Computing and Informatics, Vol. 30, 2011, 57–87

EMAIL ANALYSIS AND INFORMATION EXTRACTION FOR ENTERPRISE BENEFIT

Michal Laclavík, Štefan Dlugolinský, Martin Šeleng
Marcel Kvassay, Emil Gatial, Zoltán Balogh, Ladislav Hluchý

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: `michal.laclavik@savba.sk`

Manuscript received 22 October 2010; revised

Abstract. In spite of rapid advances in multimedia and interactive technologies, enterprise users prefer to battle with email spam and overload rather than lose the benefits of communicating, collaborating and solving business tasks over email. Many aspects of email have significantly improved over time, but its overall integration with the enterprise environment remained practically the same. In this paper we describe and evaluate a light-weight approach to enterprise email communication analysis and information extraction. We provide several use cases exploiting the extracted information, such as the enrichment of emails with relevant contextual information, social network extraction and its subsequent search, creation of semantic objects as well as the relationship between email analysis and information extraction on one hand, and email protocols and email servers on the other. The proposed approach was partially tested on several small and medium enterprises (SMEs) and seems to be promising for enterprise interoperability and collaboration in SMEs that depend on emails to accomplish their daily business tasks.

Keywords: Email, information extraction, trees, graphs, social networks, context, recommendation

1 INTRODUCTION

Recent reports [1] confirm that email is still number one online activity [4]. According to Radicati surveys, corporate users send and receive an average of 110 messages per day [5], out of which about one third are messages sent. These statistics are quite stable and did not change much in the last 5 years [3], but the needed volume for storage and email accounts penetration is increasing, reaching almost 3 billion email accounts over the world (with 1.6 account per user). This includes 730 million corporate accounts, which should grow to almost 1 billion in 2014 [5]. In 2001, corporate users received just about 25 email messages per day and sent about 13 messages [4]. Findings from 2003 [6] also show that 80 % of users prefer email for business communication. Pew/internet report [2] from 2008 says that in the USA, 62 % employees could be considered *Networked Workers*, using the internet or email at their workplace on a regular basis. There is also a growing trend in wireless email accounts including both the enterprise and consumer accounts with the total of about 307 million in 2010, expected to grow rapidly over the next four years to 1.4 billion wireless accounts worldwide [5]. These findings indicate that the importance of email in business may not be adequately addressed in the research.

Information created by business entities can represent an asset or a liability, depending on how well it is managed. Email is not different in this respect: it can be a highly efficient and useful tool for communication, but only if the information it contains can be managed effectively. Email is rarely a standalone information source; it often points to further information such as files (e.g., saved attachments), links to items on the web, and references to other resources. Email is currently used as a conduit for many functions [7, 8], including alerting, archiving, task management, collaboration and interoperability.

1.1 Content Analysis, Extraction and Semantics in Emails

Several existing commercial solutions analyze and partly "understand" the email content. For instance, Gmail focuses mainly on context sensitive advertisements, but can also detect events, addresses or package tracking numbers. Similarly, Zimbra or ClearContext try to recognize some objects in the text. Xobni focuses mainly on the extraction of contact data from email signatures [9, 10]. One of the first attempts to apply Semantic Web technologies to email was performed by McDowell [11], who tried to solve the problems arising in one-to-many communication tasks, such as event planning, by including Semantic Web formal data in the message. One of the most significant attempts to analyse and understand the email communication has been performed with Semanta [12]. It applies speech act theory to the email communication processes, eventually giving a formal structure and semantics to ad-hoc workflows, which are characteristic of email communication. Speech act theory was applied also by [14] for "email acts" classification. For content analysis and semantics, it is also important to have available corpuses with pre-annotated data. Not many of them are available. To our best knowledge, the only publicly available email corpus with enterprise emails is the Enron corpus [13]. More information on email corpuses, semantics and content analysis is provided in our previous work [9, 10].

1.2 Social Networks in Emails

Email communication analysis allows the extraction of social networks with links to people, organizations, locations, topics or time. Social networks included in the email archives represent a level of semantics beyond speech-acts, and are becoming increasingly valuable assets in organizations, enterprises and communities, though to date they have been little explored. However, email social networks have been studied to some extent. For example, communication on the Apache Web Server mailing lists and its relation to CVS activity was studied in [15]. This work also introduces the problem of identifying email users' aliases. Extracting social networks and contact information from emails and the Web and combining this information is discussed in [16]. Similarly, new email clients, e.g. Postbox or plug-ins Xobni[1], try to connect email social networks with web social networks like LinkedIn or Facebook. Xobni in addition exploits email social networks to help the user manage contacts and attachments. We have also performed some experiments with the extraction of social networks from large email archives and network transformations using a semantic model [17]. A related research effort [18] exploits social networks to identify relations and tests the proposed approaches on the Enron corpus. There is a lot of research focusing on social networks in the context of web social networking applications, but the email social networks are different. In the email archives it is possible to discover the level of interactions (number of messages exchanged, time, relation to content and possibly discovered semantics) that goes beyond what is captured by the current social networking sites, and the potential of these differences for better information and knowledge management still needs to be explored. We are using a similar approach to that of IBM Galaxy [19] in the Nepomuk[2] project, where the concept of multidimensional social network was introduced. In this paper we show the initial results of exploiting the email social network in order to support a better understanding of the email content as well as enabling novel applications such as contact, product, service, partner or supplier search within organizations or communities.

1.3 Contextual Recommendation

Efforts to link emails with knowledge or context-sensitive information have been attempted in several tools [9] such as kMail, Zimbra, Gmail or Xobni. Additional R & D prototypes have been developed to address specific aspects of the general email communication problem (e.g. task management, information archiving, collabora-

[1] `http://www.xobni.com/`
[2] `http://nepomuk.semanticdesktop.org/`

tion, etc.), such as Telenotes, ContactMap, TaskMaster, Snarf, Remail, Priorities or recent Semanta. Our recommendation components build on the extracted information in the form of semantic trees and social networks. Deeper insights into the related work are provided in our previous analysis [9, 10].

1.4 Paper Contribution and Structure

In Section 2 we discuss our approach to information extraction and analysis of email communication as well as its main concepts based on key-value pairs, semantic trees and social networks. In Section 3 we discuss the use cases and prototypes for contextual recommendation and email social network search based on the extracted information. We also discuss Acoma system which can integrate with email infrastructure and describe prototypes and techniques. In evaluation and experiments Section 4 we provide the customization and the relevance evaluation of the approach, as well as the inference evaluation on social networks. Finally we conclude our findings in Section 5.

2 TECHNIQUES FOR INFORMATION EXTRACTION AND ANALYSIS

Information Extraction (IE) techniques [20] usually focus on the five main tasks of information extraction defined by the series of Message Understanding Conferences (MUC):

Named entity recognition (NE) – finding entities. Finds and classifies the names, places, etc.

Coreference resolution (CO) – aliases and pronouns referencing the entities. Discovers the identity relations between entities.

Template element construction (TE) – properties or attributes of entities. Adds descriptive information to NE results (using CO).

Template relation construction (TR) – relations between entities. Finds relations between NE entities.

Scenario template production (ST) – events involving entities. Fits TE and TR results into specified event scenarios.

Several advanced state-of-the-art systems such as GATE [21], KIM [22], C-PANKOW [23] or knowItAll [24] exist and are able to fulfill some of these tasks. C-PANKOW or knowItAll focus on general information domain such as web, and are not applicable to the enterprise-specific content. Usually if we want to get the best results we need to apply Natural Language Processing (NLP) techniques to decompose the sentences, and do Part of Speech Tagging (POS) to distinguish the nouns, pronouns, adjectives and verbs. Then we can apply the techniques of NE on nouns, CO on pronouns and TE on adjectives. The problem occurs if we want to apply these techniques on languages where the basic NLP support (e.g. stemmers or POS tagging) does not exist. NLP techniques such as those used in knowItAll, C-PANKOW or GATE are applicable mainly to English. Further problems arise if we apply the existing techniques to business documents (e.g. related to interoperability) or email content, which is specific for each enterprise or business sector, and very different from web documents or news articles, where NLP techniques are usually tested. This forced us to explore other approaches to information extraction in the context of business interoperability and email communication.

We found out that a lighter approach – *pattern-* and *gazetteer-based* detection – was not only much simpler, but also easily adaptable to the unique information extraction needs in different enterprises. Patterns are created manually by an expert or skilled programmer familiar with regular expressions. In customization evaluation (see 4.1) we have conducted experiments, which show that the required manual effort is not too high. Pattern- and gazetteer-based extractors generate key-value pairs (object type – object value), that are used to form semantic trees (see 2.2). It is possible to connect the extracted pairs and trees into a so-called multidimensional social network representing the email communication. Such semantic graphs or networks can be exploited to return the relevant results and discover the relations among business objects by relatively simple algorithms, such as spread of activation. We discuss this in Section 2.3.

Although we prefer the pattern-based detection as a light-weight and flexible approach, our general software framework can integrate any information extraction tool (including NLP), so long as it provides key-value pairs as output. In this respect, we have for instance succeeded in integrating our prototype with the GATE system and also used the Ontotext standalone gazetteer[3] originally developed for GATE.

