XHTML with RDFa as a Semantic Document Format for CCTS Modelled Documents and Its Application for Social Services

Konstantin Hyppönen, Miika Alonen, Sami Korhonen, and Virpi Hotti

University of Eastern Finland, School of Computing
P.O.B. 1627, FI-70211, Kuopio, Finland
{Konstantin.Hypponen,Miika.Alonen,Sami.S.Korhonen,Virpi.Hotti}@uef.fi

Abstract. For achieving semantic interoperability, messages or documents exchanged electronically between systems are commonly modelled using standard specifications, such as the UN/CEFACT CCTS (core components technical specification). However, additional requirements, such as the need for layout markup or common metadata for certain archiving scenarios might be applied to the documents. Furthermore, the management of resulting artefacts, i.e., core components, XML schemas and related infrastructure, could be cumbersome in some cases. This paper investigates the use of the W3C XHTML+RDFa (extensible hypertext markup language with resource description framework attributes) for representing both the layout and semantics of documents modelled according to CCTS. The paper focuses on the validation of XHTML+RDFa documents against a core components library represented as an ontology. In addition, the paper illustrates and validates this demand-driven solution in the scope of the Finnish National Project for IT in Social Services.

1 Introduction

For a long time, documents have been the main information exchange mechanism in public administration. As the service processes get automated, documents become replaced or aided by electronic message exchange. The standard requirement for the electronic messages is that they must convey necessary semantic meaning of their contents in a form understandable by the information systems involved.

In many cases, however, additional requirements are applied to the message exchange system, such as the generation of human-readable versions of the messages, or archiving them according to a certain archiving plan. This might require the addition of layout markup and certain document management metadata to the messages.

Nešić [12] defines a semantic document as a "uniquely identified and semantically annotated composite resource". The document is a content unit (CU) built of smaller CUs, which in turn can be either composite or atomic. Every CU

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 229–240, 2012.

should be understandable by both humans and machines. Furthermore, the content of semantic documents should be completely queryable, with addressable elements (i.e., CUs) of different granularity.

Message exchange can be modelled using standardised specifications [13,14,18] which describe processes and models for designing common building blocks used in different messages. The specifications also include standard ways of designing XML schemas for business documents and information entities by reusing these building blocks [13,17,19]. The resulting schemas are purely semantic-oriented and are targeted specifically at message exchange between information systems. Therefore, they do not include provisions for any layout markup. Human-readable versions of the messages must be generated using separate style sheets or other mechanisms.

On the contrary, document formats used for preparing documents in common text processors (such as OOXML or ODF) or for publishing the documents online (such as HTML or PDF) concentrate mostly on layout features. They are not suitable as such for information exchange, as they lack precise semantic markup. However, semantic markup and metadata can be added to them using annotation tools, thus bringing semantics and layout markup together. Storing semantic annotations inside a document usually requires the extension of the document format schema, which is not always possible. Uren et al. [20] define requirements for annotation tools and provide a summarizing overview of them with regard to these requirements. Eriksson [4] describes a technology for ontology-based annotation of PDF documents. Individuals of ontology classes are created from highlighted areas of PDF documents. A bidirectional link is then established between the individual in an ontology and the PDF annotation.

Decker et al. [3] argue that flexible semantic interoperability cannot be achieved with XML, and suggest using RDF (resource description framework) based technologies for defining the semantics of information exchange. Combinations of XML and RDF technologies are also available. A knowledge model of the business domain could be represented in a form of ontology and linked to the schemas, for adding more precise semantics to schema elements. For example, the W3C recommendation SAWSDL (semantic annotations for the web services description language) defines a way of providing semantic annotations to WSDL and XML schema constructs [5]. Among other mechanisms, annotations can be implemented as links to ontology classes.

W3C provides a recommendation for a way to embed semantic markup in XML (e.g. XHTML) documents [1]. RDFa (RDF attributes) is a specification of attributes for adding semantic metadata to any markup language. RDFa can be used in conjunction with XHTML for extending the basic layout-oriented HTML markup with semantic elements. However, the applicability of XHTML and RDFa for message exchange designed with standard modelling methods such as the UN/CEFACT Core Components Technical Specification (CCTS) [18] has not been examined closely.

