Re-format page sizes

Large scale validation of millions of UBL Invoices with XML Schema and Schematron

Mikkel Hippe **Brun**

Brian **Nielsen**

Christian **Lanng**

Bryan **Rasmussen**

November 2005

Abstract

Since February 1st 2005, millions of invoices have been exchanged between the private sector and the public sector in Denmark. This paper focuses on real life problems, experiences and solutions with syntactical and semantical validation of millions of electronic invoices. Localization and documentation for regional and national use is a massive and important assignment. I.e. decisions on the use of identifiers have to be specified and local payment methods must be mapped to the international standard. The result is a message with many internal integrity constraints that cannot be validated with the UBL schemas alone. In order to provide even stronger validation, non-normative supplementary schemas have been developed. These schemas perform stronger validation based on decisions about the use of national identifiers for companies and persons. In addition to the use of XML schema – Schematron is used for the validation of internal referential integrity constraints. Experiences and theoretical considerations on the localization of international vocabularies are discussed.

RenderX
XSL • FO
formatter

Re-format page sizes

Table of Contents

RenderX
XSL • FO
formatter

Re-format page sizes

1. Electronic invoicing in Denmark

1.1. Background

Denmark has since February 1st 2005, mandated the exchange of electronic invoices between the private sector and the public sector. The initiative has been backed by legislation [VTUstatute] and it is probably worldwide one of the first XML Schemas (and perhaps the last), which has been printed on yellow paper in a book of law. A law text is not a suitable medium to document an exchange format such as an invoice. The effect of the law has never the less been that the whole public sector now receives invoices electronically. A large task remains in harvesting all of the benefits with the initiative. Larger public institutions must implement effective invoice approval workflows. Public opinion has been positive to the overall idea, but delayed payments due to ineffective workflows has also lead to frustration with private companies. The Danish Agency for Governmental Management (Økonomistyrelsen) [Link] is working intensely with all parts of the public sector in order to guide them on the implementation of effective workflows. Private companies are also experiencing a high demand from the public sector on systems that can automate the workflow with features like automatic clearing of invoices with registered orders.

1.2. Business case

The business case behind the initiative [BC] was developed by KPMG on behalf of the Ministry of Finance. The business case showed that more than 10 minutes could be saved in the handling of each invoice once a public sector institution has implemented an electronic workflow. The total volume of invoices is approximately 18 million and this is equivalent to a potential savings of more than 2000 man-years of work per year or approximately 117 million USD. The exact figures can be disputed and has been disputed [Deloite], but the fact remains that the initiative has boosted the exchange of electronic business documents. Private companies as well as public institutions are now pushing for the introduction of more business documents to support an extended procurement process.

1.3. The network infrastructure and addressing mechanism

The network infrastructure for exchanging invoices is based on traditional EDI technology. A network of VANS[1] operators handles the delivery of invoices from supplier to buyer. The addressing mechanism uses EAN-location numbers to identify the trading partners.

All public institutions are required by law to be connected to one of the five VANS operators. Private companies wishing to send invoices to the public sector are likewise required to be connected to the VANS-network. The VANS operators share a database of EAN location numbers.

1. An invoice is sent to a VANS-operator
2. The VANS-operator looks up the EAN-location number in a database
3. The invoice is perhaps sent to another VANS-operator
4. The invoice is sent to a public authority

[1] VANS - Value Added Network Service

RenderX
XSL • FO
formatter

Re-format page sizes

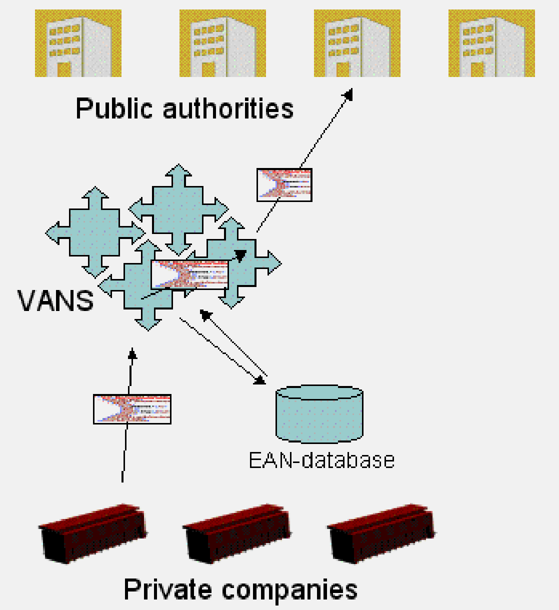


Figure 1. The infrastructure of the VANS-network

1.4. The numbers

The development in the numbers of invoices exchanged have to a large extend meet expectations. However the number of invoices received by Scanning Agencies is still relatively high. A number of barriers to full electronic invoicing have been identified in relationship with small and medium sized enterprises. The addressing of these barriers is however outside the scope of this article.

