Lightweight Semantic Approach for Enterprise Search and Interoperability

Michal Laclavík[1], Štefan Dlugolinský[1], Martin Šeleng[1], Marek Ciglan[1], Martin Tomašek[2], Marcel Kvassay[1], Ladislav Hluchý[1]

[1]Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9, 845 07 Bratislava, Slovakia
laclavik.ui@savba.sk
[2]InterSoft, a.s., Floriánska 19, 040 01 Košice, Slovakia
martin.tomasek@intersoft.sk

Abstract. In this paper we describe a lightweight approach for semantic interoperability suitable for small and micro enterprises. The approach is based on reusing the existing enterprise infrastructure – emails and documents – and enables lightweight semantic search and recommendation in order to fulfill interoperability tasks.

Keywords: interoperability, small enterprises, lightweight semantics, email.

1 Introduction

Semantic interoperability is challenging especially for small and micro enterprises. There is a variety of formats for data and document exchange depending on the country, business sector, used IT technology and software. Standards, such as EDI[1], Core Components, ebXML are established but are seldom used by SMEs and micro enterprises mainly due to their complexity. We try to address this in the VENIS[2] project, which aims at providing a new level of interoperability between large and small enterprises based on *Virtual Enterprise* paradigm. One of the sub-goals is achieving semantic interoperability through a lightweight semantic approach inspired by Web 2.0 applications. The project will also benefit from well-established web and email technologies, their widespread use and the involvement of users in the loop.

Semantic interoperability can also be addressed by standards from semantic web like RDF(S) or OWL. While the existing semantic technologies bring in interesting possibilities, there are many unresolved problems which prevent semantic technologies from building on OWL standards and logical inference from being widely used. These problems include the exponential complexity of inference techniques on rich models; unavailability of exhaustive semantic description of the problem area; and the problem of contradictory knowledge.

[1] http://en.wikipedia.org/wiki/Electronic_data_interchange
[2] http://www.venis-project.eu/

In contrast to rich logic-based semantics, mass collaboration sites such as Wikipedia, Twitter, Delicious, YouTube or Facebook create **lightweight semantics** in the form of tags and annotations and serve as examples of networks of interconnected people and entities. On these sites, we can see that the lightweight vision of semantic web can work and it can also be useful in Enterprise 2.0 to achieve the desired interoperability and build the Semantic Enterprise.

Such lightweight semantics may be more appreciated by the end users, since it is easier to manage and **can deal with contradictory knowledge,** as well as **provide scalable inference** based on graph algorithms. Since lightweight semantic networks share the same mathematical properties (small-world properties) with other complex networks, for example social networks and the internet, we can exploit a wide range of graph mining methods developed in recent years to analyze the semantic networks created by the users, as well as a number of interoperability activities or semi-automatic IE (Information Extraction).

In VENIS, the focus of semantic interoperability is to allow the users to create, manage, modify and share simple metadata in the form of tags and annotations. VENIS will also monitor and store the interoperability activities, events which are related to the shared and interoperating data, tasks and semantics. This will enable users to build **semantic networks**, which **can then be used for search, recommendation or business process management**. We will focus on sharing and reusing tags and thus achieve **semantic interoperability driven by the user's needs**.

2 Lightweight semantic approach

VENIS is addressing semantic interoperability by applying lightweight semantics based on tags or annotations. Users or enterprises will be able to attach tags or annotations to their data and documents either manually or automatically via annotation API using IE methods. We will deliver an automatic solution which can learn from user annotations and user interaction. Tags or annotations can either be hidden or shared to interoperating parties. Large enterprises can have rich semantics inside their repositories, but share with others only selected valuable meta-data that can be viewed, updated and reused by the interoperating SMEs. Semantic tags and annotations are stored and used in the form of semantic networks of interconnected repository items, entities, tags, annotations, users and events, which can be used for recommendation of resources, information, knowledge and business activities. Thus the **user-driven lightweight semantic** will serve as a basis for a **simplified semantic interoperability between large and small enterprises**.

We build on the email-based interoperability achieved in the Commius project [3], where the extracted semantics autonomously detected entities like people, addresses, contact data and organizations, although the extraction of other business objects (products, services or invoice numbers) had to be customized with the help of the developers. Customization was fast [3], but necessary. In VENIS we plan to involve the user in the loop to interact with the semantics. In Commius, the extracted semantics in the form of key-value pairs and semantic trees was further converted into the standard Core Component XML, which was often incomplete. Since SMEs rarely use this standard, we had to map Core Component XML back into the concepts of a specific enterprise vocabulary [4]. We believe that a simpler approach based on lightweight semantics and user-in-the-loop needs to be used where the interoperating parties can more easily share, adapt, modify and exploit such semantics.

