published in: Proceedings of 2nd International Workshop on Knowledge Management in E-Government,
Trauner Verlag Linz, ISBN 3-8548-7246-1, pp 196-206

Ontologies for Managing Knowledge about Forms for Government Processes

Johann Eder

University of Klagenfurt
Department of Informatics Systems
Klagenfurt, Austria
eder@isys.uni-klu.ac.at

Michele Missikoff

IASI-CNR-Rome
LEKS - Laboratory for Enterprise Knowledge and Systems
missikoff@iasi.rm.cnr.it

Abstract *Governments and public administrations often have definition authority for the meaning of terms through legalistic processes and documents as well as through their position on. Many of these definitions are used to create and interpret forms which are typically used for documenting, supporting and guiding processes of public administrations. Such forms also represent the interface for citizens and enterprises to administration procedures. We propose the use of ontologies to help organizing and documenting this knowledge and to describe the relationships between concepts and terms. These ontologies can then be used to provide semantic underpinnings of the terms used in document structures, in particular in structured and semistrucured documents (forms) especially those using XML to describe the document structure.*

1. Introduction

Knowledge representation is a complex activity and can be carried out at different levels of detail, mainly depending on the objectives, the context, and the users involved [7]. We intend to focus on the cooperation that takes place between parties within an e-Government environment [5,9,11]. One of the main characteristics of e-Government transactions is that they take place mainly through the Internet, by using web solutions, and are essentially concerned with the exchange of information.

Government agencies, public administration organizations, often have the authority to coin terms and concepts, supplying interpretation of their meaning. However, federal structures (like town, state, nation), distributed competencies (between different ministries, provincial governments), and distributed evolution of concepts and terms over time makes it a demanding task to keep track about terms in the enormous amount of produced documents and their (exact) meaning, and their semantic relationships.

Information exchange and cooperation typically takes place by exchanging (semistrucured) documents typically called forms. The electronic representation and processing of forms has already some tradition in the history of governmental data processing. Forms typically contain several kinds of data: *case data* containing data for the case at hand, *process data* describing the history of the administrative process including information of the process steps performed, decisions taken as well as persons involved.

Administrative processes frequently require the exchange of forms between cooperating partners. In principle, the content and organization of documents and processes of two different organizations are different. Such differences make it difficult for a computer to process them. Therefore, semantic reconciliation of business documents and interoperability of workflow systems are a key issues of the electronic government processes, in all ways, be it within different agencies of the public administration (G2G), in the relationship with business organizations (G2B and B2G), as well as in the relation to the citizens (C2G). On the side of semantic reconciliation, the possibility of using a domain ontology is emerging as a promising solution.

An ontology [1,14] is a representation of a shared conceptualization of a particular domain. Such a common conceptualization is necessary for every communication process. Explicitly considering ontologies allows to reason about sameness as well as differentness of concepts and allows to derive mappings for establishing semantically correct communication channels. Furthermore, ontology servers help an organization to keep track of all terms and notions used in it's documents, together with the history of their semantics and may even provide mechanisms for resolving disputes about the meaning of terms.

The remainder of this paper is organised as follows: In section 2 we discuss the notion of ontologies and the development of ontologies in e-government. In section 3 we briefly present XML as a means for describing government forms, and in section 4 we show how XML forms can be linked to an ontology and discuss some applications of this approach and finally, in section 5 we draw some conclusions.

2. Ontologies in governmental domains

In cooperating to work together (or in interacting in a social setting), people and organizations must communicate among themselves. However, due to different contexts and backgrounds, they can bear different viewpoints, assumptions and way of expressing their mind, even if acting in the same domain and dealing with the same problem. They may use different jargon and terminology, often ambiguously and imprecisely.

This is caused by the absence of a shared understanding that leads to a poor communication between people and organizations. In particular, when ICT solutions are involved, this lack of a shared understanding impacts in a negative way on the quality of:

- effectiveness of people's cooperation, through the web or a dedicated groupware
- enterprise organization and business (or administrative) processes
- user requirements and system specification
- the interoperability among systems

The goals of an ontology is to reduce (or, hopefully, eliminate) conceptual and terminological confusion. This is achieved by identifying and properly defining a set of relevant concepts that characterize a given application domain. The above considerations apply also to the interactions between public offices and citizens.