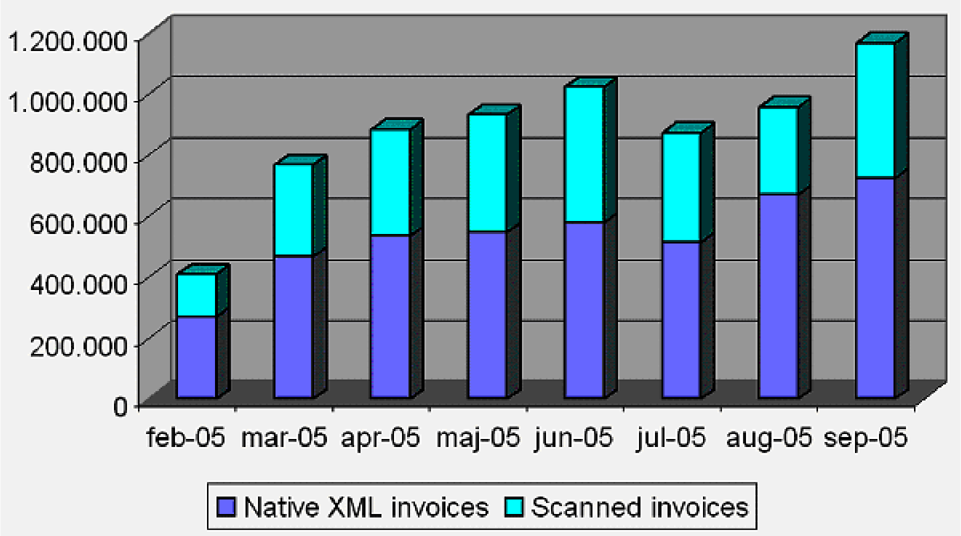


Figure 2. The development in the number of invoices

RenderX
XSL • FO
formatter

Re-format page sizes

1.5. UBL is gaining momentum in Europe

The Danish invoice format is a localized version of the UBL invoice and it is based on an early release (version 0.7) from the UBL TC. The decision to use UBL 0.7 was due to time constraints in the legislative process. There was not enough time between UBL 1.0 was approved as an OASIS standard on November 8th 2004 and the time when the Danish legislation had to be ready on November 11th.

Denmark is not the only adopter of UBL in Europe. A Spanish localization subcommittee has existed for several years and a number of sector-specific adoptions has also exist for some time.

Recently the Swedish National Financial Management Authority (ESV) [http://www.esv.se] announced [Swe] that 4 billion SEK could be saved during a five-year period in the handling of 11 million invoices. The format proposed is called "Svefakturan" and is based on UBL 1.0. ESV intends to start developing a package solution for the agencies by procuring a framework agreement for a switchboard (VANS provider) for electronic invoicing along with IT-products and support. The solution should be ready by the end of 2006.

The European commission has also indirectly played an active role via its IDABC [http://europa.eu.int/idabc/en/home] program in providing input to the next version of UBL 2.0. The eProcurement Experts Group provided a list of 120 requirements to UBL 2.0 and these requirements were discussed with members of the UBL TC in a number of face-to-face meetings and telephone conference calls. Especially especially the UK Office of Government Commerce and the Norwegian Ministry of Modernization put much effort into this work. The UBL TC has now accepted most of these requirements, thus ensuring that the next version of UBL meets the demands of Europe, and has the potential of gaining widespread adoption.

2. Validation: Problems and solutions

2.1. Preconditions causing confusion about validation

The schemas referenced in the Danish legislation were based on the original 0.7 UBL schemas. An extra set of strongly validating schemas was also supplied, but the use of these schemas was not required. They were supplied as tools to aide the developers in their development. Strictly speaking you could not claim to be in conformance with the requirements of the law without your instance being able to validate with these supplementary schemas. But even with the strict schemas it was possible to produce instances that were not in accordance with the legislation. The reason for this being that the XML Schema could not express all the business rules and integrity constraints needed for an invoice used in our scenario. But such technicalities were very often lost in the communication. Many developers blindly trusted the "lax" schemas validate instances as being conformant to the law.

