American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
http://asrjetsjournal.org/

Data Mapping for XBRL: A Systematic Literature Review

Henderson Acosta Bragança[a], Paulo Caetano da Silva[b*], Nacles Bernadino Pirajá Gomes[c]

[a,b]*Salvador University (UNIFACS) Avenida Tancredo Neves – 2131 – Salvador – BA – Brazil*

[c]*Federal University of Bahia (UFBa) Avenida Milton Santos, s/n – Salvador - BA – Brazil*

[a]*Email: henderson.braganca@gmail.com*

[b]*Email: paulo.caetano@unifacs.br*

[c]*Email: nacles@gmail.com*

Abstract

It is evident the growth of the use of eXtensible Business Reporting Language (XBRL) technology in the context of financial reports on the Internet, either for its advantages and benefits or by government impositions, however, the data to be transported by this language are mostly stored in structures defined as database, some relational other NoSQL. The need to integrate XBRL technology with other data storage technologies has been growing continuously, and research is needed to seek a solution for mapping data between these environments. The possible difficulties in integrating XBRL with other technologies, relational database or NoSQL, CSV files, JSON, need to be mapped and overcome. Generating XBRL documents from the database can be costly, since there is no native alternative that the database manager system exports from the database manager system, the data in XBRL. For this, specific third-party systems are needed to generate XBRL documents. Generally, these systems are proprietary and have a high cost. Integrate these different technologies adds complexity, since these documents do not connect to the database manager system. These difficulties cause performance and storage problems and in cases of large data, such as data delivery to government agencies, complexity increases. Thus, it is essential to study techniques and methods that allow us to infer a solution to perform this integration and/or mapping, preferably in a generic way, that includes the XBRL data structure and the main data models currently used, i.e. Relational DBMS, NoSQL, JSON or CSV files. It is expected, in this work, through a systematic literature review, to identify the state of the art concerning the mapping of XBRL data.

Keywords: XBRL; XML; Data Integration; Data Mapping.

---

* Corresponding author.

1. Introduction

Transporting financial and accounting data over the Internet while maintaining its structural and semantic integrity has been achieved using the eXtensible Business Reporting Language (XBRL)[1]. XBRL is derived from XML (Extensible Markup Language) which is considered by the W3C[2] as an important means of exchanging data on the Internet, since it allows the exchange of platform-independent data [1], the XML technology design favors simplicity, generality and usability [2], and these attributes provide the use of this XML model on the Internet and, later, as a means of integration and exchange of data, for which techniques and alternatives were proposed for the integration and mapping between XML and databases.

However, these proposals add a computational cost because, in some cases, it is necessary to create parallel databases with the data coming from xml documents, in addition to storing the XML document to safeguard the integrity of the information. In other cases, it is necessary to develop specific applications to mediate the exchange of data between different environments, i.e.XML, relational bases, CSV[3], NoSQL[4] and JSON[5].

The XBRL language, was developed for the exchange of financial information on the Internet, being consolidated as the standard to be used by government agencies in several countries [3], this language has the ability to transport financial data together with its semantics, something indispensable for a correct analysis of information, excluding the need to "decipher" the information provided without its context [4].

XBRL is a technology that has become an international standard for the exchange of financial data, e.g. the US-SEC, The European Committee of Banking Supervisors (CEBS), Bank of Spain, Bank of Japan, are just a few examples of government agencies that have adopted XBRL for data exchange between their supervised organizations[6]. In Brazil, in view of the normative impositions, as is the case of law 12.527/11 that provides for access to information from states, municipalities and the Federal District, popularly called the Transparency Law, consistent with the Normative Instruction 896/17 of the National Treasury Secretariat[7], directs the governmental spheres to the use of XBRL in their publications related to the financial statements with the control bodies.

Therefore, major drivers in the use of the XBRL language have been governments. The Accounting and Tax Information System of the Brazilian Public Sector - SICONFI[8] requires the delivery of accounting and financial information in XBRL and for this federation units need to enter the accounting and financial data in proprietary systems, in many cases manually to generate the XBRL document, since the connection to the data source, e.g.

[1] https://www.xbrl.org/

[2] https://www.w3.org/

[3] Comma Separated Values

[4] It's an acronym for "Not Only SQL"

[5] JSON is the Java acronym Script Object Notation, is a pattern of exchanging data between systems.

[6] https://www.xbrl.org/the-standard/why/ten-countries-with-open-data/

[7] https://www.gov.br/tesouronacional/pt-br

[8] https://siconfi.tesouro.gov.br/siconfi/index.jsf

relational databases, it is not trivial. This generates human and computational resource costs from the low interoperability between XBRL environments and data sources, which implies additional financial costs.

Both governments at different levels (federal, state and municipal), and private organizations, do not have *free and open-source tools* that allow the export of financial and accounting information in the XBRL format, especially when extracted directly from the databases. Also, the reception of data in XBRL format and its insertion in relational databases or even in NoSQL databases remains a difficulty for private organizations and governments.

Therefore, through a systematic literature review (SLR), because it is a method that allows a broad view of the object researched [5], it seeks to understand how data is mapping, whether from relational databases or other diverse sources to be instantiated in XBRL. As a SLR results, it is expected, in a future work, if any XBRL data mapping solution is found for other formats, extend it to cover the widest possible variety of data formats. In the event of the absence of a solution, it is intended to propose, also in a later work, a way to solve this problem. This is part of the effort to have reliable data with its characteristics preserved, excluding ambiguities and duplicities in the exchange of financial and accounting information on the Internet.

