Towards Ontology-based E-mail Fraud Detection

Koen Kerremans¹, Yan Tang², Rita Temmerman¹, Gang Zhao²
¹CVC, Erasmushogeschool Brussel, Belgium
{koen.kerremans, rita.temmerman}@ehb.be
²STARLab, Computer Science, Vrije Universiteit Brussel, Belgium
{yan.tang, gang.zhao}@vub.ac.be

Abstract. The European FF POIROT project (IST-2001-38248) aims at developing applications for tackling financial fraud, using formal ontological repositories as well as multilingual terminological resources. In this article, we want to focus on the development cycle towards an application recognizing several types of e-mail fraud, such as phishing, Nigerian advance fee fraud and lottery scam. The development cycle covers four tracks of development – language engineering, terminology engineering, knowledge engineering and system engineering. These development tracks are preceded by a problem determination phase and followed by a deployment phase. Each development track is supported by a methodology. All methodologies and phases in the development cycle will be discussed in detail.

1 Introduction

The European FF POIROT project (IST-2001-38248) aims at developing applications for tackling financial fraud, using formal ontological repositories as well as multilingual terminological resources. In this article, we focus on the development cycle towards an application recognizing several types of e-mail fraud. The development cycle, shown in figure 1, covers four tracks of development – language, terminology, knowledge and system engineering. These development tracks are preceded by a problem determination phase and followed by a deployment phase. Each development track is supported by its own methodology. All methodologies and phases in the development cycle will be discussed in the following sections. Section 2 will be devoted to a brief discussion of the problem of e-mail fraud. In section 3, we discuss the development tracks of respectively system engineering (section 3.1), terminology engineering (section 3.2), knowledge engineering (section 3.3) and language engineering (section 3.4). Section 4 covers the deployment phase and solution. In section 5, we conclude.

Note that in the development cycle, some interaction is established between the development tracks of terminology, knowledge and language engineering (see figure 1). For instance, the dotted arrow going from ‘Termbase’ to ‘Ontology development’ indicates that the results of the terminological analysis are used in the development phase of the ontology. This will be further explained in section 3.2. The dotted arrow which links the resulting ontology to the search phase in the language engineering track suggests that the content of the ontology (i.e. the concepts and relations) is used
to look up typical natural language patterns (i.e. words and/or expressions) that are identified as potential fraud indicators in e-mail messages. We will discuss this issue in sections 2 and 3.3.

Fig. 1. Overview of the development tracks in FF POIROT

2 Problem & Knowledge Space – Problem Determination

On-line fraud is one of the fastest growing forms of scam. Most often, it is spread through unsolicited e-mails (i.e. spam) as this is inexpensive and allows fraudsters to reach in a short period of time a large number of potential victims. Phishing is a type of on-line fraud in which unsuspecting web surfers end up at an illegally duplicated website of a legitimate bank and are bereft of confidential personal information (such as credit card numbers). According to the Anti-Phishing Working Group, an industry association focused on eliminating the identity theft and fraud that result from the growing problem of phishing and email spoofing, the number of ‘phishing-incidents’ keeps on growing. In the month of March 2005, 2870 active phishing websites were reported to the organisation and the average monthly growth rate of websites between July 2004 and March 2005 was 28%.

Phishing messages are sent via spam e-mails. They are well-structured and give addressees the impression that the e-mail was sent by their own banks or internet providers. A typical scenario outlined in a phishing message is a notification of the bank, stating that due to a technical malfunctioning of the bank’s internal system, due to security reasons or specific problems with credit card numbers, … the addressee is asked to click on a hyperlink directing them to the bank’s website (which is actually the fraudulent duplicate) and to register again (by specifying confidential information such as credit card numbers, PIN-codes or even passwords for on-line services).

Another form of e-mail fraud is Nigerian Advance Fee Fraud – also known as the 419 scam – a famous worldwide fraud. In a Nigerian fraud scheme, an addressee receives an unsolicited e-mail, fax, or letter containing a (confidential) business proposal and is promised a financial compensation for his assistance in transferring a huge sum of money to an overseas bank account. Once targeted victims respond to the proposal, they are lured into paying an advance fee to enable the business transaction.2

The lottery scam is another well-known form of advance fee fraud circulating throughout the world, affecting a large number of e-mail users. In this fraud scheme, an addressee receives an unsolicited e-mail claiming that he is the lucky winner of a lottery win. When responding to the fraud message, the addressee receives an application asking for his bank account information or requiring a very sizeable application fee payment. The fraudster obviously disappears once he receives the money.