Our results. We propose the use of XHTML+RDFa as a document format (carrier) for messages modelled according to the CCTS specification. In addition,

we outline the structure of the validation service which ensures that the documents follow their defined structures, and performs other document type specific checks. A proof-of-concept implementation of the validation service is described. The applicability of this approach is examined in connection with requirements for the document exchange needed in Finnish social services.

The rest of the paper is structured as follows. Section 2 introduces the requirements for documents used in social services and provides an overview of the core components types defined in CCTS. Section 3 explains how core components can be represented in RDF, and Sect. 4 describes a validation service for XHTML+RDFa documents with CCTS-based artefacts. Section 5 discusses the pros and cons of our approach, and Sect. 6 provides concluding remarks.

2 Documents in Social Services

One of the goals of the IT project for social services in Finland (Tikesos[1]) is to model all the document types that are used in Finnish social services. Document structures have been defined by the social service experts on the content level. The social services system in Finland uses about 200 different types of documents [11]. The documents are created in standard social work, where they are used as part of social service processes. As the processes get automated, the document contents are analysed and modelled for document processing systems. The documents can be used for information exchange between systems.

In addition to information exchange, one of the main requirements is to archive all documents for future use. This is demanded by the Finnish Archive Law [6], which is applied to all public authorities. A centralised archive is currently being developed for storing all social services documents in Finland [11]. When implemented, the archive will to be accessed by social services information systems, for archiving the documents (as shown in Fig. 1) and fetching them later if needed. We note that the archiving requirement is of utmost importance, as the future IT infrastructure for social services in Finland is built around the centralized archive.

There are no international de jure or de facto standards for a document format for social services. Therefore, we set out to define a document format suitable for our needs. The following basic requirements are placed on the document format for Finnish social services:

1. A nationally defined set of metadata must be implemented.
2. Content units of different granularity should be marked in the document, so that they are addressable.
3. It should be easy for an information system to parse the document contents.
4. A human-readable version of the document should be easily produced.
5. The validation of the document structure and content should be possible, according to validation service requirements.

[1] `www.tikesos.fi`

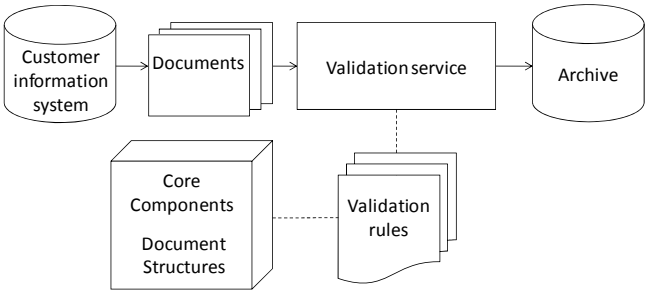

Fig. 1. The centralised archive validates and stores documents produced by customer information systems

The common building blocks of the documents used in social services were identified, analysed and modelled according to the UN/CEFACT CCTS [18]. The CCTS model was selected because it is widely used in Europe for similar projects [8], and has the status of an ISO standard. We provide here a short introduction to the concepts defined in CCTS. Only the concepts relevant to this paper are described.

CCTS defines a method for designing of common semantic building blocks called core components. The components form a language used in information exchange by business partners. A number of different core component types (CCTS artefacts) is defined (see Fig. 2).

