XBRL-YAML: Decreasing the Size of XBRL-based Financial Documents

Abstract— Data representation variety promotes wider adoption of data standards in companies. In financial domain, currently, XBRL Consortium has specified the XBRL standard conceptually through Object Information Model (OIM) and recommends the building of financial reports with different technologies (e.g., XML, JSON). Thus, software engineers can decide what technology better fits in their software context. This paper proposes the XBRL-YAML, an alternative to represent financial reports using YAML technology and following the XBRL standard. Initially, we have built a UML diagram to represent the XBRL standard based on OIM. From this diagram, two logic representations are shown: one with JSON syntax (already in use worldwide), and another with YAML (proposed in this paper). An assessment is performed in order to verify the number of characters inside XBRL reports which are built with XML, JSON and YAML technologies. As result, we found that XBRL-YAML decreases the size of financial documents in average 10%, comparing with XML, and 4%, with JSON. We conclude the use of XBRL-YAML is a promising approach to software engineers, whose analysis of XBRL document size is an important issue.

Keywords – eXtensible Business Reporting Language (XBRL); Financial Data; Data Representation; XBRL-JSON; XBRL-YAML.

I. INTRODUCTION

Software engineering is a part of computer science that handle issues about software development and data management, whose tasks are based on systematic application of engineering methodologies and approaches (Pressman & Maxim, 2019). In turn, data management makes feasible which way to go in order to better manipulate, store and query data (DAMA, n.d.).

In Data domain, currently, the "data-centric" concept has been becoming wider (The Data-Centric Manifesto, n.d.). This concept claims that data is the central part of any software, turning its representation an important part in Software Engineering.

In literature, there are several computing languages for data representation, e.g., *eXtensible Markup Language* (XML) (M.A.P.d.Silva & P.C.d.Silva, 2014), *Javascript Object Notation* (JSON) (XBRL International Consortium, n.d.), and *Ain't Markup Language* (YAML) (Charles Xu & Dmitry Ilyevskiy, 2019), which one describes and claims its features as advantages among them, e.g., human-readable, lighter, easy to represent, easy to understand, less syntax, etc. In this way, the system analyst must choose which one is better to represent data from his specific domain.

In financial domain, *XBRL Consortium* – an international organization - has specified the XBRL standard for representing financial business data. Initially, this standard has been developed on XML technology. However, currently, the consortium has adopted Open Information Model (OIM), a syntax-independent model, to represent XBRL standard (XBRL International Consortium, n.d.). Thus, XBRL-based financial data can be formatted by different technologies.

In order to facilitate the XBRL use and representation in more set of information systems and databases, the consortium has extended the XBRL format to CSV, JSON and HTML technologies (called XBRL-CSV, XBRL-JSON and iXBRL, respectively) (XBRL International Consortium, n.d.). In state-of-practice, there are tools and recommendations which manage XBRL documents on JSON syntax (XBRL International Consortium, n.d.)(Silva, M. A. P., 2018). Though these works bring to financial domain important contribution, making feasible the use of a wider variety of technologies in financial domain, there is no a study that proposes the use of XBRL standard on YAML technology.

YAML is a computing language widely used in Software Engineering to exchange large amount of data among software in a lightweight way (P. P. A. Haleem & M. P. Sebastian, 2008). Considering that XBRL documents usually have extensive size (M.A.P.d.Silva & P.C.d.Silva, 2014), this paper proposes the XBRL-YAML, which allows a XBRL representation based on YAML syntax. To make this proposal feasible, we have modeled an UML-based OIM diagram representing the XBRL standard. After that, we represent each component (from XBRL standard) with JSON and YAML technology. In order to validate the proposal of this paper, an assessment is performed in a real-world financial document, which is represented in three technologies, such as: XML (original file), JSON and YAML. The main results identified are discussed and suggest the use of YAML syntax can decrease the XBRL document size, in average 10% (if compare to XML syntax) and 4% (if compare to JSON syntax).

This paper is organized as follows: Section 2 describes the basic concepts of the XBRL standard, while Section 3 presents and discusses the XBRL-YAML specification. Section 4 discusses the assessment of the proposed solution. Section 5 shows the results. Section 6 presents some limitation of the proposal. Finally, Section 7 presents the final considerations of this article and suggestions for future work.

II. BACKGROUND AND RELATED WORKS

XBRL International Consortium defines the XBRL standard based on W3C technologies. Figure 1 depicts that XBRL standard is structured with three kinds of documents: *instance, schemas* and *linkbases*. XBRL *schema* defines the financial business concepts that can compose the business reporting. XBRL *instance* is the business reporting, i.e., a document that contains all financial business information. XBRL *linkbase* represents additional semantics related to data from XBRL instance (e.g., a financial data is the result of a calculation) (XBRL International Consortium, n.d.).