Despite the fact that scams such as the advance fee fraud cases are known worldwide, a lot of people still fall into the trap. In the problem determination phase (see figure 1), we therefore specify the need for an ontology-based application which, unlike an ordinary spam filter, is able to understand the content of e-mail messages. Based on a series of fraud indicators, the application will be able to recognize the forms of fraud mentioned above (i.e. ‘phishing’, ‘Nigerian fraud’ and ‘lottery scam’) and will notify the addressee that a certain mail may be fraudulent, while leaving the user to decide whether or not to respond to the mail. More details follow in sections 3.1 and 4.

The analysis of the problem and knowledge space during the problem determination phase results, on the one hand, in a list of system requirements which are consulted during system engineering and, on the other hand, in a knowledge scope which determines the terminology, knowledge and language engineering development tracks. These two issues will be further discussed in section 3.

3 Development Tracks

This section deals with the four development tracks that follow the problem determination phase. Section 3.1 covers the system engineering track, section 3.2 the terminology engineering track, section 3.3 the knowledge engineering track and, finally, section 3.4, the language engineering track.

2 See: http://www1.ifccfbi.gov/strategy/nls.asp
3.1 System Engineering

The aim of this development track is to design and develop the knowledge processor by looking into its functional aspects derived from questions like how the knowledge system is going to be implemented and how the knowledge developed in the knowledge engineering track is going to be deployed. Figure 1 shows the main tasks in knowledge system development. The system requirements phase pertains to the performance requirements of the system, the analysis of the user perspective of the system behavior as well as the functionality and system perspective of the processing tasks. System design seeks to identify the software components of knowledge processors and their interfaces. The system development stage builds and integrates components of knowledge processors and runtime knowledge repositories. It also takes the output of the knowledge deployment stage along the knowledge engineering track and makes it available to the knowledge processor via dedicated knowledge repositories. Finally, deployment refers to the evolution from Alpha to Beta testing of the intelligence and behavior of the knowledge system.

In the context of this article, the system requirements phase specifies an intelligent agent that takes e-mail content in French and English as input and passes it through a user-defined language model of regular expressions used as indicators for recognizing phishing scams, Nigerian fraud letters and lottery scam e-mails (section 3.4). The application should be able to recognize the type of fraud and should present its conclusion and results to the user. In order to succeed, an ontological repository of fraud indicators is created and used as a stable classification model for the language-depending, constantly changing model of regular expressions.

3.2 Terminology Engineering

The FF POIROT terminological resource incorporates, amongst others, detailed definitions of fraud terminology as well as rich semantic annotations. On the one hand, this resource is designed to support the ontology modeling process, which has led in FF POIROT to the development of a methodology called Termontography (section 3.2.1). On the other hand, parts of the terminological information – such as the fraud terminology and definitions – are integrated into the knowledge-based systems in order to explain, for instance, to the user in natural language, the reasoning processes for detecting and intercepting fraud. The use of terminological information with respect to the e-mail fraud detection application, will be discussed in section 3.2.2.

3.2.1 Termontography

Termontography combines theories and methods of the sociocognitive analysis [1] with methods in ontology engineering. The motivation for combining termontography and ontology development derives from the view that existing methodologies in terminology compilation [2], [3], [1] and (text-based, application- and/or task-driven) ontology development have important commonalities [4].

As is shown in figure 1, six methodological phases are identified in the Termontography approach: analysis, information gathering, search, refinement, verification and
validation [4]. The resource resulting from this workflow is an ontologically-structured terminological knowledge repository.

A fundamental issue in Termontography is that terminographers need a solid reference framework to scope their terminology work. Scoping implies determining which linguistic words/patterns are considered relevant terms given the applications, users and purposes of the terminological resource. This insight was also at the basis of the OncoTerm project, in which the scope of the multilingual terminological database was determined by the knowledge specified on the conceptual level [5]. In Termontology, requirements for terminology bases are therefore first translated into language-independent frameworks of interrelated categories which can provide, depending on their granularity level, detailed information with respect to the extraction of terms and knowledge rich contexts [6] from a domain-specific corpus of texts [7].