Email communication typically lacks strict structure, but in many cases it carries structured or semi-structured information. This applies to business communication and especially to interoperability (or transaction) emails. Such emails typically contain business data or documents with objects and properties such as company names, amounts, product codes, bank account numbers, product quantities or prices. As already mentioned, the state-of-the-art IE techniques are usually meant for web documents or news. Email is different in the sense that email archives often contain references to enterprise-specific business tasks, processes, products, services or transactions (often time-related) and threads of communication. Social networks of interacting people can be extracted from the emails as well. They may be enriched with the related business objects, thus giving rise to multi-dimensional social networks. Our approach is based on the idea that a substantial portion of this valuable information can be extracted using the light-weight pattern-based approach in combination with gazetteers to fulfil the following IE tasks:

[3] `http://www.ontotext.com/downloads/index.html`

NE: entity detection

CO: aliases detection (e.g. product by detecting the product code, or customer by detecting his/her email or phone)

TE: attributes and properties by the segmentation of messages and grouping of properties or using relations in the email messages

TR: relations between entities gathered from external systems (e.g. email-customer) but also relations based on multidimensional social network extracted from email communication

ST: mainly related to email itself, time of sending and the related tasks or transactions.

The extracted data goes beyond the entities and their properties; it includes the relationship graph as well. It can be further analyzed for a more precise interpretation of the intent of the messages, and the results passed on to new tools assisting the enterprises in their business tasks. In this paper we show several possible ways how to use the extracted information.

In addition to the pattern-based extraction, we detect entities (NE task) using gazetteers. Gazetteers are lists of objects of concrete type represented by strings that can be matched in the text. The well-known gazetteers from the existing IE tools are gazetteers for geographical location names, lists of organizations or people. In the business context the most important are the gazetteers representing products, services, customers or suppliers. Such lists can be created from the existing legacy systems in organizations but can also be partially extracted from the email archives using predefined patterns. We have been developing and improving our information extraction approach over the past few years. Our tool, Ontea, is being continuously developed as an open source project[4]. User interface of the recent Ontea version, which focuses primarily on emails, is shown in Figure 1.

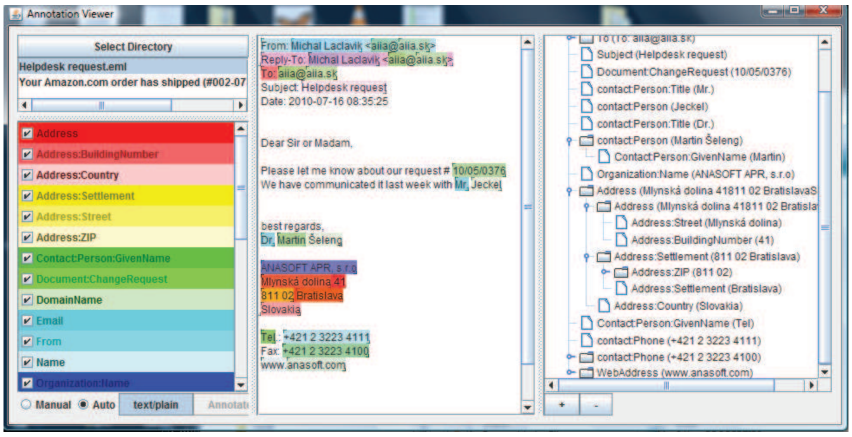
The power of the Ontea approach is in its simplicity [25] (compared to more advanced but heavy solutions such as GATE), as well as in its ability to define transformation chains and to connect to information system environment (databases, documents, intranets, and internet). In addition, Ontea supports email decomposition and analysis of email header, body and attachments. Supported attachment types are emails in .eml format, MS Word, PDF and text files, which are converted into a plain text. After an attachment or its content (archive file attachment) is converted to plain text, it is further processed like text in email.

We have tested our approach in the context of 6 organizations. Within the Commius project[5], we have tested it on the emails from Softeco, Aitek and Techfin SMEs (Italy), and from the Fedit technology center (Spain). Our relevance and social network evaluation in Section 4.2 is based on Fedit emails. In a related AIIA[6]

[4] `http://ontea.sourceforge.net/`

[5] `http://www.commius.eu/`

[6] `http://aiia.ui.sav.sk/`



```
From: Michal Laclavik <aiia@aiia.sk>
Reply-To: Michal Laclavik <aiia@aiia.sk>
To: aiia@aiia.sk
Subject: Helpdesk request
Date: 2010-07-16 08:35:25


Dear Sir or Madam,

Please let me know about our request # 10/05/0376
We have communicated it last week with Mr. Jeckel


best regards,
Dr. Martin Šeleng

ANASOFT APR, s.r.o
Mlynská dolina 41
811 02 Bratislava
Slovakia

Tel.: +421 2 3223 4111
Fax: +421 2 3223 4100
www.anasoft.com
```

```
To (To: aiia@aiia.sk)
Subject (Helpdesk request)
Document:ChangeRequest (10/05/0376)
contact:Person:Title (Mr.)
contact:Person (Jeckel)
contact:Person:Title (Dr.)
contact:Person (Martin Šeleng)
    Contact:Person:GivenName (Martin)
Organization:Name (ANASOFT APR, s.r.o)
Address (Mlynská dolina 41811 02 BratislavaS
    Address (Mlynská dolina 41811 02 Bratisla
        Address:Street (Mlynská dolina)
        Address:BuildingNumber (41)
    Address:Settlement (811 02 Bratislava)
        Address:ZIP (811 02)
        Address:Settlement (Bratislava)
    Address:Country (Slovakia)
Contact:Person:GivenName (Tel)
contact:Phone (+421 2 3223 4111)
contact:Phone (+421 2 3223 4100)
WebAddress (www.anasoft.com)
```

Fig. 1. Email message processed by the Ontea information extraction tool. Detected objects are highlighted. On the right, several objects such as address or person are grouped into a hierarchical tree.

Slovak national project, we have tested our information extraction on the emails of two organizations: the first was an SME (Anasoft), the second an academic internet provider (SANET). These experiments showed that *pattern-* and *gazetteer-based* IE approach can be customized for a specific application area (where the objects need to be discovered in the email communication) in a few hours.

Thus the aim of the Ontea IE is to create and share the patterns and gazetteers for objects and object properties. We believe the best approach for IE in the enterprise context is the pattern- and gazetteer-based extraction, for several reasons:

- Patterns can be adapted for enterprise business needs. For example, the formats of product codes may differ between companies, but within one company they tend to be well-defined and adhered to.
- Patterns can be defined, improved, evaluated and shared for a group of SMEs or for a community around a similar business model or industry type. This role is best suited for IT providers of these SMEs and communities.
- Patterns are easily adaptable to different languages when compared to advanced NLP tools, since NLP requires lemmatizators, stemmers or Part of Speech (POS) tagging tools, which are not available for many languages.
- Emails naturally contain many objects easily extractable by patterns, such as email addresses, phone numbers, people names, company names, dates, websites, addresses or other contact details. In addition, social networks, interactions and message passing data can be extracted from email headers using patterns.
- Business content usually consists of many objects and properties that can be similarly extracted via regular expression patterns, such as amounts, product codes, bank account numbers or customers.

- Patterns can be combined together with gazetteer results. For example, we have used this approach on person and company names extraction, where we defined a gazetteer list for given names, company legal classification abbreviations as well as for some common company-name specific words like Bank of, Hotel, Association, etc.

2.1 Key-Value Pair-Based Information Extraction

Early results of our approach were already presented in [25], where we also presented various information retrieval techniques for aliases and disambiguation as well as key-value transformations and integration with a variety of tools. We have also provided the state of the art of the relevant information extraction and semantic annotation techniques.

The main concept is key-value pairs representing object type (key) and value (matched text). For example, in Figure 1 we can see key-value pair, where the key is *contact:Person* and value is *Martin Šeleng*, which was extracted by the regular expression pattern of two words starting with capital letters (in this case also by identifying the title *Dr.* at the beginning). Please note that there are also other key-value pairs like *contact:Person:GivenName – Martin*, which are sub-parts of previous key-value pair. This key-value pair was extracted by the gazetteer for given names. We extract all the key-value pair candidates using the predefined patterns or gazetteers independently, often maybe identifying the same key-value pair by several techniques. Positions of key-value pairs are known and can be used for further processing, for example deleting the contact:Person pair candidate if GivenName is not matched inside. Here is an example of such configuration in Ontea tool:

```
ontea.core.gazetteer.Gazetteer {  gazetteer/lists.def }
ontea.core.xmlregex.XMLRegexExtractor {  patterns/person.xml }
ontea.transform.resultset.RuleTransformer {
  contact:Person* =>
  contact:Person:GivenName =>
  contact:Person:GivenName * => contact:Person:GivenName
}
```

We will not explain the above example in detail, but as you can see, gazetteer, pattern extractor and key-value set transformation is configured. Each block in configuration starts with Java class name implementing extraction or transformation interfaces and the brackets{} contain input for the implemented class. The Java classes are loaded dynamically by name, which allows to add more extraction and transformation tools into the system as plug-ins implementations of the interfaces. Similarly the Gate application can be plugged-in.

The example below shows fragment of patterns for the detection of postal address, which define macros for various address parts, such as city or zip, which are then incorporated into the postal address pattern. Macros can be reused. Based on the overlap of the extracted key-value pairs, we can build the trees, which can be seen on the right side of Figure 1 and in Figure 2. Trees are discussed in the next section.

