Tools for Email based Recommendation in Enterprise*

Michal Laclavík[1], Martin Šeleng[1], Štefan Dlugolinsky[1],
Emil Gatial[1], Ladislav Hluchý[1]

[1] Institute of Informatics, Slovak Academy of Sciences,
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
{laclavik.ui, martin.seleng, stefan.dlugolinsky, emil.gatial, hluchy.ui}@savba.sk

Abstract. Even in Web 2.0 era, email is still the most popular application on the internet. Beset by many problems, such as spam or information overload, yet it yields significant benefits especially to enterprise users when communicating, collaborating or solving business tasks. The email standards, content, services and clients improved a lot, but the integration with the environment and enterprise context remained pretty much the same. We believe that this can be improved by introducing our work in progress – the Acoma context-sensitive recommendation tool. Acoma processes emails on the server or desktop side and attaches the relevant information from various sources to the email messages. It can be used with any email client or mobile device since it is hooked up to email as a proxy to email protocols. In order to provide useful recommendations, emails need to be processed and business objects need to be identified. Thus the paper also discusses the object identification using the information extraction techniques based on the Ontea tool, as well as its customization in the enterprise context.

Keywords: email, information extraction, context, recommendation.

1 Introduction

According to recent surveys, information workers send and receive an average of 133 messages per day [1]. They talk about 'living' in email, spending an average of 21% of their time working with the email client. In 2001, information workers received just about 20 email messages per day and sent about 6 messages [2]. While the number of received messages is increasing, sending remains pretty much on the same level [3]. Findings from 2003 also show that 80% of users prefer email [4] for business communication.

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

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

contains can be managed effectively. One of the main drawbacks of email usage today is its insufficient integration into the collective environment. 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 [5] [6], including alerting, archiving, task management, collaboration and interoperability.

Efforts to link emails with knowledge or context-sensitive information have been attempted in several tools [7] 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, collaboration, etc.) [7]: Telenotes, ContactMap, TaskMaster, Snarf, Remail, Priorities or recent Semanta.

The main difference between Acoma and other attempts lies in its integration with the email protocols, which permits Acoma to work with any email client or mobile device without forcing the user to change his or her working practices. Acoma uses the community approach to adapt to new email types and new user activities related to email. It also provides customizable solution for the integration with enterprise environments. Deeper insights into the related work are provided in our previous analysis [7] of the existing approaches to improving or analyzing emails.

In this paper we discuss the vision and exploitation possibilities of the Acoma System, our enterprise recommender solution on top of email communication, and the current state of its implementation, which is a work in progress. The most important part of the tool is to identify business objects in text. Thus we also present the Ontea tool used for the pattern-based information extraction. We conducted several experiments in Small and Medium Enterprises (SMEs) regarding the tool customization for specific applications. Both Acoma[1] and Ontea[2] tools are developed as open source projects under *sourceforge.net*.

In chapter 2 we discuss the Acoma approach and its basic architecture along with subsystems for information extraction and hints recommendation. The chapter concludes with the ways how to adjust Acoma for specific applications. In chapter 3 we discuss the possible applications of our solution. We revisit the example email from chapter 2 and use it to demonstrate various components of the solution. The end of chapter 3 deals with the customization experiments in the applications that used real enterprise emails.

2 Approach and architecture

The ambition of Acoma 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 to 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. In this way the system can be used with any email client or even mobile device, without requiring

[1] http://acoma.sourceforge.net/, Last access: June 2010
[2] http://ontea.sourceforge.net/, Last access: June 2010

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.

Email communication passes through the Acoma system, where it is processed and additional information is added to the messages as HTML links or text attachments (see Figure 1). These additions contain relevant information and knowledge, hints or links to business resources such as document repositories, databases or information systems needed in the detected business context. The context is detected using the pattern-based information extraction [8] discussed in section 2.2.