When a citizen or an enterprise interacts with a public office, one of the key elements of this interaction is a form. Generally, associated to a form there are instructions aiming at explaining the meaning of the fields to be filled and, sometimes, at motivating the reasons why certain information is requested. Unfortunately, very often the forms are not clear and, despite the presence of the associated instructions, the rational of the transaction escapes the comprehension of the citizen. In the current trend that moves to the Web a growing portion of the interaction of the public administration with the citizen, the forms become electronic. The citizen who tries to fill in a form, and more generally who interacts with the public administration, via the Web, will need to be supported by tools able to give an intelligent help. These tools will take advantages of the services offered by an ontology management system. But the benefits of online ontologies are not limited to the interactions with the citizen.

Significant benefits will be also obtained within the government offices, in the design of the interactions among them and in the administrative processes.

An ontology is a shared understanding of some domain of interest [14]. It entails some sort of world view with respect to a given domain, gathering a set of concepts (concerning entities, attributes, processes, etc.), together with their definitions and their inter-relationships; this is also referred to as a conceptualisation (of a given domain.)

As briefly anticipated, ontologies aim at facilitating the:

- Understanding
- Communication
- Cooperation

among people and organizations, possibly with the help of computers.

An Ontology may have different degrees of formality and different techniques used to construct it, but, necessarily, it includes a vocabulary of terms with their meaning (definitions) and their relationships.

In this perspective, an ontology is a domain vocabulary containing a set of precise definitions, or axioms, that:

- provide the meaning of the terms,
- enable a consistent interpretation of the terms defined in the vocabulary.

In our work, we decided to model domain concepts and relations according to OPAL, a methodology for ontology representation. OPAL represents the basis on which SymOntos [12], an ontology management system, has been developed by LEKS (Laboratory for Enterprise Knowledge and Systems), at IASI-CNR.

According to OPAL, concepts are organised by means of three primary modeling ideas: Actor, Processes, and Object. More precisely, we have:

Actor: any relevant entity of the domain that is able to activate or perform processes (e.g, Citizen or Fishing_license_office);

Object: a passive entity on which a process operates (e.g, License);

Process: an activity aimed at the satisfaction of a goal (e.g, Renewing_a_license);

Besides the above primary modeling ideas, OPAL proposes the following secondary modeling ideas.

Goal – is a desired state of the affairs that an actor seeks to reach. (Go_fishing)

State – is a characteristic pattern of values that instance variables of an entity can assume. (e.g., Invalid_license)

Rule – is an expression that is aimed at restraining the possible values of an instance of a concept (constraint rule) or that allow to derive new information (production rule). (e.g., "license is for resident only")

Information Component - a cluster of information pertaining to the information structure of an Actor or an Object (e.g., Citizen_address);

Information Element - atomic information element that is part of an Information Component (e.g., Citizen_age, Permit_validity);

Elementary Action - activity that represents a process component that is not further decomposable (e.g., Defining_range_of_prices).

The above modeling ideas are necessary for defining (unary) concepts. According to OPAL, concepts are linked together by means of a number of semantic relations, that can be seen as vertical or horizontal. Vertical relations are: *Broader* (B), that gathers the more general concepts; *PartOf* (Pa), and InstaceOf (with an evident meaning). Horizontal relations are: *Similarity* (S), that gathers the similar concepts (with an associated similarity degree); *Predication* (Pr), that link *Information Components* and *Elements* to the current concept, and the (generic) *Relatedness* (R), to link the other related concepts.

The primary modeling ideas allow the designer to partition the concept model in three main sets: *Actor, Process, Object*, populated according to the *kind* of the concept. When a concept is entered in the ontology, the above list of information must be firstly given. Therefore, according to OPAL, an intentional, structural (i.e., without a reference to instances and with a simplified view, without goal/state/event references) definition of a concept *c*, is represented by the following 8-tuple:

$$c = (n, k, d, B, Pa, S, Pr, R),$$

where: *n* is the name of the concept; *k* is the kind, i.e., on of the modeling ideas of OPAL (Actor, Process, …); *d* is the description, explaining the meaning of the concept, generally in natural language. Then we have the set of concepts *B, Pa, S, Pr, R* related to *c* as reported above.

Conceptual relations play a key role since they allow concepts to be inter-linked according to their semantics. The set of concepts, together with their links, forms a semantic network [1].

In Fig. 1 we provide an example of a concept structured according to OPAL follows.

