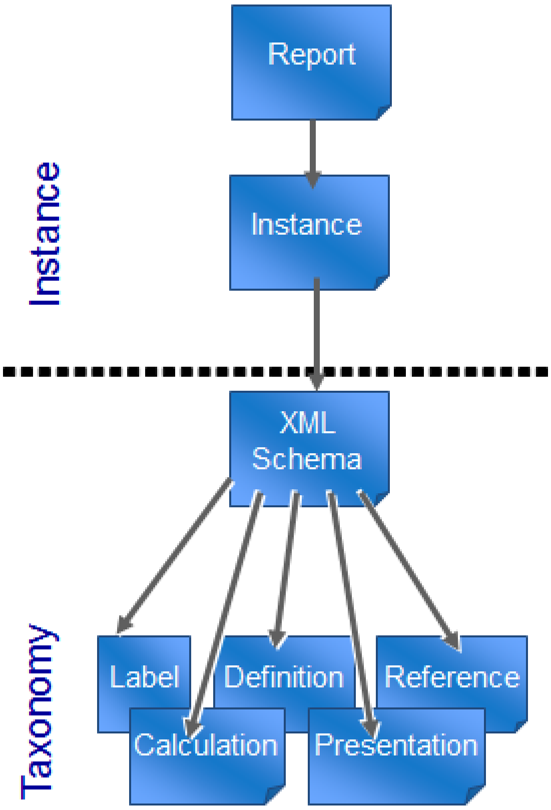
Do-it-Yourself XBRL – Is it Worth the Risk and Expense?



Are your developers suggesting an in-house, DIY approach for meeting the government-mandated SuperStream deadline for implementation of XBRL? Is this approach making you uneasy? The real risk is not just failing to meet the July 1, 2013, deadline. A DIY approach will also be slow and expensive due to the sheer effort involved in developing and maintaining financial reporting systems.

The SuperStream reforms will affect superannuation funds and require rollovers to be exchanged using a new data and e-commerce standard. This means that from July 1, 2013, APRA funds and retirement savings account providers must exchange rollovers in a new XBRL-based format. From July 2014, all large and medium employers will be required to send contributions in the new format, and all super funds, including self-managed super funds (SMSFs), will be required to receive contributions in the new standard.

Time has almost run out for companies to analyse the business case for buying XBRL software versus the cost and risks associated with developing and maintaining your own XBRL solution. **By February 2013 companies should really be in the testing phase if they are going to make the July 1 deadline.**

This article describes some of the pitfalls of using the DIY approach in preparing your superannuation systems for SuperStream. **We also discuss a cost-effective solution using a proven SBR-specific API which requires minimal training and absolutely no knowledge of XBRL.**

1. Will Your XBRL Processor be Error Free and Scalable?

Enabling financial reporting systems for XBRL is a very complex undertaking. It involves much more than creating simple XBRL instances with Java/C#/C++ and your favourite XML parser. If a DIY XBRL processor does not fully support the XBRL Dimension 1.0 specification, then it is very likely that some **errors will go undetected** during the construction of the XBRL instance.

Scalability and performance are other issues that need to be considered. When documents are small, performance is not usually an issue. But experience has shown that, once a document reporting regime (e.g. XBRL) is implemented, documents start to grow significantly in size and **system performance declines dramatically**.

2. How Will You Validate Your XBRL Documents?

If you don't have an XBRL processor which can fully validate an XBRL instance according to both the XBRL 2.1 and Dimension 1.0 then you can't be sure that your document is XBRL valid. For example, **if your SuperStream Member Rollover document is invalid and rejected how much time will you have left to rectify and re-submit the document before the 3 day limit to finalise the member transfer!**

3. How Will You Handle Taxonomy Changes Which Can Occur at Least Quarterly?

Compliance and business requirements are subject to ongoing change, requiring corresponding updates to taxonomies. Whether this happens annually, quarterly or even more often, the effect is still the same - the software which creates the XBRL instance documents needs to be modified. If your process of creating the XBRL documents involves hardcoding within the application then the process of updating can easily become **more complex and costly**.

4. How Will Your XBRL Processor Handle Constant Changes to the XBRL Standard?

Like taxonomies, **the XBRL specification is constantly changing**. New specifications are released and, inevitably, taxonomy designers take advantage of these new specifications**. A DIY solution will require frequent maintenance.**

Would your DIY XBRL processor easily handle XBRL formulae? If not, then how are you going to validate your instance document against the rules expressed in the XBRL formulae before submission if XBRL formulae are introduced to SuperStream?

5. Is There a Better Way Than DIY?

Yes, through a licensing arrangement with Fujitsu. We have licensed the Treasury API (developed by Fujitsu for SBR) and packaged it with our XWand XBRL engine to:

- Lower the cost of development and maintenance
- Dramatically shorten the time for implementation
- Provide a proven SBR-compliant solution
- Allow for minimal training
- Require no knowledge of XBRL

The API takes care of the full lifecycle – from creation and implementation through to validation and maintenance. To give you some idea of the saving provided by the API, **the TFN lookup implementation time can be reduced by 80 - 90%**!

So which approach will you use?

This article was written by Peter Campbell, Principal XBRL Consultant, Fujitsu Australia Software Technology. Read Peter's blog on this topic for an expanded version of this article or the XBRL page on the Fujitsu Australia website for demonstrations of the API in action.