Organizing Federal E-Government Schemas

Erik Wilde[1] and Willy Müller[2]
[1]ETH Zürich (Swiss Federal Institute of Technology)
[2]Informatikstrategieorgan Bund (Federal Office for IT Strategy)

TIK Report 212 (February 2005)

Available at http://dret.net/netdret/publications#wil05e

Abstract

In this paper we present an approach to organize e-government schemas in Switzerland. On the political side, Switzerland is a challenging environment for any federation-wide harmonization and cooperation, because many authorities are organized independently. On the technical side, we describe an approach which aims at increasing the federation-wide cooperation through providing interested parties with a low barrier-to-entry, and with clearly visible benefits through the continuous evolution of a directory of e-government schemas. This paper describes a *light-weight Semantic Web* approach, enabling schema authors to create namespace descriptions that provide a minimal semantic description of the namespace's subject. Using these namespace descriptions, RDF data is extracted and serves as source for a highly interlinked directory of e-government schemas in Switzerland.

1 Introduction

The subject of e-government has become very popular in recent years, partly driven by the advances in private-sector IT integration, and on the other hand by an increasing need to cut costs and still be able to handle IT in an increasingly complex environment. It could be argued that e-government discovered the economic potential and technical possibilities of IT quite a while after the private sector, and therefore issues such as *Enterprise Application Integration (EAI)* and *semantic interoperability* are not yet as popular in the e-government area as they are in the private sector (Quirchmayr and Tagg [18] describe some differences in information integration between the private and the public sector). However, the majority of activities and publications are rather isolated within particular areas or dedicated applications, or they address issues in a closely coupled scenario.

While Switzerland has a centrally planned e-government strategy and associated Web site[1], a 2003 report [19] claimed that e-government implementation and adoption are rather slow. Some of the reasons for this are described in Section 2, and since then attempts has been started to improve the vertical and horizontal cooperation between government authorities.

While there is no central agency defining all e-government standards, it is generally accepted that data is exchanged using *XML* [6], which is defined using *XML Schema* [2, 22].

[1]http//www.e-gov.admin.ch/

However, for successful e-government the description of semantics as described by Klischewski and Jeenicke [12] is essential, otherwise the XML structures cannot be processed beyond validation.

Because full-scale semantic integration is expensive and requires rather close coupling of the players involved, we follow a *light-weight Semantic Web* approach. This means that our semantic descriptions of resources provide a limited understanding of these resources, but one that aims at striking the balance between the effort to create semantic descriptions, and the benefit they offer to users. While we currently target XML Schemas (as described in Section 3), our namespace description approach (described in Section 3.1) is applicable to any vocabulary, and in fact one of the vocabularies used by our approach is an RDF Schema, but also described by a namespace description.

2 E-Government in Switzerland

Historically, Switzerland came into existence as a loose union of small, independent states called *cantons*. They had one thing in common: their struggle for independence. Switzerland of today cannot deny its roots. More rights than in most other countries are delegated to cantons and even communes. There are some 7 million Swiss inhabitants who live in 26 cantons subdivided into 2'873 communes. For example, each commune calculates its own taxes, but taxes are collected by the cantons; they deliver part of it to the communes and another part to the confederation.

E-government in Switzerland has to confront these rather special circumstances. As might be expected, there is no central e-government program and no central funding, but a collection of small and very small independent e-government programs and local initiatives. But nevertheless the different players — communes, cantons, and the confederation — have to work together, even more so than in other countries. That is the reason why *Government to Government (G2G)* is an important aspect of the e-government strategy in Switzerland[2]. G2G is similar to, but not identical with B2B. The legal framework makes the difference. Data privacy laws for example define intricate rules who may access, maintain, and transfer what data.

To help the different players to better interoperate, *eCH* was founded. eCH is an association of the confederation, cantons, communes, universities and IT players with the common goal to promote e-government standards in Switzerland. The hundreds of big, small, and even tiny e-government projects are designing and using more and more XML Schemas, and many of these might be of common interest. In this highly fragmented landscape, even for e-government insiders it is hard to know what others are doing. There is a high probability that things are reinvented in different places on different levels. eCH therefore maintains and provides access to a central view of the work already done, so that others can profit. This central view associates XML Schemas with the corresponding human readable documentation so that others can use them correctly. The first step is to make accessible the XML Schemas that are published by eCH under eCH namespace definitions. Others may be integrated in a second step.