RenderX
XSL • FO
formatter

Re-format page sizes

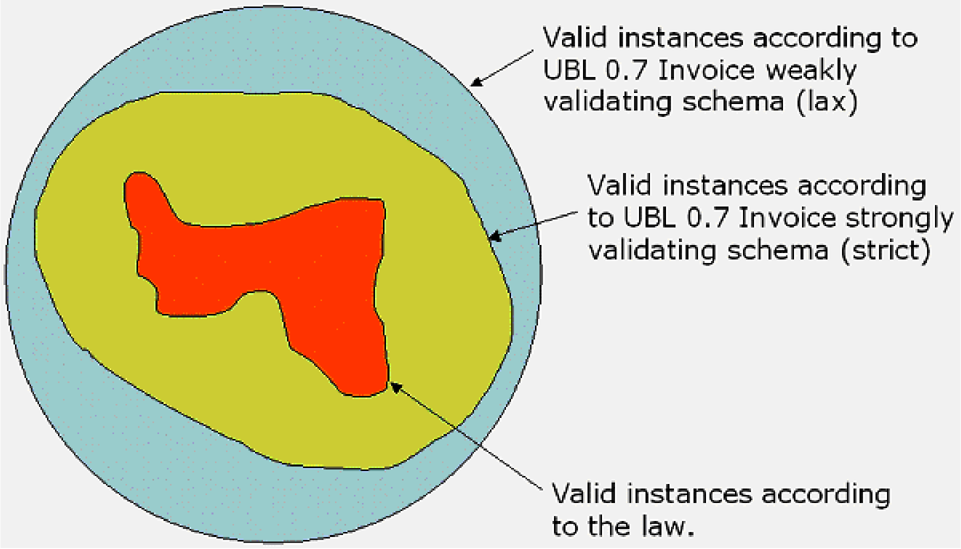

Valid instances according to
UBL 0.7 Invoice weakly
validating schema (lax)
Valid instances according
to UBL 0.7 Invoice strongly
validating schema (strict)
Valid instances according
to the law.

Figure 3. Original schemas supplied for developers. The "strict" schema is a subset of the "lax" schema.

Ideally there would only have been one set of schemas and less confusion. But the speed in which the legislation was rolled out did not allow us go through an official standardization process with the strongly validating schemas.

2.2. Developers with little or no prior exposure to XML and XML Schema

It may be that XML has conquered the world and that XML is everywhere, but that does not imply that all developers has worked with XML or even understands the most fundamental aspects of XML and validation.

In the first weeks after February 1st - invoices were sent that were not even well formed. The invoices had been generated with tools that did not support XML, and the developers had obviously not even tried to validate the messages. This caused enormous problems in the VANS-network because the VANS-operator could not determine whom an invoice was meant to be sent to.

Other developers blindly trusted the parser. They believed that their instances were in conformance with the law when they got an "ok" from the parser. But this was most often not the case. The XML Schema can of course only validate the syntax, and with the UBL Schemas to a lesser extend the data types.

All weaknesses in the schemas that could be abused were abused. For instance obligatory elements that have string or tokenized string data types were not constrained to be non-empty.

```
<com:ReferencedOrder>
   <com:BuyersOrderID/>
   <com:SellersOrderID/>
   <com:IssueDate>0001-01-01</com:IssueDate>
</com:ReferencedOrder>
```

Like in the example above developers will often just leave obligatory elements as empty or use default values like a useless date, and the XML Schema will accept this.

Re-format page sizes

2.3. Parsing in VANS network due to missing envelope for messages

Although traditional EDI technology was reapplied to XML-based electronic invoicing, experiences about enveloping were missed. EAN locations numbers were used as the central addressing mechanism, but the VANS operators had not agreed on an envelope format for the messages. This meant that each VANS-operator had to retrieve the destination EAN location number from within each message. Of course this solution proved to be prone to errors and the VANS-operators had severe problems in the beginning with not being able to locate EAN locations numbers in invalid or not well-formed messages. Thousands of messages had to be manually handled.