As VENIS is primarily focused on the handling of documents in a free text format, we will provide an API for automatic annotation and enable plugging in the existing IE tools like GATE[3], Stanford NER[4] or Ontea[5] [3]. VENIS enables automatic processing by creating an API, into which various annotation tools can be plugged, but we will also include simpler approaches based on gazetteers and user interactions, which can be handled in SMEs without the technical skills needed for customization of advanced IE tools. We shall permit the creation of gazetteers (list of objects of a specific type) defined by users. Moreover, if a user annotates some text string as a product, this string can later be recognized as a product by automatic annotation too.

2.1 Rule based Named Entity Recognition

IE techniques [1] usually focus on five main tasks of IE as defined by the Message Understanding Conferences (MUC): *1. Named entity recognition* (NE) - Finds and classifies the names, places, etc.; *2. Coreference resolution* (CO) – Finds aliases and pronouns referencing the same entity and discovers the identity relations between entities*; 3. Template element construction* (TE) – Finds properties or attributes of entities and adds descriptive information to NE results (using CO); *4. Template relation construction* (TR) - Finds relations between NE entities. *5. Scenario template production* (ST) – Finds events involving entities and fits TE and TR results into specified event scenarios.

In [3] we described in detail the state of the art in information extraction and advantages of pattern-based information extraction, thus we will not address it here. We assume that IE techniques are in place and provide useful Named entity recognition results (the first IE task). We assume the results are based on key – value pairs representing NEs. The work presented in this paper helps in the relation discovery among entities and thus mainly solves the forth IE task focused on relations – TR. Anyhow, since we do not distinguish between NE or TE and all entities are treated as key–value pairs, the results of the relation discovery are always entities related to one or several entities. This means the discovered entities represent either relations (TR), entity aliases (CO) or entity properties (TE).

For IE we use Ontea IE techniques [3] developed in the Commius project, but any other IE tool providing key-value pairs with their positions in the text can be used. Ontea is based on regular expressions and gazetteers, through which the key-value pairs (object type – object value) are extracted from the source text. Key-value pairs form a semantic tree of the annotations, and trees from multiple documents form a network of entities, i.e. a graph structure. Detailed examples can be found in our previous work [2, 3]. The Ontea IE tool is able to connect to other extraction/annotation tools like GATE[6], Stanford CoreNLP[7] or WM Wikifier[8]. In the

[3] http://gate.ac.uk/
[4] http://nlp.stanford.edu/ner/index.shtml
[5] http://ontea.sourceforge.net/
[6] http://gate.ac.uk/
[7] http://nlp.stanford.edu/software/corenlp.shtml#About
[8] http://www.nzdl.org/wikification/

experiment presented in chapter 4 we have used pure gazetteers and regular expressions with some extra rules to support better user interaction.

2.2 Semantic Trees and Graphs

Annotations (key-value pairs) representing entities form trees and graphs/networks which can be built from any text collection. A document is then represented by a document node, its paragraph and sentences nodes, and such documents are interconnected by the nodes representing the entities present in multiple documents. Entities found in multiple documents occur only once in the graph (they are unique) and they connect by edges to their respective sentences, paragraphs and documents. In the future we would also like to add activity graph to this structure, e.g. when and by whom the document was updated, downloaded, sent, re-shared. The underlying idea is that any other internal or external data can be included, and the semantic inference and relation discovery will work on such networks. For the relation discovery we are using spreading activation algorithm described in [2], with one example of enterprise search also described in chapter 4 below. We manipulate these graphs/networks data using graph APIs and databases compatible with Blueprints[9] like Neo4j[10] or SGDB[11].

2.3 Email parsing

The implemented prototype is able to access several kinds of remote and local email collections. It uses Java Mail API[12] to access email collections by IMAP, SMTP and POP3 protocols. The prototype can also access local mbox and PST collections. PST collections are read using the java-libpst library[13].

A connected email collection is parsed email by email and each email is parsed separately. There are three types of output from the email parser: *envelope*, *body text* and *attachment* (explained below). Emails are parsed with the help of Apache Tika[14], which can recognize several file formats and parse file metadata as well as its textual content. Each of the parsed email parts consists of metadata and textual content provided by Tika except the *envelope* part, which consists only of metadata built from the message headers. Attachments are parsed recursively, because some types (e.g. zip archives) can contain other files. If such files are discovered during the recursive attachment parsing, they are treated just like they were a regular attachment, i.e. they are parsed as *attachment* part with metadata and textual content.