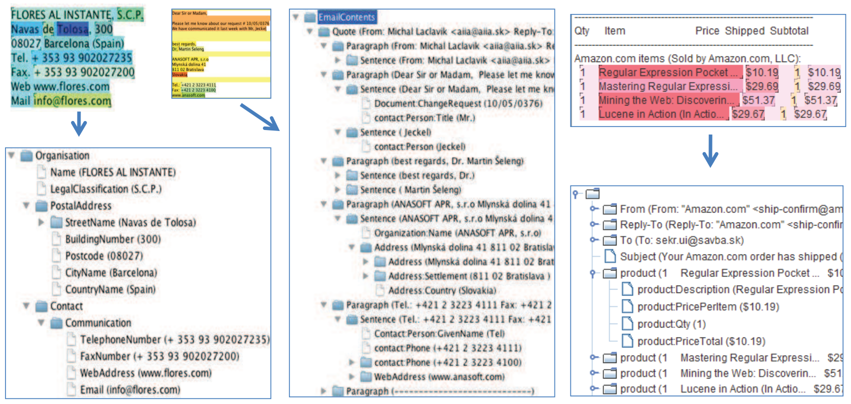
```
<?xml version="1.0" encoding="UTF-8"?>
<patterns>
  <pattern name="Postcode" class="Postcode">
    <regexp><![CDATA[(?:[0-9]{3} *[0-9]{2})]]></regexp>
  </pattern>
  <pattern name="CityName" class="CityName">
    <regexp><![CDATA[(?:\p{Lu}\p{Ll}+(?: \p{L}+)*)]]></regexp>
  </pattern>
  ....
  <pattern name="PostcodeCityLine">
    <regexp>
      <![CDATA[(?:\p{pattern:Postcode}+\p{pattern:CityName}
        (?: +\(\p{pattern:CountryName}\))?)]]>
    </regexp>
  </pattern>
    ....
</patterns>
```

2.2 Semantic Trees

Our approach to building semantic trees is quite novel and is not mentioned in the state of the art information on the extraction tools elsewhere. The extracted key-value pairs are formed into hierarchical trees depending on their positions in the text and the key names logical dependencies according to a pre-defined XML schema and tree transformation rules. Hierarchical trees give us extra information about the relationships among the result instances. For instance, an Organisation result is a sub tree that contains nested results like Name, Email, TelephoneNumber, StreetName, BuildingNumber, PostCode, CityName, etc. From the tree we can see that its sub-components are related to each other since they have a common parent (Figure 2 left).

In the Commius project, we use modified Core Components specification[7] to represent the information extraction results; but constructing a result tree compliant to CoreComponents specification does not involve just the information extraction. It also requires subsequent transformations of the hierarchical tree of results, for which we use several tree transformers. We define tree transformation rules, which call the tree transformers and can create, delete, move, rename or reorder the tree nodes (information extraction results). The tree on left side of the Figure 2 represents the Organization object compliant to the Core Components XML schema. Such XML

[7] http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf



```
FLORES AL INSTANTE, S.C.P.
Navas de Tolosa, 300
08027 Barcelona (Spain)
Tel. + 353 93 902027235
Fax. + 353 93 902027200
Web www.flores.com
Mail info@flores.com
```

```
▼ Organisation
    Name (FLORES AL INSTANTE)
    LegalClassification (S.C.P.)
  ▼ PostalAddress
    ▶ StreetName (Navas de Tolosa)
      BuildingNumber (300)
      Postcode (08027)
      CityName (Barcelona)
      CountryName (Spain)
  ▼ Contact
    ▼ Communication
        TelephoneNumber (+ 353 93 902027235)
        FaxNumber (+ 353 93 902027200)
        WebAddress (www.flores.com)
        Email (info@flores.com)
```

```
Dear Sir or Madam,

Please let me know about our request # 10/05/0376
We have communicated it last week with Mr. Jeckel


best regards,
Dr. Martin Šeleng

ANASOFT APR, s.r.o
Mlynská dolina 41
811 02 Bratislava
Slovakia

Tel.: +421 2 3223 4111
Fax: +421 2 3223 4100
www.anasoft.com
```

```
▼ EmailContents
  ▼ Quote (From: Michal Laclavik <aiia@aiia.sk> Reply-To:
    ▼ Paragraph (From: Michal Laclavik <aiia@aiia.sk> Re
      ▶ Sentence (From: Michal Laclavik <aiia@aiia.sk>
    ▼ Paragraph (Dear Sir or Madam,  Please let me know
      ▼ Sentence (Dear Sir or Madam,  Please let me kn
          Document:ChangeRequest (10/05/0376)
          contact:Person:Title (Mr.)
      ▼ Sentence ( Jeckel)
          contact:Person (Jeckel)
    ▼ Paragraph (best regards, Dr. Martin Šeleng)
      ▶ Sentence (best regards, Dr.)
      ▶ Sentence ( Martin Šeleng)
    ▼ Paragraph (ANASOFT APR, s.r.o Mlynská dolina 41
      ▼ Sentence (ANASOFT APR, s.r.o Mlynská dolina 4
          Organization:Name (ANASOFT APR, s.r.o)
        ▼ Address (Mlynská dolina 41 811 02 Bratisla
          ▶ Address (Mlynská dolina 41 811 02 Brat
          ▶ Address:Settlement (811 02 Bratislava )
            Address:Country (Slovakia)
    ▼ Paragraph (Tel.: +421 2 3223 4111 Fax: +421 2
      ▼ Sentence (Tel.: +421 2 3223 4111 Fax: +421
          Contact:Person:GivenName (Tel)
          contact:Phone (+421 2 3223 4111)
        ▶ contact:Phone (+421 2 3223 4100)
        ▶ WebAddress (www.anasoft.com)
    ▶ Paragraph (----------------------------)
```

```
-------------------------------------------------------------
Qty    Item                    Price  Shipped  Subtotal
-------------------------------------------------------------
Amazon.com items (Sold by Amazon.com, LLC):
  1    Regular Expression Pocket ...  $10.19   1  $10.19
  1    Mastering Regular Expressi...  $29.69   1  $29.69
  1    Mining the Web: Discoverin...  $51.37   1  $51.37
  1    Lucene in Action (In Actio...  $29.67   1  $29.67
```

```
    From (From: "Amazon.com" <ship-confirm@am
    Reply-To (Reply-To: "Amazon.com" <ship-confir
    To (To: sekr.ui@savba.sk)
    Subject (Your Amazon.com order has shipped (
    product (1    Regular Expression Pocket ...   $10
        product:Description (Regular Expression Po
        product:PricePerItem ($10.19)
        product:Qty (1)
        product:PriceTotal ($10.19)
    product (1    Mastering Regular Expressi...   $29
    product (1    Mining the Web: Discoverin...   $51
    product (1    Lucene in Action (In Actio...   $29.67
```

Fig. 2. Left: An example of a hierarchical sub-tree of information extraction results with a corresponding source text. The hierarchy is determined by the result positions and tree transformation rules, as well as the extraction pattern nesting. The result is based on Spanish email from Fedit application. Middle: An example of a hierarchical tree built upon email segmenter results. Email text is divided into quotes, paragraphs and sentences. Example is based on the same email as seen in Figure 1. Right: The Amazon shipment email processed by the Ontea tool. We can see the generated product tree.

objects are then used in Commius to identify document types [28] and determine the current processes [29].

We build hierarchical trees from the ordered lists of results. Results in the lists are ordered by their position relative to the beginning of the source text and by their length (or end position). The length of the result is important when sorting the list items, because we can decide whether one result contains another one and put the parent result before its child results in the list. If we have such ordered list, then it is simple to produce a hierarchical tree from it. The basic hierarchy of results can be acquired from the results produced by email segmenters. We use several simple segmenters such as email quotation segmenter, paragraph segmenter and sentence segmenter. These segmenters divide email text into structured segments as information extraction results from which we can build a hierarchical tree (Figure 2 in the middle). Segmenters make a hierarchy starting up with a quotation block, followed by paragraphs, sentences and results. This hierarchy describes the relation of results on the basis of their closeness to each other, where we assume that results found in a sentence are semantically more closer than results in a paragraph. Information extraction results produced by the segmenters are exploited in recommendation prototype (Section 3.1) as well as in the social network inference algorithm described in Section 2.3.

Moreover, the hierarchy of the results in the tree is also revealed by the key-value pair results. This is done by exploiting the information extraction model we use.

The extraction model lets us create complex extraction patterns consisting of other patterns/macros, which together represent a composite structure. A good example is the postal address pattern, where we have several macros defined for street name, street number, city name, etc. and we use them in the address pattern (see patterns example in Section 2.1).

We are able to build rich objects using our tree- and graph-oriented approach and the transformation techniques described above. Complex structures such as those shown in Figure 2 can be transformed into XML according to standards such as Core Components, but can also be used to create rich ontology instances and thus serve not only the needs of information extraction but for semantic annotation [25] as well. One semantic tree represents one email message, but when we connect the trees from the whole email archive (they are interconnected through shared objects), then we get a multi-dimensional social network or graph, which can be used for knowledge management or for inferring hidden relationships, as discussed in the next section.

2.3 Email Social Networks and Graph Inference

Information Extractor (IE) primarily handles the straightforward cases of structured objects like postal address where the sub-components (street name, ZIP code, city, etc.) immediately follow one another in a predefined sequence captured by regular expressions and macros. Graph inference is meant for the tasks beyond the reach of this method. One such example is to gauge the relationship between the telephone numbers and people which, in general, need not immediately follow one another in the email text. The task to assign telephone numbers to people is then a form of uncertain inference, which we are accomplishing through the application of spreading activation to the multidimensional social network graph provided by the information extractor.

We implemented our *social network extractor and analyzer* in Java on top of the information extraction tool Ontea and open-source graphical library Jung[8]. The novelty of our approach is in the application of the spreading activation algorithm to the twin tasks of reconstructing the social network from emails, and then efficiently searching the social graph. The prototype implementation described below is a work in progress. The evaluation of our initial experiments is provided in Section 4.3.

In [26] we have introduced two approaches to building multidimensional social network graphs depending on the format of the information extraction results. Here we only use the more advanced one where IE extracts complex objects (e.g. an address) consisting of simpler objects (e.g. a street name and a ZIP code), and preserves the information about their physical proximity by building a hierarchical tree that includes the nodes representing the sentences, paragraphs and blocks of the message in which they were found. Graph is built from hierarchical tree that can be seen in Figure 1 in the right column or in Figure 2. If the same object (string) is

[8] `http://jung.sourceforge.net/`

found in several messages, it is connected to all of them, but represented as a single object (node) in the graph. If the same object is found multiple times in the same message, it has multiple links to that particular message (or to its parts), which was our way of recording the strength of the bond.