This problem is now being addressed with the implementation of an ebMS based enveloping mechanism. This means that the addressing information will now be part of the ebMS header. This model should of course have been in place at the outset of the implementation. The VANS network will now be able to handle any message format that can be placed in an ebMS envelope. The new ebMS enveloping format including positive and negative acknowledgments will be in operation as of December 1st 2005.

This problem is now be

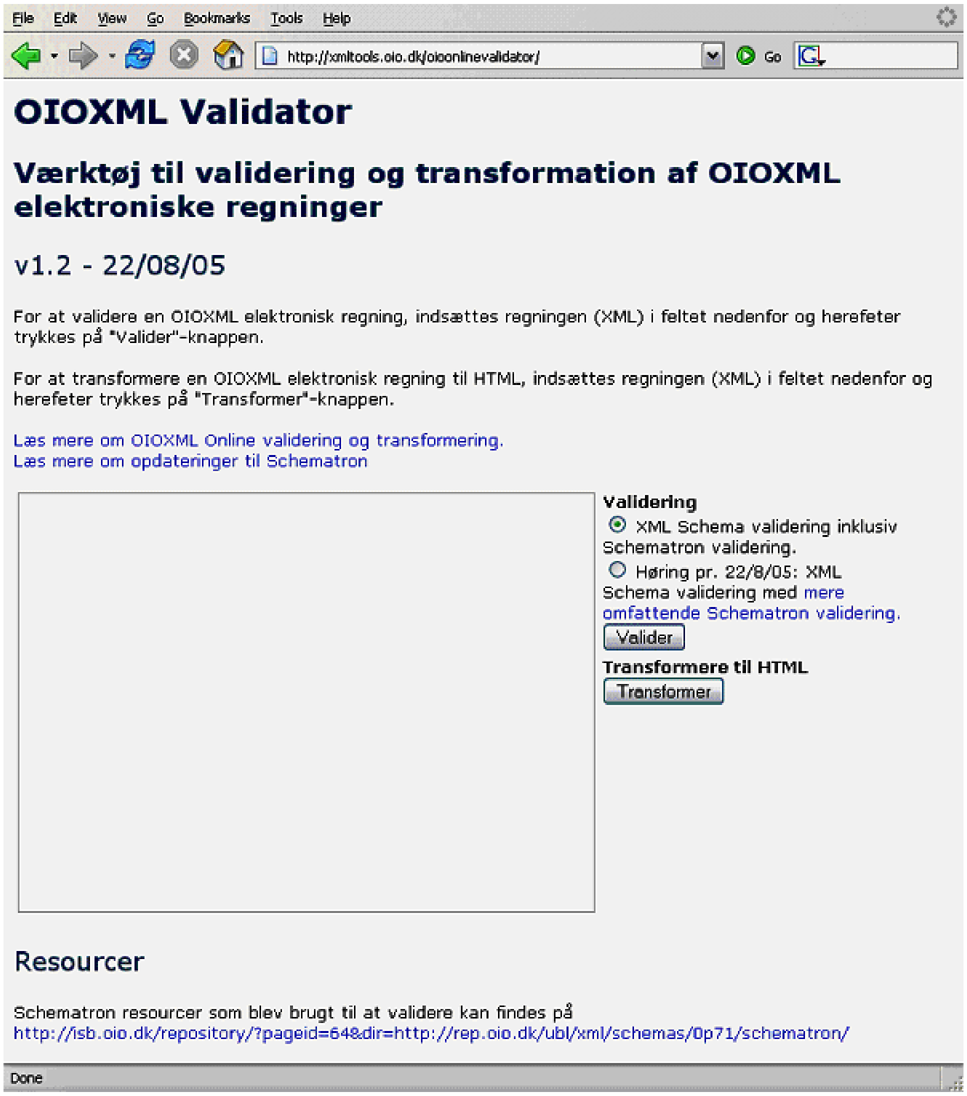
2.4. Online validation tool

The problem with the developer community's inability to work with XML and XML Schema was addressed by developing an online validation tool. The tool allows developers to post an instance to a central server using an html form, and de server will reply any detected errors or warnings.

The tools has been extremely valuable to the developer community and the need for support dropped drastically once we made it a requirement that developers had tested their instances prior to contacting us with questions.

RenderX
XSL • FO
formatter

Re-format page sizes



OIOXML Validator

Værktøj til validering og transformation af OIOXML elektroniske regninger

v1.2 - 22/08/05

For at validere en OIOXML elektronisk regning, indsættes regningen (XML) i feltet nedenfor og herefeter trykkes på "Valider"-knappen.