[2] http://internet.isb.admin.ch/internet/egovernment/00677

3 Describing Schemas

While an XML Schema itself is self-containing in the sense that it describes all syntactic constraints for a class of XML documents (possibly importing, including or redefining other XML Schema definitions), it does not convey any information about the semantics of XML documents. In many application scenarios, however, only having an XML Schema without its semantics is not very useful, because documents then can be validated, but not processed any further (which requires an understanding of a document's contents).

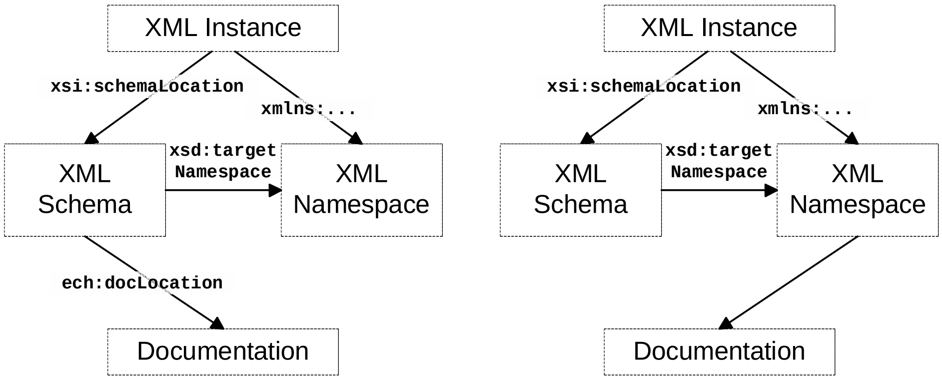

Figure 1: XML Schema Descriptions Access

For most application scenarios, the semantics of a schema are described in some informal way, in most cases using simple prose. The understanding of prose description requires a human, but at least it would be nice to have some standard mechanism how to associate this human-readable description with the XML Schema definition in a standard way. There is no established way for doing this, and there are two main approaches, shown in Figure 1, to solve this problem:

- *Via the Schema:* It is possible to embed additional information within the schema. This could either be done directly using the XML Schema `annotation` element, but this is probably more appropriate for technical schema documentation rather than semantic information intended for users of the schema (this approach is not shown in Figure 1).

  The other possibility is to define attribute(s) on the `schema` element (shown as `ech:docLocation` in Figure 1) that associate the documentation with the schema; very similar to the way in which the `xsi` namespace and the `schemaLocation` attribute associate the schema with an instance.

- *Via the Namespace:* Since the `targetNamespace` of a schema defines a namespace name according to the *XML Namespaces* [5] recommendation, and this namespace name must be a *Universal Resource Identifier (URI)* [1], it is easily possible to associate additional information for a schema through the namespace name (even though technically the namespace name does not need to reference an existing resource). Whether the resource referenced by the namespace name is the documentation itself, or is used to reference the documentation (Figure 1 shows the latter approach) is a question discussed in detail in Section 3.1.

While both approaches solve the problem, we chose to adapt the second approach, because it better separates the schema implementation from the schema description. In order to make this approach work, the eCH rules state that namespace names for eCH schemas must reference existing resources, and that these resources must follow guidelines set by eCH. These guidelines are discussed in the following sections. Because they are generic enough to not only serve as XML Schema descriptions, but as descriptions for any vocabulary associated with a namespace name, we subsequently refer to them as *namespace descriptions*.

3.1 Namespace Descriptions

The question how XML namespaces should be described has been discussed since the namespace recommendation's first publication. Despite the fact that the specification itself does not require any resource to be available at a namespace's URI, it is convenient if this is the case. Several approaches have been used, but even the namespaces of W3C recommendations are described in very different ways, ranging from 404 errors for namespace names (which is perfectly legal) to carefully hand-crafted HTML pages giving references to the XML and XML Namespaces recommendations, and all relevant namespace-specific documents.

The approach of having elaborate HTML pages serving as namespace descriptions is useful for humans, but makes it very hard to process this information automatically. In the environment described in Section 2, however, it would be very beneficial to have machine-readable namespace descriptions, because these could be collected and compiled into a comprehensive database of namespaces and related resources.

In an effort to create a namespace description language to combine the human-readability of HTML documents which machine-readable semantics, the *Resource Directory Description Language (RDDL)* [4] was invented in 2001, mainly intended as a format for namespace documents. RDDL is a simple but ingenious combination of *XHTML* [16] (for human-readable information), *XLink* [7] (for machine-readable information), and a set of predefined roles for describing the semantics of XLinks and the resources they point to. However, even though there was no competition from other formats, RDDL never really caught on.

