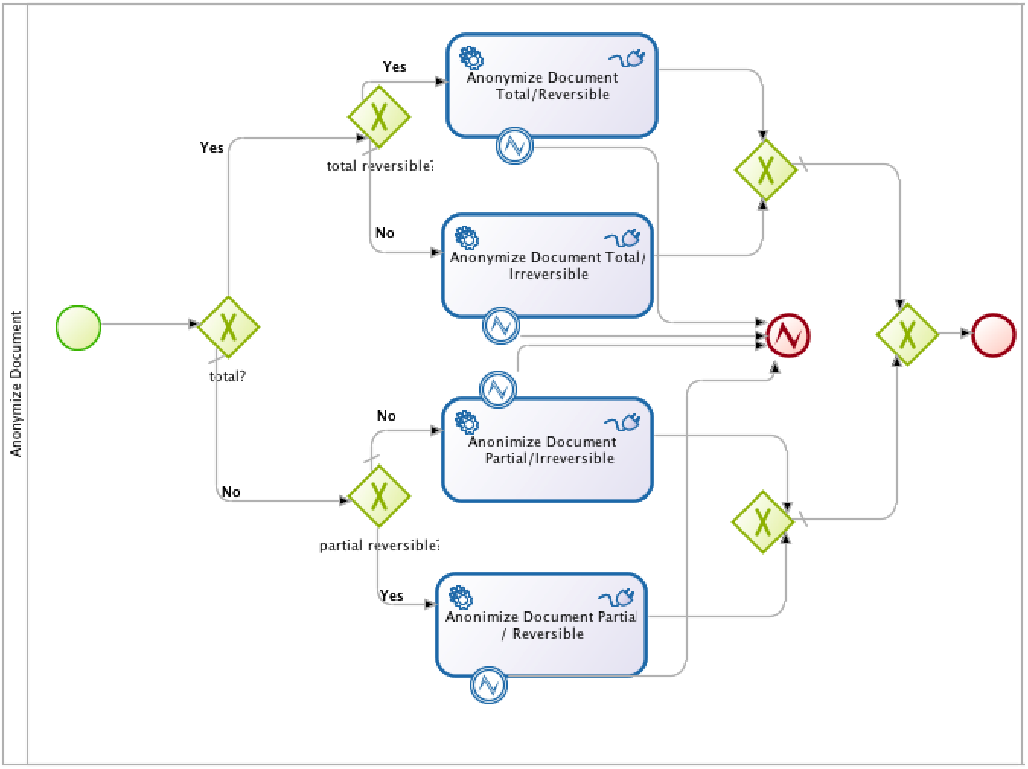
Anonymize Document
Yes
Anonymize Document Total/Reversible
Yes
total reversible?
No
Anonymize Document Total/ Irreversible
total?
No
Anonimize Document Partial/Irreversible
No
partial reversible?
Yes
Anonimize Document Partial / Reversible

Fig. 6. “Anonymize Document” subprocess

is approved, the real anonymization is done (Anonymize Document subprocess in Figure 6). In this step, some of the anonymization kinds is done: partial or total (depending on whether some or all types of named entities are anonymized, respectively), reversible or irreversible (depending on whether the original document can be reversed or not, respectively).