Additional lightweight semantic information is put into the *message* and *attachment* part metadata by Ontea, which extracts this information from the related textual content. After the metadata is enhanced by IE, the email represented by the parsed parts is ready to be indexed.

[9] https://github.com/tinkerpop/blueprints/wiki/
[10] http://neo4j.org/
[11] http://ups.savba.sk/~marek/sgdb.html
[12] http://www.oracle.com/technetwork/java/javamail/index.html
[13] https://github.com/rjohnsondev/java-libpst
[14] http://tika.apache.org/

2.4 Email indexing

We implemented a special indexer in our prototype, which builds indexing requests for Solr[15]. For each email being indexed, the indexer gets as input one *envelope* parsed part, one or more *body text* parsed parts and zero or more *attachment* parts. Parsed results are not directly added to the index; they are pre-processed first. There are two document types, which are put into the Solr index. The first is *message* and the second is *attachment*. The *message* document is built from the *envelope* parsed part and all the parsed *body text* parts. The metadata of the *envelope* and *body text* parts is merged into one and so all the *body text* parts textual content is concatenated. The *attachment* document is built from the *attachment* parsed parts by enhancing their metadata by *envelope* metadata.

A filter is applied on metadata, to filter and rename particular metadata being added into the *message* and *attachment* Solr documents as index fields. It is because not all the metadata fields are required in the index.

Textual content is put into the index in a special field *text_general*, which is later processed in Solr by standard tokenizer, stop word filter and lower case filter. When the *message* and *attachment* documents are ready, they are sent to Solr and indexed.

3 Use Case

The InterSoft use case is related to the allocation of development resources to various providers and customers at the same time. Customers usually ask large providers of software solutions to fulfill their complex projects. Since the providers often do not have all the required development resources available, they need to find suitable subcontractors and involve them in the collaboration so as to complete the project for the customer.

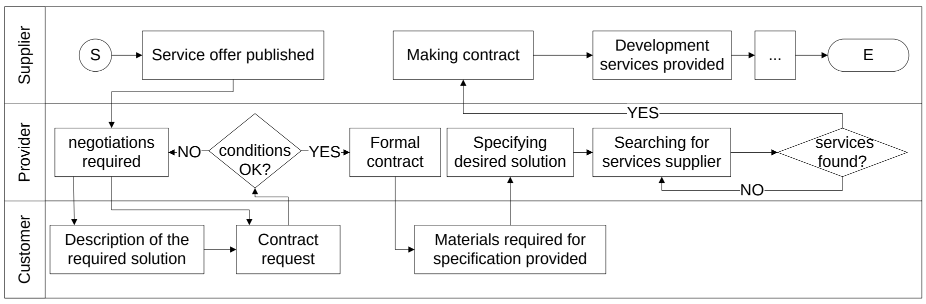

Supplier
Provider
Customer
S
Service offer published
Making contract
Development services provided
...
E
YES
negotiations required
NO
conditions OK?
YES
Formal contract
Specifying desired solution
Searching for services supplier
services found?
NO
Description of the required solution
Contract request
Materials required for specification provided

Figure 1: Diagram of InterSoft use case

In such collaboration process there is currently no information system used either by InterSoft or by the large providers subcontracting to InterSoft. E-mail communication is the only method to manage the collaboration, exchange and sharing of the electronic materials (documents, software) related to the outsourced projects.