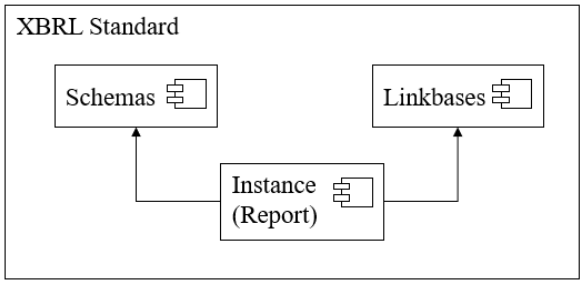

Figure 1. UML diagram of XBRL Standard.

A. *Related Works*

This section canvases works that discuss the data representation with YAML technology. Two task guide our search: (i) if YAML has already been used in XBRL domain, and (ii) which kind of data YAML can represent.

Xu & Ilyevskiy (Charles Xu & Dmitry Ilyevskiy, 2019) uses YAML for setting representation in software development tools. Haleem & Sebastian (P. P. A. Haleem & M. P. Sebastian, 2008) format general data (using YAML) in order to exchange through wireless environment. Yahui (Y. Yahui, 2012) evaluate the data-exchange among companies also using YAML. Paoli (Jean Paoli, 2015) proposes the use of YAML for representing open data through Internet. Zervoudakis et al., 2013. perfom a study for YAML representation of ontology modeling data. Musyaffa et al. (2016) represent RESTful web services through YAML technology. Wang et al. (2017) model Cloud APIs, using YAML, for executing tests over them. Kuzniar et al. (2018) represent Health data with YAML.

We conclude there is no approach or tool that uses or enables the use of YAML in financial domain, neither in XBRL domain. Another conclusion is that YAML can be used in different kinds of data. In the following section, the proposal of this paper is presented.

III. THE XBRL-YAML

In XBRL-JSON recommendations (XBRL International Consortium, n.d.), *XBRL International Consortium* has specified objects just for composing the *XBRL instance*. In this way, XBRL-YAML also follows this formal specification (i.e., XBRL-YAML just can represent XBRL *instance*).

Initially, as XBRL-JSON follows the OIM-based XBRL standard, a UML diagram has been built in order to represent XBRL *instances* (from OIM perspective). After that, we have developed a financial document based on XBRL-JSON recommendations and XBRL-YAML (proposed in this paper).

Figure 2 depicts *XBRL instance* (or *report*) is composed by two objects (*Facts* and *DocumentInfo*), both are explained in the following. *Fact* is an individual piece of information in a report, (e.g., reporting that profit at Acme Inc. in 2013 was $10m). *Fact* is composed by other information (i.e., *value* and *accuracy*) and two other objects (*Aspect* and *Link*). *Aspect* represents the aspects associated with a *Fact*, which contains the following properties: *Concept* is a definition that provides the meaning for a *fact* (e.g., "Profit", "Turnover", "Assets"), *entity* (the owner of this Fact), *period* (the time period that respective fact is valid), *unit* (currency), language (which language the concept is defined), *noteId* (additional information as a footnote), *SQName* (colon-separated representation of a URI/value pair). *Link* represents semantical information about respective Fact (e.g., label, calculation, presentation).

DocumentInfo is composed by two information (document type and taxonomy), and two objects (*Feature* and *Prefixe*). *DocumentType* defines the document type (XML, JSON or CSV). *Feature* represents additional constraints defined by other specifications (e.g., xAudit, GeoXBRL). *Prefixe* is a shorthand notation to represent namespaces (e.g., it is common to use a notation such as "ifrs-full:Profit", in this case, "ifrs-full" is a shorthand for the http://xbrl.ifrs.org/taxonomy/2016-03-31/ifrs-full namespace. Finally, *taxonomy* is a collection of concept definitions. Typically, a taxonomy will correspond to a particular reporting domain (e.g, taxonomies exist for many accounting standards such as IFRS, for various regional GAAP standards, as well as for reporting requirements of individual regulators, government agencies and large enterprises).

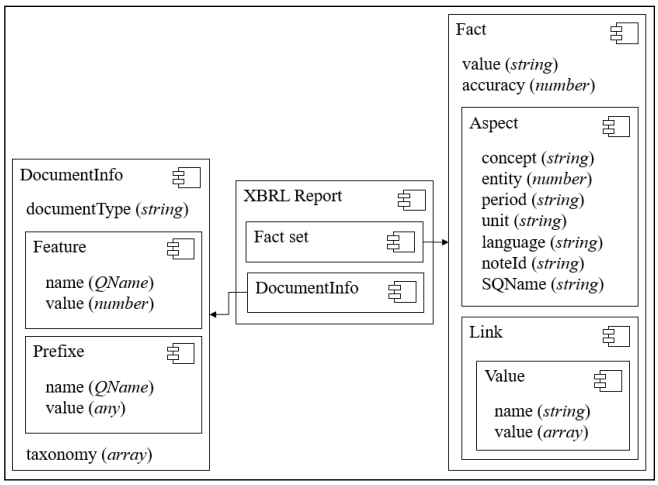

Figure 2. UML diagram of the OIM-based XBRL instance.