<table>
<tr><th colspan="2"><u>Fisherman</u></th></tr>
<tr><td><u>Def</u>: someone who has the license to fish on rivers and lakes of the County</td><td>Kind: Actor</td></tr>
<tr><td><u>Broader</u>: sportsman, citizen
<u>PartOf</u>: fishing_club
<u>Similar</u>: scubadiver [0.6]
<u>Predication</u>: identity_info, residence_info</td><td><u>Related-objects</u> :fishing_license, payment, fishing_pond
<u>Related-actors</u>: county_permits_office, roomService
<u>Related-processes</u>: reserving, paying, billing,</td></tr>
<tr><td colspan="2">(all reported terms, except in the row below the concept name, correspond to concepts in the ontology)</td></tr>
</table>

Figure 1 – the Fisherman concept in OPAL

3. Forms in XML

There are no generally accepted standards for electronic forms, although many attempts have been made. We find heterogeneity of the documents used on several levels, from the choice of the code for characters up to semantic heterogeneity.

For the lower levels, XML, the extensible markup language [10,13], is developing as generally accepted standard. It can be soon taken for granted that it is possible to send XML documents to a partner and this partner is equipped with software to process XML documents.

For typical forms of e-governement processes, XML seems an excellent choice for representing these forms with their integrated use of formatted and unformatted data, their different choice of free text, fill-in-the-blanks text, tabular data, etc.

XML models semistructured data as trees, whereby nodes contain data and the markup names (tags) label the nodes (elements). The tags are interpreted as schema information and therefore the tree contains the schema information of the whole document. The interface to the XML tree-model of a document is called the Document Object Model (DOM) also standardized by the W3C.

```
<!ELEMENT request (person, fishing_ground, accomodation?)>
<!ELEMENT person (first_name, last_name, home_address?)>
<!ELEMENT home_address (address)>
<!ELEMENT accomodation ((hotel | pension | appartment), acc-address)>
<!ELEMENT acc_address (address)>
<!ELEMENT address ANY>
...
```

