

# THE ARGUMENTS FOR EXTENSIBLE CONTENT

## INTRODUCTION

This document will attempt to summarize the experiments of the Danish UBL project with an Extensible Content area, and our arguments for why such an element would be useful for UBL as a whole.

## History of use of ExtensibleContent in Danish UBL

This section is based on the section named ‘**The Danish Experience with use of xsd:any**’ in the document named <http://www.oasis-open.org/committees/download.php/14806/A%20PROPOSAL%20FOR%20UBL%20LOCALIZATION%20RULES.pdf>

### Initial usage of ExtensibleContent:

The use of ExtensibleContent by the Danish super-standardization of UBL 0.7 was haphazard, based on a meeting with the Danish Agency for Governmental Management in which they told me that they had been using note fields in an invoice to pass around escaped XML fragments needed by their applications for tracking of the XML document through the system, reading in those escaped XML fragments, unescaping them, loading them in as an XML Dom and so forth as needed.

The reason why the format could not just be extended with custom markup at relevant places was that validation of an Invoice did not just take place at the entry into the system and then exit, but rather all through the system at various points the number and purpose of which I have absolutely no knowledge about.

Given this, and given that the system used by the Danish Agency for Governmental Management was the main one that governmental invoices would have to pass through and thus was the system through which the most savings could be expected I suggested the simple expedient of adding an element ExtensibleContent that would require one child element in any namespace not among the Danish UBL namespaces, and that the validation of this child element should be set to skip validation.

This was a pragmatic decision on my part, given the experience I found the various developers had with working with XML related technologies.

Rules were specified that any information found in there that was not understood by an application should be ignored, the ExtensibleContent element itself was considered to be an agreed up element between trading partners, if anyone wanted to use it to exchange information between trading partners not currently specified in the UBL message itself.

That, as far as I was concerned, was the whole of it – although I did consider making some demonstrations of passing around xhtml displays of the message in the area.

## **Project experiences of usefulness of the ExtensibleContent element**

As it turned out the ExtensibleContent element proved to be very useful, so useful in fact that if we had not had the Danish UBL project might have run into major problems.

One of the defining characteristics of the Danish UBL project that may prove to be an uncommon one is that it was legally mandated. Basically all organizations had to invoice the Government using the Danish version of UBL. However after the date of the implementation was passed we found out from various utility organizations that our law mandating UBL was in conflict with a law applying to their industry mandating what information had to be sent with an Invoice.

Obviously such a situation has the potential of becoming politically untenable. One can only imagine the negative headlines such a collision could have provoked. Luckily we had a solution provided by our use of ExtensibleContent, which allowed the utility industry to identify additions to the legally mandated format without breaking the law requiring invoicing information to be sent in that format.

## **Current Arguments**

We have four current arguments for the use of an ExtensibleContent element, validated via xsd:any.

1. The first is the legal mandate argument, it is impossible for one to be sure that there will not turn out to be some additional requirement not covered by a particular subset on any particular document exchanging entity required to work with that subset.
2. The second is the need to extend instances for application purposes such as discussed in the section of this document named “Initial usage of ExtensibleContent”. I do not consider this to be important enough of an issue, a programmer competent in XML usage could manage any number of non-problematic solutions, dependent on their own tastes and needs.
3. The third has not been outlined above or in the localization document from which the above was taken, from the perspective of the Danish UBL project localization and subsetting can be developed in a far quicker process with an extensible format. Basically the idea is that with the finalization of UBL 2.0 a subset can adopt it, the implementation can consist of schematron validation laid over the elements defined in the UBL namespace, defining restrictions of values, or cardinalities within the range specified by the schemas, while extensions required by the subset can be placed in the ExtensibleContent area. This is a possible way of providing for national requirements of customization.
4. The fourth argument has not been discussed before; we would suggest a method of preliminary production of additional semantics in UBL messages. In any new version of a format, whether this is a new version of subset, or of UBL itself, there will need to be a heavy political process around standardization of the new version. It can be, especially in a governmental subset such as the Danish one, that certain trading partners (for example, governmental ministries) will be ready to exchange documents in the new version or have a need to do so before the standardization process is complete.

This can also apply in situations where the standardization is complete, but the required date of exchange for the format is at some future date. In these cases the extending parts of the new version can be sent in the old version.