2.3.1 Cumulative Edge Scorer with Attenuation

The structure depicted in Figure 3 can be visualized as a multipartite graph provided the objects of each data type are considered a separate partition. The objects in each partition have no connections among themselves, only to objects in other partitions, which is advantageous from the point of view of computational complexity. For the reconstruction of social network, our prototype *Cumulative Edge Scorer with Attenuation* implements a simplified breadth-first variant of spreading activation. The reconstruction consists in assigning the potential attributes, such as postal addresses, telephone numbers, etc. to the identified primary entities – people or organizations. For the purpose of testing our prototype we have chosen the task of assigning telephone numbers to people.

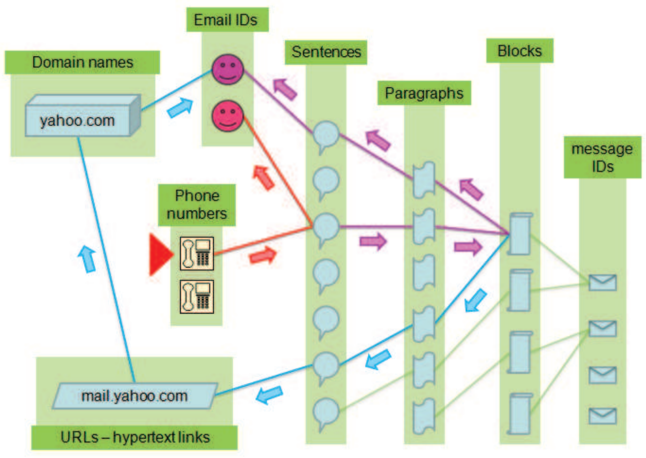


Fig. 3. Spreading activation in a multipartite graph. It starts at one attribute instance (phone number) indicated by the red arrow (center left) and flows towards candidate primary entities (email IDs) by a variety of ways. Since the activation gets attenuated each time it passes through an edge, the shortest path will carry over the greatest increment. But the longer paths can be so numerous that – depending on the value of attenuation and other parameters – their accumulated contribution will ultimately prevail

We were able to simplify our implementation of spreading activation by letting each node fire only once. This was possible because in our case the spreading activation always started from a single node – the attribute instance that we were trying to assign to a primary entity. Similarly, we did not restrict the maximum value of the activation value that a node could accumulate. Such a restriction is natural and appropriate e.g. in neural networks, but in our case it would be counter-productive since we were using the accumulated activation as a score to determine the most likely "owner" of the attribute instance. The result of these simplifications is an elegant and compact procedure described by the following JAVA-based pseudo-code.

Breadth-first variant of Spreading Activation. *Current_wave* collection (implemented as a *HashMap* with the node as key and its *accumulated_activation* as value) contains the nodes that are going to fire. At the beginning, it only contains one node – the attribute instance that we are trying to assign to a primary entity. In the next step, all its neighbouring nodes are placed into the *new_wave* collection as candidates for firing in the next iteration, and their accumulated activation is incremented by the value of *activation*, which reflects both the edge attenuation and the degree of the firing node. The iteration is completed when all the neighbours of all the nodes in the *current_wave* are processed. After each iteration, the number of iterations is incremented and the nodes in the *current_wave* are transferred into the fired collection so they do not fire again. The *current_wave* is then refilled by the *new_wave.get_activated_vertices()* method that returns the nodes meeting the activation criteria. The nodes representing the primary entities never fire. They accumulate in the *new_wave* and are returned at the end of the algorithm by the *get_output_type_vertices()* method. Their accumulated activation becomes their score, and the primary entity with the highest score will "own" the attribute instance.

```
do
  for (Vertex v : current_wave)
    activation = v.accumulated_activation();
    activation = activation/(attenuation * v.degree());
    for (Edge e : v.outgoingEdges())
      Vertex w = neighbouring_vertex(v,e);
      if (!has_fired(w))
        new_wave.smart_add(w, activation);
  total_iterations++;
  fired.add(current_wave);
  current_wave = new_wave.get_activated_vertices();
  new_wave.remove_activated_vertices();
while (!done());
return  new_wave.get_output_type_vertices();
```

The resulting scores of primary entities for each attribute instance (internally stored in nested HashMaps) can be output in the form of a hierarchical XML. This XML can be converted to HTML by XSL transformations. Results of spreading activation algorithm are also exploited in Email Social Network Prototype discussed in Section 3.2 and shown in Figure 6.

3 PROTOTYPES AND USE CASES

In this section we discuss the Acoma system, which integrates email analysis and information extraction approaches with recommendation and email social network search. We also describe prototypes for contextual recommendation and email social network search on simple use cases.

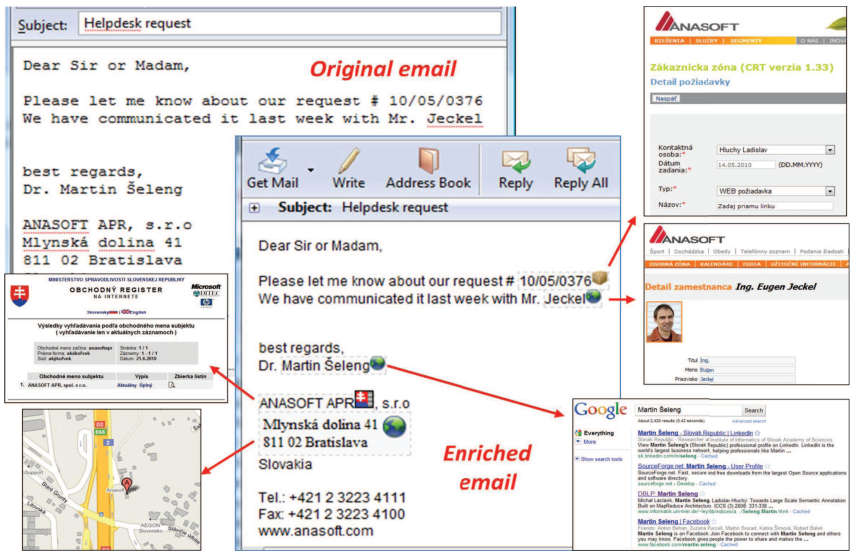
3.1 Contextual Recommendation Prototype and Use Case

We have developed simple contextual recommendation, built on top of the extracted data. The data needed for the recommender are in the form of extracted key-value pairs formed into semantic trees discussed in Section 2.2. For this kind of recommendation we do not use social networks or graph data, which are used in the next use case. Context for the recommender is key-value and tree based representation of email message. When the information extraction results are available (Figure 1), the email is further processed and additional information is added to the message in the form of HTML links or text attachments (see Figure 4). These additions contain further relevant information and knowledge, hints or links to business resources such as document repositories, databases or information systems needed in the detected business context. When checking for new emails, the users may receive added information/hints either in the modified (enriched) email as an attachment to the original unmodified message (this applies mainly to mobile devices), or in the original message as an attachment to the enriched HTML email with the additional information as seen in Figure 4. In this way the users can configure the email enrichment process according to their needs depending on device, email client or user preferences.

Our solution can successfully process mainly formal, system-generated emails such as Amazon shipment in Figure 2 on the right side. Formal emails are present in many cases of system-to-person communication, for example when purchasing or ordering goods and services on the Internet, in case of transaction status emails or transaction notifications. Typical examples are bank statements, hotel or air-ticket reservation confirmations, invoices to be paid, or shipped goods notifications. Such formal emails with fixed structure are easier to analyze and enrich with context-relevant information. They are often used in business tasks of SMEs and currently need to be processed manually, while we are able to process them semi-automatically. Our experiments, such as the example in Figure 4, have demonstrated that even regular person-to-person emails contain many structural parts and objects which can be detected using patterns or gazetteers.

Our solution, which connects to the email infrastructure and enriches the messages according to the context provided by the information extraction, is called Acoma [31] and is developed as an open source project[9]. Its internal architecture and modularity is described in Section 3.3.

[9] `http://acoma.sourceforge.net/`



Subject: Helpdesk request

```
Dear Sir or Madam,

Please let me know about our request # 10/05/0376
We have communicated it last week with Mr. Jeckel


best regards,
Dr. Martin Šeleng

ANASOFT APR, s.r.o
Mlynská dolina 41
811 02 Bratislava
```

Subject: Helpdesk request

Dear Sir or Madam,

Please let me know about our request # 10/05/0376
We have communicated it last week with Mr. Jeckel

best regards,
Dr. Martin Šeleng

ANASOFT APR, s.r.o
Mlynská dolina 41
811 02 Bratislava
Slovakia

Tel.: +421 2 3223 4111
Fax: +421 2 3223 4100
www.anasoft.com

Fig. 4. The same email as in Figure 1 is now enriched and changed into an HTML message with recommendations and links that can help process the email and the tasks it represents. For example, the user may go to the commercial register to find further information about the company, see its address on the map, or search for person in Google or some legacy system such as CRM.

3.1.1 Example of Use

A sample email processed by the Acoma system can be seen in Figure 4. It is based on a modified Anasoft Helpdesk request visible in Figure 1 (we changed the private info and translated the email into English). Acoma replaced the original plain text message by an HTML message including the recommendation hints with links. The links can be placed either inside the message text or on the right side of the message depending on the configuration. (The original email is attached and available to the user if needed.) Thus, the email message contains suggestions for actions to be taken to process the email.

Actions such as accessing a help desk request or a person in a legacy system, showing an address on a map or accessing company information in the commercial registry can be performed by clicking on the recommended hint represented by a box with icon.

The modified email also contains the link that will display the Graphical User Interface (GUI) related to the message in the browser. The GUI contains similar or extended information related to hints and provides the dynamic and interactive functionality, which cannot be achieved by the static HTML included in the email.