Figure 2 – request-fishing-license.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE request_fishing_license SYSTEM "request_fishing_license.dtd">
<request>
  <person>
    <first_name>John </first_name>   <last_name> Doe </last_name>
  </person>
  <fishing_ground> Wörther See North </fishing_ground>
  <accomodation>
    <Hotel> Bellevue </Hotel> <acc_address> Velden </acc_address>
  </accomodation>
    ...
</request>
```

Figure 3 – request-fishing-license.xml

The XML document can be constrained and further documented by a Document Type Definition (DTD), which is part of the XML language. A DTD is a context-free grammar for the document, and the schema of a document instance can be validated against a DTD by an XML parser. The DTD is an optional part of an XML document but in this paper we assume that every document has a DTD.

A DTD can describe several disjoint sets of document schemas - for example one single DTD can validate an order-form as well as an application form. Thereby the root-element of the document is used as an entry-point to the DTD to put together the grammar for validating the schema-tree. Elements may share common sub-structures.

In Fig.2. we show a simple document type description for XML forms for requesting a fishing license, and in Fig.3 we show an example of such a request in XML.

4. Linking XML and Ontologies

4.1. Meta-Structure

In contrast to HTML documents, which provide only meta-information for the representation (rendering) of documents, XML provides structural information through tags. These tags, however, are purely syntactical and they only represent the meaning by means of their names. For enriching the semantic content of such documents, we can link the elements of XML documents, in particular the definition of elements in document type descriptions to the terms we introduced and managed in our ontologies.

In figure 4 we show the combined meta-schema for document type descriptions and ontologies. A DTD consists of several (root-) components, which are the entry points for parsing XML documents and top level tags in documents. Components may recursively consist of other components, showing the typical hierarchical structure of DTDs. Furthermore we introduce a generalisation relationship between components of DTDs, as in [4,10]. The components are always associated with tags which might appear in XML-documents.

As outlined in section 2, an ontology consists primarily of concepts together with the semantical relationships between them (a concept is composed of other concepts, is related to other concepts, is similar to other concepts, etc.) The components of a DTD represent concepts which are described in the ontology and we represent this relationship in our meta-structure. This combination now allows one to navigate from a tag in an XML document to the corresponding concept in the ontology. Therefore, it opens the possibility to looking up the exact definition, finding similar concepts, or more general or more specialized concepts. The association can also be used in the other direction. For an example, it is possible to find all forms in which a certain concept is involved.

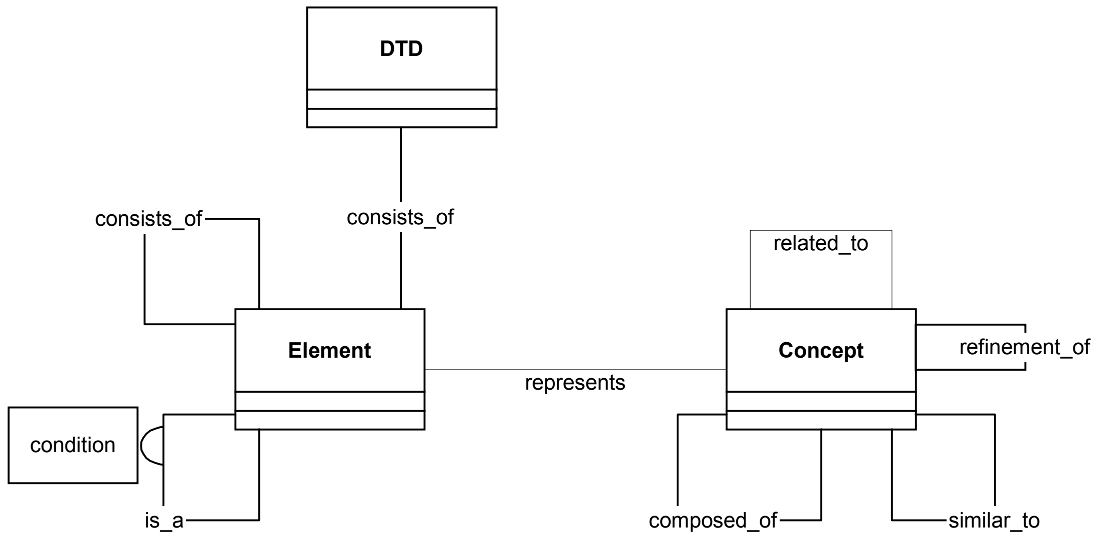

Figure 4: Meta-structure for form description

4.2. Applications

Now we want to present some of the intended applications of this meta-structure which make use of the combination of document type descriptions and ontologies.

4.2.1. Data Dictionary

The first use is to see the ontology as a data dictionary for the developed DTDs and for all XML documents. The developers of DTDs are requested to relate the tags they use to the concepts represented in the ontology. By this discipline a close connection between a rigorous representation of knowledge and information, and the development of legal and procedural terminology and structures can be maintained.

Such a structure can also be used to document the evolution of concepts and terms on one hand and the evolution of forms and documents on the other hand, when the structure also contains temporal information. Furthermore, the structure provides tremendous support for the maintenance of forms when legal definitions change. It is easily possibly to identify all forms which are affected by such changes.

4.2.2. Help System

The ontology contains a clear and precise definition of terms. It can therefore be used to automatically generate Help Systems which support users in the comprehension of language used in administrative forms. We do not think that a help system for casual users can be generated automatically this way, but surely for educated users.

4.2.3. Form transformations

Administrative processes typically involve several actors from different organizations and several forms which in general are not designed together. When such processes need to cooperate by exchanging electronic documents, with the intention to avoid media breaks, the transformation of forms between several formats (i.e. structural representation) is necessary.

The development of such transformation firstly consists of identifying the format into which a given form has to be transformed, and secondly the identification of the correspondence of the components of the different forms. After that the actual transformations of elements can be developed. The first two steps can benefit tremendously from our meta-structure, if the developers of the different forms relate the DTDs to the same ontology. Now the semantic description of the meaning of tags is no longer simply in their name, but different named tags may represent the same concept. Given a DTD it is easy to find other DTDs which represent the same or similar concepts, and it is easy to discover which components of different forms are semantically equivalent by considering the semantic relationships of the concepts of the ontology they represent.