For at transformere en OIOXML elektronisk regning til HTML, indsættes regningen (XML) i feltet nedenfor og herefeter trykkes på "Transformer"-knappen.

Læs mere om OIOXML Online validering og transformering.
Læs mere om opdateringer til Schematron

Validering

● XML Schema validering inklusiv Schematron validering.

○ Høring pr. 22/8/05: XML Schema validering med mere omfattende Schematron validering.

Transformere til HTML

Resourcer

Schematron resourcer som blev brugt til at validere kan findes på http://isb.oio.dk/repository/?pageid=64&dir=http://rep.oio.dk/ubl/xml/schemas/0p71/schematron/

Figure 4. Online validation tool.

The online validation tool is now the de facto reference implementation on validation.

2.5. Schematron validation

Almost any large XML standard is bound to define connections between elements and attributes that cannot be modeled in XML Schema. In UBL the most common occurrence of this kind of connection is that an attribute on an element containing a textnode will be used to define the type of the textnode's content. In order to check as many integrity rules as possible, a third validation tool was developed using Schematron. This tool allows contingent validation.

Once we started using Schematron for determining the type of an element's textual content we discovered that some of the various types defined were normally understood to have algorithmic requirements to determine their validity. For example an EAN location number has a specific algorithm (modulus 10 calculation) to determine its validity. The following example demonstrates how this validation check is performed using Schematron.

RenderX
XSL • FO
formatter
]

Re-format page sizes

```
<sch:rule context="*[@schemeID]">
    <sch:report test="@schemeID='EAN' and string-length(.) != 13">
    WARNING: EAN numbers are 13 digits in length</sch:report>

    <sch:report test="@schemeID='EAN' and . != (. + 1) - 1">
    WARNING: EAN numbers are 13 digits in length</sch:report>

    <sch:report test="@schemeID='EAN' and substring(.,13,1)!=0 and ((((10 -
       substring((substring(.,1,1) * 1 + substring(.,2,1) * 3) +
      (substring(.,3,1) * 1 + substring(.,4,1) * 3) + (substring(.,5,1) * 1 +

       substring(.,6,1) * 3) + (substring(.,7,1) * 1 + substring(.,8,1) * 3)
+
      (substring(.,9,1) * 1 + substring(.,10,1) * 3) + (substring(.,11,1) * 1
 +
       substring(.,12,1) * 3),string-length((substring(.,1,1) * 1 +
       substring(.,2,1) * 3) + (substring(.,3,1) * 1 + substring(.,4,1) * 3)
+
      (substring(.,5,1) * 1 + substring(.,6,1) * 3) + (substring(.,7,1) * 1 +

       substring(.,8,1) * 3) + (substring(.,9,1) * 1 + substring(.,10,1) * 3)
 +
      (substring(.,11,1) * 1 + substring(.,12,1) * 3)),1)) +
     ((substring(.,1,1) * 1 + substring(.,2,1) * 3) + (substring(.,3,1) * 1 +

       substring(.,4,1) * 3) + (substring(.,5,1) * 1 + substring(.,6,1) * 3)
+
      (substring(.,7,1) * 1 + substring(.,8,1) * 3) + (substring(.,9,1) * 1 +

       substring(.,10,1) * 3) + (substring(.,11,1) * 1 +
       substring(.,12,1) * 3))) - ((substring(.,1,1) * 1 +
       substring(.,2,1) * 3) + (substring(.,3,1) * 1 + substring(.,4,1) * 3)
+
      (substring(.,5,1) * 1 + substring(.,6,1) * 3) + (substring(.,7,1) * 1 +

       substring(.,8,1) * 3) + (substring(.,9,1) * 1 + substring(.,10,1) * 3)
 +
      (substring(.,11,1) * 1 + substring(.,12,1) * 3))) != substring(.,13,1)
)">
    WARNING: there is an improperly formatted EAN number.</sch:report>

    <sch:report test="@schemeID='EAN' and substring(.,13,1) =0 and
       substring((substring(.,1,1) * 1 + substring(.,2,1) * 3) +
      (substring(.,3,1) * 1 + substring(.,4,1) * 3) + (substring(.,5,1) * 1 +

       substring(.,6,1) * 3) + (substring(.,7,1) * 1 + substring(.,8,1) * 3)
+
      (substring(.,9,1) * 1 + substring(.,10,1) * 3) + (substring(.,11,1) * 1
 +
       substring(.,12,1) * 3),string-length((substring(.,1,1) * 1 +
       substring(.,2,1) * 3) + (substring(.,3,1) * 1 + substring(.,4,1) * 3)
+
      (substring(.,5,1) * 1 + substring(.,6,1) * 3) + (substring(.,7,1) * 1 +
```