Finally, it is possible to see the anonymized document (View Anonymized Doc-

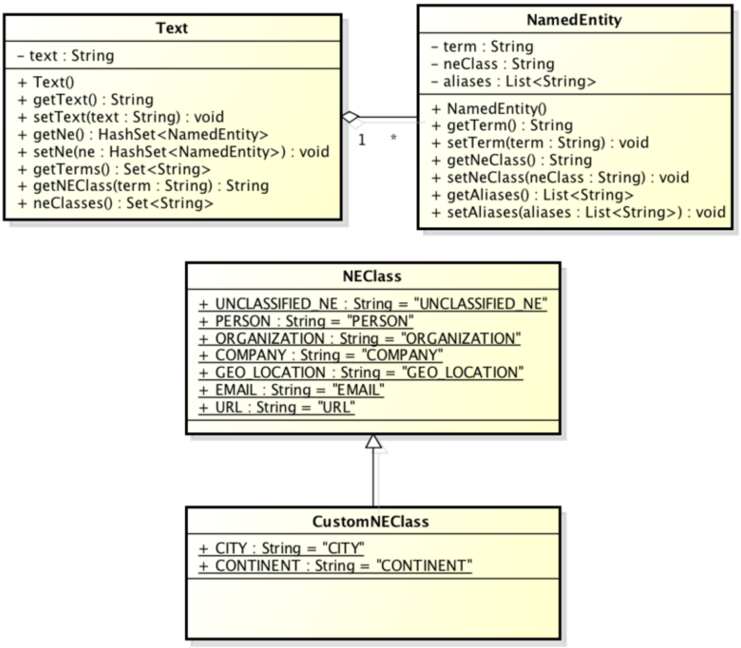
Text
- text : String
+ Text()
+ getText() : String
+ setText(text : String) : void
+ getNe() : HashSet<NamedEntity>
+ setNe(ne : HashSet<NamedEntity>) : void
+ getTerms() : Set<String>
+ getNEClass(term : String) : String
+ neClasses() : Set<String>
1
*
NamedEntity
- term : String
- neClass : String
- aliases : List<String>
+ NamedEntity()
+ getTerm() : String
+ setTerm(term : String) : void
+ getNeClass() : String
+ setNeClass(neClass : String) : void
+ getAliases() : List<String>
+ setAliases(aliases : List<String>) : void
NEClass
+ UNCLASSIFIED_NE : String = "UNCLASSIFIED_NE"
+ PERSON : String = "PERSON"
+ ORGANIZATION : String = "ORGANIZATION"
+ COMPANY : String = "COMPANY"
+ GEO_LOCATION : String = "GEO_LOCATION"
+ EMAIL : String = "EMAIL"
+ URL : String = "URL"
CustomNEClass
+ CITY : String = "CITY"
+ CONTINENT : String = "CONTINENT"

Fig. 7. Data model

ument task). If the previous steps require the use of other tools and the connection (e.g. in the case of web services) or processing fails, the problem can be analyzed (Analyze Problems task) to evaluate if it is necessary or not to retry the anonymization process.

4.2 Information View

The unstructured text flows through the process described in the functional view. For this reason the document is not specified directly on the BPMN model. We have defined a simple data structure that is depicted in Figure 7 for representing the unstructured text. It consists of an class `Text` which contain the string (`String`) representing the document in its current state (anonymized or not), and a structure containing the set of named entities the have been identified. A named entity is represented as a class `NamedEntity` with three attributes:

- **term**: string that contains the term of interest, e.g. `John Doe`
- **neClass**: string that contains the classification of the named entity, e.g. `ORGANIZATION`, `GEO_LOCATION`, or `PERSON`
- **aliases**: list of equivalent terms to the word of interest, when clusters are defined

4.3 Development View

As described before, there are several technological alternatives for implementing the most important steps of the process.

TreeTagger [21]. It is a free tool that uses inductive techniques to tag text in several languages after a training process. Several configuration files are provided with the training results of different languages, including Spanish. The tagger identifies nouns, numbers, verbs, alphanumeric codes, and dozens of other linguistic forms. The distribution also includes adapters for different platforms and a graphical configuration utility.

FreeLing [15]. It is a tools suite for natural language processing. Several configuration files are provided with the training results of different languages, including Spanish. It provides some features for the detection of sentences, numbers and dates, morphological analysis, detection of multi-word phrases, recognition and classification of named entities. It is distributed as both a C++ library and a command line utility.

Apache OpenNLP [7]. It is a tools suite for natural language processing based on machine learning. Like most of the tools based on inductive techniques, it requires training and several configuration files are already provided. Some of the tools it provides are for the detection of sentences and named entities.

OpenCalais [17]. It is a semantic processor of unstructured documents developed by Thomson Reuters corporation, allowing, among other things, to identify and classify named entities. Free access is provided for both personal and commercial use through web services. It has some limitation in terms of a maximum number of queries per day allowed for each registered user. In some tests performed with this utility, we observed very good effectiveness in the identification of named entities, which coincides with the results of the study conducted for NER systems in [13].

LingPipe [1]. It is a commercial framework that can be integrated into other applications, providing a wide range of services such as grammatical tagging, recognition of named entities, clustering, spell checking, among others. LingPipe can be used free of charge for academic purposes.

We can notice that the tools we have evaluated have both the possibility of identify and classify named entities. This allows, for example, determine whether a named entity is of type “person”, or whether it is a geographical location. Some tools, such as OpenCalais are quite broad, classifying entities with great detail. Others are more rudimentary or limited, merely identifying names of people and places. The architecture allows to define the classification criteria for each tool when defining its correspondent adapter, just listing the types of named entities that will be classified by the tool and then mapped to the types defined within the adapter.

