interTALENT UNIDEB

interTALENT
UNIDEB

Szerkesztő:
Mándy Zsuzsanna

Tördelés és grafika:
Meszesán László

ISBN 978-963-473-953-1

Kiadó:
Debreceni Egyetem

TIBOR ROSKÓ

University of Debrecen

Supervisor: Dr. Attila Adamkó

ELECTRA SIGNATURE: ÜGYFÉLKAPU, ADOBE XMP BASED DOCUMENT SIGNING SERVICE

Introduction

I am Tibor Roskó, a Computer Scientist (MSc), from the University of Debrecen. My research topic is designing, implementing new possible usages of metadata based on semantic Web and using a graph database if it is more suitable for data structure, data storing.

I have many projects with my supervisor, Dr. Attila Adamkó, these focus on the possibilities of semantic Web, and some of them are finished with useful results. We would like to build up a common infrastructure that these products are able to communicate with each other and other semantic Web applications using the semantic Web. I would like to highlight and present two main projects in my InterTalent paper: FOAF (Friend Of A Friend) using as a digital business card and naturally, Electra Signature.

I talked about a possible usage of Electra Signature at the conference and made an interactive presentation, the audience also could try out its demo. Unfortunately, the time limit was too short, but I tried to talk about information is needed to everybody could draw a complete figure about the infrastructure and services of Electra. There was given an opportunity to describe our talks in the abstract of the conference, so I could extend my presentation in this. I drew up some information about FOAF and the workflows of Electra, so the audience could meet my future plans before the talk. It was a big effort because I had enough time to talk about the important things, like unique values, advantages of Ügyfélkapu and naturally, present the demo.

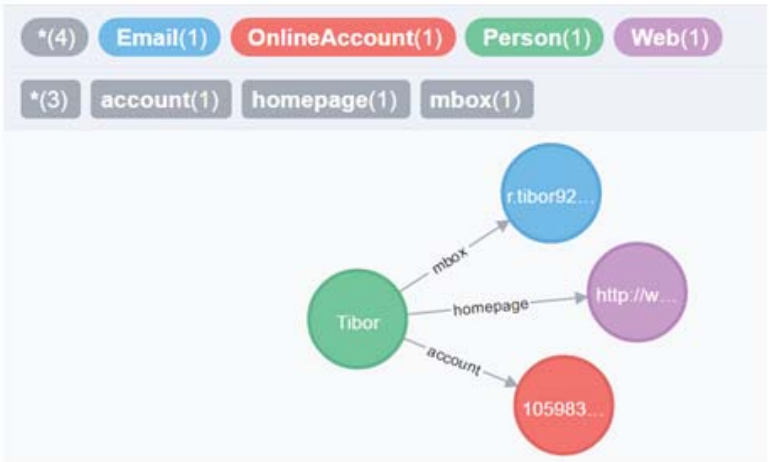
I am sure, conferences are always great chances to talk about our results, knowledge because it can be more understandable instead of a written paper, the audience can ask immediately if they do not understand anything. Personally, I like international events because I can share my knowledge with not only Hungarians but many foreign people, like at this conference.

I am especially happy about I can create this paper because who could not come to the InterTalent, also can meet my results.

Semantic Web

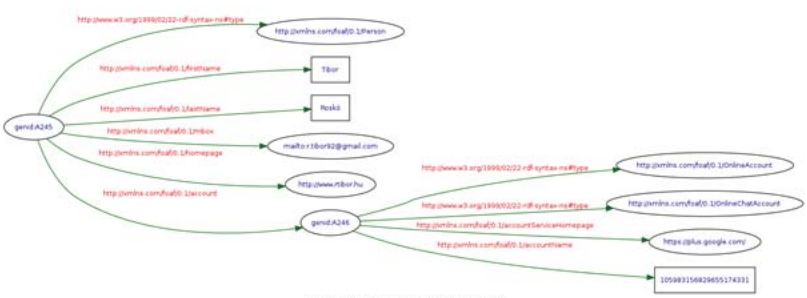
This is the future of the Web because it focuses on data instead of documents. The classic Web is the Web of the documents, but nowadays it is not enough to store docs on the Web or in the cloud because there is too much information in order to be able to process them without computers. Add semantics to the data to enable computers to understand the information behind the data. For example, computer can recognize the content of a picture, there is a cat in a pic, it can define there is a vertebrate animal in the picture. This is a very basic example, what can be solved with semantics. In the first step, common formats need to be used for integration and combination of data and common languages to define them such as RDF (Resource Description Framework), OWL (Web Ontology Language), FOAF and graph databases.