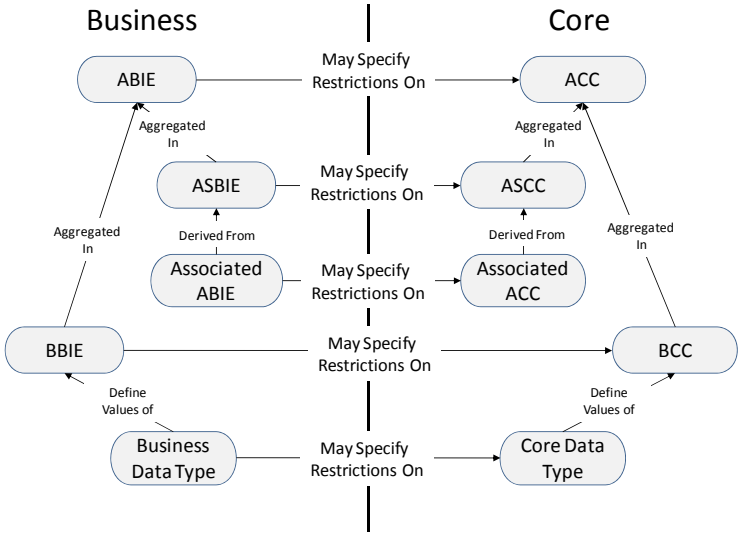

Fig. 2. Relationships of different types of core components (figure from [18])

An Aggregate Core Component (ACC) is an object class describing a concept with clearly defined semantics. The object class contains a number of properties related to this concept. An example of an ACC is Address.

A Basic Core Component (BCC) is a property appearing in a certain object class (ACC). BCCs are based on simple data types (called Core Data Types) such as Text, Number, Amount. For example, the street name in an address can be represented as a BCC.

An Associated Core Component (ASCC) is a property appearing in a certain object class (ACC). The property is based on another object class (another ACC). For example, the validity period of an address could be represented as an ASCC, based on the ACC Period.

An Aggregate Business Information Entity (ABIE) is an object class which is a qualified version of an ACC. It may have a narrower meaning than the parent ACC. Moreover, any property of the ACC may be qualified in the ABIE. An example of an ABIE is Trade_ Address (an address representation used in trade-related business documents).

Basic Business Information Entities (BBIE) and Associated Business Information Entities (ASBIE) are properties of ABIE, constructed by qualifying the corresponding BCC and ASCC properties of the parent ACC.

The resulting library of core components defined for social services in Finland has around 200 different object classes (ACCs and ABIEs), with about 1000 properties in them altogether. The core components are used in some 200 different documents, split in 15 groups (applications, decisions, notifications, agreements etc.).

Although XML schemas for all business documents can be produced relatively easily, their applicability for our requirements is questionable. For instance, it is rather difficult to handle documents based on 200 different schemas in a single archive. The systems which access the archive must support these schemas in order to parse the document contents. Furthermore, document structures tend to change from time to time, and with the updated versions the number of different schemas can easily reach thousands in a span of a few decades. The documents based on old schemas remain in the archive and should be accessible by users, at least as human-readable versions. Producing human-readable document outputs can be also troublesome, as the layout information is not included in documents based on standard CCTS-compliant schemas, and separate schema-specific style sheets are needed. A separate human-readable output generator could be implemented within the archive. However, the generator must handle all the different schemas on which the documents in the archive are based.

3 From Core Components towards RDFa

CCTS does not define a way in which core components should be stored. UN/CEFACT distributes their international core components library in a table format, and provides specifications for the representation of business information entities as XML schemas. We present a transformation to an RDFS/OWL

representation from any core component library. The representation is aimed at simplifying the generation and validation of semantic XHTML+RDFa documents using existing core components. Business parties can model their documents according to the standard CCTS method, and use XHTML with RDF attributes as the data exchange medium.

The RDFS/OWL representation of a core component library is generated in a two-step process. First, a general XML serialisation of the library is produced, based on a custom schema. Second, the serialisation is transformed to the RDFS/OWL representation through an XSLT script. The script transforms core components into RDF descriptions that are subclasses of the following OWL types: owl:class, owl:objectProperty and owl:datatypeProperty (see Fig. 3). Unique URIs for RDF descriptions are generated from CCTS names of core components (qualifiers, object class terms, property and representation terms).