We have evidenced that the tools have different effectiveness levels, particularly when using the models available for Spanish (we have used the generic models trained on different corpus, not specific ones). The tools mostly used statistic models for identifying named entities, so the results are not deterministic, but there is a certain margin of error that is reflected as false positives (terms that are identified as an entity and are not) and false negatives (terms that are not identified as an entity when they should be).

Given this context, we developed a mechanism to use several of these tools together: the MultiNER adapter. This adapter allows to use several NER tools in

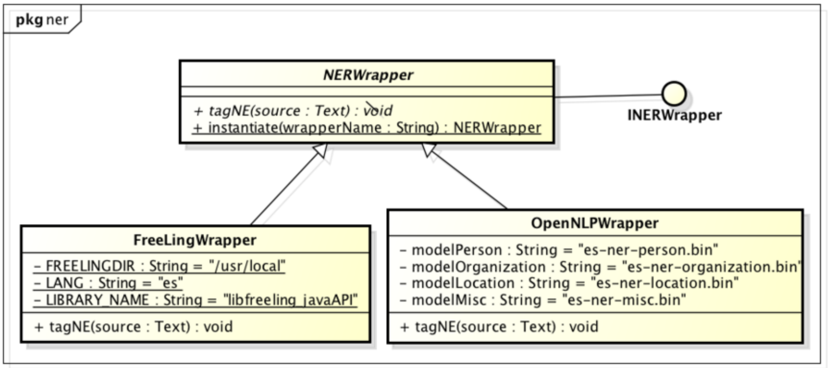
pkg ner
NERWrapper
+ tagNE(source : Text) : void
+ instantiate(wrapperName : String) : NERWrapper
INERWrapper
FreeLingWrapper
- FREELINGDIR : String = "/usr/local"
- LANG : String = "es"
- LIBRARY_NAME : String = "libfreeling_javaAPI"
+ tagNE(source : Text) : void
OpenNLPWrapper
- modelPerson : String = "es-ner-person.bin"
- modelOrganization : String = "es-ner-organization.bin"
- modelLocation : String = "es-ner-location.bin"
- modelMisc : String = "es-ner-misc.bin"
+ tagNE(source : Text) : void

Fig. 8. External tools adapter

parallel and records how many tools identify a particular entity. Depending on this value, the MultiNER defines a threshold above which an entity is considered valid, so it is possible to reduce "false positives". However, we need further evaluation for the definition of a correct threshold.

Considering the different technological alternatives, the architecture is benefited from the existence of a generic interaction mechanism, so that it is possible to choose between alternatives simply by modifying the initial configuration step. This mechanism also provides extensibility for the inclusion of new tools or the combination of the existent ones.

For this purpose we define a generic adapter that allows to encapsulate the characteristics of each tool, providing a unified interface to the anonymization system. This solution is depicted in Figure 8. The complexity and specific aspects of a certain tool are encapsulated and implemented by adapter classes (e.g. `FreeLingWrapper`), from which the external tool specific API is called. These adapters implement a generic interface `INERWrapper` providing a method for the recognition of named entities. In addition it is proposed that each adapter classes inherits from a generic adapter `NERWRapper`, which provides common methods, e.g. `instantiate` which is intended to initialize (using introspection/reflection) the corresponding adapters so that they can be exchanged dynamically at runtime.

Finally, we need to configure the tools to be used, and even enable or disable certain specific steps of the anonymization process according to user requirements. Therefore, the architecture can be dynamically adapted by editing XML configuration files. Figure 9 partially shows the content of a configuration file where multiple NER tools are defined, their versions and corresponding adapters.