In January 2004, RDDL 2.0 was released, leaving behind XLink and using two additional attributes to augment XHTML `a` elements with semantics. However, RDDL 2.0 used an entirely non-standard approach to representing semantics. To rectify this problem, the *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)* [8] language was developed, which in addition to RDDL 2.0 also defines a way how to expose the machine-readable information as *RDF* [13]. GRDDL is not yet part of the W3C's standardization activities, but has been published as part of the *W3C Semantic Web Activity*.

Thus, the latest approach to the namespace description problem is GRDDL, which is conceptually very similar to RDDL (embedding machine-readable within human-readable information), but more generalized and also based on RDF rather than XLink. The GRDDL model assumes that the machine readable information of an XHTML page is extracted by using an XSLT program that transforms the GRDDL/XHTML into RDF statements. Even though this adds little to the expressiveness of RDDL, it is a little bit easier to handle (because the machine-readable information can be encoded at the users discretion, as long as it can be transformed into RDF), and may gain more popularity because it uses RDF rather than XLink.

Because GRDDL is the latest and most promising candidate for namespace descriptions, the eCH concept for namespace descriptions is based on GRDDL. Basically, a namespace description is a collection of machine-readable information about how other resources are related to the namespace, such as schemas defining the namespace's vocabulary, documentation for the namespace application, and similar issues. In order to make GRDDL work, this vocabulary of how linked resources related to the namespace must be well-known.

3.2 Description Roles

GRDDL descriptions are not meant to replace any existing documentation, nor is it planned to have schemas being completely documented in GRDDL documents. Instead, the GRDDL namespace description serves as a supplement to the documentation, providing machine-readable documentation that can be used to compile a directory of eCH schemas. To make this directory as rich as possible, we have defined a number of roles that must or can be used to describe schemas. These roles describe how a particular resource relates to the namespace being described.

The following description roles thus constitute that fraction of namespace description that we expect to be available in a machine-readable way, so that it can be collected and processed. Since eCH's goal is to create a useful directory rather than a perfect representation of all available information, we have opted for a rather small number of roles:

- *Namespace:* This is described namespace (for example, an XML Schema's `targetNamespace`), and in our scenario, the namespace description must be accessible through this URI.
- *Schema:* A schema definition must exist for a namespace description to make any sense. The language of the schema is not predefined, but in most cases XML Schema will be used.
- *Documentation:* This refers to human-readable documentation and may either reference Web page or site describing the semantics of the schema, or other references (such as references to books or standards) which do so.
- *Namespace Prefix:* Even though a namespace prefix is a purely local matter to associate a namespace declaration with a qualified name, life is easier for developers if schemas consistently are consistently used with the same namespace prefix. It is therefore possible to define the recommended schema namespace prefix, even though this is only a recommendation and it cannot be guaranteed that the namespace will always be used with this prefix.
- *Transformations:* If there exist transformations (i.e. XSLT programs) for transforming XML documents into other representations (for example HTML pages), then these transformations should be made available through this role.
- *Contact Information:* For many schemas there will be contact information such as persons (through email, Web pages, phone, ...) or other contacts such as working groups, departments, projects, or similar entities which also may be reachable through some URI-referenceable resource.
- *Versioning:* If a schema is a new version of another schema, then this should be made explicit by referring to the previous version.

- *Dependencies:* Other dependencies than versioning may include schema mechanisms such as include, redefine, or import, and are also dependencies that should be made explicit by including them in the namespace description.

Because these resource roles have been chosen to be as easily understandable to schema creators as possible, they may also be interpreted differently by different people. Therefore, each of these roles can be augmented with an additional description, which serves as an additional augmentation of the resource's role and is used to describe the author's intention beyond the basic semantics of the predefined vocabulary.

In fact, because namespace descriptions can define any vocabulary, it would be perfectly possible to describe additional and/or more detailed resource roles with a new vocabulary, and then describe this vocabulary with a namespace description. Thus, our basic vocabulary could be extended without leaving the overall framework of namespace descriptions.

Technically, these resource roles are defined in a simple XML document, which serves as configuration for defining the eCH GRDDL namespace description format described in the following section, and for the RDF Schema used for the information extraction process described in Section 4.

3.3 Creating Descriptions

Schema authors must create GRDDL descriptions for any eCH schemas they are creating and they must follow the guidelines requiring certain kinds of roles to be present. However, schema authors are free how they create the GRDDL. While many choose to write their GRDDL by hand, or simply take another GRDDL and modify it, others don't want to "learn" GRDDL, even though it is very easy to learn (some attributes embedded into XHTML).

For these users, we provide a simple XML Schema (the outline of the schema structure is shown in Figure 2) that can be used to capture all the information required for a GRDDL document, and an *XSL Transformations (XSLT)* [11] program to generate the GRDDL that then serves as namespace description. This schema for namespace descriptions is generated from the list of resource roles described in the previous section.