When checking new emails, users may receive the modified (enriched) email as an attachment to the original unmodified message (applies mainly to mobile devices), or they may receive the original message as an attachment to the enriched HTML email with additional information as seen in Figure 1. In this way the users can configure Acoma according to their needs depending on their devices, email clients or personal preferences.

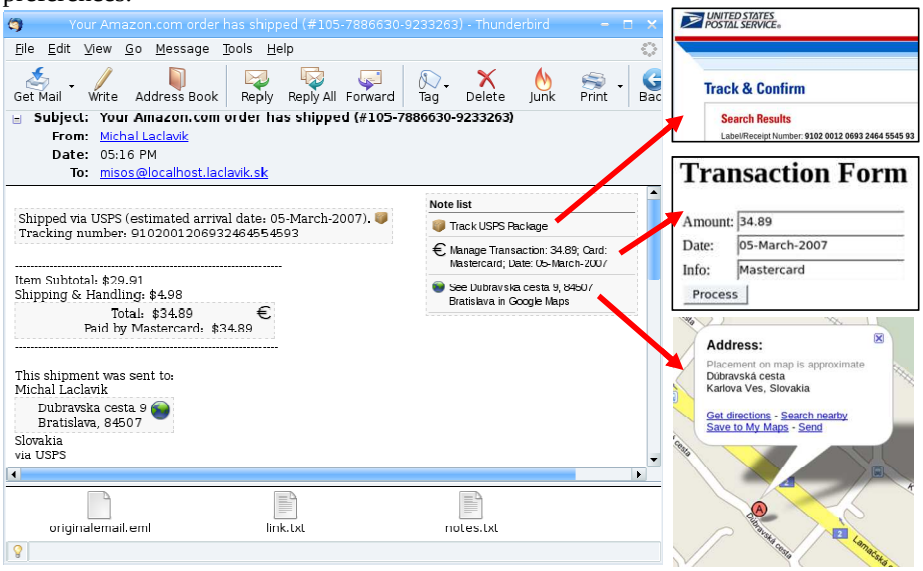

Fig. 1. Fragment of Amazon.com shipment confirmation email processed by Acoma.

Acoma can successfully process mainly formal, system-generated emails. Formal emails are also 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 notification. Examples are bank statements, hotel or air-ticket reservation confirmations, invoices to be paid, or shipped goods notifications. Such formal email with fixed structure is easier to analyze and provide context-relevant information. In addition, formal emails are often used in business tasks of SMEs and currently need to be processed manually, while Acoma is able to process them semi-automatically. However, experiments have shown that regular

person-to-person emails contain many structural parts and objects which can be detected using patterns or gazetteers (object lists).

2.1 Information Extraction

In order to provide useful recommendations, the most important task is to identify objects and object properties in the text and thus to formalize email message content and context. Processed email introduced in Figure 1 can be seen in Figure 2.