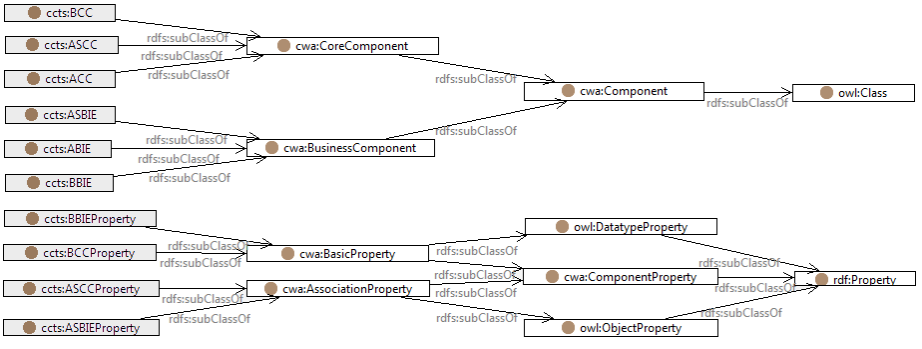

Fig. 3. CCTS artefacts represented as OWL classes. The classes showed on grey background are not in the model designed for validation, but may be used for more precise representation of the CCTS model.

Core component cardinalities are represented in each corresponding OWL class as property restrictions. CCTS cardinalities maxOccurs and minOccurs are represented as OWL properties: owl:maxCardinality, owl:minCardinality and owl:cardinality in conjunction with owl:Restriction as shown in Table 1.

In our model, most core data types are transformed into corresponding XML schema data types. However, if a data type contains additional elements (data

Table 1. Core components cardinalities represented as RDF/OWL restrictions

maxOccurs	minOccurs	Restriction	owl:Restriction
1	1	exactly 1	owl:cardinality
unbounded	k	Min k	owl:minCardinality
1	0	Max 1	owl:maxCardinality
n	k	Max n, Min k	owl:maxCardinality, owl:minCardinality

type attributes in CCTS such as code list identifier in the Code data type), it is represented as a class. Possible values for basic and associated core components are defined in properties as rdfs:range.

Definitions and examples for the core components are represented through the introduction of properties ccts:definition and ccts:example. Additional properties for useful information could be added accordingly. The resulting model is stored as a business ontology for its future use in the validation service.

It is also possible to create a CCTS specific OWL model which contains exactly the same information as the original CCTS model [2]. This type of model can be used for the construction of core components and business documents, including the automatic generation of XML schemas. However, in the case of XHTML+RDFa validation such model is not necessarily needed.

4 Validation of XHTML+RDFa Business Documents

Social services processes include a substantial amount of processing rules, part of which is visible on the document level. For example, there can be conditionally mandatory parts of the document, with the condition defined by code values. Documents could also include some unconditionally mandatory fields, such as the social security number of the customer. Most of such rules and checks can be implemented on the form level. However, as there could be several different implementations of the same forms, it must be ensured that invalid documents cannot be submitted to the archive. A validation service is a part of the archive that checks incoming documents against a number of rules. In addition to the standard validation of the document structure against its schema, the validation service might check the following:

1. Correctness of code values and code lists used in the document (similar to UBL code list value validation [14])
2. Conditionally mandatory fields and other rule-based checks
3. Adherence to the basic formatting requirements

The validation of business documents is based on the closed world assumption [15] because a document is in fact a snapshot of the information collected on a certain clearly defined issue at a single moment. Although more information may be available elsewhere, it is out of the scope of the document and should not be considered in validation. There have been several proposals for adding integrity constraints into OWL by defining semantics for the constraints [10,16]. Standard OWL reasoning cannot be used for strict validation because restrictions are used for inferring new information rather than for checking integrity constraint violations.

Our solution for the validation of XHTML+RDFa business documents works as follows:

1. validate the XHTML structure
2. extract the RDF graph from the XHTML markup enriched with RDFa attributes

3. compare the resulting graph with the corresponding RDFS/OWL schema
4. perform additional rule-based checks.