A second schema — also generated from the list of resource roles described in the previous section — is required for describing the attributes (this schema does not define any elements) that convey the machine-readable information within the GRDDL document. Some of the attributes are defined to appear on XHTML `a` elements, augmenting the link with well-defined semantics. A special case here is the prefix information, which is a string rather than a link to an external resource. Using the `data` URI scheme [14], however, it is possible to express this information using a link, too. Other attributes appear on XHTML `div` elements and are used to enclose textual descriptions. Using these attributes, the eCH namespace description contains all the information that we want to make accessible in machine-readable form.

Figure 3 shows the complete model of how namespace descriptions are used. In this figure, we assume that the namespace description is generated (by using the eCH namespace description schema shown in Figure 2), so that the actual GRDDL document located at the namespace URI of the schema is an XHTML document generated by XSLT. The generated XHTML contains information about associated resources (the schema described, documenta-

[3] Please note that the element in the lower choice all use the `descType` type which is shown in the labeled rectangle with the dashed lines.

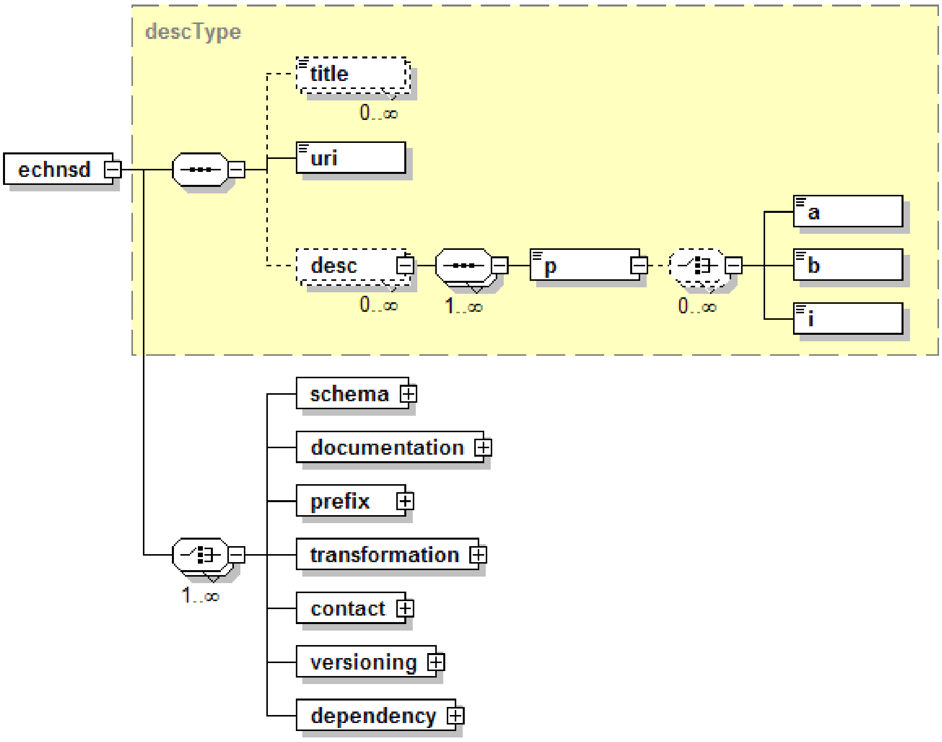

Figure 2: Schema for eCH Namespace Descriptions[3]

tion, transformations, the namespace prefix), and some of these associated resources may be eCH namespace descriptions themselves (versioning and other dependencies).

While we currently only generate GRDDL, it would be easy to update the XSLT program to also generate other machine-readable formats. Figure 4 shows how the XML-based descriptions are described in XML. The XML in the figure is only a fragment of the description, showing the part which links to and describes the schema for the namespace. The GRDDL fragment shows the text tat is being generated from the XML fragment. While the GRDDL uses normal XHTML elements, it also contains attributes from the `echns` namespace, which is the namespace carrying the machine-readable semantics for eCH namespace descriptions. From the GRDDL code containing the `echns` attributes, the next step is to generate RDF, which is shown in the third fragment and described in the following section.

4 Harvesting Descriptions

Harvesting descriptions is a task that requires all namespace descriptions to be located and collected for further processing. Namespace descriptions can be found by reference (known namespace descriptions that point to previously unknown namespace descriptions) or through submitting them to the harvesting process. Submission is necessary because a simple crawler would depend on a fully connected network of namespace descriptions, but instead it is possible that some namespace descriptions are not referenced by others and therefore would never be found by a crawler.

