Journal of Information Systems and Technology Management – Jistem USP
Vol. 22, 2025, e202522007
ISSN online: 1807-1775
DOI: 10.4301/S1807-1775202522007

Original Article

CONCEPTUAL MODEL FOR GENERATION OF XBRL INSTANCES

Henderson Acosta. Bragança[1] https://orcid.org/0000-0002-1922-9935
Silas Pinho Ladislau[2] https://orcid.org/0000-0002-1922-5141
Paulo Caetano da Silva[1] https://orcid.org/0000-0002-5038-2460
Daniel José Diaz[3] https://orcid.org/0000-0001-5459-1297

[1]Salvador University (UNIFACS), Salvador, BA, Brazil

[2]Legislative Assembly of the State of Rondônia, Porto Velho, RO, Brazil

[3]National University of Rosario, Faculty of Economic Sciences and Statistics, Rosario, Santa Fe, Argentina

ABSTRACT

The need to integrate eXtensible Business Reporting Language (XBRL) technology with data storage technologies is growing continuously. The growth in the use of eXtensible Business Reporting Language (XBRL) technology in the context of financial reporting on the internet has occurred due to its advantages and benefits or due to government impositions e.g. the Brazilian Public Sector Accounting and Tax Information System – Siconfi, which requires the delivery of accounting and financial information from the Federated States and Municipalities in XBRL. The growth in use has increased the need for solutions that allow the easy use of the language, in order to generate XBRL instances with the organization's data stored in relational databases, NoSQL (Not Only SQL) or even in CSV files. This work proposes an open-source Extract, Transform and Load (ETL) conceptual solution planned in two stages of execution in order to facilitate maintenance, evolutions and the possibility of extension to understand the most diverse taxonomies XBRL.

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

Manuscript first received: 2024-12-01. Manuscript accepted: 2025-09-01

Address for correspondence:

Henderson A. Bragança, Salvador University (UNIFACS), Salvador, BA, Brazil.
Email: henderson.braganca@gmail.com
Silas P. Ladislau, Legislative Assembly of the State of Rondônia, Porto Velho, RO, Brazil. Email: splhead@gmail.com
Paulo Caetano da Silva, Salvador University (UNIFACS), Salvador, BA, Brazil.
Email: caetano.paulo@animaeducacao.com.br
Daniel José Diaz, National University of Rosario, Faculty of Economic Sciences and Statistics, Rosario, Santa Fe, Argentina.
Email: djdiaz@yahoo.com

INTRODUCTION

For information to be contextualized in a given domain, it must be interconnected with other information, be it contextual, past information, experiences, or future information (Cerqueira & Silva, 2021) and (Beelitz, 2017). In the financial and accounting domain the language that meets these requirements is the XBRL (eXtensible Business Reporting Language)(XBRL The Business Reporting Standard [XBRL], 2023).

In 2008 Silva stated that the XBRL language was becoming a technological standard for the exchange, storage and dissemination of financial information on the Internet (Silva, Silva, Santos, & Cruz, 2008) and (Gray & Miller, 2009)since April 2005, the Securities and Exchange Commission (SEC, and in fact the forecast was confirmed, the adoption of the language by relevant institutions has driven the use of XBRL, as is the case of US-SEC (U.S. Securities and Exchange Commission [US-SEC], 2023) that pointed to the XBRL language as the key to its modernization (Gray & Miller, 2009)since April 2005, the Securities and Exchange Commission (SEC.

The XBRL language, because it is robust and completely represents the financial and accounting domain, brings with it the complexity inherent in solutions aimed at complex problems, such as the transport of financial and accounting information with all its contextual framework.

The generation of XBRL[1] instances using financial and accounting data repositories as a source is not trivial (Asimadi, Reiff-Marganiec, Donnelly, Baker, & Fang, 2017) only proprietary tools available on the market are able to automatically extract data stored in different repositories and using various storage technologies (e.g. relational, csv, json etc.) The absence of open-source tools may be making it difficult to use language in a massive way, especially in The Brazilian territory, whose only project is that of the Treasury Secretariat Nacional (STN), called SICONFI (Brazilian Public Sector Accounting and Fiscal Information System)(SICONFI, 2023). SICONFI requires federal entities to provide accounting and financial information in XBRL format. To achieve this, proprietary systems are used where, in many cases, accounting and financial data must be entered manually to generate the XBRL instance, since data extraction, especially from relational databases, is not trivial.

