The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information

Christian Beelitz

53773 Hennef, Germany
contact@christian-beelitz.de

Abstract. Knowledge is linked information. Information that cannot be connected with other current information or information stored in the past is useless for its consumers.
But, previous research has shown that XBRL on an XML basis lacks an idea for linkage of financial information. Currently there is an ongoing discussion in the XBRL community about expanding the technical basis of XBRL from XML to more semantic-web compatible formats like JSON.
This article introduces the pros and cons of switching the technical basis of XBRL from XML to JSON regarding linkage of financial information. It defines and explains the concepts of intra-linkage (within the same domain / taxonomy-framework / namespace) and inter-linkage (between different domains / taxonomy-frameworks / namespaces). At the end there is an indication that the implementation of XBRL-JSON comes at a high price with even less beneficial impact on (intra-)linkage of information compared to XBRL-XML.
The contribution of this paper to the XBRL community consists of increasing awareness for the linkage problems of XBRL inherited by the underlying data formats (XML, JSON), explaining the importance of intra-linkage over inter-linkage and providing some thoughts how to cope with the intra-linkage problem.

Keywords: XBRL, financial reporting, XML, JSON, linkage, knowledge framework

1 Introduction

Due to the several knowledge frameworks such as the knowledge-stairway by North [1] information has to be interconnected with other information be it with related current information (context), information from the past (experience) or information about the future (expectations). To give a simple but practical example:

Imagine a financial supervisory authority receiving a regulatory report from a supervised company containing the figure 1.000.000,00€. First this is just data but by its surrounding context, e.g. as part of a balance sheet, this figure is assigned to a meaning. This meaning can be ‘eligible own funds’ for instance. By adding a meaning the figure evolves to information. By linking this information to other current information (e.g. a ‘minimum solvency capital requirement’ (mcr) of 800.000,00€) and infor-

mation from the past (e.g. ‘eligible own funds’ (eof) of 1.500.000,00€ and an mcr of 800.000,00€ in the previous year) the responsible supervisor can assess a decrease of the eof/mcr-ratio from 187,5% to 125,0%. He now knows that there is a tendency that this company might get in trouble to fulfill its eof/mcr-ratio in the next year. Contacting the company and investigating the cause could be a wise and sustainable decision in this situation.

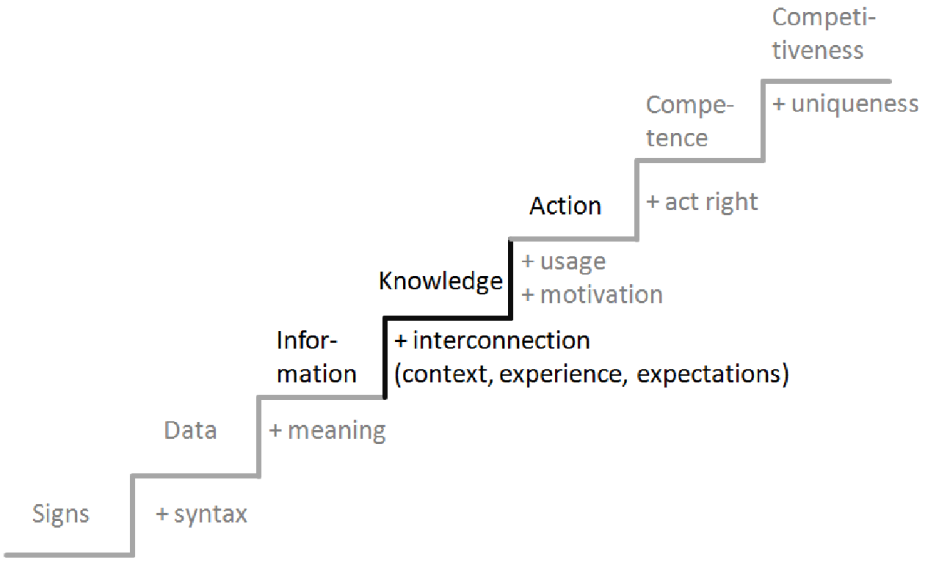

Fig. 1 Knowledge-stairway based on North [1]

Unfortunately XBRL on the technical basis of XML (XBRL-XML) does not natively support the linkage of filings neither from different financial entities nor from different reporting periods. [2]