Harvesting descriptions is rather easy and efficient, because we strongly suggest schema

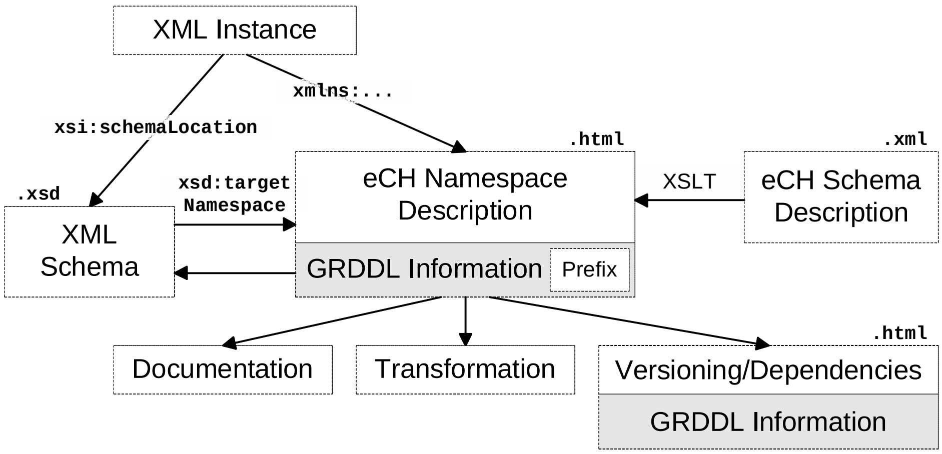

Figure 3: eCH Namespace Description Model

authors to store their namespace descriptions on the eCH Web server. Resources pointed to from the namespace descriptions may be located on other servers, but at least the machine-readable namespace descriptions are on one central server where they can be managed and archived.

As pointed out in Section 3.1, namespace descriptions are GRDDL documents, which means they are XHTML with embedded semantic information. This makes automated processing of namespace descriptions rather easy, because the XHTML can be ignored, and only the semantic information has to be extracted. This task is best done by either XSLT or *XQuery* [3], and we decided to use XSLT because it is stable and well-known to many XML-oriented developers. This means that harvesting eCH namespace descriptions means collecting GRDDL documents, and then use XSLT to generate RDF from these documents.

Figure 5 shows the full RDF graph of the eCH namespace description shown in Figure 4. It is obvious that the RDF graph is not very complex, but it contains all information necessary to compile the namespace description directory that has been the goal of eCH. The RDF Schema for the RDF is generated from the list of resource roles described in the Section 3.2.

As pointed out in Section 3.3, we do not require schema authors to generate their namespace description. They can generate it, in which case the schema and the XSLT for the generation will guarantee an error-free namespace description. However, if the namespace description is generated by hand (either by scratch or by copying and modifying an existing documentation), errors may be introduced, so that the harvesting also needs to check whether the harvested documents are valid.

Validity in our context means that the harvested descriptions must be valid GRDDL, and that they satisfy all requirements for eCH namespace descriptions as detailed in Section 3.2. If harvested descriptions are invalid, they are excluded from further processing stages and the description originator is contacted, if possible. Contacting the description originator and having the description corrected is a manual process, but it does not interfere with the overall process of harvesting and subsequently publishing all valid namespace descriptions.

Validation (as well as RDF generation) in the current implementation is done using an *XSLT 2.0* [11] program. The reason for this is that XML Schema does not provide a reasonable way of validating a schema that is tightly integrated with a host language, and that XSLT 1.0

eCH Namespace Description XML:

```
<schema type="xsd" id="xsd">
 <title lang="en">XML Schema for eCH Namespace description</title>
 <uri>http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd</uri>
 <desc lang="title">
  <p>The schema for eCH Namespace description ...</p>
 </desc>
</schema>
```

eCH Namespace Description GRDDL:

```
<h4 echns:title="http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd"
    xml:lang="en">XML Schema for eCH Namespace description</h4>
<h4>URI:
    <code><a href="http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd"
    echns:role="schema" echns:type="xsd">
    http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd</a></code></h4>
<fieldset>
 <legend>Description</legend>
 <div xml:lang="en"
  echns:description="http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd">
  <p>The schema for eCH Namespace description ...</p>
 </div>
</fieldset>
```