This work is organized as follows: in section 2 the XBRL language is conceptualized, in section 3 the methodology used for research in search repositories is presented, in section 4 the results of the research carried out are shown. The findings, analysis and discussion of the proposals found in the selected papers in sections 5 and 6 are presented, and in section 7 the main challenges are presented. Finally, in section 8, the final remarks.

2.XBRL

XBRL is an extended open language of the XML language, specifically designed for the exchange and analysis of financial information on the Internet [6]. Its basic structure consists of: (i) instance document, which contains the data to be reported, along with information from a specific context and (ii) taxonomy with the definition of the concepts of accounting and financial terms and their semantic relationships [7].

In the instance document the financial facts are contained, it is in this document that the accounting and financial data are entered and for this follows a structure that is defined in the taxonomy [7].

Taxonomy is described by a *Schema* XML document [6], which contains the concepts related to accounting and financial data. These concepts have a name and a type [7]. Another important component of taxonomy is known as *linkbase*. Taxonomy *linkbases*, based on XLink [6], express the semantic relationships between the concepts and between them and the instance document, associating the concepts with their documentation [7].

In 2021, the consortium responsible for XBRL defined new ways to use the standard, defining as a final recommendation in addition to XML's extended XBRL, the extended XBRL of JSON and CSV[9].

XBRL has well-established and documented benefits for Riccio and his colleagues [8] XBRL provides

[9] https://www.xbrl.org/news/2021-saw-new-specifications-launched-lift-off-for-xbrl-json-and-xbrl-csv

technology platform independence, interoperability, and efficient preparation of financial reporting.

3. Methodology

Given the scope and importance that XBRL has achieved in the financial market, it is salutary to know the state of the art of techniques and applications aimed at generating the document composed of contextualized facts, called XBRL instance, from various data sources. The objective is to know the proposals of mapping and/or integration of data for XBRL, to know about the existing views regarding the instantiation of data in markup language, such as XBRL.

The formulation of the research question was based on the identification of all the studies that deal with the mapping of data to XBRL and from XBRL to different data storage formats, regardless of the application segment. Based on this premise, the research question formulated for this systematic review is as follows: (Q1) Are there solutions for integration or mapping of data for XBRL and vice versa? Based on this question, secondary questions were defined to have a better understanding of the problem: (Q1.1) What is the need to map data from various sources to XBRL and vice versa? (Q1.2) What technologies are used to map XBRL data? and (Q1.3) Are there proposals efficient for including any type of data source and XBRL?

From these research questions, a search *string was planned* that could facilitate finding the largest number of papers related to the theme of this work. Because most of the works were published in English, the terms were placed in this language to achieve greater quantity. Making use of the resources of search engines, the *string* was defined following the Boolean logic in the elaboration and application of the search.

(XBRL AND (Integration OR Mapping OR Framework OR Middleware) AND (Relational OR Data OR Database OR NoSQL OR CSV OR JSON))

The searches were carried out in the main digital libraries available in the computing area: AIS[10], IEEE Xplore[11], ACM Digital Library[12], Science Direct[13] and Springer Link[14]. We complement the search using Google Scholar, thus giving a greater scope of search in search of works not yet identified and that could be in little known repositories.

To select the papers that are related to the research questions, the inclusion and exclusion criteria were then defined.

Inclusion Criteria:

- (CI1) The paper addresses the themes of research questions.

[10] https://aisel.aisnet.org/

[11] https://ieeexplore.ieee.org/

[12] https://dl.acm.org/

[13] https://www.sciencedirect.com/

[14] https://link.springer.com/

- (CI2) The paper deals with difficulties, critical issues or challenges related to research issues.
- (CI3) The paper proposes some methodology, conceptual model or study framework related to research questions.

Exclusion Criteria:

- (CE1) The main focus is not related to this research.
- (CE2) The paper is prior to 2015.
- (CE3) The paper is not available for download.

For the conduction of the work, the systematic review selected the studies, following five steps: (1) execution of the search; (2) application of the first filter, (3) application of the second filter and (4) application of the third filter (5) reading and analysis of the selected works [9].

By performing the first step, we do not limit the search with any type of criterion beyond *the* predefined string. At this moment, the titles of papers that are consistent with the research issue were identified, which were inserted in the Mendeley[15] bibliography manager and in the tool to help systematically review the Parsifal[16] literature.

Then, the first filter was applied, i.e. papers selected through the inclusion and exclusion criteria, through the reading of the *abstracts of* the works. In the second filter, the work resulting from the first filter was fully read, reapplying the inclusion and exclusion criteria. At this stage, the papers that did not meet the inclusion criteria, as well as those that were outside the delimited cut-off date, papers prior to 2015, were rejected. Finally, papers that were not available for download were rejected.

It is also essential to evaluate the quality of the work [10], so criteria were defined to verify the quality presented in the papers and to meet at least 80% of the following questions:

- (QQ1) Is the solution clearly detailed?
- (QQ2) Aren't the proposed technologies obsolete?
- (QQ3) Are the results and conclusions clearly explained?
- (QQ4) Is there a clear description of the environment of the proposed solution?
- (QQ5) Are the research objectives and motivation clearly defined?
- (QQ6) Does the study present an XBRL data mapping solution?

[15] https://www.mendeley.com

[16] https://parsif.al

4. Findings

Only 4 studies resulted from the research with the *string defined* in the methodology. Therefore, due to the need to seek a solution for mapping and integration between XBRL and other data storage models and because XBRL is a language derived from XML, the original *string* was adapted to XML, so that it could identify other related works, focusing on XML, which assists in the solution of research questions. Thus, searches related to mapping and integrating data to XML can meet expectations for data mapping to XBRL.