RenderX
XSL • FO
formatter

Re-format page sizes

```
        substring(.,8,1) * 3) + (substring(.,9,1) * 1 + substring(.,10,1) * 3)
 +
       (substring(.,11,1) * 1 + substring(.,12,1) * 3)),1) != 0">
     WARNING: there is an improperly formatted EAN number. </sch:report>
</sch:rule>
```

The Schematron validation was quickly implemented as part of the online validation tool and the Schematron rules has gradually been extended and improved. The developer community has been very positive about the tools but only a minority of developers has themselves implemented Schematron validation in their programs.

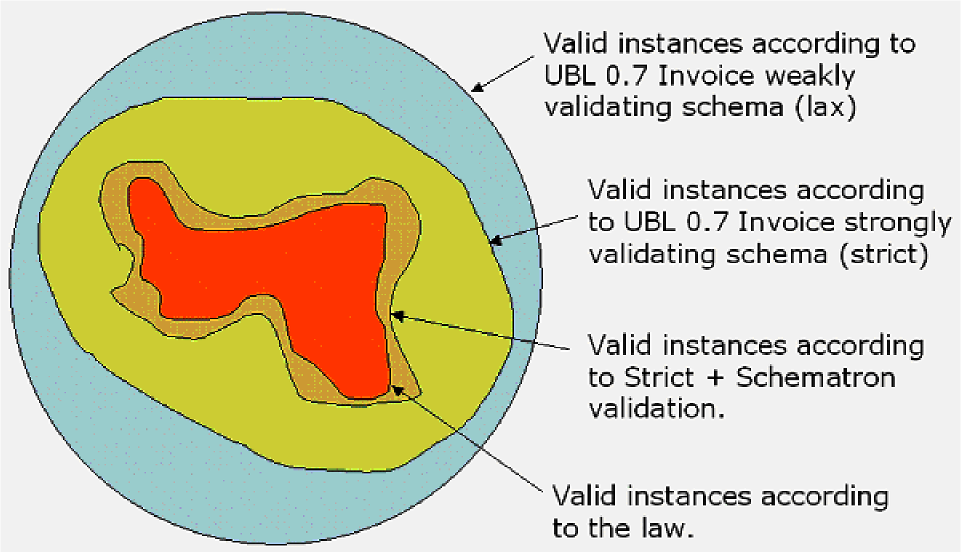

Valid instances according to UBL 0.7 Invoice weakly validating schema (lax)
Valid instances according to UBL 0.7 Invoice strongly validating schema (strict)
Valid instances according to Strict + Schematron validation.
Valid instances according to the law.

Figure 5. The figure illustrates how the constraints on the content of an invoice instance become tighter and tighter as stronger validation is employed. The strict validation, supplemented by Schematron validation ensures a closer coherence with the statute than the lax validation or strict validation alone.

The Schematron schemas are provided in two forms:

- Native Schematron schemas
- An XSLT stylesheet

Schematron has capabilities that will not work the same in all applications, for example if one uses the XSLT implementation it is possible to use the XSLT functions not defined in Xpath.

In order to make Schematron schemas that would work in non-XSLT implementations of Schematron (despite the fact that we were using the XSLT implementation) we made sure to keep ourselves to the absolute simplest Schematron set of functionalities, just rules, reports, assertions, and pure XPATH. This often led to the Schematron schema being more verbose than necessary if focused on an XSLT implementation only.

The Danish developer community is gradually seeing the value of Schematron based validation. Some implementations allows for Schematron to be an extension on XML Schema where the Schematron schema fragments are imbedded as application info in the XML Schema. This is attractive for documentation purposes because there is a direct relationship between the XML Schema construct and the extended validation performed by Schematron. A simple stylesheet can

RenderX
XSL • FO
formatter

then extract the Schematron fragments into a single document that can be handled by other implementations. The imbedded Schematron will not affect the XML Schema validation.