For our purposes, a top-level metamodel is created to separate the classes and properties used for validation (classes in namespace cwa in Fig. 3) from standard OWL structures. The metamodel defines new types for classes and properties that are used in validation to check whether the extracted RDF graph is consistent with the corresponding RDFS/OWL schema. The closed world constraint validation service depicted in Fig. 4 can check XHTML+RDFa documents for unknown predicates and classes, unexpected namespaces, ill-formed literals and cardinality constraints.

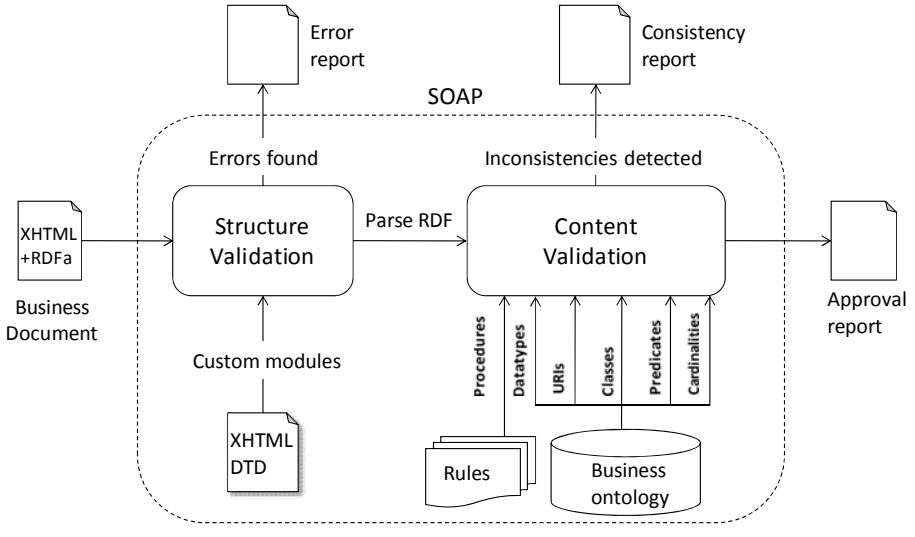

Fig. 4. Closed world constraint validation service

XHTML is based on a modular framework [9] that can be used to restrict or extend the language. The document can be validated against a custom schema or document type definition (DTD) for XHTML. For instance, the scripting module can be left out, as it is rarely relevant for business documents. In addition, the requirements for archiving might prohibit any references to external resources, as these can get outdated after the document has been archived.

An XHTML+RDFa document is first sent to the validation service wrapped in a SOAP request. The document is then validated against the (potentially customised) XHTML+RDFa DTD. For documents with impaired basic XHTML structure, a SOAP response with relevant error details is generated.

After the validation of the basic document structure, the document is accepted for the content validation, where the embedded RDF graph is extracted from it. The extracted RDF triples are then inspected against the business ontology

which contains the core components, business information entities and business document structures.

The content validation is based on a set of inferencing rules and custom modules implemented using the Jena framework [7]. The triples extracted from the document are merged with the business ontology, and the validation rules are applied to the resulting model through inferencing. The inferencing rules for validation are implemented in the Jena rules language. Jena was selected mostly because of its licensing terms and speed; other rule engines could have been used instead. Most rules are designed to detect incorrect references between components (such as using a wrong BBIE in an ABIE), check the cardinalities and data types of literals, but there is also a number of more general rules, such as checks for unknown namespaces. As an example, the following rule is used for data type checks:

```
[DatatypeRangeError:
   (?documentFragment cwa:isFragmentOf ?s)
   (?s ?p ?o)
   (?p rdf:type cwa:BasicProperty)
   (?p rdfs:range ?range)
   notCastableAs(?o, ?range)
   makeSkolem(?errorNode,?p,?o,?range)
->
   (?errorNode rdf:type cwa:Failure)
   (?errorNode cwa:FailureMessage "Datatype error")
   (?errorNode cwa:onProperty ?p)
   (?errorNode cwa:found ?o)
   (?errorNode cwa:expecting ?range)
]
```