A total of 393 papers distributed in the repositories as illustrated in Figure 1 were found, of these papers, 24 were present on more than one platform.

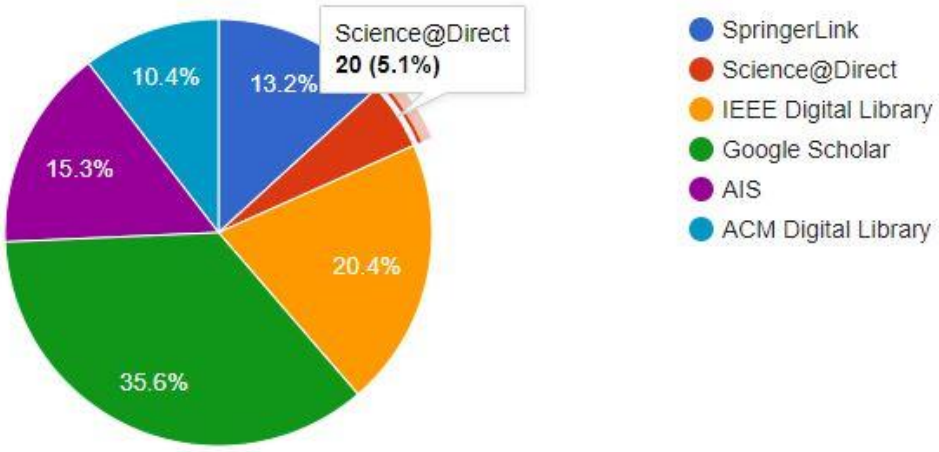

Figure 1: Distribution of the 393 papers found in the research repositories.

Because they were before 2015, 177 papers were excluded and 188 did not have the main focus related to this research. Therefore, there were 28 papers left to be analyzed and with the following distribution in the repositories: SpringerLink with 5 papers (17.85%), Science Direct with 3 papers (10.71%), IEEE with 7 papers (25%), Google Scholar with 7 papers (25%), AIS with 1 paper (3.57%) and ACM with 5 papers (17.85%). The objectives and solutions proposed in the 4 papers that deal specifically with XBRL are presented in Table 1 and the solutions and proposals of the 24 papers dealing with XML, JSON and CSV are presented in Table 2.

Table 1: Objectives and proposed solutions of the works related to XBRL

Paper	Main Objectives	Proposed Solution
[4]	Develop a tool to transform data from CSV files into XBRL instance.	It presents a solution titled XBRL-ETL ENGINE that generates XBRL instances in accordance with the SICONFI[17] taxonomy.

[17] https://siconfi.tesouro.gov.br/siconfi/index.jsf

[11]	Identify storage and validation alternatives for XBRL instances.	The authors claim that the most efficient way to store XBRL instances is by using DBMS (Database Management Systems) capabilities that natively internally use the XML model, in parallel with in-memory processing and the use of proprietary platforms to generate XBRL instances. Although the authors have stated that the best solution for storing data from XBRL instances is in native XML DBMS, the literature holds the opposite [12] [13]. In addition, no solution is proposed for mapping other data sources to XBRL.
[14]	Develop a semantic approach to integrate, process, and query financial information embedded in XBRL instances.	Introduces an approach to integrating instantiated financial records into XBRL into a semantically identifiable format that allows you to run queries.
[15]	Develop a method for retrieving information on XBRL instances.	It presents a method called X-IM that allows data interoperability by mapping elements with different but semantically identical names on XBRL instances.

Studies dealing with the problem of mapping and XBRL integration are scarce, as evidenced in Table 1, so the investigation to XML was extended.

Table 2: Objectives and proposed solutions of selected papers dealing with XML

Paper	Main Objectives	Proposed Solution
[16]	Discusses methods of integrating data and their challenges using ETL[18].	A case study was presented in which data from a bank in operation was exported and validated in XML. The process was carried out through an ETL flow to be applied in any BI (Business Intelligence) tool.
[17]	Analyze the advantages of using XML in data exchange, following with hierarchical XML analysis to represent a relational schema of data. It also aims to develop a mapping algorithm that exposes	Presented a conceptual framework for mapping the data contained in relational databases and data in XML instances, preserving the data structure, integrity, and constraints of the relational schema.

[18] ETL is the acronym for Extract, Transform and LOad (Extraction, Transformation and Load)

	the data structure of the relational database while preserving semantic integrity and constraints.	
[18]	Perform data mapping on XML instances for relational databases.	An approach entitled Mini-XML was presented with the main advantage of mapping and storing the relationships of XML nodes. It considers more efficiency in storage time and disk space consumption when compared to the S-XML[19] proposal.
[19]	It proposes an analysis of the advantages and disadvantages of different approaches, such as node labeling or hybrid labeling based on models such as S-XML, XMap, XParent, Mini-XML, for exporting data from XML instances to relational databases.	Limitations and advantages of each mapping approach were pointed out, where two components common to all mappings were found, node labeling and data mapping. Between the analyzed approaches, at least two tables are required to map data from XML instances to relational databases. These two tables store the path expression and path details with the data of the leaf node.
[20]	Proposes a single syntax for querying data in XML structures, relational databases, and hybrid structures.	A translator was presented capable of converting SQL queries to meet data recovery in relational structures, XML and suggests a hybrid structure that allows modeling the data in XML and representing it in relational tables, thus emerging a hybrid DBMS.
[21]	It seeks to develop a multimodal processing framework for relational data and XML and design a join algorithm.	It presents a query algorithm titled XJOIN that, according to the proposal, outperforms traditional algorithms of XML queries and relational databases at runtime.
[22]	It aims to identify different approaches and techniques that map fuzzy XML schemas to relational databases or object-oriented databases. In addition, it hopes to identify different fuzzy models, XML data models and	It presents a review of fuzzy literature on XML, relational databases, and object-oriented databases.