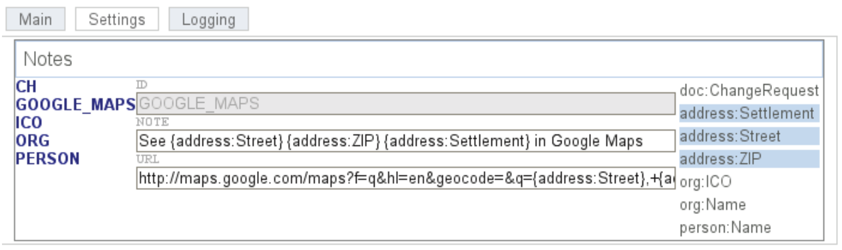
It can also integrate the legacy systems for which web GUI is not available. GUI offers extensive functionality but is not discussed in this paper.

Depending on the settings and the email client, Acoma can modify the email messages to include the text, html attachments or just a link to the message GUI.

3.1.2 Recommendation Hints

Information extraction results provide the necessary context for recommendation. Recommendation subsystem is quite simple. It just matches and fills in the text of hints with key-value pairs discovered by IE. It iterates over predefined hint templates and if the required key-value pair (keys are in {} brackets) for a hint is found, the hint template is filled in with the extracted information (value of the key-value pair). The results are then passed on for message post-processing to be included in the email as HTML-formatted output or simple text attachment with links.

Table 1 lists hint templates; in Table 2 the hints are replaced by the key-value pairs extracted from the email using the Ontea IE discussed in Sections 2 and 2.1. Hint templates can be defined through special user interfaces (Figure 5), where the same information as that available in Table 1 can be defined using the GUI.



ID: GOOGLE_MAPS
NOTE: See {address:Street} {address:ZIP} {address:Settlement} in Google Maps
URL: http://maps.google.com/maps?f=q&hl=en&geocode=&q={address:Street},+{a

Fig. 5. Interface for editing generic hint templates, with *See address on the map* hint template displayed. Objects that have to be found in order to fire the hint are selected on the right.

The first hint *show address on the map* is more complex than others, because it requires three objects to be found in the text of the message (street, settlement and zip). This hint is also shown in Figure 5. The problem arises if more than one street or one city name are discovered in the text of the email. In that case we first need to decide, which street and city belong to which address. We are solving this by email segmentation and trees (Section 2.2), and the required objects had to appear only once in common subtree, which can be paragraph, sentence or complex object in tree structure. If we will find all the required objects within one sub-tree (see Figure 1, for the address sub-tree, middle part of the Figure 2 for sentence or paragraph subtree or right part of Figure 2 for product sub-trees) the hint is fired. If we cannot locate the single required key-value pairs for a given hint template within one common sub-tree, the hint is not fired.

Hint Templates
See {address:Street}, {address:ZIP} {address:Settlement} in Google Maps http://.../maps?q={address:Street},+{address:ZIP}+{address:Settlement} address:Street, address:ZIP, address:Settlement link
Organization {org:Name} in commercial register http://orsr.sk/hladaj_subjekt.asp?OBMENO={org:Name}&PF=0&SID=0&R=on org:Name orsr

See request # {doc:ChangeRequest}
http://www.ana.sk/crt/main/task_detail.aspx?taskId={doc:ChangeRequest}
doc:ChangeRequest
product

See person: {person:Name}
http://www.google.com/search?q={person:Name}
person:Name
link

Table 1. List of hint templates used in the example shown in Figure 4. First line: text of hint; second line: link (URL); third line: objects to be found in the text in order to match the hint; last line: hint type/icon. Text in brackets {} are object types (keys) detected by the IE and replaced by concrete values in the generated hints.

Generated Hints
See Mlynská dolina 41, 811 02 Bratislava in Google Maps http://...google.com/maps?q=Mlynsk%C3%A1+dolina+41,+811+02+Bratislava link, 174, 209
Organization ANASOFT APR in commercial register http://orsr.sk/...asp?OBMENO=ANASOFT+APR&PF=0&SID=0&R=on orsr,155,166
See request # 10/04/0647 http://www.ana.sk/crt/main/task_detail.aspx?taskId=43048 product,59,69
See person: Martin Šeleng http://www.google.com/search?q=Martin+%C5%A0eleng link,140,153 See person: Jeckel https://intranet.ana.sk/.../DispForm.aspx?ID=61 link,113,119

Table 2. List of generated hints based on the templates from Table 1. First line: text of hint; second line: link (URL) to external system; third line: hint type/icon, start and end position in text.

Table 2 shows the hints generated from the hint templates in Table 1. The first hint is the already mentioned multi-object match hint, which can be used for *show address on map*, *process transaction* or *manage contact details* user actions. The second hint is the simplest hint requiring only one object (organization name) in the text; the same name is then reused for generating the link to the commercial registry. The third hint is again related to a single detected object, *change request ID* from the *Anasoft Helpdesk* application, but please note that while we are detecting the ID *10/04/0647* in the text, to access the intranet system we need to convert this ID value into *43048* in order to generate the appropriate URL. This is done by an extension, which is not described in this paper but uses the idea of key-value pair transformations mentioned in Section 2.1 and described previously in [25]. Similarly, the last hint appears twice in the email, since two people were detected. If a person is detected as an Anasoft employee, the link to his/her Anasoft intranet card is generated, otherwise just a link to Google Search is offered.

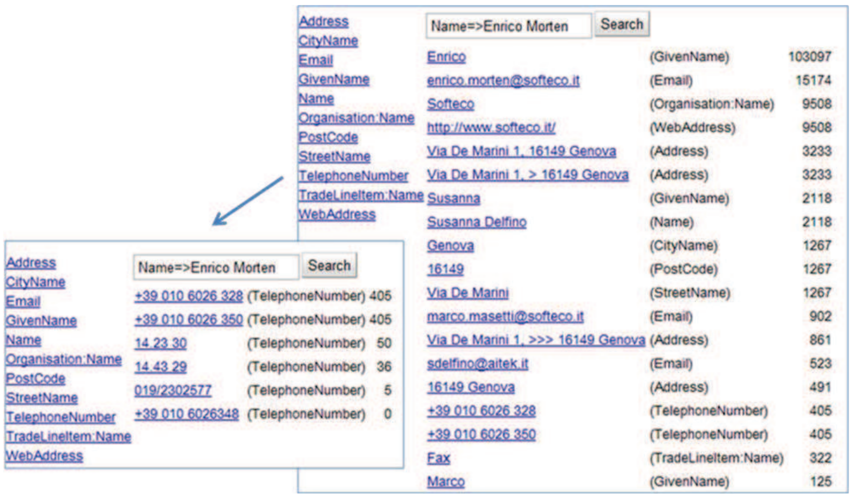
3.2 Email Social Networks Search Prototype and Use Case

Email Social Network prototype can be also integrated as a module into Acoma system, where it exploits email analysis and extraction in form of key-value pairs, semantic trees as well as their interconnection into graph of social network. It exploits the spreading activation inference algorithm discussed in Section 2.3.

Email Social Network Search Module enables the user to search real world objects mentioned in an email or email archive, and their relations. Exploiting the module the user can discover personal email addresses, telephone numbers, company names, company products and so on. When the user accesses the module GUI, all the objects related to the current email message (i.e. people, products, contact details) are shown as results and the user can navigate deeper into the object graph by clicking on any object. The user can, for instance, select a persons name to get the company where this person works, then select the company to get its products, and so on.

Figure 6 shows the user interface of the Email Social Network Search module. Its input is a key-value pair representing the business object (in this case, *person*) extracted from the email. When pressing the search button, the spreading activation algorithm is activated on the social network graph. The algorithm returns all the relevant objects (key-value pairs) for the searched object. For example, it returns organization names, email addresses, postal addresses, telephone numbers and locations. The left part of the screenshot shows the restriction of the search to return only the objects of a certain type (type is defined by the key of the key-value pair). In this case we wanted to get the phone number of the person *Enrico Morten*, which was quite low in the main (mixed-type) list, so the search was further restricted to return only the relevant phone numbers for this person by clicking on the type *TelephoneNumber* in the left panel of GUI.

The search algorithm can also be improved by allowing the users to delete the wrongly extracted objects or to connect the various aliases of the same object (e.g.



Enrico	(GivenName)	103097
enrico.morten@softeco.it	(Email)	15174
Softeco	(Organisation:Name)	9508
http://www.softeco.it/	(WebAddress)	9508
Via De Marini 1, 16149 Genova	(Address)	3233
Via De Marini 1, > 16149 Genova	(Address)	3233
Susanna	(GivenName)	2118
Susanna Delfino	(Name)	2118
Genova	(CityName)	1267
16149	(PostCode)	1267
Via De Marini	(StreetName)	1267
marco.masetti@softeco.it	(Email)	902
Via De Marini 1, >>> 16149 Genova	(Address)	861
sdelfino@aitek.it	(Email)	523
16149 Genova	(Address)	491
+39 010 6026 328	(TelephoneNumber)	405
+39 010 6026 350	(TelephoneNumber)	405
Fax	(TradeLineItem:Name)	322
Marco	(GivenName)	125

+39 010 6026 328	(TelephoneNumber)	405
+39 010 6026 350	(TelephoneNumber)	405
14 23 30	(TelephoneNumber)	50
14 43 29	(TelephoneNumber)	36
019/2302577	(TelephoneNumber)	5
+39 010 6026348	(TelephoneNumber)	0

Fig. 6. Email Social Network Search prototype GUI

the same company, personal names spelled differently, or the same address *Via de Marini* on the screen). Such user feedback will enable the module to learn and return better results in future. It is also possible to let the users annotate their own key-value pairs in the emails. In this way they can later easily search for the additional valuable information hidden in the emails, such as passwords, links or documents. An interesting feature would also be to extend the search to include the attachments.