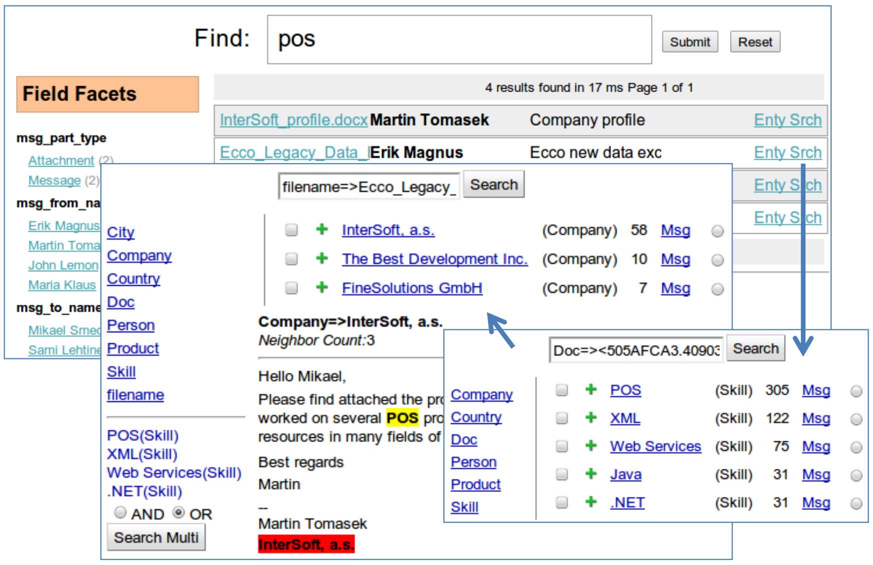
[15] http://lucene.apache.org/solr/

This use case identifies three actors or roles: *Supplier* – SME which offers a portfolio of simple services; *Customer* – requires a complex service and in general looks for a large Provider of such services; *Provider* – LE/SME which provides a complex service and needs a Supplier for simple services to build the complex service for a Customer.

The usual process is described in Figure 1. It involves activities such as matching requirements and specifications to the profiles of involved parties, contacting the involved parties, and can be extended further to formal ordering of services, invoicing and payments. We showcase a few interoperability activities in chapter 4.

4 Early Enterprise Search Prototype

In this section we discuss two innovative semantic interoperability features in VENIS: (1) Enterprise Search; (2) search and recommendation for relevant semantics in concrete interoperability tasks.



4 results found in 17 ms Page 1 of 1

InterSoft_profile.docx **Martin Tomasek** Company profile Enty Srch
Ecco_Legacy_Data_ **Erik Magnus** Ecco new data exc Enty Srch
Enty Srch
Enty Srch

InterSoft, a.s.	(Company)	58	Msg
The Best Development Inc.	(Company)	10	Msg
FineSolutions GmbH	(Company)	7	Msg

Company=>InterSoft, a.s.
Neighbor Count:3

Hello Mikael,
Please find attached the pr
worked on several **POS** pro
resources in many fields of
Best regards
Martin
–
Martin Tomasek
InterSoft, a.s.

POS	(Skill)	305	Msg
XML	(Skill)	122	Msg
Web Services	(Skill)	75	Msg
Java	(Skill)	31	Msg
.NET	(Skill)	31	Msg

Figure 2: Prototype Enterprise search user interface

In Figure 2 we can see an early implementation of search functionality. There is a search field on the top of the screen, which can be used to search for documents or emails using full-text search. Search results can be filtered by clicking on facets on the left. Several facets are predefined like *msg_part_type*. *msg_from_name*, *msg_to_name*, *msg_cc_name*, *msg_bcc_name* and *filename*. The *msg_part_type* can be either *message* or *attachment.*

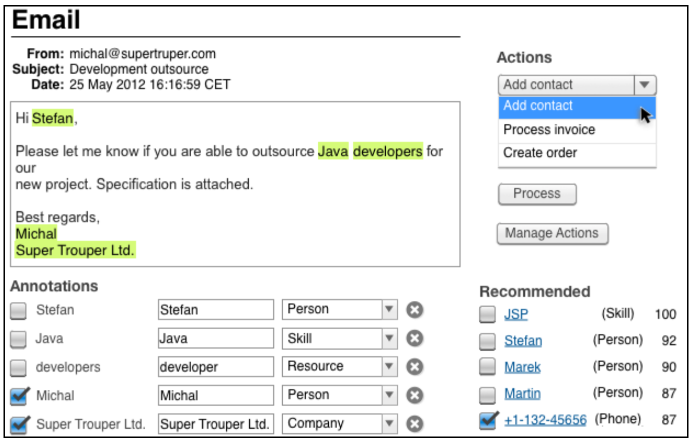
Full-text search is integrated with an entity search, where the related entities are displayed by clicking on *Enty Srch* link. Both full-text and enterprise entity searches should be even more tightly integrated later, but for now we keep two separate user interfaces for the full-text and the entity search (each with different facets).

After clicking on a document listed in the search results, the user can access the document or do any other operation with it like share it, add semantic tags and/or share them or even invoke the user interface that exploits the lightweight semantics for system-to-system interoperability with the user in the loop (see section 4.1). The email interface (Figure 3) will show email semantics, i.e. the detected entities and recommended entities for the email with the possibility to click on a desired entity (tag, person, invoice ID, customer or product) and perform entity search (as seen in Figure 2 in the two front windows).