[19] S-XML is a mapping scheme XML [42].

	fuzzy techniques integration process in different databases.	
[23]	Develop an ETL algorithm that integrates data from the WEB in XML with relational databases to facilitate analysis by DW/BI systems.	It presents a platform capable of handling the integration of complex data, e.g. semi-structured, unstructured data, chat logs, emails, images, videos, originated on the WEB in real time (real-time).
[24]	Define mapping rules for the integration of relational databases and XML instances, to provide a representation of data by reducing the redundancies typical of hierarchical models.	Introduces rules for integrating relational databases and XML instances overcoming redundancy problems of structures. As a result, a library has also been developed to address the integration of XML data into relational databases.
[25]	Create a framework for migrating relational databases to other types of databases and hierarchical structures, such as XML.	Introduces a framework for migrating relational databases to other types of databases with semi-automatic mapping and using XML.
[26]	Propose a solution for mapping XML instances to relational databases.	A mapping called XML-REG is presented that acts on reading the XML instance, handling nodes and data, and loading into the relational database.
[27]	Develop an XML instance mapping for relational databases.	It presents a mapping based on two algorithms, one of reading and the other translates xPath to SQL queries, the mapping is titled XAncestor.
[28]	Develop a connection between XML and relational databases, retrieving the data by abstracting the syntaxes of languages and structural models (XML or RDB).	Displays a system that aims to extract data independently of the query language and data storage model.
[29]	Develop a framework that translates SPARQL queries to XQuery.	It presents a framework titled SPARQL2XQuery that allows interoperability between semantic web and XML.
[30]	Develop a framework that identifies the relationships	Presents a framework for mapping between XML and relational database using a tree pattern, where it

	between the data obtained in the XML instance and the fields of the tables in relational databases.	considers an XML document as a tuple, with a single root, and all edges are far from the root. The work restricts the implications on the dependencies of relationships in the database, without considering null values.
[31]	Generate XML schemas, based on *XML Schema*, with data coming from relational databases.	Introduces a solution for translating relational databases into *XML Schema* by validating *schema structure*, data semantics, and health constraints.
[32]	Create a fuzzy XML time-time data model for relational databases.	It presents a meteorological data modeling methodology called fuzzy XML timespace and the transformation of fuzzy XML space-time data to relational database.
[33]	Discusses THE ANSI SQL JSON features investigating how different relational database systems (DBMS's) integrate them.	It presents *insert* codes and query codes (*select*) in JSON format and concludes that Oracle DBMS natively implements the ANSI SQL JSON concepts, Microsoft SQL Server partially, and PostgreSQL does not implement any ANSI SQL JSON concepts.
[34]	Proposes a way to detect errors in data by pre-checking the JSON schema.	Offers an application that validates JSON documents with a JSON schema. Provides the *github* repository with the codes in the Python programming language.
[35]	Develop an approach to integrating tree structures such as JSON and XML into relational databases.	It presents a system titled MITRA as an acronym for "Migrating Information from Trees to RelAtions". The steps followed by the algorithm to migrate the data are visually displayed, however access to the source code is not available through the link provided in the paper.
[36]	Describes the advantages and disadvantages of integrating JSON and Database Management Systems and suggestions for solving identified problems.	Identifies the advantages of storing JSON in DBMS, such as: (i) Semi-structured data storage; (ii) Databases offer costs in reduced management; (iii) Increased productivity for developers. It also identified that the most relevant problem is the lack of DBMS's with native JSON integration capability.
[37]	Develop a *schema to* represent geographic data using JSON, which meets the integration of JSON *Schema with* DbMS's	Suggests an extension of JSON (*schema*) for spatial data compatible with DBMS's NoSQL, called JS4Geo, the suggested pattern facilitates integration between JSON documents with geographic data and DBMS's

	NoSQL.	NoSQL.
[38]	Discusses the use *of the Remote Table Access* Framework (RTA) to make relational database tables available on the internet for consultation.	Describes how the RTA Framework works and for cases where data originates in *Comma Separated Values* (CSV), presents the Framework extension called *Table on Top* (ToT).

The studies investigated show the effort to overcome the difficulties related to the insertion of data from databases based on different models (Relational, NoSQL, CSV) in XML, JSON or XBRL documents. In the following section we will discuss the identified papers that make these mappings or integration.

5. Mapping data to XBRL

Studies that propose data mapping for XBRL instances were identified from this literature review. Although scarce, the work that deals with data integration and mapping, although not dealing specifically with XBRL, allowed investigating the languages that support the XBRL standard disciplined by the consortium, be they XML, JSON or CSV files.

Analyzing the work that deals specifically with XBRL, we realize that the prevailing view is that government restrictions make financial data complex. According to Belev [11], for information exchange in this domain, with the validation of data and rules, we chose to use XBRL. Belev's work [11] aims to identify the most efficient way to store the financial data contained in the XBRL instances, without defining a specific DBMS, the paper recommends the use of DBMS that natively use the XML model without convincingly presenting the analysis that supported this conclusion, contrary to the existing literature that deals specifically with the performance of DBMS's XML. In the works [4,14,15] we can identify the effort for mapping the data from various sources and instantiate it in XBRL and read the XBRL instance. The generation of the XBRL instance was successfully achieved in the work of Bragança and his colleagues [4] from CSV files, however, data validation was not developed in this work, from the taxonomy, in the instance generation.