3. Conclusion

The ability to introduce a "syntax wall" with XML schemas in the hart of our network infrastructure was crucial to the success of the project. The "syntax wall" allowed us to tighten the performed validation and thus the quality of invoices exchanged. We would to this day have had enormous problems with bilateral discussions with developers about the quality of their invoices if it were not for this mechanism.

Having exploited Schematron in the validation of schemas has also been extremely valuable. We have had the ability to gradually improve the Schematron validation, and give developers better feedback on their instances. Schematron validation is a promising candidate for being mandated in the next roll out of procurement messages based on UBL.

Acknowledgements

Many thanks to our colleagues at the Danish Agency for Governmental Management for valuable discussions on our approach to validation.

Bibliography

[FMstatute] *[Danish] Statute on electronic invoicing to public authorities. [http://purl.oclc.org/net/oioxml/ubl/einvoice/FinanceStatute]* Danish title: Bekendtgørelse om elektronisk afregning med offentlige myndigheder (Bekendtgørelse nr. 991 af 7. oktober 2004). October 7th 2004

[KPMG] *[Danish] Optimizing payment processes in the public sector. [http://purl.oclc.org/net/oioxml/ubl/einvoice/BusinessCaseDocumentation]* Danish title: Optimal betalingsformidling i den offentlige sektor. October 2003

[Law] *[Danish] Law on public payments. [http://purl.oclc.org/net/oioxml/ubl/einvoice/LawPublicPayments]* Danish title: Lov om offentlige betalinger m.v. (Lov nr. 1203 af 27/12 2003) December 27th 2003

[VTUstatute] *[English] Statute on information in the OIOXML Electronic Invoice for use with invoicing of public sector organisations (Statue no 1075 of november 11th 2004). [http://purl.oclc.org/net/oioxml/ubl/einvoice/VTUstatuteEN] / [Danish] Bekendtgørelse om information i OIOXML elektronisk regnig til brug for elektronisk afregning med offentlige myndigheder [http://purl.oclc.org/net/oioxml/ubl/einvoice/VTUstatuteDK]* November 11th 2004

RenderX
XSL • FO
formatter

Re-format page sizes

Biography

Mikkel Hippe **Brun**
Chief Consultant, XML Architect, M.Cs.
Danish National IT and Telecom Agency [http://www.itst.dk]
Copenhagen
Denmark

Mr. Mikkel Hippe Brun (M.Cs.) has been an active SGML/XML consultant to Danish business and government since 1995. Mr. Brun has been the chief technical advisor and consultant to the Danish XML Committee at the Ministry of Science, Technology and Innovation between 2001 and 2004. Mr. Brun is currently employed at the National IT and Telecom Agency at the Ministry of Science, Technology and Innovation where he is responsible for the technical e-business standardization. Mr. Brun was the primary author to the first handbook on XML Schema Naming and Design Rules published by the Danish XML Committee in 2002. Mr. Brun has been an active member of the Danish SGML/XML community since 1993 and is also member of the OASIS UBL TC.

Brian **Nielsen**
Enterprise Architect
Danish National IT and Telecom Agency [http://www.itst.dk]
Copenhagen
Denmark

Working as a Consultant on developing a common danish data definition/model for the public sector, documented in W3C XML Schema, giving both technical advice as well as facilitating the development in various domains. Background in Mathematics/Physics, later focused on computer sciencs. Has worked with meteorological data for many years.

Christian **Lanng**
e-business analyst
Ministry of Science, Technology and Innovation [http://www.vtu.dk]
Copenhagen
Denmark

Mr. Christian Lanng has worked with e-business and mobile commerce as a consultant to Danish and European businesses since 1999. From 2005 he has worked for the Danish Ministry of Science, Technology and Innovation with a focus on entry barriers for small and medium sized enterprises to e-business. He is also an active member of the Nordic Government e-business Network where the Scandinavian countries work on cross border e-business issues and standardization.

Bryan **Rasmussen**
Developer
Ministry of Science, Technology and Innovation [http://www.itst.dk]
Copenhagen
Denmark

Bryan Rasmussen has worked with XML related technologies over the last 7 years. Did neccesary technical work for various technologies used in Danish UBL implementation. Mr. Rasmussen devotes much of his time to the cause of limiting work title expansionism with the attendant hubristic faults exposed by such.

RenderX
XSL • FO
formatter