Figure 3 shows a XBRL *instance* using JSON syntax. In this example, the representation follows the objects and fields specified by OIM (shown in Figure 1).

```
{
 "report" : {
    "fact" : [
        {
            "value" : "",
            "accuracy" : "",
            "aspect" : {
                "concept"  : "",
                "entity"   : "",
                "period"   : "",
                "unit"     : "",
                "language" : "",
                "noteId"   : "",
                "SQName"   : ""
            }
        }
    ],
    "documentInfo" : {
        "documentType" : "",
        "feature" : {
            "name" : "",
            "value" : ""
        },
        "taxonomy" : [
            {
                "href" : ""
            }
        ]
    }
  }
}
```

Figure 3. XBRL instance based on JSON syntax.

Figure 4 illustrates a XBRL *instance* using YAML syntax. In this example, the

representation follows the objects and fields specified by OIM (shown in Figure 2).

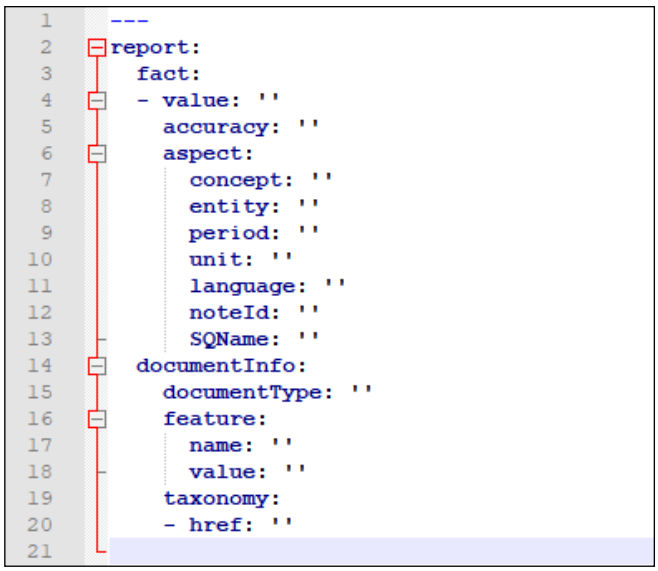

Figure 4. XBRL instance based on YAML syntax.

Thus, this study shows to XBRL community that financial reports can be built through YAML syntax, creating a new way of exchanging financial data among information systems and databases.

IV. ASSESSING THE XBRL-YAML

The scope of this assessment is to compare the number of characters inside real-world XBRL *instance*, which are represent by three technologies: XML, JSON and YAML. After that, evaluating the numerical difference among them. XBRL *linkbases* or *schemas* are not part of this assessment because XBRL-JSON are not specified to represent these documents (i.e., XBRL-JSON specification just represents XBRL *instances*).

All XBRL *instances*, assessed in this paper, are available on Security Exchange Commission (SEC) website (https://www.sec.gov/edgar.shtml). Each steps of this assessment are explained in following: (i) XBRL *instances* are download from SEC website; (ii) we verify the number of characters from original document (i.e., in XML syntax); (iii) we convert this original document from XML to JSON and YAML formats, and save them in different documents; and (iv) we verify the number of characters contained inside both JSON and YAML documents.

Table 1 presents the number of characters of some XBRL instances. In XML column the number is bigger than JSON column, which is bigger the YAML column.

TABLE 1. Number of Character.

File (XBRL *instance*)	XML	JSON	YAML
oxfo-20130930.xbrl	150,106	139,410	133,295
msft-20130331.xbrl	2,720,440	2,454,283	2,347,581
goog-20151221.xbrl	5,078,990	4,726,486	4,615,789

Clearly, the number of character decreases from XML to both format: JSON and YAML. This behavious in JSON syntax is already know in literature. However, this behavious in YAML syntax is not invested yet. Thus, YAML document has less characters than both format: JSON and XML.

This new discovery directly impacts in XBRL document size and open a new perspective of XBRL representation. Some results are debated in following.

V. RESULTS

It is known that XBRL International Consortium has already been using different technologies for formatting their financial document (i.e., XBRL-XML, XBRL-JSON, XBRL-CSV). However, this study brings to state-of-art the promising use of YAML syntax in Financial and XBRL domain.