3.3 Integration with Email Environment

The Acoma tool integrates the information extraction, contextual hint recommendation and the enrichment of the messages with email protocols and email servers. Moreover, the information extraction, recommendation and other functionality can be extended by implementing new modules which take email as input and produce recommendation hints as output. In a similar way, the hints shown in Figure 4 and described in the previous section were generated by three separate modules: generic hint template module (1st and 2nd hint), key-value pair transformation module (3rd hint) and a combination of key-value pair and URL transformation module.

Acoma architecture can be seen in Figure 7. The ambition is to support users in business tasks in the context of email communication. We do not want to force people to use new webmail interfaces, new plug-ins to desktop email clients or new email clients, but rather allow users to send and receive emails as they are used to doing it. The Acoma system is hooked to the mail server or desktop in a similar way as email antivirus programs (Figure 7). In this way it can be used with any email

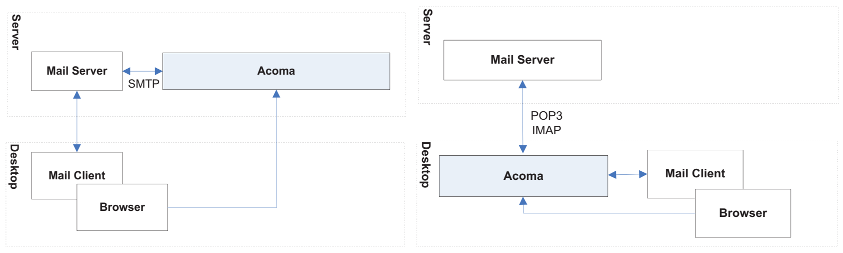


Fig. 7. Acoma on the server (left) and on the desktop (right)

client or even mobile device, without requiring changes to working practices or the adoption of new tools. By a combination of server and desktop use, the users can ensure that security and privacy issues are taken into account. Users and providers can be aware of what data is shared and passed via the communication.

3.3.1 Adaptability

Acoma is implemented using the OSGi[10] approach, which is one of the standards for modular systems in Java. We are using the Apache Felix[11] implementation of OSGi. This enables us to adapt the solution to integrate different functionalities such as context sensitive recommendation or Email Social Network Search into one integrated system accessible from user emails. We believe that such an email based system can benefit from the community approach: just as people are creating applications for Facebook or modules for different open source software, Acoma users can create, modify and share new modules, which will be able to execute or support specific email activities (for example, the Geo module for identifying and displaying the addresses on a map as shown in Figure 4). Other module examples include package tracking, event detection and adding events to calendar, Email Social Network Search module or modules for processing bank statements from a specific bank, service provider or mobile operator. This way the user can just download and add new modules to their Acoma system based on their needs, and so benefit from additional functionality. Even the service providers can create and share Acoma modules to their benefit. Users can be notified on module availability and install them upon receiving the notification. Within 6 SMEs which we have partially evaluated, we have identified the following modules: Catch a contact; Identification of the responsible person (related also to Email Social Network Search module); Customer card (CRM); Geo module; Voting module; Generic hint module; Generic key-value transformation and URL transformation modules. In addition, partner search module, attachment manager and mail template module are being developed in the scope of Commius project.

[10] `http://www.osgi.org/`
[11] `http://felix.apache.org/`

Voting, Geo and Email Social Network Search modules are implemented as prototypes. For the rest of the modules we have tested IE on emails of one or more enterprises. In this paper (Figure 4) we used the Geo module and early prototypes of simple modules. Hint templates presented in Section 3.1 will be covered by the final prototype of the generic hint module in future. When all the needed objects are detected, the hint action can be defined by an URL pointing to some existing web-based system. Functionality for dozens of hints can be created and implemented in a few hours. Customization of the information extraction is more difficult (Section 4.1) but can also be accomplished in a reasonable time-frame. We discuss this in the next section.

4 EVALUATION EXPERIMENTS

In this section we describe the relevance and customization evaluation of our Information Extraction approach. We also discuss experiments on social networks formed from the hierarchical trees and the results of the spread of activation inference in such networks. We do not evaluate the system as a whole, but we evaluate just main email analysis and information extraction approaches described in Section 2.

- Since building of patterns for key-value pair extraction is a manual process, we evaluate the customization process of key-value pair extraction. See Section 4.1.
- We also evaluate success rate of such extraction on selected object types such as people names, addresses and telephone numbers. See Section 4.2.
- Semantic trees are not directly evaluated. They have been found helpful in contextual recommendation and they are also needed in social network relation inference. Its evaluation is given in Section 4.3.

4.1 Information Extraction Customization and Experiments

In this section we describe how the proposed approach for IE can be customized for a specific enterprise or application. We have conducted several experiments in the Commius and AIIA projects, where these information extraction tools are being extended and developed.

The AIIA project focuses on two applications: Anasoft helpdesk, where the products, modules and components mentioned in the email are identified, and an appropriate contact person within the organization is chosen to deal with the email. Automatic identification of the customer contact details is performed in order to retrieve the CRM information related to the customer or change request. The second application is in SANET – the academic internet provider. The focus is on document- and task-related voting through email, in which the contacts, people, agreements and documents need to be identified.

The Commius project focuses on the enterprise interoperability for SMEs. It is important to extract organizations, people, products or transactions from the orders and invoices transmitted via email.

In the first experiment in Anasoft, we have annotated Helpdesk emails. Manual annotations were created with the Ontea tool, which supports this kind of task. Tags were created by the authors of this article (2 people), company expert (1 person) as well as real Helpdesk worker (1 person). We have selected 15 emails where 93 tags were created.

Manual annotations were also tested in the Commius project where several annotation events were organized in Italy and Spain. In Italy Softeco, Aitek and Techfin SMEs were involved. In Spain, Fedit technology center was involved. The focus of Commius annotation events was on the business aspects of perceiving and decomposing objects as well as on their mapping on UN/CEFACT Core Component[12] interoperability standard. Results are available in the Commius deliverable [27] and also in an article in this CAI journal issue [28]. The first round of experiments led us to the definition of the following customization process and to changes in the Ontea tool in order to support it. We believe the process is valid for any enterprise application where IE is needed:

1. *Emails/files browsing*, the step performed by the application users and the developer.
2. *Definition of the objects types and properties to be extracted*, the step performed by the application users and the developer.
3. *Implementation of the patterns for the objects*, performed by the developer.
4. *Transfer of automatically extracted entities to manual annotation mode.* The application user then does not have to spend too much time by annotating emails, he/she just adds the missed tags and deletes the superfluous ones.
5. *Manual annotation performed by the application user.* It is useful if the developer watches the process.
6. If we want to achieve high consistency, precision and recall, *several users should annotate the same set of emails.*
7. *Automatic annotation evaluation* using the precision and recall measures, or just visual evaluation in the tool by browsing the emails/files with the discovered tags (as in Figure 1).

In the subsequent annotation events and experiments this process was found to be valid and useful. In the context of Anasoft application we have identified the following objects to be extracted in the second step of the customization process:

- *Organization*: org:Name, org:RegistrationNo, org:TaxRegistrationNo
- *Person*: person:Name, person:Function
- *Contact info*: contact:Phone, contact:Email, contact:Webpage
- *Address*: address:ZIP, address:Street, address:Settlement
- *Product*: product:Name, product:Module, product:Component

[12] `http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf`

- *Document*: doc:Invoice, doc:Order, doc:Contract, doc:ChangeRequest
- *Inventory (rooms, computers, ...)*: inventory:ResID, inventory:ResType
- *Other business object*: bo:ID, bo:Type.

These objects are quite interesting and common for many enterprises, but only several of them (such as address, organization, contact or person) can be detected by the same patterns across enterprises. Of course, different patterns need to be defined for different languages. Patterns for products or business documents need to be defined for each enterprise. In addition, products are usually discovered using gazetteers. While the definition of objects and patterns can take just several hours, sometimes a bigger effort is required to create gazetteers (lists of products or services). Gazetteers need to be created manually or extracted from the company information systems if available. It is hard to estimate how much effort we have spent customizing the Anasoft Helpdesk application in the first round of experiments, since the customization process was not defined at that point in time, but in the second round (when the customization process was already defined and followed) the first step took us about 2 hours and the second step together with the seventh took us 3 hours.

In the SANET application, we have developed the Voting module. For this module, it was clear from the beginning what kind of objects needed to be detected in the emails: people, voting agreement, subject of email. Thus, not all the process steps were applied. The first and second steps were done within one hour. The third step was accomplished within 4 hours including the seventh evaluation step done only visually on about 100 emails. We found 2–3 emails that were problematic with regard to agreement detection. The voting module currently supports the agreement detection in Slovak language only.

Concerning the contact module, we have evaluated the contact detection on Spanish emails of the Fedit technology center. Basically all the needed steps (1, 3 and visually step 7) were performed by the developer, since the second step was given by the address structure: street name, number, ZIP code and city name. It took about 5 hours to create and fine-tune the address extraction patterns on 104 Spanish emails from Fedit, with 82–100 % precision and 81–94 % recall depending on the object type, which is reported in 4.2 in detail.

4.1.1 Customization Conclusion

Since the described customization process is manual (but supported by Ontea tool), we have provided this evaluation. It shows that customization for a small enterprise can be done in reasonable time. The provided tests were small so this can raise questions on scalability. In case of SANET application, we have run tests on larger email sets (more than 500 emails) and the results were satisfactory. We believe the relevance of extracted information can be improved also by allowing user to mark not discovered or wrongly discovered results directly in the email. This will be the focus of our future work.