The main considerations related to mapping data between relational or NoSQL databases to instances, whether XML, JSON, or CSV, and in the opposite direction, we will see in the next sections.

5.1. Mapping relational data or NoSQL to XML

Proposals related to techniques for mapping data in relational databases to XML instances were identified in jobs [18,26,27]. The works [18,27] are cited and evaluated in the work of Song and his colleagues [19], which defines the technique titled Mini-XML, described in the work of Zhu and his colleagues [18], as an approach that reduces data redundancy by storing the leaf node separately from the data table, this technique labels each node with (l,[n, d]), where l represents the level of the node, n the parent node, and d the position of the current node [19]. The paper by Song and his colleagues [19] still makes its weights related to the work of Qtaish and Ahmad [27] which brings the XAncestor technique that addresses the anchoring in three components, relational database scheme, XtoDB mapping and XtoSQL algorithm [19]. The comparison made between these two

techniques, Mini-XML and XAncestor results in the conclusion that, XAncestor can correct the excess description generated by Mini-XML with respect to the position of the node for the description of storage paths made in the database tables [19].

From the work of Song and his colleagues [19], its authors later present a data mapping technique with the name XML-REG [26]. The authors make a comparison with Mini-XML and XAncestor so that the XML-REG technique performs better than others about the storage process, query recovery process, database size, and scalability test [26].

Bikakis and his colleagues paper [29] in addition to its contribution to the creation of a framework that translates SPARQL queries into XQuery queries, notes that due to the universal adoption of XML, for web developments to thrive, as is the case of WoD (Web of Data), the ability to read/query and export data in XML instances is indispensable [29]. In this perspective, it is observed that the work conditions the evolution of the web the ability to integrate the information represented in a semi-structured way (XML/XBRL) with the most diverse data models, this view is corroborated in the work of Niewerth and Schwentick [30].

The use of framework's has been popularized by facilitating software development by providing a generic solution for a specific need, this allows the developer an abstraction in their code. This approach was explored in the works [16,20,21,23,24,25,28,15,29,30,31,14,32,4]. All these approaches strive to create a generic way to integrate XML into traditional data storage structures, whether using ETL techniques or unprecedented integration proposals.

In the paper by Bai and his colleagues [32] it was pointed out that the need for interoperability of data in homogeneous and heterogeneous databases, forces the development of ways to interconnect these databases without being lost meta-data and restrictions, so the most viable way found by the work was to use XML for this purpose, even though it is not simple to convert the data from XML instances that have the characteristic of being ordered hierarchy in unordered relational formats.

5.2. Relational or NoSQL data mapping for JSON

Java Script Object Notation (JSON) is the primary format considered by NoSQL database management systems because it provides a flexible data representation suitable for modeling data entities [37].

The documentation that deals with XBRL, available on its website (www.xbrl.org) includes The JSON-based XBRL, as a strategic initiative of the XBRL International consortium to simplify and modernize XBRL technology.

The alternative to XML-based XBRL was analyzed in the paper "The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information" [39]. Initially the advantages related to the use of JSON, in place of XML, for XBRL are identified: JSON is a data format that needs fewer characters to describe financial items; ease to be understood by humans; structure similar to that found in programming languages, making it easy and fast to use from a technical perspective; JSON supports document-oriented databases (NoSQL) [39].

However Beelitz [39] stated that the use of XBRL-JSON has the disadvantage of not being able to use a *schema file* containing the taxonomy, making it impossible to use XBRL in its fullness. However, the documentation offered by the consortium that administers XBRL states that XBRL-JSON has a file called *documentinfo* with contextual information[20] and the work of Habib and his colleagues [34] is conclusive to refute this statement, since it is dealing precisely with the search for errors or inconsistencies in *JSON schema* files, thus refuting Beelitz's statement [39].

Frozza e Mello's work [37] states that using JSON associated with DBMS's NoSQL provides ease integration because the NoSQL DBMS is document-oriented, and its structure has similarity to JSON documents.

The works [33,35,36] focus on integrating JSON documents with relational databases. In these works, integration concepts were addressed and in the case of Petković's work [33] some codes were tested, i.e., executed in controlled situations to ascertain which DBMS's accept written codes using the JSON format, so it is not an integration of the data instantiated in JSON to relational databases, but rather the verification of the DBMS's ability to interpret the JSON language for its operations. For the works [35,36] the concept of integration was addressed, and problems and advantages were identified without presenting a solution to the problem.

The investigated studies did not present JSON integration solutions with relational or NoSQL databases.

5.3. Relational or NoSQL data mapping for CSV

The perspective of integrating CSV files with relational or NoSQL databases or in the opposite direction has scarce literature in research repositories, only two studies have been identified. However, one of the works, [4], deals with the instantiation of data from CSV files for XBRL instances. In this case the need for the use of CSV is given due to the low interoperability of the DBMS Adabas[21] that allows data export only in CSV files, without the possibility of connecting to any DBMS, due to the high financial cost for acquisition of components *(software*) that allow the connection.

Doi and Toyama's second work [38] focuses on the availability of data on the internet to meet the open data project, in this case CSV is also used as a data source for the RTA framework.