5 Case Study: Anonymization of Legal Documents

Sometimes, citizens and organizations access to jurisprudence, i.e. judicial sentences defined by magistrates (judges and ministers). In some countries, these cases are binding precedents which are strongly considered in future court actions. In Uruguay's case it is an important source of knowledge and support for study by lawyers and judges.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Tools>
  <NERTools>
    <tool>
      <name>FreeLing</name>
      <version>3.0</version>
      <wrapper>org.myorg.wrapperF</wrapper>
    </tool>
    <tool>
      <name>LingPipe</name>
      <version>4.1.0</version>
      <wrapper>org.myorg.wrapperL</wrapper>
    </tool>
  </NERTools>
</Tools>
```

Fig. 9. Configuration file

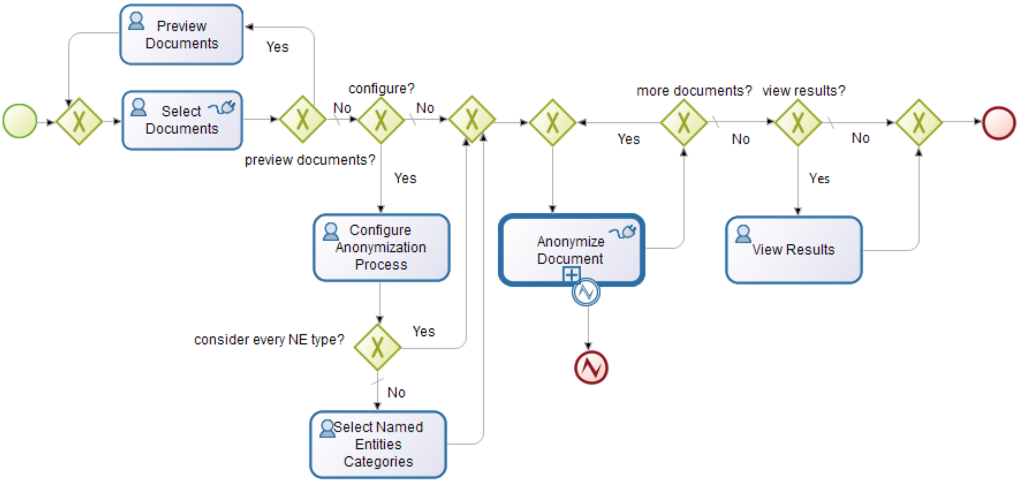

Fig. 10. “DEMO Application” process

The Judiciary of Uruguay publish jurisprudence information of the Courts of Appeals and the Supreme Court to citizens through the BJN system (Base de Jurisprudencia Nacional [4]). In such system, the identity of organizations and individuals is protected when it comes to certain types of legal proceedings. Such is the case for example of the identity of minors in proceedings related to family. Anonymization techniques are used to guarantee confidentiality. However, the process is manually performed by court officials who read each of the statements, identify sensitive data (named entities) to be anonymized, and finally replace such information with generic terms.

Based on the reference architecture defined in Section 4, we have implemented an anonymization system for legal documents. The process was executed using Bonita Open Solution [2] in his open source community edition. Beyond the implementation of the prototype, we implement a second business process that uses the anonymization process, which is shown in Figure 10.

This process is used to allow access to the BJN, select the documents to be anonymized, configure the anonymization process, process the documents with respect the given configuration and then review the results and store the anonymized documents in the BJN database. The main users of this system are the officials of the Justice Department, who are responsible for carrying out the anonymization process, so the sentences can be released to the general public, and even the citizens access them through a web system. This last aspect requires high levels of reliability of the anonymization process.

For the integration of the different tools used during the anonymization process, we developed a Java library which is invoked from the process engine using Groovy connectors. We have integrated many tools for the recognition of named entities. For each one of these tools we have implemented an adapter following the pattern presented in the last section, so it is possible to integrate libraries, console applications and web services. In this way, the use of one or another tool is transparent to the user. Figure 11 shows a deployment diagram of the prototype which illustrates the technological ecosystem involved in our solution.

- Demo Server: server in which the Demo Application is running.
- BOS Runtime 5.9: Bonita Open Solution process engine.
- Anonymization Process: anonymization process running within BOS.
- Anonymization_fat.jar: JAR archive containing libraries and dependencies for the Demo Application: Java based applications as OpenNLP [7] and LingPipe [1], as well as APIs to interact with external tools as TreeTagger [21], FreeLing [15] and OpenCalais [17].
- TreeTagger: local installation of TreeTagger.
- FreeLing: local installation of FreeLing.
- Web Browser: web browser used to access BOS user interface.
- Database Server: server containing the court decisions (MySQL 5.5).

In the “Group Named Entities” subprocess we use two different tools to group named entities into clusters. The first tool is an ad hoc component that groups entities following rules and patterns that identify equivalences between them. To cite an example, there is a rule grouping extended names of organizations with its corresponding acronym. Additionally, we also provide the alternative of using LingPipe, which is connected through an adapter analogous to the NER tools.