4.2 Relevance Evaluation of Information Extraction

The point of interest in the relevance evaluation of information extraction from emails was the extraction of personal names, postal addresses and telephone numbers since such entities can be used in any enterprise, and patterns can be reused/shared among enterprises. To conduct the experiment, we have selected a subset of 50 Spanish emails in which we manually annotated these kinds of named entities in order to have a so-called golden standard for evaluation. We then evaluated the extraction results automatically by comparing them to manual annotations. We considered the information extraction result to be relevant, when it was strictly equal to the corresponding manual annotation. It means the automatic result and the manual annotation must have occupied exactly the same position in the email text. The evaluation results are summarized in the table below.

Type	Total relevant	Total extracted	Relevant extracted	Recall [%]	Precision [%]	F1 [%]
Personal name	779	788	499	64.06	63.32	63.69
Telephone number	262	178	166	63.36	93.26	75.45
Fax Number	139	127	121	87.05	95.28	90.98
Postal address	170	134	75	44.12	55.97	49.34

Table 3. Evaluation of information extraction from emails (strict match)

As we can see in Table 3, the results except for the fax number and telephone number extraction are not high. This is due to strict comparison of the extraction results against the annotated set, where many good results were rejected. For example, there were many cases where the extracted name was almost equal to the annotated one, but it differed from annotation by prefix *D.* Although this prefix could be considered as an abbreviation of the first name, in our situation it stands for Spanish *Don (Sir)*, and it was not included in the manual annotations since we did not consider the title to be a part of personal name. Postal addresses suffered from similar drawbacks: they were written in various formats, so many were extracted partially or not at all. Partially extracted addresses were classified as false matches. For example, if only a postal code with a city name was extracted from the whole address, it was a false match, although the postal code and city were located correctly. The same problem was with telephone numbers. There were many extractions, which were not identical but overlapped the manually annotated telephone numbers. This made us search for alternative ways of evaluation that would reflect these *partial successes* as well.

In the second evaluation, we considered the information extraction result to be relevant when it intersected the manual annotation. The results of evaluation were much better (Table 4) and showed us that if we enhance the information extraction from emails, we can significantly improve the recall and precision. It would also improve the performance of our social network extractor, since it operates on the results of information extraction.

Type	Total relevant	Total extracted	Relevant extracted	Recall [%]	Precision [%]	F1 [%]
Personal name	779	788	672	86.26	85.28	85.77
Telephone number	262	178	178	67.94	100.00	80.91
Fax Number	139	127	125	89.93	98.43	93.98
Postal address	170	134	133	78.24	99.25	87.50

Table 4. Evaluation of information extraction from emails (intersect match)

4.3 Social Network Evaluation

In this section we evaluate spreading activation inference algorithm on task of assigning phone numbers to persons. For both the personal names and phone numbers the social network extractor depended on the information extractor (IE). Its only responsibility was to correctly assign the received phone numbers to the received personal names. This difference in the task (compared to IE) necessitated a different evaluation methodology. Our evaluation of social network extractor focused on user needs and was inspired by the approach used to evaluate the TREC 2006 Enterprise Track in [30]. In our case, the user needs are defined as the ability to extract attributes (phone numbers) and assign them correctly to primary entities (persons). It is not necessary to have all the occurrences of each attribute value or primary entity identified in the email archive. It is acceptable that some occurrences are missed as long as each unique attribute value is identified somewhere, and correctly assigned to its proper primary entity. While the extraction of telephone numbers was fairly straightforward, a number of problems resurfaced during the extraction of personal names. Spanish personal names often consisted of four components, which the writers seldom spelled out in full. They might use three components only, and sometimes even two. On top of that, they would sometimes write their names properly accentuated, but at other times they would write without accents. We therefore decided to accept as a valid variant of personal name any extracted string consisting of at least two correctly spelled name components (accents were not required). The telephone number was considered correctly assigned if any of the acceptable variants of its owners name appeared as the first or second in the list of potential "owners" sorted by score. The results are summarized in the tables below. The number of telephone numbers in Table 5 differs from that in Table 3, because Table 5 represents only the unique telephone numbers for the whole test set of 50 emails, and not all occurences of telephone numbers in emails. Moreover, the telephone numbers were re-formatted by leaving only numeric characters in them, before they were added into the social graph.

Out of 42 relevant unique phone numbers in the test corpus (and 32 phone numbers actually extracted), two occurred in the emails that did not contain any personal name (only company name was present). These two were then irrelevant for the task of pairing phones and personal names; this is summarized in Table 6. Since there were 40 relevant unique phone numbers, we expected 40 correct pairs consisting

Total relevant	Total found	Relevant found	Recall [%]	Precision [%]
42	32	29	69	90.6

Table 5. Quantitative evaluation of the extraction of telephone numbers for social network reconstruction

of a phone number and a personal name. This was for us the *Total relevant* for the pairing task. The number of assigned pairs (shown in the *Total Found* column) is now 30, since it excludes the two irrelevant phone numbers. The number of correctly assigned phones to persons is shown in the *Relevant found* column.

Total relevant	Total found	Relevant found	Recall [%]	Precision [%]
40	30	18	45	60

Table 6. Quantitative evaluation of the assignment of phones to personal names

Table 6 implies that out of 30 found pairs, 12 were wrong. Out of these, 6 were caused by the fact that the Information Extractor failed to extract the name of the person that actually "owned" the phone number. Further 3 errors were caused by the extraction of wrong (corrupted) phone numbers, as is implied by Table 5. This means that out of 21 pairs which the social network extractor could possibly pair correctly, it only failed 3 times. This would give the theoretical precision of $18/21 = 85.7\,\%$, which demonstrates the potential of spreading activation in reconstructing social networks. The immediate conclusion is that we need to focus primarily on improving the quality of the information extraction of text-based data in multilingual contexts.

4.3.1 Social Networks Summary

In contrast to [26], where we tested our prototype on a set of 28 emails written in English, we now tested it on a set of 50 emails written in Spanish, supplied by our Commius project partner, Fedit. Spanish emails, with their accented characters and differing cultural norms (e.g. varying ways of writing personal names) represented a challenge that made a number of hidden issues visible in our prototype code. Some of them were solved on the fly, but other require a substantial redesign of the information extraction strategy. The 60 % precision of the social network extraction is less than what we obtained with English emails in [26] (precision of 77 %), but there are clear possibilities for improvement, especially at the information extraction stage. Spreading activation can deal with the lower precision of the information extraction but cannot cope with low recall, i.e. the situations when something needed was not extracted. In this respect, the main opportunity for improvement lies in making the partially good results of the information extraction (summarized in Table 4) available for the social network extractor, since now only strict IE results (shown in Table 3) are really exploited for social network reconstruction.

4.4 Evaluation Experiments Conclusion

To conclude, the success rate (precision and recall) of the Ontea IE is quite high which reflects our previous evaluations on web documents [25] as well as our present evaluation in Table 4, where the F1 measure for the intersect match was above 80 % in all cases. The developer fine-tunes the patterns and gazetteers on the target email data set in a given enterprise, and so achieves good results. We have also proved that the approach can be customized for enterprise applications in a reasonable time-frame of several hours, provided the customization is conducted by the person skilled in regular expressions.

5 CONCLUSION

We have developed a solution for simple and customizable processing of email messages with focus on enterprise emails. We believe the work described can be used in many applications in enterprise or community environment, such as knowledge management, social networks, information management or enterprise interoperability.

In the article, we have discussed our experiments with the customization of the information extraction in several enterprises and organizations. In future we plan to extend the solution with an easy user interface for user interaction with extracted data, setting up the application-specific patterns for the information extraction, hint creation and evaluation. We believe that enterprises can benefit from the proposed approach from the day one of the system installation by offering simple functionality such as search for related objects, catch a contact or show information about organizations/people. Enterprises that want to benefit from their email archives more substantially need to customize the system. Our experiments have shown that it is possible to customize it in a reasonable time-frame.

In this paper we have also discussed the email social network extraction, which employs the spreading activation algorithm on the information extraction results. We have shown that reasonable relations among business objects can be discovered based on the information hidden in the email archives. Such information can be used to populate the empty enterprise databases when installing a new information system, CRM, BPM or Help Desk. We have also shown the use of email social networks on the Email Social Network Search prototype, which exploits multi-dimensional social networks for exploring various object-to-object or object-attribute relations.

Acknowledgment

This work is partially supported by projects AIIA APVV-0216-07, Commius FP7-213876, APVV DO7RP-0005-08, SMART ITMS: 26240120005, SMART II ITMS: 26240120029, VEGA 2/0184/10. We would also like to thank Anasoft, Fedit, Aitek, Softeco, Techfin and SANET for providing us with emails for testing, and for their support with the customization of the information extraction process.

REFERENCES

[1] PewInternet Report: Online Activities 2010. May 2010, `http://tinyurl.com/pewOnline10`.

[2] Madden, M.—Jones, S.: Networked Workers. PewInternet report, Pew Research Center; September 24, 2008, `http://tinyurl.com/pewNetWrks08`.

[3] HP, The Radicati Group, Inc.: Taming the Growth of Email – An ROI Analysis (White Paper), 2005, `http://tinyurl.com/RadicatiEmail05`.

[4] Jones, J.: Gallup: Almost All E-Mail Users Say Internet, E-Mail Have Made Lives Better. 2001, `http://tinyurl.com/Gallup01`.

[5] The Radicati Group, Inc.: Email Statistics Report, 2010; Editor: Sara Radicati; `http://tinyurl.com/RadicatiEmail10`.

[6] META Group Inc.: 80 % of Users Prefer E-Mail as Business Communication Tool. 2003, `http://tinyurl.com/MetaEmail03`.

[7] Whittaker, S.—Sidner, C.: Email Overload: Exploring Personal Information Management of Email. In Proceedings of ACM CHI96, pp. 276–283, 1996.