The CSV files use may have a boost after the availability of XBRL also in CSV, however, research related to the integration of CSV with relational or NoSQL databases or even XBRL-CSV itself was not found in known repositories.

6. Results against the research questions

The research questions were partially satisfied. The following discusses how the investigated studies satisfied the issues proposed in this work.

[20] https://www.xbrl.org/guidance/xbrl-json-tutorial/#11-xbrl-json-report-structure

[21] https://www.softwareag.com/en_corporate/platform/adabas-natural.html

(Q1) Are there data integration or mapping solutions for XBRL and vice versa? The research developed in this work identified only two studies [4,15] that address the issue. It is salutary to note that the proposals found in the research, having their focus on solving specific problems such as meeting the legislation or identifying semantic similarity in XBRL instances, that said, we can still realize that the works are limited to dealing with a single path of information, being mainly in the sense of generating the XBRL instance. Due to the scarcity of work dealing with XBRL and the consortium responsible for XBRL admitting XML, JSON and CSV files, we chose to investigate mappings related to XML, JSON and CSV in the search for works that satisfied this issue. From this perspective we identify the works [16,18,19,20,21,23,24,25,26,27,28,15, 29,30,31,14,32,33, 35,36,38] that deal with mapping data from XML or JSON instances and CSV files to databases.

Therefore, there are studies that deal with the mapping of XBRL data and mappings related to the languages admitted by the consortium responsible for XBRL.

(Q1.1) What is the need to map data from various sources to XBRL and vice versa?

In Brazil the predominant factor for the use of XBRL is the legal imposition, this rule is evidenced by Belev in Belev's work [11]. In the work of Bragança and his colleagues [4] the development of the solution was created specifically to meet Normative Instruction 896/17 of the National Treasury Secretariat (STN).

Brazilian companies that intend to trade their securities on *U.S.* stock exchanges must comply with the legislation of the *Securities and Exchange Commission* (SEC) [22] that determines the delivery of financial and accounting information in the XBRL format.

Therefore, the legal determinations of governments impose the needs on federal entities and legal entities regarding the use of XBRL, making it essential to recover and export data *in eXtensible Business Reporting Language*.

(Q1.2) What technologies are used to map XBRL data?

In the studies investigated, the discussion about the technologies used was done superficially, most of the studies did not mention or make any mention about the technologies used. Only in the work of Bragança and his colleagues [4] that the use of pentaho's ETL tool was commented. When investigating the XML and JSON languages, it was possible to notice the use of JDBC for the SGBD connection in the work of Mao and Ye [17] and the use of Microsoft SQL Server DBMS in the work of Song and his colleagues [19]. In the work of Bikakis and his colleagues [29] we identified the use of SPARQL and XQuery due to the use of these technologies the objective of the work.

(Q1.3) Are there proposals efficient for including any type of data source and XBRL?

Only a single proposal met this question, this to specifically meet a legal imposition [4], this work uses CSV files as a data source. When related to the JSON language together with CSV files, also investigated in this work, no

[22] https://www.sec.gov/

solutions were found that aimed to instantiate the data in XBRL-JSON or XBRL-CSV.

7. Key Challenges

The studies identified in this research show that the primary concern is related to the reading of XML instances, mapping of data and insertion in a relational database, while with XBRL instances the primary concern is focused on the generation of the instance.

However, the same concern is not perceived for reading, validating, and storing the data contained in XBRL instances, either using XML and JSON languages or CSV files. This problem persists and deserves the community's effort to make the exchange of financial and accounting data effective in both paths, that is, in the recovery of data contained in XBRL instances to be stored in diverse databases and in the generation of the XBRL instance from the different data formats.

The paper [40] addresses the complexity of the XML framework for data representation, this challenge can be perceived in the work that deals with reading XML documents and inserting data into relational databases. The techniques discussed, such as XAncestor [27], Mini-XML [19] and XML-REG [26] have devoted attention to identifying the nodes contained in the documents, labeling these nodes, and saving this mapping of the framework. This requires additional tables in databases and greater complexity in data recovery. Simplifying the structure of the XML document can contribute to facilitating the exchange of data and its storage [40].

It was identified that in the investigated studies, attention is mainly focused on relational databases, and the analysis with the perspective of DBMS's NoSQL is not abundant. Given that data from JSON instances is commonly stored in NoSQL databases [41] and that the XBRL consortium admits XBRL-JSON instances, studies need to be developed to identify best practices and best data mapping techniques.

Therefore, there are still challenges to overcome and unanswered problems such as exporting and retrieving validated data for XBRL documents, whether from XML, JSON, or CSV. These problems need to be addressed so that government entities' requirements and market needs can be met by consistent, efficient, and properly documented solutions.

8. Final Considerations

This literature review sought to identify the state of the art, conceptual definitions of terms, data mapping techniques, solutions based on frameworks and mishaps overcome or identified, described in the literature to generate XBRL instances and recover the data contained in it.

The research questions were partially answered, due to the studies not offering details about the technologies used and the absence of comparative results of their techniques with others available in the literature. Only the studies [17,19,22] present changes of their techniques or comparations between existing techniques. Analyzing the studies selected in this research, the only technique that presents adequate generality and a comparative study with the techniques documented in the literature is XML-REG, presented in the paper [26].

The small number of studies consistent with the questions of this research demonstrate the importance and urgency in conducting research that seeks solutions to recover data from XBRL instances and transform into formats that allow storage in relational or NoSQL environments, this concern was expressed in the studies [14,15].