eCH Namespace Description RDF:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://www.ech.ch/echnsd/echnsd-rdf/1#">
 <namespace rdf:about="http://www.ech.ch/echnsd/echnsd-ns/1">
  <schema
   rdf:resource="http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd"/>
 </namespace>
 <rdf:Description
  rdf:about="http://www.ech.ch/echnsd/echnsd-ns/echnsd-ns.xsd">
  <type>xsd</type>
  <description rdf:parseType="Literal" xml:lang="en">
   <p>The schema for eCH Namespace description ...</p>
  </description>
  <title xml:lang="en">XML Schema for eCH Namespace description</title>
 </rdf:Description>
</rdf:RDF>
```

Figure 4: eCH Namespace Description (XML → GRDDL → RDF)

does not provide any support for checking datatypes. The result of the XSLT-based validation is a report containing a list of warnings or errors raised during the validation process.

5 Publishing Descriptions

After the harvesting and validation process, the GRDDL documents are processed using XSLT, which after joining the individual RDF graphs results in a single RDF graph describing the harvested eCH namespace descriptions. This aggregated collection of descriptions is published as XML and XHTML. By using these two formats, eCH enables users to access the

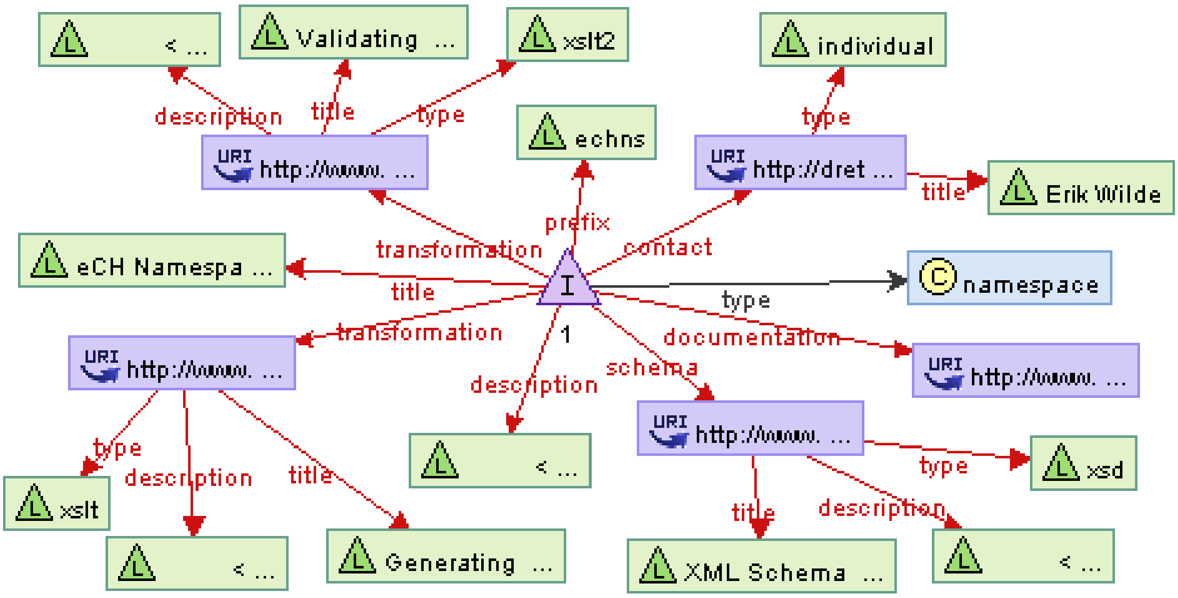

Figure 5: eCH Namespace Description RDF Graph

information automated or manually.

The XHTML pages are published as heavily crosslinked pages, enabling users to retrieve all the information present in the RDF using a regular browser. Using search features, it is possible to search for specific text in all literal information, so that access through the XHTML pages is provided through search-based retrieval as well.

For users interested in a machine-readable description of the collected data, eCH publishes the data as XML. This XML uses more traditional structures using standard ID/IDREF references rather than being based on RDF. Even though eCH uses an RDF-based data model, it was decided that an application-specific XML Schema is better suited for representing the eCH namespace descriptions. The reason for this is that RDF is not a very popular format, and is not well-suited for processing it with XPath/XSLT, whereas a suitably designed XML can be processed very simply by users with relatively little XPath/XSLT experience.

The publishing process therefore adds a fourth stage to the three representations shown in Figure 4, which is also implemented using XSLT 2.0. However, we plan to move to a RDF-specific method, because the XSLT programs do not operate on the abstract RDF graph model but on the XML syntax, and can easily be broken if the RDF serialization changes. Moving to an RDF-specific method would therefore improve the robustness of the overall system and also enable us to easily join other RDF graphs to our generated RDF, an issue described in Section 6.2.

6 Discussion

Switzerland's situation is a bit different from many other countries because of the country's strong and important federal political system. Therefore, in Switzerland it is harder than in many other countries to practice in top-down approach for organizing e-government issues.

6.1 Related Work

In the area of e-government, many countries have developed strategies and programs to accelerate the speed of introducing IT in the public sector. An interesting example relevant to the work presented in this paper is Britain's *e-Government Interoperability Framework (e-GIF)*[4], which explicitly addresses the area of improving the interoperability of e-government activities. Within this Framework, the *e-Government Metadata Standard (e-GMS)* [15] uses *ISO Dublin Core* [9] metadata elements to semantically describe resources. e-GIF also has a schema library containing some seventy schemas, all of them documented using e-GMS metadata and often accompanied by additional resources, such as documentation. However, there is no metadata describing the relations between schemas and other resources (apart from a very general classification scheme used for all schemas), and therefore the schema library is an unstructured set of schemas.

Similar to this is New Zealand's *e-Government Interoperability Framework (e-GIF)*[5], which also has some provisions about data formats and resource metadata, but does not yet make any attempts to tackle the information integration problem on the semantic level. As a third example, the European Union's *Interchange of Data Between Administrations (IDA)*[6] program focuses on interoperability, but so far also is more concerned with technical interoperability issues than with semantic interoperability. As a last example for government-based activities, the US E-Gov Framework[7] with its *Federal Enterprise Architecture (FEA)* and the associated *FEA Management System (FEAMS)* is based on manual collection and compilation of FEA-related material, which is then made available through a Web-based application.