First, this is because integration of XBRL-XML-formatted filings by Extract, Transform, Load (ETL) regularly leads to loss of meta-data about the quality and layout of a filing. [3] Beyond the XBRL 2.1 base specification for creation, exchange, and comparison of business reporting information [4] further specifications and additional linkbases like Formula 1.0 for validation and transformation of XBRL instance facts and Table Linkbase 1.0 for tabular layouts of facts have evolved. The Formulae Linkbase can guarantee some degree of data quality by applying business validation rules and arithmetic expressions to a filing while the Table Linkbase can give it a tabular layout and thereby add a form to a data-centric taxonomy approach. The main problem of this originally good idea is, that data quality assessment and visualization using formulae and table linkbase can only be performed on single, isolated XBRL-files. After ETL and shredding XBRL instances into a Data Warehouse for integration purposes the semantic uniqueness of the Discoverable Taxonomy Set (DTS) is shredded away [5] and so is information about business validation rules and layout. Even if the ETL or a taxonomy-driven approach when building the DWH analytical data model could read and preserve validation rules and template layouts the question would remain where and how to store this information in the corresponding databases.

The format of source (XBRL, XML, XLink) and target (SQL, MDX) do not naturally fit. [6] This impedes application of business rules and visualization of filings across the borders of single filings of one single entity for one single reporting period.

Second, even if the integration of XBRL-XML-formatted financial information is basically possible it still remains an effortful bespoke ETL- and Data-Warehouse-solution which does not follow a standardized approach.

Recently JavaScript Object Notation (JSON) was presented by XBRL II as one means to ease integration and ETL. [7] The Open Information Model (OIM) shall help to implement XBRL in several underlying formats (like JSON) or markup languages (like XML) to be flexible depending on the scope of application. E.g. for publishing financial information to the public another format could be technically and semantically suitable than for a financial supervisory authority which seeks to gather high-quality information with a uniform "look and feel" from its supervised financial companies.

The next section will take a closer look on the reasons for a shift from XBRL-XML to XBRL on the basis of JSON (XBRL-JSON) and explain some characteristics of the latter.

2 Characteristics of XBRL-JSON

JSON compared to XML is a lightweight data format which needs less overhead to describe financial items. While a primary item in XBRL-XML consists of a fact and a related context tag an object in XBRL-JSON is a tighter, smaller chunk of information. The following listings are an exemplary juxtaposition of a code fragment in JSON and XML respectively for the same financial concept.

Listing 1 [7] shows a "profit" fact implemented in XML:

```
<xbrli:contextRef id="c1">
    <xbrli:entity>
      <xbrli:identifier
scheme="http://standards.iso.org/iso/17442">12345</xbrli:
identifier>
    </xbrli:entity>
    <xbrli:period>
      <xbrli:startDate>2015-01-01</xbrli:startDate>
      <xbrli:endDate>2015-12-31</xbrli:endDate>
    </xbrli:period>
</xbrli:contextRef>

<xbrli:unit id="u1">
    <xbrli:measure>iso4217:USD</xbrli:measure>
</xbrli:unit>

<gaap:Profit contextRef="c1" unitRef="u1" decimals="-6"
>12000000
```

```
</gaap:Profit>
```

Listing 2 [7] shows the same a “profit” fact implemented in JSON:

```
{
  "oim:concept": "gaap:Profit",
  "oim:accuracy": -6,
  "oim:unitNumerator": [ "iso4217:USD" ],
  "oim:period": "2015-01-01/2016-01-01",
  "oim:entity": "lei:12345",
  "value": "12000000",
  "numericValue": 12000000,
}
```

The <context>- and <unit>-tag of the XML representation could be re-used but it is still obvious that much shorter and due to the simple enumeration of key-value-pairs easy to understand.

The tree-structure of an XBRL-XML document has to be parsed by the software which processes it to map the content of XML-tags to corresponding data types used by the software. [2] In contrast, the data types of XBRL-JSON like key-value-pairs or arrays are similar, if not identical, to data types used in common programming languages which makes XBRL-JSON easy and quickly to use from a technical point of view. Moreover, while XBRL-XML is regularly shredded into its granular components when loaded into a data warehouse, XBRL-JSON is compatible to being loaded into document-oriented (NoSQL) databases like MongoDB or CouchDB without breaking up its structure. [8]

Moreover, beyond pure JSON which is in discussion for XBRL a W3C recommendation called “JSON-LD 1.0”, whereas LD stands for Linked Data, was published in 2014. JSON-LD is deemed to support interconnection of information from different namespaces by using context-tags to describe the domain to which a fact belongs. JSON-LD aims to ease interconnection of heterogeneous information. [9]

3 The difference between inter- and intra-linkage

This section introduces the concepts of inter- and intra-linkage of XBRL-formatted financial information. It also argues about the impact of the underlying data format, XML or JSON, on each kind of linkage.