Also, the need to have an effective way of exporting data to XBRL instances becomes increasingly urgent, since in the studies surveyed it was noticed that government entities adopt this format for the exchange of financial information.

This work identified promising formats, as is the case of XML-REG [26], which proved to be more efficient in the use of storage space and execution speed when compared with MINI-XML and XAncestor, however, the XML-REG technique lacks implementation in production environments so that the performance pointed out in the study can be, in fact, experienced.

No research has been identified that focus on identifying and analyzing libraries of programming languages that are intended to export data to XBRL instances and recover data from these documents.

It is noticed that there is a lack of research that meets the issues raised in this work, even with the need for data exchange in the evaluated formats being continuously growing due to the evolution and expansion of the Internet and legal impositions.

References

[1] G. Jayashree and C. Priya, "Data Integration with XML ETL Processing," *2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020*, no. March, 2020, doi: 10.1109/ICCSEA49143.2020.9132936.

[2] H. Zhu, H. Yu, G. Fan, and H. Sun, "Mini-XML: An efficient mapping approach between XML and relational database," *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, pp. 839–843, 2017, doi: 10.1109/ICIS.2017.7960109.

[3] M. M. M. Dunce, P. C. da Silva, and S. Viana, "SIMILARITY EVALUATION BETWEEN CONCEPTS REPRESENTED BY XBRL," 2013. Doi: 10.5748/9788599693094-10contecsi/ps-457.

[4] H. A. Bragança, P. C. da Silva and S. P. Ladislau. "XBRL-ETL ENGINE: A DATA TRANSFORMATION TOOL FOR XBRL-SICONFI TAXONOMY Motor XBRL-ETL: A tool for data transformation based on xbrl-siconfi taxonomy," no. 1, pp. 1–19, doi: 10.5748/16CONTECSI/XBR.

[5] D. Dermeval, J. A. P. de M. Coelho, and I. I. Bittencourt, "Systematic Mapping and Systematic Review of Computer Literature in Education," *Informatics Research Methodology in Education: Quantitative Approach to Research (Volume 2),*no. 2, pp. 1–26, 2019.

[6] P. C. Silva, L. Silva, A. Santos, and M. Cruz, "The Xbrl Framework," *International Conference on Information Systems and Technology Management 5th*, pp. 4343–4365, 2008.

[7] M. G. Cerqueira and P. C. da Silva, "Coming Impacts of Xbrl Adoption in Financial Software Development Processes and Software Quality Factors: a Systematic Mapping," *Proceedings of the 13th CONTECSI International Conference on Information Systems and Technology Management*, vol. 13, pp. 3185–3209, 2016, Doi: 10.5748/9788599693124-13contecsi/ps-4103.

[8] E. Riccio, M. Sakata, O. Moreira, and L. Quoniam, "Introduction to XBRL: new language for the dissemination of business information over the Internet," Information *Science*, vol. 35, No. 3, pp. 166–182, 2006, Doi: 10.1590/s0100-19652006000300016.

[9] K. R. N. Felizardo, *Systemotic Review of Software Engineering Literature: Theory and Practice*. 2017.

[10] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, 2007.

[11] I. Belev, "Alternatives for Storing and Validating XBRL Data," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS),*, vol. 60, no. 1, pp. 191–201, 2019.

[12] A. Schmidt *et al.* , "Why and how to benchmark XML databases," *SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 30, no. 3, pp. 27–32, 2001, doi: 10.1145/603867.603872.

[13] B. Bin Yao, M. T. Özsu, and N. Khandelwal, "XBench benchmark and performance testing of XML DBMSs," *Proceedings - International Conference on Data Engineering*, vol. 20, pp. 621–632, 2004, doi: 10.1109/ICDE.2004.1320032.

[14] E. Asimadi, S. Reiff-Marganiec, B. Donnelly, J. Baker, and D. Fang, "Semantic approach to financial data integration for enabling new insights," *CEUR Workshop Proceedings*, vol. 1890, pp. 1–15, 2017.

[15] D. Liu, U. Etudo, and V. Yoon, "X-IM framework to overcome semantic heterogeneity across XBRL filings," *Journal of the Association for Information Systems*, vol. 21, no. 4, pp. 971–1000, 2020, doi: 10.17705/1jais.00626.

[16] G. Jayashree and C. Priya, "Data Integration with XML ETL Processing," *2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020*, no. March, 2020, doi: 10.1109/ICCSEA49143.2020.9132936.

[17] J. Mao and X. Ye, "Relational schema and XML schema bidirectional mapping algorithm based

on the intermediate object tree," in *2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017*, 2018, vol. 2018-Janua, pp. 2380–2383. doi: 10.1109/CompComm.2017.8322961.

[18] H. Zhu, H. Yu, G. Fan, and H. Sun, "Mini-XML: An efficient mapping approach between XML and relational database," in *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, 2017, pp. 839–843. 10.1109/ICIS.2017.7960109.

[19] E. Song, S. C. Haw, and F. F. Chua, "Handling XML to relational database transformation using model-based mapping approaches," in *2018 IEEE Conference on Open Systems, ICOS 2018*, Nov. 2019, pp. 65–70. doi: 10.1109/ICOS.2018.8632805.

[20] H. Nassiri, M. Machkour, and M. Hachimi, "One query to retrieve XML and Relational Data," *Procedia Computer Science*, vol. 134, pp. 340–345, 2018, doi: 10.1016/j.procs.2018.07.201.

[21] Y. Chen, "Worst case optimal joins on relational and XML data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2018, pp. 1833–1835. doi: 10.1145/3183713.3183721.