As mentioned in the introduction, it can be observed that the e-government area is lagging behind the private sector in terms of EAI[8]. This is mainly caused by the size and complexity of the public sector, by the huge legacy of traditional technologies and procedures, by sometimes complex legal regulations, and by the more indirect economic pressures.

Coming from the technical side, approaches such as the EU's *OntoGov* project[9] employ the strong approach of applying full-scale AI technologies. From this project, a publication by Stojanovic et al. [21] describes this strong approach to managing e-government, based on a variety of ontologies describing all relevant resources and their relations as well as the ongoing evolution of the e-government. A variety of other strong approaches is described in Wimmer [23], but most of them favor semantic richness over loose coupling and easy of use.

6.2 Future Work

The primary goal of the eCH activity was to compile a directory of namespace descriptions for the variety of schemas being used in Swiss e-government activities. We therefore use the RDF generated from the machine-readable namespace descriptions for internal purposes only. Future plans for eCH activities include more activities towards semantic descriptions, and the RDF that we now use internally could either exposed to the general public, so that others

[4] http://www.govtalk.gov.uk/schemasstandards/egif.asp
[5] http://www.e-government.govt.nz/interoperability/
[6] http://europa.eu.int/ida/
[7] http://www.whitehouse.gov/omb/egov/
[8] Most e-government activities focus on *Government to Citizen (G2C)* rather than G2G.
[9] http://www.ontogov.com/

can integrate into their semantic frameworks, or it could become part of a larger initiative towards Semantic Web technologies for e-government in Switzerland.

We are closely monitoring the development of more advanced RDF-based technologies, such as the *Web Ontology Language (OWL)* [20] and SPARQL [17]. While we currently think that the RDF-based directory approach is appropriate, further advances in RDF technologies (such as standardized APIs or query languages) or wider acceptance of RDF-based technologies may influence our future strategy for semantics description and dissemination.

6.3 Contributions

While the activities described in this paper served a very practical purpose, we also feel that the combination of different technologies and a right choice of tools can help to achieve easier ways to collaborate in federal environments, where it is always a challenge to find the right balance between common standards and individually chosen technologies. In particular, the eCH namespace description makes the following contributions:

- Our namespace description approach implements the idea of a *light-weight Semantic Web*, searching for the middle ground between the considerable effort necessary to create machine-readable descriptions for many details of an IT environment, and the absence of any machine-readable description in the plain namespace handling defined by the recommendation.

- In an effort to implement the light-weight Semantic Web as easily, standards-compliant and future-proof as possible, we employed a wide variety of Web technologies such as XML, XML Schema, XSLT, GRDDL, RDF, and RDFS. Using these technologies and combining them in the most efficient way enabled us to implement what we consider to be a gap in the XML landscape of today without too much effort.

We are currently in the starting phase and therefore were so far not able to test the extensibility of our concept (regarding the resource roles). However, since we generate a very simple XML-based description format from the list of resource roles, acceptance levels for our approach are high, and we are confident that we will be able to adapt to future requests for adding additional semantics to our framework.

7 Conclusions