The entity search and the recommendation have features similar to those used in the Email Social Network Search[16] developed in the Commius[17] project, which is being further extended in VENIS [2]. The entity search interface is displayed in Figure 2 in the two front windows. The front most shows the skills detected in the development requirement email; the one behind it the list of companies relevant for the skills. The idea is to return the relevant entities for one or more selected elements (i.e. context) and thus to deliver needed information for the interoperability task.

4.1 Proposals for legacy system integration

As we can see in Figure 3, the email user interface should show the email with the detected context displayed in the email text. The context is also displayed as a set of items which can be modified.



From: michal@supertruper.com
Subject: Development outsource
Date: 25 May 2012 16:16:59 CET

Hi Stefan,

Please let me know if you are able to outsource Java developers for
our
new project. Specification is attached.

Best regards,
Michal
Super Trouper Ltd.

☐	Stefan	Stefan	Person
☐	Java	Java	Skill
☐	developers	developer	Resource
✓	Michal	Michal	Person
✓	Super Trouper Ltd.	Super Trouper Ltd.	Company

☐	JSP	(Skill)	100
☐	Stefan	(Person)	92
☐	Marek	(Person)	90
☐	Martin	(Person)	87
✓	+1-132-45656	(Phone)	87

Figure 3: Mockup of user interface with email and context recommendation

Based on the context, VENIS will deliver relevant recommendations. Recommendations can contain information, inferred knowledge or process and activity information relevant for the email and the business activity covered by the email. The recommendation will be provided by the same engine as the already implemented entity search (Figure 2 front screens), i.e. it will reuse the underlying lightweight semantics in the form of a graph, but instead of user query, the context is detected directly from the email and its attachments. Based on this, other relevant entities (next activity, related people, products, services, or skills) will be

[16] http://ikt.ui.sav.sk/esns/
[17] http://www.commius.eu/

recommended in a way similar to the multiple entity search showed in Figure 2. Additional templates for recommendation can be created and/or adjusted even at a later stage. For example, if a request for outsourcing developers is detected, the system should restrict its recommendation only to people and skills, and omit irrelevant data.

Detected or recommended entities can be selected (as seen in Figure 3) and further processed. In this way, a system-to-system interoperability activity can be initiated by the user. For example, contact information can be stored to a database or an invoice can be submitted to and processed by a legacy system.

Users can also exploit recommendations for further entity searches. For example, if a user wants to see the skills or contact details of developer *Stefan*, he/she can click on the recommended item (Person *Stefan* in this case) and the relevant information will be displayed.

5 Conclusion and Future Work

This paper summarizes our work in progress on semantic interoperability using lightweight semantic networks. We base our approach on earlier achievements in the Commius project as well as on entity relation discovery in unstructured text.

We have provided a proof of concept implementation of the enterprise search, which exploits the lightweight semantic graph and can help with interoperability tasks communicated by email and shared documents. This was also tested on the concrete use case of the VENIS project provided by one of small enterprises – InterSoft.

We have also proposed a way to reuse the lightweight semantic networks for system-to-system interoperability with users in the loop. In the future we will try to implement the remaining features focused mainly on rich user interactions with the lightweight semantics, enabling features such as sharing of semantics, adjusting, deleting or annotating tasks. This will have an impact on better recommendation and better semantic processing of unstructured textual data involved in interoperability.

Acknowledgments: This work is supported by projects VENIS FP7-284984, CLAN APVV-0809-11 and VEGA 2/0184/10.

References

1. Cunningham, H. (2006), Information Extraction, Automatic. In: Encyclopedia of Language & Linguistics, Second Edition, volume 5, pp. 665-677. Oxford: Elsevier
2. Michal Laclavík, Marek Ciglan, Štefan Dlugolinský, Martin Šeleng, Ladislav Hluchý: Emails as Graph: Relation Discovery in Email Archive. In Email2012 workshop, WWW 2012, April 16–20, 2012, Lyon, France, pages 841-846, 2012
3. M. Laclavik, S. Dlugolinsky, M. Seleng, M. Kvassay, E. Gatial, Z. Balogh, L. Hluchy: Email Analysis and Information Extraction for Enterprise Benefit. In Computing and informatics, 2011, vol. 30, no. 1, p. 57-87. ISSN 1335-9150
4. 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