[22] M. M. Gamal, A. E. A. Ahmed, H. A. Hefny, and M. A. El-Moneim, "A literature survey on mapping between fuzzy XML databases and relational or object oriented databases," in *Proceedings of 2015 IEEE World Conference on Complex Systems, WCCS 2015*, Nov. 2016, pp. 1–6. doi: 10.1109/ICoCS.2015.7483293.

[23] R. Salem *et al.* , "Active XML-based Web data integration To cite this version : HAL Id : hal-01433718 Active XML-based Web Data Integration," vol. 15, No. 3, 2017.

[24] A. V. Lyamin and E. N. Cherepovskaya, "XML-Relational mapping using production rule system," in *2017 Intelligent Systems Conference, IntelliSys 2017*, 2018, vol. 2018-Janua, pp. 422–429. doi: 10.1109/IntelliSys.2017.8324328.

[25] A. El Alami and M. Bahaj, "Framework for a complete migration of relational databases to other types of databases(object oriented OO, object-relational OR, XML)," in *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, Nov. 2017, pp. 1–7. doi: 10.1109/AICCSA.2016.7945763.

[26] E. Song and S.-C. Haw, "XML-REG: Transforming XML Into Relational Using Hybrid-Based Mapping Approach," *IEEE Access*, vol. 8, pp. 177623–177639, 2020, doi: 10.1109/ACCESS.2020.3026006.

[27] A. Qtaish and K. Ahmad, "XAncestor: An efficient mapping approach for storing and querying XML documents in relational database using path-based technique," *Knowledge-Based Systems*, vol. 114, pp. 167–192, 2016, https://doi.org/10.1016/j.knosys.2016.10.009.

[28] H. Nassiri, M. Machkour, and M. Hachimi, "Integrating XML and Relational Data," *Procedia Computer Science*, vol. 110, pp. 422–427, 2017, doi: 10.1016/j.procs.2017.06.107.

[29] N. Bikakis, C. Tsinaraki, I. Stavrakantonakis, N. Gioldasis, and S. Christodoulakis, "The SPARQL2XQuery interoperability framework: Utilizing Mapping Schema, Schema Transformation and Query Translation to Integrate XML and the Semantic Web," *World Wide Web*, vol. 18, no. 2, pp. 403–490, Mar. 2015, doi: 10.1007/s11280-013-0257-x.

[30] M. Niewerth and T. Schwentick, "Reasoning About XML Constraints Based on XML-to-Relational Mappings," *Theory of Computing Systems*, vol. 62, no. 8, pp. 1826–1879, Nov. 2018, doi: 10.1007/s00224-018-9846-5.

[31] A. M. Maatuk, M. A. Ali, and S. Aljawarneh, "An algorithm for constructing XML Schema documents from relational databases," in *ACM International Conference Proceeding Series*, 2015, vol. 24-26-Sept. doi: 10.1145/2832987.2833007.

[32] L. Bai, L. Yan, Z. M. Ma, and C. Xu, "Incorporating fuzziness in spatiotemporal XML and transforming fuzzy spatiotemporal data from XML to relational databases," *Applied Intelligence*, vol. 43, no. 4, pp. 707–721, Dec. 2015, doi: 10.1007/s10489-015-0677-7.

[33] D. Petković, "JSON Integration in Relational Database Systems," *International Journal of Computer Applications*, vol. 168, no. 5, pp. 14–19, 2017, doi: 10.5120/ijca2017914389.

[34] A. Habib, A. Shinnar, M. Hirzel, and M. Pradel, "Finding Data Compatibility Bugs with JSON Subschema Checking," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021, pp. 620–632. doi: 10.1145/3460319.3464796.

[35] N. Yaghmazadeh, X. Wang, and I. Dillig, "Automated Migration of Data to Relational Tables Using Programming-by-Example," *Proc. VLDB Endow.* , vol. 11, no. 5, pp. 580–593, Jan. 2018, doi: 10.1145/3177732.3177735.

[36] D. Petković, "SQL/JSON Standard: Properties and Deficiencies," *Datenbank-Spektrum*, vol. 17, no. 3, pp. 277–287, 2017, doi: 10.1007/s13222-017-0267-4.

[37] A. A. Frozza and R. dos S. Mello, "JS4Geo: a canonical JSON Schema for geographic data suitable to NoSQL databases," *GeoInformatica*, vol. 24, No. 4, pp. 987–1019, 2020, Doi: 10.1007/s10707-020-00415-w.

[38] Y. Doi and M. Toyama, "ToT for CSV: Accessing Open Data CSV Files through SQL," in *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services*, 2019, pp. 423–429. doi: 10.1145/3366030.3366130.

[39] C. Beelitz, "The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information," *CEUR Workshop Proceedings*, vol. 1890, pp. 1–11, 2017.

[40] N. Language, P. Centre, and C. Republic, "' Help , my XML is too complex !' – the problem of excessive structural markup in dictionaries ' Help , my XML is too complex !' – the problem of excessive structural markup in dictionaries," pp. 137–138, 2010.

[41] R. Bahta and M. Atay, "Translating JSON data into relational data using schema-oblivious approaches," in *ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 233–236. doi: 10.1145/3299815.3314467.

[42] S. Subramaniam, S. C. Haw, and P. Kuan Hoong, "S-XML: An efficient mapping scheme for storing XML data in a relational database," in *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, 2010, vol. 2, pp. V2-149-V2-153. 10.1109/ICACTE.2010.5579277.