Inter-linkage is about the interconnection of information which stems from different domains, taxonomy-frameworks or namespaces. When inter-linking information the receiver may not expect information to be structured in the same way using the same data model and the same semantic expressions. That is because the context of information is different and different facts might have the same name or the other way around. The dictionary of information is regularly very different.

XBRL-XML, at least in some taxonomy-frameworks, considers some aspects of inter-linkage by design, but only very limited. For example in reporting frameworks

like Solvency II concepts of the CoRep and FinRep taxonomies have been re-used. [10] But other frameworks like the German HGB taxonomy for instance may use very different concepts and a different taxonomy architecture. The difference in data models, semantic expressions and structural complexity increases, of course, if XBRL-XML shall be mashed up with data from non-XBRL sources, e.g. securities databases, rating information, additional master-data etc.. Shredding the XBRL-XML formatted information, consolidating and integrating it with data from different domains can therefore cause a lot of effort.

XBRL-JSON on the other hand provides a much less complex architecture. There are, for example, no XLink-interdependencies and schema imports in JSON (in contrast to an XML-formatted Discoverable Taxonomy Set (DTS)). [11] Therefore the data model of XBRL-JSON formatted information is quite generic, flexible and expandable. [12] The granular setup of JSON allows for adding chunks of information from other domains with different underlying data-models more easily because it is neither necessary to break up a complex data structure nor is it necessary to merge information with complex data structures with complicated ETL. Thus, consolidating and integrating financial information in XBRL-JSON from different domains is the easier, faster and less error-prone alternative to XML due to its simplicity. JSON-LD (see above) containing context information about the source domain could even further ease inter-linkage of heterogeneous information. [9]

In contrast to inter-linkage, intra-linkage is about the interconnection of information within the same domain/taxonomy-framework/namespace. Main use-cases for intra-linkage are, for example, the integration of filings from several financial entities or the integration of filings from one and the same financial entity for several reporting periods. This integration of financial filings is a mandatory task for financial supervisory authorities when they want to be able to conduct benchmarking among supervised companies and variation analyses about one company over time. In the context of intra-linkage the dictionary of information among reporting entities is one and the same.

Related to intra-linkage of financial information XBRL-XML allows for deriving the data model of the target database (e.g. as component of a data warehouse, DWH) from the XBRL-taxonomy and its included XML-schema-files. For example, the metrics used and defined in the taxonomy can be seen as the role model for facts of a DWH data model and DWH dimensions can be derived from the dimensions incorporated in the taxonomy due to dimension specification. [13] The advantage of XBRL-XML based filings is that they are all bound to strict schema definitions. The receiver of filings knows in advance which information he may expect in which structure. Beyond, the receiver of XBRL-XML filings may even rely on data quality to a certain degree thanks to formulae linkbase which allows for the execution of business validations. [14] And, thanks to table linkbase (which is another meta-data specification in XBRL-XML next to already mentioned formulae specification) single filings can be instantly visualized since the table linkbase contains valuable rendering intructions. [15] But, there are not only advantages (schema definitions, business validations and visualization features as mentioned above) in XBRL-XML. The complexity and meta-data richness of XBRL-XML taxonomies and filings cause some downsides. First, a

complex data model leads to complex and regularly bespoke (that is, individually developed) ETL processes and secondly meta-data like formulae (for validations) and table linkbase (for rendering) get lost during ETL. This is because current databases are not prepared to store XBRL formulae or table linkbase information, at least not in its original form without rebuilding this logic manually. [3, 16, 17]

In the context of intra-linkage XBRL-JSON seems less powerful. Indeed, JSON is easy to process during ETL as it is built upon commonly understandable data types (integer, string etc.) and objects (arrays, key/value-maps etc.). But the simplicity, light weight and flexibility of JSON comes at the price of less assurance of structure and content of filings. A receiver of information, e.g. a financial supervisory authority, may no longer rely on the same data model or content respectively used and sent by company A and company B(, C etc.). Because of the absence of strict schemas and validation rules content and quality of filings will be less uniform. In fact, XBRL-JSON formatted information can still be loaded to analytical Warehouses technically easy, especially when the target are document-oriented databases which can reflect the submitted data almost one-to-one. [18] But, less assertion about structure, content and quality increases the risk of misinterpretation during analysis because the semantic dictionary of the reporting framework is no longer reflected by the underlying data model as it is with XBRL-XML. The risk of losing validation rules and rendering information through ETL is not given when using XBRL-JSON but that is just because these concepts are not foreseen in JSON at all. Due to this trivial cause there is not even a theoretical chance (unlike XBRL-XML) to transfer this meta-data into an integrated XBRL data store. Another though rather practical drawback from the receivers' perspectives is that IT-systems for submission, receipt and validation of "traditional" XBRL-XML instance files have been built throughout the last years with high personnel, temporal and financial efforts. Their motivation to spend a similar effort on implementing XBRL-JSON redundantly is likely to be rather limited.