3.2.2 Terminology Engineering in E-mail Fraud Detection

For lack of space this section will only briefly discuss the most fundamental steps in the terminology engineering process supporting the ontological repository as well as the e-mail fraud detection application.

A first important step in this development process is the determination of the knowledge scope. Given the problem and knowledge space outlined in section 2, the aim of the terminological resource is to provide, in natural language, definitions and terms related to the fraud types of ‘phishing’, ‘Nigerian fraud’ and ‘lottery scam’. Apart from that information, also explanations of what a user should do in case he received one of these e-mails will be added to the terminological resource as this information will be shown to the receiver by the knowledge based system (section 4).

All the terminological information described in the previous paragraph is manually derived from textual material collected from websites on e-mail fraud. Apart from the terminology, all definitions and other relevant information are added to the terminological resource, including the references to their sources. From the definitions, important term-relation-term patterns are manually extracted or highlighted in order to support the abstraction towards a first conceptual model in the knowledge engineering track (section 3.3). This is shown in table 1 for the term ‘Nigerian advance fee fraud’.

<table>
<thead>
<tr>
<th>English Term</th>
<th>Nigerian advance fee fraud</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>the target receives an unsolicited fax, email, or letter often concerning Nigeria or another African nation containing either a money laundering or other illegal proposal OR you may receive a Legal and Legitimate business proposal by normal means. […]</td>
</tr>
<tr>
<td>Reference</td>
<td><a href="http://home.rica.net/alphae/419coal">http://home.rica.net/alphae/419coal</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analysis</th>
<th>target Receives unsolicited fax, email or letter Contains email or letter proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unsolicited fax, Contains email or letter proposal</td>
</tr>
</tbody>
</table>

Table 1. Example of a terminological record
3.3 Knowledge Engineering

The knowledge engineering track aims at developing an application specific ontology of fraud forensics, used for recognizing the forms of e-mail fraud outlined in section 2. In the framework of the FF POIROT project, the Application Knowledge Engineering Methodology (henceforward: AKEM) has been worked out for developing application-oriented ontologies. This methodology will be explained in section 3.3.1. Section 3.3.2 is devoted to the discussion of the application ontology of fraud forensics.

3.3.1 Application Knowledge Engineering Methodology

AKEM is based on the DOGMA ontology representation framework which defines an ontology as a set of lexons and their commitments in particular applications. Lexons represent relationship types between two object types and are constructed as follows: <Context, Term1, Role1, Term2, Role2>. They capture the underlying concepts and relationships while commitments link them to a particular application or task requirement with specific constraints and instantiations [8], [9], [10].

The AKEM methodology organizes a geographically distributed, multidisciplinary team of domain experts, knowledge analysts and engineers in a methodical traceable development cycle – shown in figure 1 – of knowledge scoping, knowledge analysis, ontology development and deployment, similar to RUP [11]. In AKEM, knowledge scoping identifies that part of the universe of discourse, used for ontology modeling and development. Stories result from the knowledge scoping activity and are used to convey business cases and scenarios (see figure 1). From these stories, a constituent model is created during knowledge analysis to describe how the application semantics are decomposed and how each constituent is elaborated in the description of business logic. The ontology development activity refers to the process of creating ontologies to capture the meta knowledge. During deployment, certain concepts and relationships will be selected and constrained to form commitments or networks of application semantics in view of specific processing tasks [12], [13].

3.3.2 Knowledge Engineering in E-mail Fraud Detection

This section discusses the most important steps in the application-oriented knowledge engineering process, following the AKEM methodology. Each step will be illustrated by means of an example taken from Nigerian fraud. Note that the same methodology has been applied for constructing conceptual models for phishing and lottery scam.