If we want to understand why it is so great, we have to see behind the things. Inspect the Web from the side of IoT (Internet of Things), and we can realize how many documents, records are created every day. There is an estimation, says about 50 000 GB/second data will be created in 2018, this is a very huge number, so computers must be used almost for all methods to be able to manage huge data. RDF offers chances to link the cognate information, like user and its contacts or publications. It has a relative simple format to implement this connection, it came from the human grammar: object – has – subject. It builds up a graph from these relations, it is known that graphs are more suitable to manage reusable information, like contacts instead of relational databases. I will talk about these advantages in the section of Neo4j. Now turn back to the world of semantics for a minute. I mentioned the OWL, this is also a basis of semantic Web. Vocabularies can be defined to describe a special context, its specific knowledge, for example IT, medicine or just our life. In the first sentences, I talked about how a computer can realize what is in a picture, it uses similar vocabularies to answer this question. So, we can see everything is related to everything in this world, I am sure it is not a problem because it goes similarly in the real life. I would like to illustrate this process with a figure about an RDF graph.



2: Neo4j graph (by Author)

Continue with the Neo4j graph database solution, it is the gate of my FOAF project with the previously described semantic Web infrastructures. This is an open source database with community and enterprise editions. Enterprise edition offers more advanced services such as scalability, clustering and more advanced security and backup management. Security management is available from the 3.1 version of Enterprise edition, we can add for example roles to our users, like admin or reader of a database. However, its real advantage is the compatibility with semantic Web infrastructures. As I mentioned the most important point is the IoT in the semantics because the number of these sensors is exponentially growing. It causes problems in mostly the sensors data storing, it is easy to store, but if you would like to use them, have to save in an easy processable format, for example in a graph. Thanks to its schema independent structure, NULL values do not have to be compulsively registered, so data mining can be easier, it may use fewer resources. I also want to present a figure about it, so you can compare it to the previous RDF figure, they are so similar, almost equal.



1: RDF graph (by Author)

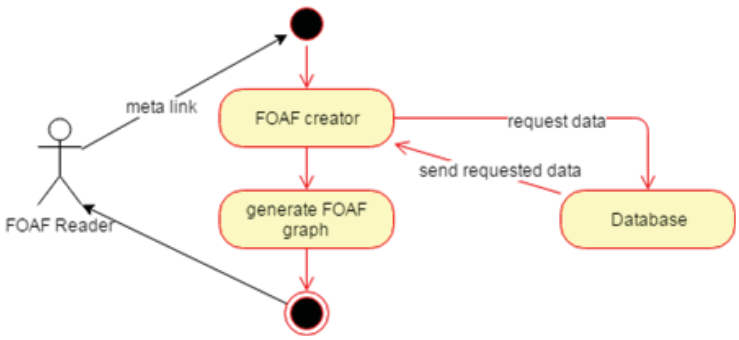
And now, let's talk about the FOAF. This is a machine-readable ontology describing people and their relations to objects, subjects. It is based on RDF and OWL, so we can create semantic Web relation graphs are more advanced processable by computers, applications such as CRM (Customer Relationship Management), and ERP (Enterprise Resource Planning) systems. As we can see it is designed for creating primarily personal info sets, this is

absolutely useful because the Web is built up from mostly personal contents. Nowadays, correctly connecting to these contents is very important to be able to support the chance of data mining, getting information or easier processing methods. The future of data extractions will be stated by correctly built basics. Unfortunately, there was not thought in the close past, how big this amount of records will be, nowadays it is realized, and this amount is growing extremely by the big number of sensors. There was also recognized, this amount of records is not processable manually, so it is needed to use algorithms. I designed a simple method with my supervisor to share personal information in a semantic Web format. The goal, these info sets can be processed by applications automatically, without human actions. This algorithm supports the cooperation between already existing data warehouses because it uses them instead of creating a new database from these data. The final goal is reaching already existing databases work together to avoid data duplication and other anomalies in storing and managing processes. So, here is my absolute simple method to take the first step:

1. Read necessary information from existed databases.
2. Build up the FOAF graph from them.
3. Send this graph to the client.