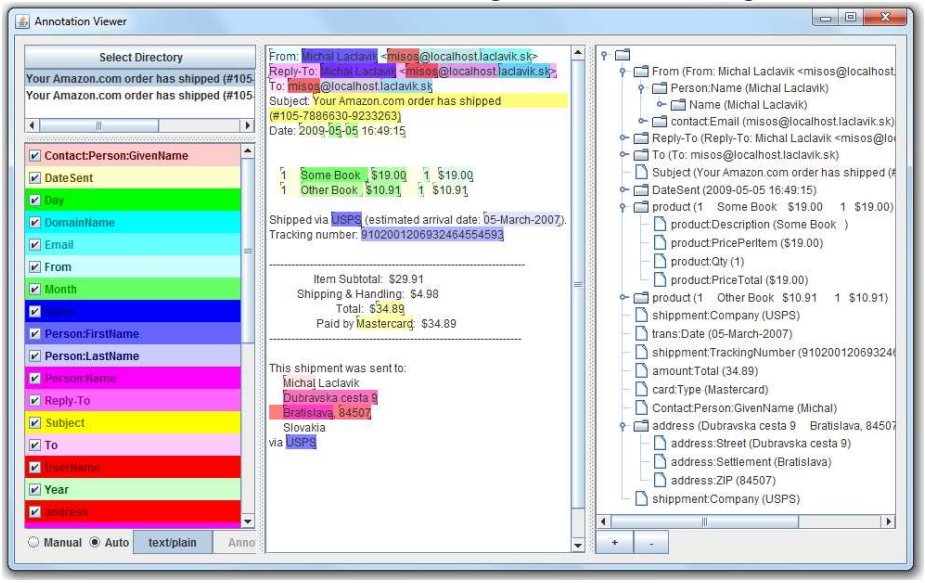

Fig. 2. Email message processed by the Ontea information extraction tool. Detected objects are highlighted. On the right, several objects such as address or product are grouped into a hierarchical tree.

As for object identification, we use the information extraction (IE) techniques. We have developed the Ontea [8] extraction and annotation tool, which uses regular expression patterns and gazetteers[3]. These patterns and gazetteers generate key-value pairs (object type – object value) such as those seen in Figure 2. Key-value pairs are then used in hint templates described in section 2.2.

While this is a quite simple approach, it is valuable for enterprise context, where business objects included in email communication differ from enterprise to enterprise. State of the art IE approaches such as C-PANKOW [9] or knowItAll [10] focus on general information domains such as web, and are not applicable to enterprise content. Tools such as GATE[4] [11] offer the framework for special JAPE patterns as well as other advanced Natural Language Processing (NLP) techniques, but require more knowledge for customization. Regular expression patterns can be created by

[3] Gazetteer is simply a list of keywords (e.g. credit card types or shipping companies' lists) representing an object type, which are matched against the text of emails.

[4] http://gate.ac.uk/, Last access: June 2010

many developers and do not require special knowledge of NLP techniques. NLP techniques such as those used in knowItAll, C-PANKOW or GATE are applicable mainly to English, while regular expressions and gazetteers are applicable to any language. We have tested Ontea on English, Spanish, Italian and Slovak emails. Our approach can actually integrate any IE tool, including NLP techniques, as long as it provides key-value pairs as output. We succeeded, for example, in integrating Ontea with GATE system and we use the Ontotext standalone gazetteer[5] originally developed for GATE.

The power of the Ontea approach is in its simplicity [8] compared to more advanced but heavy solutions such as GATE, as well as in its ability of transformation chaining and connection with other information systems (databases, documents, intranets, and internet). In addition, Ontea also supports email decomposition and understanding of email header, body and attachments. In chapter 3.2 the Ontea tool customization evaluation on emails in several SMEs is provided.

2.2 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 hints templates and if the required key-value pair (keys are in {} brackets) for a hint is found, the hint template is filled in with extracted information (value of key-value pair). The results are then passed 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 key-value pairs extracted from the email using the Ontea IE discussed in the previous section.

Table 1. List of hint templates used in the example shown in Figure 1. 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 generated hints.

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
Track USPS Package http://...smi.usps.com/...Inquiry.do?origTrackNum={shippment:TrackingNumber} shippment:TrackingNumber, shippment:Company:USPS product
Manage Transaction: {amount:Total}; Card: {card:Type}; Date: {trans:Date} http://.../trans?amount={amount:Total}&info={card:Type}&date={trans:Date} amount:Total, card:Type Money

[5] http://www.ontotext.com/downloads/index.html, Last access: June 2010

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.

Generated Hints
See Dubravska cesta 9, 84507 Bratislava in Google Maps http://maps.google.com/maps?q=Dubravska+cesta+9,+84507+Bratislava link,454,494
Track USPS Package http://...smi.usps.com/...Inquiry.do?origTrackNum=9102001206932464554593 product,13,98
Manage Transaction: 34.89; Card: Mastercard; Date: 05-March-2007 http://.../trans?amount=34.89&info=Mastercard&date=05-March-2007 money,280,321

Currently, the hint templates have to be defined and edited in the configuration file, but we are working on the user interface, where users will be able to define new hint templates suitable for the tasks they need to perform with the email.