[8] Fisher, D.—Brush, A. J.—Gleave, E.—Smith, M. A.: Revisiting Whittaker & Sidner's "Email Overload" Ten Years Later. In CSCW2006, New York ACM Press 2006.

[9] Laclavík, M.—Maynard, D.: Motivating Intelligent Email in Business: An Investigation Into Current Trends for Email Processing and Communication Research. In E3C Workshop; IEEE Conference on Commerce and Enterprise Computing; DOI 10.1109/CEC.2009.47, 2009, pp. 476–482.

[10] Balzert, S.—Burkhart, T.—Kalaboukas, K.—Carpenter, M.—Laclavík, M.—Marin, C.—Mehandjiev, N.—Sonnhalter, K.—Ziemann, J.: Appendix to D2.1.2: State of the Art in Interoperability Technology, Commius project deliverable (2009), `http://tinyurl.com/CommiusSoTA`.

[11] McDowell, L.—Etzioni, O.—Halevy, A.—Levy, H.: Semantic Email. in-WWW '04: Proceedings of the 13th international conference on World Wide Web. New York, NY, USA: ACM, 2004, pp. 244–254.

[12] Scerri, S. et al.: Improving Email Conversation Efficiency through Semantically Enhanced Email. In: 18th International Conference on Database and Expert Systems Applications, IEEE Computer Society 2007, pp. 490–494.

[13] Klimt, B.—Yang, Y.: Introducing the Enron Corpus. CEAS, 2004, `http://www.ceas.cc/papers-2004/168.pdf`, `http://www.cs.cmu.edu/~enron/`.

[14] Carvalho, V. R.—Cohen, W. W.: On the Collective Classification of Email "Speech Acts". In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, New York, ACM 2006, pp. 345–352.

[15] Bird, C.—Gourley, A.—Devanbu, P.—Gertz, M.—Swaminathan, A.: Mining Email Social Networks. In: MSR '06: Proceedings of the 2006 International Workshop on Mining Software Repositories, ACM, New York 2006, pp. 137–143.

[16] Culotta, A.—Bekkerman, R.—McCallum, A.: Extracting Social Networks and Contact Information from Email and the Web. In: CEAS '04: Proceedings

of the First Conference on Email and Anti-Spam 2004, `http://www.ceas.cc/papers-2004/176.pdf`.

[17] Laclavík, M.—Šeleng, M.—Ciglan, M.—Hluchý, L.: Supporting Collaboration by Large Scale Email Analysis. In: Cracow '08 Grid Workshop: Proceedings, Academic Computer Centre CYFRONET AGH, Kraków 2009, pp. 382–387, ISBN 978-83-61433-00-2.

[18] Diehl, C. P.—Namata, G.—Getoor, L.: Relationship Identification for Social Network Discovery. In: The AAAI 2008 Workshop on Enhanced Messaging, 2008.

[19] Judge, J.—Sogrin, M.—Troussov, A.: Galaxy: IBM Ontological Network Miner. In: 1st Conference on Social Semantic Web, Volume P-113 of Lecture Notes in Informatics (LNI) series (ISSN 16175468, ISBN 9783-88579207-9), 2007.

[20] Cunningham, H.: Information Extraction, Automatic. In: Keith Brown (Editor-in-Chief), Encyclopedia of Language & Linguistics, Second Edition: Elsevier Oxford, Volume 5, pp. 665–677.

[21] Cunningham, H.—Maynard, D.—Bontcheva, K.—Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02), Philadelphia.

[22] Kiryakov, A.—Popov, B.—Terziev, I.—Manov, D.—Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Elsevier's Journal of Web Semantics, Vol. 2, 2005, No. 1, `http://www.ontotext.com/kim/semanticannotation.html`.

[23] Cimiano, P.—Ladwig, G.—Staab, S.: Gimme' the Context: Context-Driven Automatic Semantic Annotation With C-Pankow. In WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA. ACM Press. ISBN 1-59593-046-9, 2005, pp. 332–341.

[24] Etzioni, O.—Cafarella, M.—Downey, D.—Kok, S.—Popescu, A.—Shaked, T.—Soderland, S.—Weld, D.—Yates, A.: Web-Scale Information Extraction in Knowitall (Preliminary Results). In WWW '04, 2004, pp. 100–110, `http://doi.acm.org/10.1145/988672.988687`.

[25] Laclavík, M.—Šeleng, M.—Ciglan, M.—Hluchý, L.: Ontea: Platform for Pattern Based Automated Semantic Annotation. Computing and Informatics, Vol. 28, 2009, pp. 555–579.

[26] Kvassay, M.—Laclavík, M.—Dlugolinský, Š.: Reconstructing Social Networks from Emails. In: Pokorný, J., SnáŠel, V., Richta, K. (Eds.): DATESO '10, 2010, pp. 50–59, ISBN 978-80-7378-116-3.

[27] Mehandjiev, N.—Gouvas, P.—Marin, C.: D5.3.1 Visual Mapping Tool (initial version). Commius project deliverable, `www.commius.eu`, 2009.

[28] Marin, C. A.—Carpenter, M.—Wajid, U.—Mehandjiev, N.: Devolved Ontology in Practice for a Seamless Semantic Alignment within Dynamic Collaboration Networks of SMEs. In Computing and Informatics, Vol. 30, 2011, No. 1, pp. 31–55.

[29] Burkhart, T.—Werth, D.—Loos, P.—Dorn, C.: A Flexible Approach Towards Self-Adapting Process Recommendations. In Computing and Informatics, Vol. 30, 2011, No. 1, pp. 89–111.

[30] Soboroff, I.—de Vries, A. P.—Craswell, N.: Overview of the TREC 2006 Enterprise Track. URL: `http://tinyurl.com/TrecEnt06`.

[31] Laclavík, M.—Šeleng, M.—Dlugolinský, Š.—Gatial, E.—Hluchý, L.: Tools for Email based Recommendation in Enterprise. In CENTERIS – Conference on ENTERprise Information Systems, Berlin: Springer, 2010, part I, p. 209–218. ISBN 978-3-642-16401-9. ISSN 1865-0929.

Michal Laclavík is a researcher at Institute of Informatics, Slovak Academy of Sciences. In 1999 he received his MSc degree in Computer Science and Physics. He received his Ph. D. degree in Applied Informatics with focus on Knowledge Oriented Technologies in 2006. He is the author and co-author of more than 100 publications with more than 100 citations, and participates in the Pellucid, K-Wf Grid and Commius European projects, as well as in several national projects. He has strong scientific and development expertise in email analysis, information extraction and information retrieval. He also gives lectures on Information Retrieval at Slovak University of Technology. Finally, he is one of the initiators of the Commius project focusing on email based interoperability for SMEs.

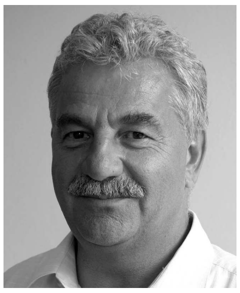
Štefan Dlugolinský is a researcher at the Institute of Informatics of Slovak Academy of Sciences. In 2009, he received his M. Eng. degree in information systems at the Faculty of Informatics and Information Technology, Slovak University of Technology. He is the author and co-author of several research papers and participates in European and national research projects. His research interests include email analysis and processing, information extraction and semantic annotation.

Martin Šeleng is a researcher at Institute of Informatics, Slovak Academy of Sciences. In 1999 he obtained his M. Sc. degree in Mathematics and Computer Sciences at the Faculty of Mathematics and Physics. He worked previously at Faculty of Economic and Informatics at the Economic University as a researcher and a teacher in the field of mathematics, statistics and computer science. He has strong expertise with email related system development and information system evaluation. He has been employed at the Institute since 2006. He is the author and co-author of several scientific papers and participates in the K-Wf Grid and Commius European projects and in several national projects. He teaches Information Retrieval at the Faculty of Informatics and Information Technologies. His research interests include email communication and large scale information processing.

Marcel Kvassay is a research assistant at the Institue of Informatics of Slovak Academy of Sciences. He graduated from the Faculty of Electrical Engineering of Slovak University of Technology in 1991. He spent a number of years as free-lance programmer and software development methodologist, working with a variety of programming languages, software development environments and methodologies. His research interests include social networks, graph theory, semantics, intelligent and distributed technologies, machine learning, knowledge management and representation.

Emil Gatial is researcher at the Institute of Informatics, Slovak Academy of Sciences. He received his M. Sc. degree in Computer Science in 2002. He received his Ph. D. degree in Applied Informatics in 2010. He is the author and co-author of several scientific papers. He participated in several European and national research projects. He has expertise in mobile agents, ontology and information processing.

Zoltán Balogh is a researcher at the Institute of Informatics of the Slovak Academy of Sciences. In 1999 he received his M. Sc. degree in Quantitative Management and Informatics. He received his Ph. D. degree in Applied Informatics in 2007. He is the author and co-author of several scientific papers and publications. He participates in several EU ICT international research projects (Pellucid, K-Wf Grid, Commius, Secricom) as well as in many national projects (VEGA, APVV, SPVV). His research focus is in ontology-based modeling, instance-based learning, web services and cloud computing and application of these technologies in real business systems.



Ladislav Hluchý is the Director of the Institute of Informatics of the Slovak Academy of Sciences and also the Head of the Department of Parallel and Distributed Computing at the institute. He received M. Sc. and Ph. D. degrees, both in Computer Science. He is R & D Project Manager, Work-package Leader in a number of 4FP, 5FP, 6FP and 7FP projects, as well as in Slovak R & D projects (VEGA, APVV). He is a member of IEEE. He is also (co-)author of scientific books and numerous scientific papers, contributions and invited lectures at international scientific conferences and workshops. He is also a supervisor and consultant for Ph. D., master and bachelor studies.