For the purpose of testing the prototype, we take a set of judicial sentences which are publicly available in the BJN system (not anonymized) and we execute the automatic anonymization process. We got mixed results. In some cases the system had good levels of recognition of named entities and sensible data, and in others we found false positives. One of the sources of error is that within these sentences it is common to use capitalized terms that are not necessarily named entities. These could be listed and excluded from the anonymization process using rules and patterns. In addition, sentences contain many references to literature or authors related to the law, which can be erroneously identified as sensitive data, affecting the semantics of the document to be anonymized. It is also common to find Latin terminology, which confuses the tools when they are trained over a Spanish corpus. These problems can be solved with a better training of the tools we use. We can also include a validation step taken by an expert user (already supported

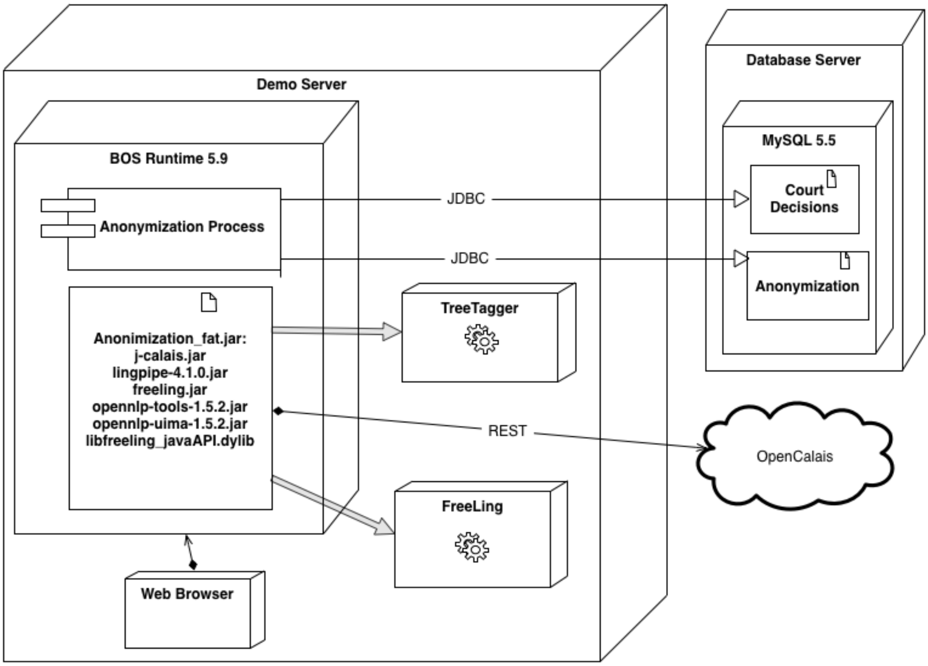

Fig. 11. Deployment view of the prototype

by the proposed architecture) to give some feedback. Finally, we got better results using the MultiNER adapter than using separate tools. However, we need a more qualitative assessment of the results for stronger conclusions.

6 Conclusions and Future Work

Document management introduces significant challenges to organizations, which are strengthened by the existence of specific laws punishing unauthorized information disclosure of individuals or legal entities. In this context, document anonymization is a process of great interest to organizations because it allows to comply with the law while allowing valuable information to be available to other organizations and individuals.

In this paper we have addressed the problem of anonymizing unstructured documents from a technological perspective, introducing the software architecture of a generic anonymization system, which can be used as a reference for stakeholders when developing similar kind of systems. This proposal is based on previous architectures, fully adopting their ideas and overcoming their limitations. In particular, the anonymization problem was described as a business process, modeled using BPMN, which provides a high level description of the process as well as enough flexibility to be easily adapted to different usage scenarios. It also allows the use of a BPMS for process execution, providing basic security and auditing infrastructure. The architecture also provides extensibility in terms of the different tools required during the process. The implementation also combines different techniques for the recognition of named entities through the MultiNER adapter.

Besides that the process considers user's feedback to the system to improve the identification of sensitive data using rules and patterns, feedback could be directly provided by the statistical models used by the natural language processing tools. Entities recognition and feedback phases could benefit from using a morphological tagger to categorize words within the documents. In this way, false positives could be reduced, because if a word is classified in a certain category (e.g. if it is recognized as a verb), it can be discharged as a possible entity with greater certainty. Moreover, we can integrate a spell checker to analyze those words that cannot be classified in any category, improving effectiveness. A further qualitative analysis is required with respect to the results of the MultiNER adapter to determine its adequacy. Although the proposal considered unstructured documents, it would be desirable to extend the ideas to structured documents such as the XML-based standards used in the biomedical domain, and PDF documents for which it is surely required the integration of some specific text recognition technology.