5. Conclusions

We presented an approach for describing government forms by means of XML-documents and provided a meta-architecture for storing of XML-forms and their components. The meaning of the elements, components and attributes of these forms is given by linking them to an ontology of the domain which describes the concepts of the domain and their relationships.

This solution has several potential uses, e.g.

1. First it is an aid for development of forms and thus for the documentation of the data dimension of governmental processes.
2. Knowledge management for automated help system
3. Support structure for defining mappings between forms

This concept is an important step in our project to create a methodology and a set of software prototypes aimed at supporting the cooperation of different organizations by using integrated ontology and web solutions [6]. The cooperation may take place at two different levels of interoperability between actors in administrative processes: deep and shallow. Shallow interoperability consists mainly in the exchange of documents representing case data; deep interoperability implies a certain level of workflow integration [2,3], that is necessary when the cooperating organizations share a certain amount of their knowledge, in particular for what concern their way of doing business (at least in the part that requires interoperability with external subjects). In deep interoperability the organizations typically exchange also process data which are much more specific than general case data [3].

Deep interoperability requires a better understanding of the ways the different partners carry on their processes. Assuming that shallow interoperability has been solved, here we concentrate on business and administrative processes and the workflows that implement them. This level requires the explicit modeling of workflows and the possibility of verifying their compatibility [8]. It is possible that suggestions for business process reengineering (BPR) emerge in this phase, to increase the interoperability potential of an organization (but also to gain on efficiency).

At both levels, shallow and deep interoperability, we believe that the use of a domain ontology, agreed and accepted by all the partners, is an important enabling solution. In a nutshell we find all these problems in the way forms are developed, interpreted and treated in processes of public administration. Elements of forms frequently refer to terms defined in legal documents (laws, regulations, provisions, findings, etc.). Keeping track of these relationships is a cumbersome but necessary task. Changes in legal documents, evolution of terms are often difficult to reproduce in all forms referring to these documents. On the other hand, it is often necessary to integrate data stemming from different forms with differences in the used elements. For such an integration it is mandatory to reason about the meaning of terms and to decide how to relate elements of one form to another form. When all elements of the forms are connected to same ontology, such an integration is facilitated to a high degree. Even if the integration will not be made full automatic, the developer of transformation programs which map the content of one form to a different form will receive great support for this complex task. Adaequate technical support for defining transformers[4,10], like composition and specialization of transformers based on composition and specialization structures of the involved forms, contribute also to the productivity increase in this process. For the semantic interpretation of transformation programs also refereal to ontologies is very helpful, as well as for indexing transformers with the goal of easy retrieval.

6. References

[1] R.J. Brachman; “On the epistemological status of semantic networks”; in "AssociativeNetworks - Representation and use of Knowledge by Computers", N.V.Findler (Ed.);Academic Press, New York, 1979.

[2] J. Eder, H. Groiss, W. Liebhart; "The Workflow Management System Panta Rhei",In: A. Dogac, L. Kalinichenko, T. Öszu, A.Sheth (Eds.): Workflow Management Systems and Interoperability, Springer-Verlag, 1998

[3] J. Eder and H. Groiss; "Telecooperation through World Wide Workflows", in: Proc. IFIP'98 World Computer Congress - Telecooperation Conference, Vienna, Österreichische Computer Gesellschaft, 1998.

[4] J.Eder, W.Strametz;"Composition of XML-transformations"; technical report ISYS-01-01, University of Klagenfurt, 2001.

[5] A.K. Elmargamid, and W.J McIver, "The ongoing March Toward Digital Government", IEEE Computer, Vol.34(2), 2001.

[6] A. Formica, R. Pizzicannella; "On the Reification of Business Processes"; 1st International Workshop on Enterprise Management and Resource Planning Systems:Methods, Tools and Architectures (EMRPS'99); Venice, 25-26 November 1999

[7] M.Missikoff, "Management of Knowledge and Intangible Assets in the Context of Dynamic Business Environments: a Position Paper", Workshop on New Organisations and Knowledge Management, Bruxelles, 24-24 May, 2000.

[8] A. Formica, M. Missikoff; "Analysis and a preliminary proposal of an Ontology in the tourism domain", FETISH Workshop, Rome, 23 June, 2000

[9] A. Engel; "IT-gestützte Vorgangsbearbeitung in der öffentlichen Verwaltung", in Lenk/Traunmüller (eds.) Öffentliche Verwaltung und Informationstechnik, Decker, Heidelberg, 1999.

[10] W. Strametz; "Komposition und Spezialisierung für XML Transformationen", Diplomarbeit, Universität Klagenfurt, 2000.

[11] R. Traunmüller; "Systeme zur IT-Unterstützung", in Lenk/Traunmüller (eds.) Öffentliche Verwaltung und Informationstechnik, Decker, Heidelberg, 1999.

[12] www.symontos.org

[13] www.w3c.org

[14] M. Uschold, M. Gruninger; “Ontologies: Principles, Methods and Applications”; The KnowledgeEngineering Review, V.11, N.2, 1996.