Using YAML, the financial taxonomist diminishes the size of XBRL instances in average of 10% (if compare to XML syntax) and 4% (if compare to JSON syntax). This decreasing is a positive index, which must be considered by software engineers or system analysts, in decision-make about what syntax should use in a financial information system. This scenario turns YAML a promising alternative, because XBRL instances are commonly very large.

VI. LIMITATIONS

This study identifies a limitation about using YAML syntax in XBRL domain. The other syntaxes (i.e., XML and JSON) have database based on their format. This allows the storage of the XBRL-XML and XBRL-JSON documents, natively, in which there is no need of format change to store the respective documents.

VII. CONCLUSION

This paper introduces the XBRL-YAML, an alternative syntax for representing XBRL-based financial reports. Use of XBRL-YAML decreases the XBRL document size in 10%, comparing to XML syntax, and 4%, comparing to JSON syntax. Since performance matters to financial information systems, minor size of financial documents increases the runtime processing among software and databases.

As future investigation, we propose a library that allows software to work with XBRL-YAML documents in memory.

ACKNOWLEDGEMENT

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) – Finance Code 1712205.

REFERENCES

A. Kuzniar et al., "A portable and scalable workflow for detecting structural variants in whole-genome sequencing data," 2018 IEEE 14th International Conference on e-Science (e-Science), Amsterdam, 2018, pp. 303-304. Doi: 10.1109/eScience.2018.00064.

Charles Xu and Dmitry Ilyevskiy. 2019. Isopod: An Expressive DSL for Kubernetes Configuration. In Proceedings of the ACM Symposium on Cloud Computing (SoCC '19). Association for Computing Machinery, New York, NY, USA, 483. DOI:https://doi.org/10.1145/3357223.3365759.

DAMA Internacional. The Global Data Management Community. (n.d.) [online]. Accessed on Jan.28, 2020. Available on https://dama.org.

F. A. Musyaffa, L. Halilaj, R. Siebes, F. Orlandi and S. Auer, "Minimally Invasive Semantification of Light Weight Service Descriptions," 2016 IEEE International Conference on Web Services (ICWS), San Francisco, CA, 2016, pp. 672-677. Doi: 10.1109/ICWS.2016.93.

Fokion Zervoudakis, David S. Rosenblum, Sebastian Elbaum, and Anthony Finkelstein. 2013. Cascading verification: an integrated method for domain-specific model checking. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2013). Association for Computing Machinery, New York, NY, USA, 400–410. DOI:https://doi.org/10.1145/2491411.2491454.

Jean Paoli. 2015. Documents as Data, Data as Documents: What we learned about Semi-Structured Information for our Open World of Cloud & Devices. In Proc. 2015 ACM Symposium on Document Engineering (DocEng '15). Association for Computing Machinery, New York, NY, USA, 1. DOI:https://doi.org/10.1145/2682571.2797070.

J. Wang, X. Bai, L. Li, Z. Ji and H. Ma, "A Model-Based Framework for Cloud API Testing," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, 2017, pp. 60-65. Doi: 10.1109/COMPSAC.2017.24.

M. A. P. d. Silva and P. C. d. Silva, "Analytical Processing for Forensic Analysis," 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, Ulm, 2014, pp. 364-371. Doi: 10.1109/EDOCW.2014.60.

P. P. A. Haleem and M. P. Sebastian, "A method for the improvisation of messaging in wireless mobile devices," 2008 IET International Conference on Wireless, Mobile and Multimedia Networks, Beijing, 2008, pp. 109-112. Doi: 10.1049/cp:20080157.

R. Pressman, B. Maxim (2019) Software Engineering: A Practitioner's Approach. McGraw-Hill Education, ISBN: 1259872971.

Silva, M. A. P. (2018). XBRL-JSON Converter. [online]. Accessed: Jan.28, 2020. Available on https://xbrlframework.herokuapp.com/.

The Data-Centric Manifesto. (n.d.) [online]. Accessed on Jan.28, 2020. Available on http://datacentricmanifesto.org/.

XBRL International Consortium. XBRL-JSON: mapping from Open Information Model 1.0. (n.d.) http://www.xbrl.org/Specification/xbrl-json/CR-2019-06-12/xbrl-json-CR-2019-06-12.html. [online]. Accessed: Jan.28, 2020. Available on http://www.xbrl.org/Specification/xbrl-json/CR-2019-06-12/xbrl-json-CR-2019-06-12.html.

XBRL International Consortium. The Business Reporting Standard. (n.d.) Accessed: Jan.28, 2020. Available on http://xbrl.org/.

Y. Yahui, "Impact data-exchange based on XML," 2012 7th International Conference on Computer Science & Education (ICCSE), Melbourne, VIC, 2012, pp. 1147-1149. Doi: 10.1109/ICCSE.2012.6295268.