This rule infers several new triples describing a node of type `cwa:Failure` if the content is based on a wrong data type. For example, an ill-formed date in a business document would create the following new triples:

```
_:X1 cwa:expecting xsd:date
_:X1 cwa:found '2011-AUGUST-02'
_:X1 cwa:onProperty sos:endDate
_:X1 cwa:FailureMessage 'Datatype error'
_:X1 rdf:type cwa:Failure
```

At the end of the validation process, the inferred model is queried for failures. A report is constructed and returned as a SOAP response. The report contains either a list of failures or the approval of the document. A business document is considered valid if its basic XHTML structure is correct and the document is semantically consistent. A document with about 200 triples is validated in some 300 milliseconds; performance could be additionally improved by storing supported business document schemas in RAM.

5 Discussion

Compared with document exchange based on document type specific XML schemas, our approach has a number of advantages. We list the advantages and illustrate their impact on document exchange and document management.

First, layout information is integrated in the document. There is no need to develop and maintain separate scripts (e.g., XSLT) for producing human-readable versions of the documents. Instead, documents can be viewed in any web browser. At the same time, separate CSS style sheets can be used for fine-tuning the layout, or additional scripts written for converting the documents to other formats, such as PDF/A, which is better suitable for long-time archiving.

Second, all documents are based on a single schema (XHTML+RDFa DTD). This simplifies the management of documents in a single archive used by a number of different information systems. In the Finnish social services system the use of separate schemas for separate document types would mean that the documents in the archive would be based on some 200 schemas, not counting their versions. It is a strong requirement for a single information system to implement support even for a part of them. In the case of a single schema, the system will be able to show the document to the user without resorting to display format generation services, even if it cannot process the semantic markup.

Third, the version management of document structures is easier, because semantics are clearly separated from the document container. A new version of a document does not require a new version of a schema. In a purely schema-based system a new version of a commonly used core component may influence a number of documents and, in turn, force the generation of new versions for their schemas. We note that the management of the core component library itself does not become easier. However, there is no need to keep track of schema namespace changes spawned by the versioning of business information entities.

Finally, semantic markup can be applied only to the information that has clear impact on the automatic processing of information. We noticed that in some social services processes the amount of such information is rather low. This information can be stored as pure XHTML, and semantic markup may be added gradually in the future if the need arises.

A drawback of this approach is that the validation of the document structures becomes more difficult. In the previous section, however, we showed how this drawback can be addressed with closed world validation based on semantic technologies.

There is also a number of new requirements placed on the information systems that need to exchange information. They must produce documents with XTHML+RDFa markup instead of somewhat more straightforward generation of simple XML document instances. Importing the information from a document is also different, as it is not based on parsing the XML structure. Instead, the RDF graph stored in the document must be processed and imported. We note, however, that tool support for such functionality is currently sufficient. Furthermore, the extracted RDF graph can be even transformed to a more traditional XML representation, usable in legacy applications.

6 Conclusion and Future Work

When there is need for the interoperability of several systems, such as customer information systems, document type specific XML schemas are often used. However, if the number of schemas is big, or schemas are updated often, their management might become troublesome. In addition, scripts (e.g., XSLT) for producing human-readable versions of the documents and other schema-specific infrastructure have to be maintained. In the paper, we present a solution for exchanging CCTS-modelled documents using the XHTML+RDFa document format. In social services applications, where the exchange of human-readable content is sufficient in many cases, our solution can facilitate the incremental addition of semantic markup to the documents. Still, the validation of semantic markup which represents CCTS artefacts is possible to aid the interoperability of different systems.

The feedback received from customer information system developers was more in favour of the traditional schema-based approach. However, the system developers found some aspects of XHTML+RDFa, such as the change management model and the possibility for gradual addition of semantic markup, beneficial. The impact of the proposed solution on other aspects of the social services IT infrastructure has still to be analysed in more detail. We assume that the switch to XHTML+RDFa influences the design of at least the following services: end-user interfaces, code list server, digital signatures, statistics and research services, and message communication protocols. Currently, the use of XHTML+RDFa in Finnish social services is still under consideration.