The knowledge scope in AKEM is specified and documented by means of a story. In the context of e-mail fraud detection, a story provides a structured presentation of the information that one encounters in a typical fraud case. Figure 2 shows parts of the story derived from a typical Nigerian fraud case.
Fig. 2. Semantic scoping with the AKEM story editor

From the story, knowledge is decomposed and elaborated in the knowledge analysis phase (figure 1). During knowledge decomposition, knowledge constituents, derived from the story, are structured hierarchically in a four-layered Wigmore chart [14]. The top layer refers to the hypothesis of the existence of fraud. The second layer consists of supporting postulates. The third layer specifies evidences in the postulates, whereas the fourth layer provides the facts that express or embody the evidence. Knowledge elaboration refers to the process in which each knowledge constituent is translated into a controlled language of logical statements (figure 3).

```
IF Fact (1.1.1.1) the addressee surprised by the email
OR IF Fact (1.1.1.2) the addressee is unknown addressor
OR IF Fact (1.1.1.3) the addressor introduce himself or herself
OR IF Fact (1.1.1.4) the addressor impersonal reference to addresser
OR IF Fact (1.1.1.5) the addressor describe the acquirement of addresses
THEN Evidence (1.1.1) the addressor send unsolicited email to addressee
IF Evidence (1.1.1) the addressor send unsolicited email to addressee
THEN Postulate (1.1) email is unsolicited as the medium of the fraud scheme
```

Fig. 3. An example of the knowledge elaboration.

The ontology development phase follows the knowledge analysis phase (figure 1). It seeks to define the concepts and relationships underlying the knowledge elabora-
tion. It is here that also the results of the terminology engineering process are used to extract, abstract and organize concepts and relations into an ontological repository (figure 4).

Knowledge Unit

<table>
<thead>
<tr>
<th>Subject</th>
<th>Reference</th>
<th>Ontology Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>FFO-1-3</td>
<td>Term</td>
</tr>
<tr>
<td>IF (1.1) the addresser is added to that in or her OR IF (1.2) the addressee is added to provide current address THEN (1.3) A proposal is made to initiate the capital AND IF the proposition is such or intended for the individual has more than 25% or suggestion for maintenance THEN (1.7) the proposal is intended to appear actually known [ \text{ if the addresser knows the person of the proposal} ]</td>
<td>Address</td>
<td>InCapacity</td>
</tr>
</tbody>
</table>

Fig. 4. Extraction and abstraction of lexons

Abstraction refers to the process in which highlighted keywords denoting objects and relationships are formalized into lexons. The purpose of this process is only to model those concepts and relationships which are explicitly verbalized in the knowledge elaboration deliverable. The ‘Ontology Specification’ in figure 4 shows parts of the lexon base created during abstraction.

3.4 Language Engineering

The fraud ontology resulting from the knowledge engineering track is used to support language engineering in the overall methodology (figure 1). The purpose of the language engineering track is to create a model of common linguistic patterns (words and/or expressions) which are used by fraudsters to create fraudulent e-mails. In contrast to terminology engineering, in which only texts are analyzed that provide information about the fraud types, the language engineering track is solely focused on the actual fraud e-mails and is only concerned with finding the linguistic structures that express fraud evidences in texts. The methodological steps of knowledge scoping, information gathering, search, verification and validation are the same as in the terminology engineering process.

Table 2. List of some fraud indicators for Nigerian fraud, phishing and lottery scam

<table>
<thead>
<tr>
<th>Nigerian fraud</th>
<th>Phishing</th>
<th>Lottery scam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsolicited e-mail</td>
<td>Unsolicited e-mail</td>
<td>Unsolicited e-mail</td>
</tr>
<tr>
<td>Solicitation</td>
<td>UrgentResponse</td>
<td>WonLottery</td>
</tr>
<tr>
<td>Dead person</td>
<td>SeekPrivateInformation</td>
<td>RandomSearchEmail</td>
</tr>
<tr>
<td>Capital</td>
<td>HyperlinkToForm</td>
<td>ConfidentialInformation</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Table 2 shows a survey of some of the most important fraud indicators that will allow a knowledge based system to conclude on the fraudulent nature of an e-mail message. During language engineering, these fraud indicators or so-called ‘red-flags’ are used to highlight, annotate or extract, from an archive of fraud e-mails, the keywords or expressions needed to recognize the fraud.

The extracted linguistic structures are classified according to the fraud indicators. In a next step, the expressions and words linked to a specific red flag are translated into regular expressions. In this study, we did not explore the use of any natural language processing/understanding engine. In a preliminary stage, the regular expressions engine was considered sufficient to explore the ontology-based natural language engineering for fraud detection. A previous similar approach of ontology-based information extraction was explored by the Data Extraction Group at Brigham Young University.3

The search phase in the language engineering track is further divided into three steps. First of all, a semantic model is defined, based on the results from the knowledge analysis and ontology development (section 3.3). Secondly, the syntactic and lexical structures are analyzed that express the semantic model in the light of representative samples. Finally, the syntactic and lexical descriptions are formalized into regular expressions. Table 3 shows how syntactic and lexical structures in French (see ‘B’ in table 3) of the fraud indicator ‘Capital’ (see ‘A’ in table 3) are translated into a regular expression rule (see ‘C’ in table 3).