2.3 Adaptability

Acoma is implemented using the OSGi[6] approach, which is a standard for modular systems. We are using the Apache Felix[7] implementation of OSGi. Such a solution enables us to adapt the solution even when it does not have the configuration user interfaces for hints. We believe that such a recommendation 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 1). Other module examples include package tracking, event detection and adding events to calendar 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; Customer card (CRM); Geo module; Voting module; Generic hint module. In addition, partner search module, attachment manager and mail template module are being developed in scope of Commius[8] project.

Voting and Geo modules are implemented as prototypes. For the rest of the modules we have tested IE on emails of one or more enterprises. For the purpose of this paper, we were able to create simple modules (or hint templates presented in chapter 2.2, which will be covered by generic hint module in future) like tracking

[6] http://www.osgi.org/, Last access: June 2010
[7] http://felix.apache.org/, Last access: June 2010
[8] http://www.commius.eu/, Last access: June 2010

number or transaction detection showed in Figure 1 in less than one hour. When all the needed objects are detected, the hint action can be defined by URL to existing web-based system. Functionality for dozens of hints can be created and implemented in a few hours. Customization of information extraction is more difficult and we discuss it in section 3.3.

3 Applications and Experiments

We believe the Acoma approach can be used in many applications in enterprise or community environment, such as Knowledge Management, Social Networks, Information Management or Enterprise Interoperability. In this chapter we first discuss the example used in this paper and then review the information extraction experiments.

3.1 Example of Use

An sample email processed by the Acoma system can be seen in Figure 1. The example is based on a modified email from Amazon.com (we removed the private info and fitted the text to the screen). Acoma replaced the original plain text message by an HTML message including the recommendation hints with links, both in the text where it belongs, and on the right side of the message. (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 tracking of a shipped package, entering a transaction in the enterprise information system, or showing the address on the map can be taken by the user via clicking on the recommended hint represented by a box with icon.

The modified email also contains the link that will display Graphical User Interface (GUI) related to the message in the browser. The GUI contains similar or extended information related to hints or modules and it provides the dynamic and interactive functionality which cannot be achieved by the static HTML included in the email. It can also integrate the legacy system where the web GUI is not available. GUI offers extensive functionality but is not a focus of this paper.

Depending on the settings and the email client, Acoma can modify email messages to include the text, html attachments or just link to the message GUI. In Figure 2, the same Amazon email message is shown within the Ontea IE tool. It took us about one hour to configure/customize Acoma and Ontea tools for the Amazon email shown in Figures 1 and 2. In the next section we discuss the customization of information extraction in real enterprise applications.

3.2 Information Extraction Customization and Experiments

In this section we describe how the proposed approach for IE can be customized for specific enterprise or application. We have conducted several experiments within

the Commius[9] and AIIA[10] projects, where the Acoma and Ontea tools are being developed.

The AIIA project focuses on two applications: *Anasoft helpdesk*, where products, modules and components are identified and appropriate contact within the organization is detected to deal with the email. Also, the identification of contact details and customer is needed in order to show 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, where 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 orders or invoices communicated via email.

In the first experiment in Anasoft, we have annotated Helpdesk emails. Manual annotations were created by the Ontea tool, which supports this kind of task. Tags were created by the authors of this article as well as real Helpdesk workers. 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[11] interoperability standard. Results are available in the Commius deliverable [12]. 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. Defining 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 tags.
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 2).

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:

[9] http://www.commius.eu/, Last access: June 2010
[10] http://aiia.ui.sav.sk/, Last access: June 2010
[11] http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf, Last access: June 2010

- *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
- *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) available. Gazetteers need to be created manually or extracted from company information systems if available. It is hard to estimate how much effort we have spent customizing the Anasoft Helpdesk application since the customization process was not defined at that point in time, but in the second round of experiments (where the customization process was already defined and followed) the first step took us about 2 hours and the second step together with the 7th 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 needs 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 7th 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 only in Slovak language.