We implemented a prototype of a validation service for XHTML+RDFa documents with CCTS-based content. The service reports problems in both the basic XHTML structure and CCTS content. We plan to develop the service further to improve its report generator, as currently the service does not indicate the line numbers in which ill-formed RDFa attributes and values are located. In addition, we plan to investigate the possibilities of using XHTML+RDFa documents with CCTS-based content in the implementation of business rules.

Acknowledgement. This work was supported by the Tikesos project (National Project for IT in Social Services).

References

1. Adida, B., Birbeck, M., McCarron, S., Pemberton, S.: RDFa in XHTML: Syntax and processing. a collection of attributes and processing rules for extending XHTML to support RDF (2008), http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014
2. Biersteker, H., Hodgson, R.: The Netherlands Ministry of Justice Metadata Workbench: Composing XML message schemas from OWL models. Enterprise Data Journal (May 2010), http://www.enterprisedatajournal.com/article/netherlands-ministry-justice-metadata-workbench-composing-xml-message-schemas-owl-models.htm

3. Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: The semantic web: the roles of XML and RDF. IEEE Internet Computing 4(5), 63–73 (2000)
4. Eriksson, H.: The semantic-document approach to combining documents and ontologies. International Journal of Human-Computer Studies 65(7), 624–639 (2007)
5. Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema (2007), http://www.w3.org/TR/2007/REC-sawsdl-20070828/
6. Finnish Government: Archives Act (831/1994 Arkistolaki) (1994)
7. Jena: A semantic web framework for Java, http://jena.sourceforge.net/
8. Laudi, A.: The semantic interoperability centre europe: Reuse and the negotiation of meaning. In: Charalabidis, Y. (ed.) Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business, pp. 144–161. IGI Global (2010)
9. McCarron, S., Ishikawa, M.: XHTML 1.1 - module-based XHTML. W3C Recommendation, 2 edn. (November 2010), http://www.w3.org/TR/xhtml11/
10. Motik, B., Horrocks, I., Sattler, U.: Bridging the gap between OWL and relational databases. Web Semantics: Science, Services and Agents on the World Wide Web 7(2), 74–89 (2009)
11. Mykkänen, J., Hyppönen, K., Kortelainen, P., Lehmuskoski, A., Hotti, V.: National interoperability approach for social services information management in Finland. In: Charalabidis, Y. (ed.) Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business, pp. 254–278. IGI Global (2010)
12. Nešić, S.: Semantic document model to enhance data and knowledge interoperability. In: Devedžić, V., Gaševič, D. (eds.) Web 2.0 & Semantic Web, vol. 6, pp. 135–160. Springer, US (2009)
13. NIEM Technical Architecture Committee (NTAC): National information exchange model naming and design rules. Version 1.3 (October 2008), http://www.niem.gov/pdf/NIEM-NDR-1-3.pdf
14. OASIS Universal Business Language (UBL) TC: Universal Business Language v2.0 (December 2006), http://docs.oasis-open.org/ubl/os-UBL-2.0/
15. Reiter, R.: On closed world data bases. In: Gaillaire, H., Minker, J. (eds.) Logic and Data Bases, pp. 55–76. Plenum Press, New York (1978)
16. Sirin, E., Tao, J.: Towards integrity constraints in OWL. In: Proceedings of the Workshop on OWL: Experiences and Directions, OWLED 2009 (2009)
17. Thompson, H.S., Beech, D., Maloney, M., Mendelsohn, N.: XML schema part 1: Structures second edition. W3C Recommendation (October 2004), http://www.w3.org/TR/xmlschema-1/
18. United Nations. Centre for Trade Facilitation and Electronic Business: Core components technical specification. Version 3.0 (September 2009), http://unece.org/cefact/codesfortrade/CCTS/CCTS-Version3.pdf
19. United Nations. Centre for Trade Facilitation and Electronic Business: XML naming and design rules technical specification. Version 3.0 (December 2009), http://unece.org/cefact/xml/UNCEFACT+XML+NDR+V3p0.pdf
20. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web 4, 14–28 (2006)