To sum up, the expectable pros and cons of using XML or respectively JSON as underlying data format for XBRL are opposed in a tabular view. The table includes a differentiation between inter- and intra-linkage perspectives as well.

Table 1. Comparison of XBRL-XML and XBRL-JSON

		XBRL-XML	XBRL-JSON
Inter-Linkage	Pros	1. Usage of namespaces allows semantic distinction between data points 2. Re-Use of schema files among taxonomies allows for a consolidated data model to some degree	1. Usage of contexts (in JSON-LD specification for Linked Data) 2. Easy to understand due to common data types 3. Easy to load and consolidate due to document-oriented databases and flexible (noSQL-)data-models
	Cons	1. Deviations between data models interfere derivation of a consolidated	1. Semantic context information only provided when using JSON-LD specification

		data model 2. Each data model/XML schema depends on a dedicated shredding and ETL process	2. No schema-based dictionary
Intra-Linkage	Pros	1. Given Data Model including dimensions allows taxonomy-driven derivation of analytical data models 2. Consolidation of information is less error-prone since business validations guarantee a minimum-standard for data quality 3. Business validations (formulae linkbase) and rendering instructions (table linkbase) can potentially be used in a DWH/BI-Solution if they could be stored and processed 4. XML-based ETL processes already implemented on receivers' sides	1. Lightweight ETL process due to simple data model 2. Modeling of an integrated target-data-model needless due to potential use of document-oriented databases
	Cons	1. Taxonomical meta-data for business validations and rendering instructions are at risk to get lost during ETL process	1. Taxonomical meta-data for business validations and rendering instructions not provided in uniform technical syntax at all, thus less data quality expectable and, as to that, nothing to transfer to a DWH/BI-Solution 2. Minimum content and comparability of filings cannot be assured due to lack of a strict model and schema files 3. Effort to switch from proven XML-based validation and ETL processes on receivers' side

The number and character of pros and cons slightly indicates that XBRL-JSON might have an overall advantage over XBRL-XML related to inter-linkage of information,

especially due simplicity and its flexibility as it is not bound to one single strict data model and schemas.

On the other hand, the advantage of flexibility vanishes if intra-linkage is required. The received information is based on the same reporting framework, thus it is adequate and sensible to use a uniform strict data model to assure consistency between filings (among entities and among time) and a certain level of data quality. Moreover, XBRL-XML supports application of business validations and rendering of filings within the borders of a reporting framework. There is no reason obvious why one should give up this additional information in advance. Hence XBRL-XML is clearly recommended for intra-linkage of financial filings.

And finally, before mashing up financial information from several heterogeneous sources, it seems logical that information from one and the same source (e.g. companies reporting to a financial supervisory authority under the same reporting framework) should be properly integrated first (including all the useful meta-data which XBRL currently provides). Hence, at the moment and in the perspective of information integration and knowledge management, no compulsory need for switching the basis of XBRL from XML to JSON is given. There may be plenty good reasons for other purposes but these potential purposes are not in the scope of this paper. This paper is about integration of financial information to gain better knowledge and for this purpose XBRL-XML is the best (and cheapest because already implemented) tool we, the XBRL community, currently have.

4 Conclusion

The goal of this paper and contribution to the XBRL community is to stress the importance of integration of financial information among financial entities and over time. Due to several knowledge frameworks only integrated information leads to knowledge which is, in the context of financial supervisory authorities, the prerequisite for sophisticated and sensible supervisory action. The main question of this paper is if the current discussion about XBRL-JSON can contribute to improve the information integration capabilities of XBRL. Therefore, in its analytical section the study has introduced and explained two perspectives of information integration (inter-linkage versus intra-linkage) and has argued which underlying data format, XML or JSON, is better for each perspective. It has turned out that JSON has a slight advantage in terms of inter-linkage, while it is vice-versa in terms of inter-linkage. Since intra-linkage (linkage of information within the same reporting framework) is deemed the primary task (before mashing information up with data from other, foreign sources), XBRL-XML is the winner of this comparison and this paper strongly advises to keep and/or improve XBRL-XML for the purpose of information integration.