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 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.

To conclude, the success rate (precision and recall) of the Ontea IE is quite high [8] since the developer fine-tunes the patterns and gazetteers on the target email data set. We have also proved that the approach can be customized for enterprise applications in a reasonable time-frame of several hours.

4 Conclusion

The first version of the Acoma system was tested in an administrative application [13] and earned positive feedback by not forcing people to change their way of working. In [13] we identified potential target groups and suggestions for further

improvement of Acoma (mainly in the area of user-friendliness) – which are now partially implemented. In the paper we presented an updated version of Acoma (still under development), which is suitable mainly for processing the system-generated emails with partially formal structure, where objects can be discovered using pattern-based detection. We have also discussed experiments with the customization of the information extraction in several enterprises and organizations. We have shown that the solution can be customized for specific enterprise applications within a reasonable time-frame of a few hours. In future we plan to extend the solution with an easy user interface for setting up the application-specific patterns for information extraction, hint creation and evaluation.

References

1. HP, The Radicati Group, Inc.: Taming the Growth of Email – An ROI Analysis (White Paper), http://www.radicati.com/wp/wp-content/uploads/2008/09/hp_whitepaper.pdf, 2005
2. Jeffrey Jones, Gallup: Almost All E-Mail Users Say Internet, E-Mail Have Made Lives Better, http://www.gallup.com/poll/4711/Almost-All-EMail-Users-Say-Internet-EMail-Made-Lives-Better.aspx, 2001
3. A. Lantz; Does the Use of E-Mail Change Over Time?; International Journal of Human-Computer Interaction, Volume 15, Issue 3 June 2003 , pages 419 – 431
4. META Group Inc.: 80% of Users Prefer E-Mail as Business Communication Tool http://www.mariosalexandrou.com/technology-trends/2003/80-percent-of-users-prefer-email.asp, 2003
5. S. Whittaker, C. Sidner: Email Overload: Exploring Personal Information Management of Email. In Proceedings of ACM CHI'96, 276-283, 1996
6. D. Fisher, A.J. Brush, E. Gleave & M.A. Smith: Revisiting Whittaker & Sidner's "email overload" ten years later. In CSCW2006, New York ACM Press, 2006
7. Michal Laclavík, Diana Maynard: Motivating intelligent email in business: an investigation into current trends for email processing and communication research; In Workshop on Emails in e-Commerce and Enterprise Context (E3C); IEEE Conference on Commerce and Enterprise Computing; DOI 10.1109/CEC.2009.47; pp. 476-482, 2009
8. Michal Laclavik, Martin Seleng, Marek Ciglan, Ladislav Hluchy: Ontea: Platform for Pattern based Automated Semantic Annotation; In Computing and Informatics, Vol. 28, 2009, 555–579
9. Cimiano P., Ladwig G., Staab S.: Gimme' the Context: Context-Driven Automatic Semantic Annotation With C-Pankow. In WWW'05, New York, NY, USA. ACM Press, 2005, ISBN 1-59593-046-9, pp. 332–341.
10. 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, 100-110, http://doi.acm.org/10.1145/988672.988687
11. Cunningham H., Maynard D., Bontcheva K., Tablan V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of ACL'02, Philadelphia, 2002.
12. Nikolay Mehandjiev, Panagiotis Gouvas, Cesar Marin: D5.3.1 Visual Mapping Tool (initial version); Commius project deliverable, www.commius.eu, (2009)
13. Michal Laclavik, Martn Seleng, Ladislav Hluchy: ACoMA: Network Enterprise Interoperability and Collaboration using E-mail Communication; In Proceedings of eChallenges 2007; IOS Press, 2007 Amsterdam ISBN 978-1-58603-801-4, p 1078-1085