In this paper, we present a generic framework for describing namespaces. Our application area are e-government schemas, but we believe that our framework is flexible enough to provide support in other application areas as well. The current Web architecture [10] does not define a data format for namespace descriptions, but recommends that namespace names should point to some kind of description. Our GRDDL-based approach follows this basic recommendation, and adds additional facets such as a controlled vocabulary (which is configurable), and a harvesting process joining a given set of namespace descriptions into an RDF graph representing the semantics of the resources associated with the namespace descriptions.

References

[1] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. Internet draft standard RFC 2396, August 1998.

[2] Paul V. Biron and Ashok Malhotra. XML Schema Part 2: Datatypes Second Edition. World Wide Web Consortium, Recommendation REC-xmlschema-2-20041028, October 2004.

[3] Scott Boag, Don Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie, and Jérôme Siméon. XQuery 1.0: An XML Query Language. World Wide Web Consortium, Working Draft WD-xquery-20050211, February 2005.

[4] Jonathan Borden and Tim Bray. Resource Directory Description Language (RDDL), February 2002.

[5] Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in XML. World Wide Web Consortium, Recommendation REC-xml-names-19990114, January 1999.

[6] Tim Bray, Jean Paoli, C. Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible Markup Language (XML) 1.0 (Third Edition). World Wide Web Consortium, Recommendation REC-xml-20040204, February 2004.

[7] Steven J. DeRose, Eve Maler, and David Orchard. XML Linking Language (XLink) Version 1.0. World Wide Web Consortium, Recommendation REC-xlink-20010627, June 2001.

[8] Dominique Hazaël-Massieux and Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). World Wide Web Consortium, W3C Coordination Group Note NOTE-grddl-20040413, April 2004.

[9] International Organization for Standardization. Information and Documentation — The Dublin Core Metadata Element Set. ISO 15836, November 2003.

[10] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One. World Wide Web Consortium, Recommendation REC-webarch-20041215, December 2004.

[11] Michael Kay. XSL Transformations (XSLT) Version 2.0. World Wide Web Consortium, Working Draft WD-xslt20-20050211, February 2005.

[12] Ralf Klischewski and Martti Jeenicke. Semantic Web Technologies for Information Management within e-Government Services. In *Proceedings of the 37rd Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 2004. IEEE Computer Society Press.

[13] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium, Recommendation REC-rdf-syntax-19990222, February 1999.

[14] Larry Masinter. The "data" URL scheme. Internet proposed standard RFC 2397, August 1998.

[15] Office of the e-Envoy. e-Government Metadata Standard Version 3.0, April 2004.

[16] Steven Pemberton. XHTML 1.0: The Extensible HyperText Markup Language. World Wide Web Consortium, Recommendation REC-xhtml1-20000126, January 2000.

[17] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. World Wide Web Consortium, Working Draft WD-rdf-sparql-query-20041012, October 2004.

[18] Gerald Quirchmayr and Roger Tagg. An Architectural Concept for Knowledge Integration in Inter-Administration Computing. In Maria A. Wimmer, editor, *Proceedings of 3rd International Workshop on Knowledge Management in e-Government*, volume 7 of *Schriftenreihe Informatik*, pages 67–77, Copenhagen, Denmark, May 2002. Trauner Verlag.

[19] Kuno Schedler and Lukas Summermatter. 2. Bericht zum Stand von E-Government in der Schweiz 2003 — Teil Bund. Technical Report IDT-Working Paper Nr. 8c, Institut für Öffentliche Dienstleistungen und Tourismus, Universität St. Gallen, St. Gallen, Switzerland, February 2004.

[20] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. OWL Web Ontology Language Guide. World Wide Web Consortium, Recommendation REC-owl-guide-20040210, February 2004.

[21] Ljiljana Stojanovic, Andreas Abecker, Nenad Stojanovic, and Rudi Studer. On Managing Changes in the ontology-based E-Government. In Robert Meersman and Zahir Tari, editors, *Proceedings of 2004 CoopIS, DOA, and ODBASE, OTM Confederated International Conferences*, volume 3291 of *Lecture Notes in Computer Science*, pages 1080–1097, Agia Napa, Cyprus, October 2004. Springer-Verlag.

[22] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. XML Schema Part 1: Structures Second Edition. World Wide Web Consortium, Recommendation REC-xmlschema-1-20041028, October 2004.

[23] Maria A. Wimmer, editor. *Proceedings of 5th IFIP International Working Conference on Knowledge Management in Electronic Government*, volume 3035 of *Lecture Notes in Computer Science*, Krems, Austria, May 2004. Springer-Verlag.