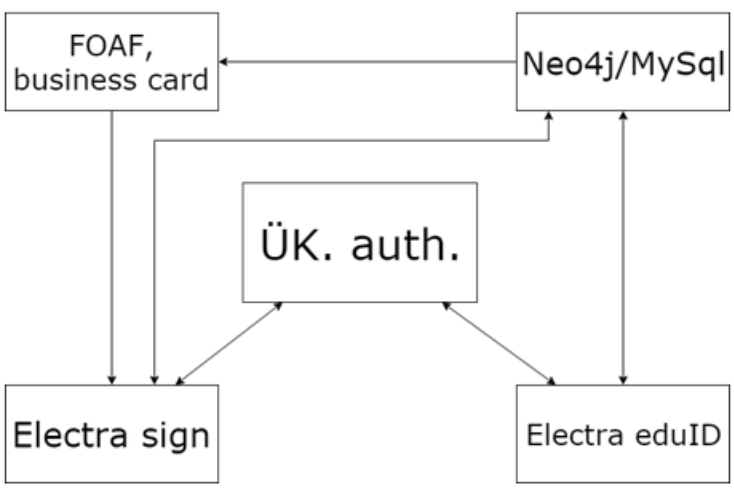
This is an abstract, to use, implement it in a Web programming language, like PHP, Java. It is better than you create manually FOAF because the method always generates an up-to-date graph from databases. In this algorithm, there will not be implemented new databases and the number of already implemented databases will be reduced because various info sets can be built up from them without redundancies.

Here is a simple example about its working process:



3: FOAF algorithm workflow (by Author)

Its unconcealed goal to support the cooperation between semantic Web applications, so a possible solution was designed for the Electra Signature. Signing parties' Ügyfélkapu e-mail addresses can be gotten in a trusted way, in a well-formed semantic Web format. Electra is able to process this info set automatically, without human actions. To get our partners' details in a trusted tunnel is very important to avoid illegal acts, and thanks to the opportunity of Ügyfélkapu, details do not have to be gotten manually, Electra can import the contacts via FOAF. The figure below presents this cooperation.



4: Future plan (by Author)

Electra Signature

This is an open source, component and government authentication based document signing service for creating primarily electronic agreements. It can be done by its unique values, which are not available in other products in the market. These other products mean the unlimited available solutions such as certificate based or biometric handwritten electronic signatures because limited services may offer opportunities to set sign order or other advanced features, but these cannot be used unlimited by for example Hungarian citizens. Electra has two main unique features are sign order set and vice add options for signing parties of a document. As I highlighted, we would like to implement the paper agreement flow in electronic official administration context, so it is focused on primarily these properties.

Electra Signature has two essential infrastructures are Ügyfélkapu government authentication and Adobe Extensible Metadata Platform (XMP).

Ügyfélkapu is an electronic official administration system for primarily Hungarian citizens, but foreign people also can register an account that has a valid identity document, like passport. One of the e-signature conditions is done by this service, the identity verification of signee has to be ensured in the case of sign and in the future to e-signature be official. Official and non profit organizations can connect to the central system of Ügyfélkapu, so the University of Debrecen can host the Electra Signature. That means, the service will be managed by the University, but everyone can access it that have an Ügyfélkapu account.
Adobe Extensible Metadata Platform is an RDF based structure to add meta information to primarily PDF/a files, but you can use it in for example JPGs too. This is an ISO (International Organization for Standardization) standard so hardware suppliers also use it in cameras, mobile phones to add metadata of photos to the captures. This is a semantic Web format with RDF base and thanks to this fact, advanced Meta tags can be built up with XML (Extensible Markup Language) schemas. We also used this property to implement Meta structure of Electra. Here is its XSD (XML Schema Definition) schema:

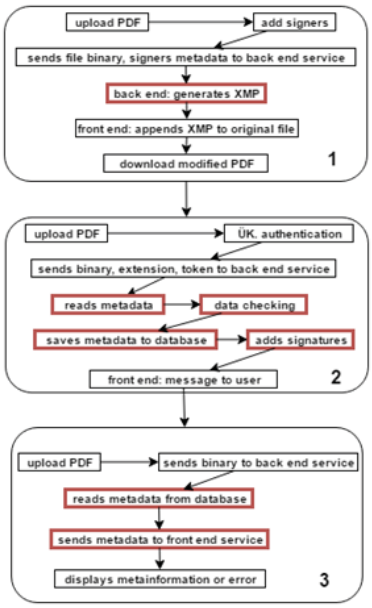
Now, some words about its working process, there are three sub-services: metadata adding, signing and checking. These are connected to each other, but they can work independently in other applications by component based design.

In this figure, the complete working process can be seen, how sub-services cooperate with each other:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:sign="http://w2.inf.unideb.hu/~maszatweb/signature/"
elementFormDefault="qualified" targetNamespace="http://w2.inf.unideb.hu/~maszatweb/signature/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:annotation>
      <xs:documentation>Electra sign signature schema - 2016</xs:documentation>
   </xs:annotation>

   <xs:simpleType name="email_type">
      <xs:restriction base="xs:string">
         <xs:pattern value="[^@]+@[^\.]+\..+"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:element name="signatures">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="signature" minOccurs='1' maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="email" type="sign:email_type" minOccurs='1'
maxOccurs="1"></xs:element>
                     <xs:element name="level" type="xs:positiveInteger" minOccurs='1'
maxOccurs="1"></xs:element>
                     <xs:element name="vices" minOccurs='0' maxOccurs="1">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="vice" minOccurs='1' maxOccurs="5">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="email" type="sign:email_type"
minOccurs='1' maxOccurs="1"></xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```



5: Electra Signature workflow (by Author)

The first step is adding signing parties' Ügyfélkapu e-mail addresses, sign levels and vices to a document will be signed via Electra Signature. Thanks to TCPDF (an open source PDF generator, implemented by Technick.com Ltd and redesigned by me), metadata adding process can be easily managed. After the submission of info set, it will be represented in the RDF structure and appended to the original PDF/a file. Electra is a client-server implementation, so you can recognize TCPDF runs in the background, and the given RDF structure will be added to the file by the client. It is also important than the component based design because different clients can be implemented easily, independently from the back end service in the future.

For example, beyond the online browser client, we can implement mobile, command line and other clients too.

In the second step, signature will be born, I am sure this is the main service, the first is the preparation, the third is the checking and the point is done in the second. In this step, a third party was brought, Ügyfélkapu service between the back end and client. So, this is a little complex service, but it is not incomprehensible. It starts with an Ügyfélkapu token check process, if the signing party is verified by the government service, it goes to the next process to check user is added to the uploaded file as a signee. If everything is okay, signature timestamp will be registered, and it gives back a message to the client. This notification will be presented to the signing party about success of its sign process. I would like to mention an important thing, timestamp will be provided by the local NTP (Network Time Protocol) service is official from the side of law. And finally, let's meet the checking process: we can get the information about signing parties' signatures in this step. If an Electra signed document is uploaded, then it gives back an info set about all signatures of the doc and a notice if all signatures are done. If everybody signed a document, it will be marked as a valid document.

All of these services can be tried out in the DEMO, but you cannot make an official, valid document, this option will be available from the real system, of course.

Summary

In first, I would like to thank my supervisor for supporting me to implement this service, and a big thanks to everybody who attended at this InterTalent conference and made the chance I could present my results.

You could meet my research results, my two main projects: FOAF and Electra Signature. I would like to highlight the possibilities of Electra in this summary.

Electra service is based on Adobe XMP and it has Neo4j database connection too, so Electra has many chances to cooperate with semantic Web applications. Beyond the cooperation, an other important thing, automatic data process also can be easily implemented.

Thanks to the Ügyfélkapu authentication and unique solutions of Electra, sign order set and vice add options, paper based different signing policies can be represented in electronic environment. This opportunity allows creating total paperless, electronic official administration systems for almost all organizations.

I talked about how Electra can be used at the University of Debrecen as an electronic official administration system at the InterTalent conference. Our goal is supporting administration become redundancy free, less resource needed service. In the section of Electra Signature, I described the basis of Electra service will be at the University, but everybody can access it, unlike nowadays how ERP systems operate.