Such an improvement could, for example, be to retain some relevant meta-data (like business validations or rendering instructions) through ETL processes and to potentially allow for this meta-data to be applied to more than one single file (e.g. beyond the borders of one report per one entity per one period). This could perhaps be achieved, if databases regularly used for data-warehouses could natively understand

XBRL business validations and BI-Tools could natively understand XBRL rendering instructions “out of the box” (without depending on additional tools of bespoke ETL solutions). But, this is just an outlook and a task for further research. [19]

Nevertheless, it is to say that, in this paper, XBRL-JSON has been examined only with regard to integration of financial information. It is explicitly not the intention of this paper to lessen the potential benefits of JSON for other purposes in general.

References

1. North K (2010) Wissensorientierte Unternehmensführung: Wertschöpfung Durch Wissen. Springer-Verlag
2. Garcia R, Gil R (2010) Triplificating and linking XBRL financial data. In: Proceedings of the 6th International Conference on Semantic Systems. ACM, Graz, Austria, pp 1–8
3. Spies M (2010) An ontology modelling perspective on business reporting. Information Systems 35(4): 404–416
4. XBRL International Incorporated [XII XBRL 2.1]. http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html. Accessed 30 Apr 2016
5. XBRL International Incorporated [XII Formua Overview]. http://www.xbrl.org/WGN/XBRL-formula-overview/PWD-2011-12-21/XBRL-formula-overview-WGN-PWD-2011-12-21.html. Accessed 30 Apr 2016
6. Silva PC, dos Santos Souza F, Cesario Times V (2009) LMDQL: link-based and multidimensional query language. In: Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP. ACM, Hong Kong, China, pp 89–96
7. Paul Warren (2016) xBRL-JSON: making XBRL easier. https://www.xbrl.org/xbrl-json-making-xbrl-easier/
8. Quackit (2017) List of JSON Databases. http://www.quackit.com/json/tutorial/list_of_json_databases.cfm. Accessed 29 Mar 2017
9. Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Niklas Lindström JSON-LD 1.0: A JSON-based Serialization for Linked Data. https://www.w3.org/TR/json-ld/. Accessed 29 Mar 2017
10. EIOPA Reporting formats: DPM, XBRL and Validations. https://eiopa.europa.eu/regulation-supervision/insurance/reporting-format. Accessed 29 Mar 2017
11. XBRL II Extensible Business Reporting Language (XBRL) 2.1: Recommendation 31 December 2003 with errata corrections to 20 February 2013. http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html. Accessed 29 Mar 2017
12. XBRL II xBRL-JSON: mapping from Open Information Model 1.0: Candidate Recommendation 14 December 2016. http://www.xbrl.org/Specification/xbrl-json/CR-2016-12-14/xbrl-json-CR-2016-12-14.html. Accessed 29 Mar 2017
13. XBRL II XBRL Dimensions 1.0. https://specifications.xbrl.org/work-product-index-group-dimensions-dimensions.html. Accessed 29 Mar 2017
14. XBRL II XBRL Formula Overview 1.0: Public Working Draft 21 December 2011. http://www.xbrl.org/WGN/XBRL-formula-overview/PWD-2011-12-21/XBRL-formula-overview-WGN-PWD-2011-12-21.html. Accessed 29 Mar 2017
15. XBRL II Table Linkbase 1.0: Recommendation 18 March 2014 with errata corrections to 09 March 2016. http://www.xbrl.org/Specification/table-

linkbase/REC-2014-03-18/table-linkbase-REC-2014-03-18+corrected-errata-2016-03-09.html. Accessed 29 Mar 2017
16. Fernando Cezar R. Borges, Paulo Caetano da Silva (2012) A framework for processing business financial rules. In: Proceedings of the 18th Brazilian symposium on Multimedia and the web. ACM, S&#227, o Paulo/SP, Brazil, pp 47–50
17. Nutz A, Strauß M (2002) eXtensible Business Reporting Language (XBRL). Wirtschaftsinf 44(5): 447–457. doi: 10.1007/BF03250866
18. Apache Software Foundation (2017) Apache CouchDB 2.0 Documentation: 12. JSON Structure Reference. http://docs.couchdb.org/en/2.0.0/json-structure.html. Accessed 29 Mar 2017
19. Gräning A, Felden C, Piechocki M (2011) Status Quo und Potenziale der eXtensible Business Reporting Language für die Wirtschaftsinformatik. WIRTSCHAFTSINFORMATIK 53(4): 225–234. doi: 10.1007/s11576-011-0282-2