Table 3. Regular expression of the Nigerian fraud indicator ‘Capital’ covering French samples

<table>
<thead>
<tr>
<th>A. Fraud indicator</th>
<th>B. Syntactic and lexical structures</th>
<th>C. Regular expression</th>
</tr>
</thead>
</table>

4 Deployment - Solution

During deployment, all components are integrated into the system environment. The e-mail fraud detector runs behind a web page which contains a text area into which users can paste the e-mail message they want to analyze.

---

3 See: http://www.deg.byu.edu/
Fig. 5. Result of the e-mail fraud detection

Under the text area is a choice list where the user needs to select the language of the e-mail. Currently, there are two languages which the system supports: English and French.

Results of the analysis are shown in a separate web page, divided into four blocks (see figure 5). In the first block (‘fraud notification’), the application specifies if the e-mail is fraudulent or not. In the second block, the type of fraud is named and defined. The third block shows the e-mail with highlighted words/phrases denoting fraud indicators. Finally, the fourth block is for the fraud specialists to check which fraud indicators have been identified in the e-mail message.

The e-mail fraud detector has been developed in Java (system requirements: CPU 800MHZ, Windows or Linux, Apache Tomcat 2.0, JRE 5.0). The system runs behind a webpage, which will soon be added to the FF POIROT website (http://www.ffpoirot.org). Current experiments show promising results with respect to the application’s precision. We intend to fine-tune the language model by analyzing more e-mails and to report on the application’s results.

5 Conclusion

In this article, we have focused on the development cycle towards an application recognizing different types of e-mail fraud, such as phishing, Nigerian advance fee fraud and lottery scam. The development cycle covers four tracks of development – language engineering, terminology engineering, knowledge engineering and system engineering. These development tracks are preceded by a problem determination
phase and followed by a deployment phase. Each development track is supported by a specific methodology.

This work is done in the context of intelligent system development for more complex domains of knowledge engineering and ontology-based information extraction, such as information extraction from emails or web pages based on ontology models provided by intelligence analysts and domain experts for fraud detection and investigation.

Email frauds are selected as a simple use case for a walk-through of development cycle. The problem space is reasonably well structured semantically, narrow in linguistic coverage. An alternative approach thus can be mainly based on the key words processing without going into conceptual modeling and systematic linguistic engineering. In addition, there can be a large collection of corpora, with which statistics-based approach to text processing is viable. The recognition of email frauds can be treated as text clustering. Considering relatively narrow linguistic coverage, the ontology model we created of frauds is deployed in the regular expressions rather than on a sophisticated natural language processing engine and experimental results prove effective [15].

The success of key word processing and text clustering banks on the prominent linguistic features of fraudulent emails, without going deeper into semantic models of frauds. This can be problematic in two main senses. One is the form and expression of frauds are very dynamic with fast mutations in different dimensions. The other is its forms are most deceptively genuine. The approach to cope with the dynamics of fraud, suggested in this paper, is one of methodological knowledge engineering: layering and modularization to encapsulate changes and maximize partial reuses with multi-viewpoint conceptual and linguistic modeling. The conceptual model, terminology and linguistic structure evolve at different paces, due to different factors and in different lifecycles. It recognizes three facts in the domain of fraud examination. Firstly, no matter how fast the form mutates, fraud schemes are well understood and remain stable. The semantic model of fraud essentials is often partially (at least) common among various types of fraud. Secondly, there is often no large corpus available for machine training in fraud detection, though the domain knowledge of them is readily available and can be explicitly modeled. This is particularly true when speedy reaction is needed from fraud examiners. Thirdly, precision is important than recall in face of deceptive and noisy objects under examination. The study of the detection of email frauds illustrates the potential added-value of application ontology in the competition of staying ahead of the game with fraudsters.

Acknowledgements

The study is partially funded by the European Commission in the project of FF POIROT (IST-2001-38248) (www.ffpoirot.org).
References