I hope, you could correctly meet my results, and I could arouse your interest want to know more about my projects. If I am right, do not hesitate to visit my site and contact

me at the www.rtibor.hu.

Bibliography

2015. évi CCXXII. tv. (Downloaded: 05.10.2016.). (Web: http://njt.hu/cgi_bin/njt_doc.cgi?docid=193173.316586).
Adobe XMP. (Downloaded: 05.10.2016.). (Web: http://www.adobe.com/products/xmp.html).
Bruno, E. J. (2013). PHP Takes on Business-Critical Apps. IMPACT Assessment. (Downloaded: 05.10.2016.). (Web: http://static.zend.com/topics/Zend-Impact-Assessment-PHP-July-2013.pdf).
DigitDoc. Biometrikus aláírás (Biometric hand-written signature). (Downloaded: 05.10.2016.). (Web: https://digitdoc.hu/megoldasaink/biometrikus-kezi-alairas/ismerteto).
DigitDoc. Hogyan működik az e-aláírás (How e-signature works). (Downloaded: 05.10.2016.). (Web: http://www.digitdoc.hu/hatteranyagok/digitalis-hitelesites/hogyan-mukodik).
Digitoll. Bizalmi szolgáltatások (Trusted services). (Downloaded: 05.10.2016.). (Web: http://ds.digitoll.co.hu/hitelesitesszolgaltatasok.php).
e-Szignó. Tanúsítvány fajták (Types of certificates). (Downloaded: 05.10.2016.). (Web: https://e-szigno.hu/hitelesites-szolgaltatas/tanusitvanyok/tanusitvany-fajtak.html).
e-Szignó. Időbélyegzés (Timestamp). (Downloaded: 05.10.2016.). (Web: https://e-szigno.hu/tudasbazis/idobelyegzes.html).
eIDAS 910/2014/EU rendelet. (Downloaded: 05.10.2016.). (Web: http://eur-lex.europa.eu/legal-content/HU/TXT/?uri=CELEX%3A32014R0910).
Gamma, E. (1994). Design Patterns. Addison-Wesley.
Gasiorowski-Denis, E. (2012). Adobe Extensible Metadata Platform (XMP) becomes an ISO standard. (Downloaded: 05.10.2016.). (Web: http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1525).
Hidasi. (2009). Ellenjegyzés vagy tanú (Countersign or wittness). (Downloaded: 05.10.2016.). (Web: http://www.hidasi.hu/index.php?option=com_content&view=article&id=183&catid=20&lang=hu).
Martin, R. C. (2009). Clean Code. Prentice Hall.
Németh, G. (2013). Aláírás, bélyegző használat (Usage of signature, stamp). (Downloaded: 05.10.2016.). (Web: http://blog.drnemethlaw.hu/belyegzo-es-stemplinyomkodas-kek-tinta-vagy-nyomtatott-betuk-mitol-eros-egy-okirat-es-mi-az-ami-nem-szamit-az-alairaskor/).
Németh, G. Hibák a szerződésben (Mistakes in the agreement). (Downloaded: 05.10.2016.). (Web: http://blog.drnemethlaw.hu/tag/hibak-a-szerzodesben/).
Netlock. Minősített tanúsítvány (Qualified certificates). (Downloaded: 05.10.2016.). (Web: https://www.netlock.hu/html/minositett.html).
Neo4j official page: http://neo4j.com
Newcomer, E. (2002). Understanding Web Services. Addison-Wesley.
OWL official page: https://www.w3.org/2001/sw/wiki/OWL
Pethő, A., Folláth, J., Huszti, A. (2010). Informatikai biztonság és kriptográfia (IT security and cryptography). Debreceni Egyetem.
Pugh, K. (2006). Interface-Oriented Design. The Pragmatic Programmers LLC.
RDF official page: https://www.w3.org/RDF

TechNote 0003: Metaadat in PDF/A-1. (2008). PDF/a Competence Center.
Tidwell, D. (2001). Programming Web Services with SOAP. O'Reilly.
Wiegand, S. (2013). Using OWL with Neo4j. (Downloaded: 05.10.2016.). (http://neo4j.com/blog/using-owl-with-neo4j).