We have shown how this kind of systems can be developed with reasonable costs, since there are available tools to perform various tasks of the anonymization process, such as the recognition of named entities and clustering. The prototype was implemented using open-source technologies and it is focused on the anonymization of jurisprudence of the Judiciary of Uruguay. We observed that it is feasible to apply a semi-automatic process and that it speeds up the traditional anonymization process. However, court decisions have peculiarities for which the recognition of named entities based on models trained using generic corpus, do not provide good results. In this sense, it is interesting to explore the use of a specific corpus to get better results in this application domain and language. Furthermore, it is possible to optimize the process by identifying rules and patterns that apply to this kind of documents. For example, a domain expert could identify terms to exclude from entities recognition. This kind of custom processing is already supported by the reference architecture.

References

[1] Alias-I, *LingPipe* (2003).
URL http://alias-i.com/lingpipe/

[2] Bonitasoft, *Bonita Open Solution v6.2, community edition* (2013).
URL http://es.bonitasoft.com/

[3] Estados Unidos de América, *Health insurance portability and accountability act* (1996).
URL http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf

[4] República Oriental del Uruguay, Poder Judicial, *Base de jurisprudencia nacional pública* (2011).
URL http://bjn.poderjudicial.gub.uy/

[5] República Oriental del Uruguay, Poder Legislativo, *Ley n 18.331: Protección de datos personales y acción de "habeas data"* (2008).
URL http://www.parlamento.gub.uy/leyes/ley18331.htm

[6] España, "Ley 14/2007, de 3 de julio, de investigación biomédica," Colección Textos legales, Ministerio de Sanidad y Consumo, 2007.

[7] Apache Software Foundation, *Apache OpenNLP* (2000).
URL http://opennlp.apache.org

[8] García, L., "Un etiquetador morfológico para el espaol de Cuba," Master's thesis, Universidad de Oriente - Santiago de Cuba - Facultad de Matemática y Computación (2008).

[9] Gardner, J. and L. Xiong, *Hide: An integrated system for health information de-identification*, in: *Proc. Twenty-First IEEE Intl. Symposium on Computer-Based Medical Systems* (2008), pp. 254–259.

[10] Gardner, J. and L. Xiong, *An integrated framework for de-identifying unstructured medical data*, Data Knowl. Eng. **68** (2009), pp. 1441–1451.

[11] Object Management Group, *Business Process Model and Notation (BPMN) version 2.0* (2011). URL http://www.omg.org/spec/BPMN/2.0/

[12] Iglesias, A., E. Castro, R. Pérez, L. Castao, P. Martínez, J. Gómez-Pérez, S. Kohler and R. Melero, *Mostas: Un etiquetador morfo-semántico, anonimizador y corrector de historiales clínicos*, Procesamiento de Lenguaje Natural **41** (2008).

[13] Marrero, M., S. Sánchez-Cuadrado, J. Lara and G. Andreadakis, *Evaluation of named entity extraction systems*, Advances in Computational Linguistics. Research in Computing Science **41** (2009), pp. 47–58.

[14] Marrero, M., J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. Gómez-Berbís, *Named Entity Recognition: Fallacies, challenges and opportunities*, Computer Standards & Interfaces **35** (2013).

[15] Padró, L., *FreeLing* (2003). URL http://nlp.lsi.upc.edu/freeling/

[16] Pérez-Laínez, R., C. Pablo-Sánchez and A. Iglesias, "ANONIMYTEXT : Anonymization of unstructured documents," SCITEPRESS Digital Library, 2008 pp. 284–287.

[17] Thomson Reuters, *OpenCalais* (2008). URL http://www.opencalais.com

[18] Rodríguez, J., "Sistema de Clustering de Named Entities," Master's thesis, Universidad Politécnica de Catalua (2008).

[19] Rozanski, N. and E. Woods, "Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives," Addison-Wesley Professional, 2005.

[20] Russo, P., "Gestión Documental en las Organizaciones," UOC (Universitat Oberta de Catalunya), 2009.

[21] Schmid, H., *TreeTagger* (1994). URL http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[22] Vico, H., "Definición de una Arquitectura de Referencia para Anonimizar Documentos," Master's thesis, Universidad de la República, Uruguay (2013). URL https://www.fing.edu.uy/node/9166

[23] Weske, M., "Business Process Management - Concepts, Languages, Architectures, 2nd Edition," Springer, 2012, I-XV, 1-403 pp.