Initiatives to address the absence of solutions that have the ability to generate XBRL instances can be observed in (Bragança, Ladislau, da Silva, & da Silva, 2019), proposing an ETL tool[2] (Bragança *et. al*., 2019), however the proposed solution collects only data from CSV files, restricting the adoption capacity of the solution. It can be seen (Bragança *et. al*., 2019) there is a lack in the executive powers of Brazil, whether at the federal, state or municipal levels, for solutions that meet the demand in the generation of XBRL instances to be delivered to SICONFI.

Thus, an open source ETL solution is proposed in this work divided into two stages of execution in order to facilitate the development of new functionalities, the first stage is intended to connect to the data source and the second to link the data with the taxonomy[3] enabling the generation of the XBRL instance. The concept presented in this proposal seeks to meet the most diverse scenarios with reusable configurations and with reduced complexity. The solution was planned in order to facilitate maintenance and extension to generate instances of the most varied taxonomies.

This work is organized as follows: section 2 presents important concepts to the understanding of this work, in section 3 the details of the proposal are shown. In section 5, future work is presented, and finally, in section 4 the final considerations.

1 XBRL instances are XML documents based on the XBRL specification that contains the financial/accounting data to be reported.
2 ETL is the acronym for Extract, Transform and Load.
3 The taxonomy is a file responsible for the concepts of the financial and accounting terms.

THEORETICAL FOUNDATION

This proposal uses relational database technologies, NoSQL[4] and CSV[5] files as a data source, integrating with the XBRL language, which by XML (W3C, 2023) based language, uses XML Schema and Linkbases technologies. In the following sections will be presented the main concepts related to these technologies.

XBRL Language

The XBRL (eXtensible Business Reporting Language) language is an extension of the Extensible Markup Language (XML) language recommended by W3C (W3C, 2023) as an important language for exchanging data on the Internet (Bragança, Caetano, & Bernadino, 2022).

XBRL was developed to maintain the integrity of financial and accounting data at the time of exchange between organizations, providing a standard syntax and semantics of this domain. Its objective is to allow information to be represented and validated through a taxonomy guaranteed its integrity, even when transferred between countries and entities that have different accounting standards (Dunce, Silva, & Viana, 2013).

The basic structure of XBRL is composed of: (i) instance document, in this document the accounting and financial data are contained, along with the information of the specific context in which they are inserted and (ii) the taxonomy with the definition of the concepts of accounting and financial terms and their semantic relationships, the taxonomy is described by an XML Schema (W3C, 2023) document, which defines the vocabulary that will be used to represent the data to be exchanged, and by Linkbases, which are Xlink-based (W3C, 2023) documents, which express the semantic relationship between the concepts defined in the Schema document (Cerqueira & Silva, 2016).

The consortium responsible for XBRL defined in 2021 new ways of using the standard, defining as a final recommendation in addition to xml's extended XBRL, the Extended XBRL of JSON and CSV (Bragança *et. al*., 2022). XBRL in JSON and CSV are limited to instance, and the file containing the taxonomy is not allowed in these formats. Also, it was not clarified by the consortium responsible for XBRL what the XBRL-JSON and XBRL-CSV instances will look like when generated from the GL taxonomy.

The robustness and integrity provided by XBRL has been exploited by government regulatory agencies and financial institutions in several countries (e.g. US-SEC (US-SEC, 2023), FERC (FERC, 2023), European Committee of Banking Supervisors (CEBS) (CEBS, 2023), Secretariat of the National Treasury of Brazil - STN.

Data Sources

Databases are part of the daily life of the modern (Navathe, 2013), so it is possible to infer that organizations, of the most varied activities (e.g. industry, entertainment or in the context of the Internet), make use of different database technology s to remain competitive (Navathe, 2013).

---

4 NoSQL stands for "Not Only SQL" and refers to a type of non-relational database that stores data in flexible formats. These databases are designed to handle large volumes of unstructured and semi-structured data, and are scalable and more flexible for modern development.

5 Comma Separated Values (values separated by comma)

The concept given by Spink for databases tells us that “a database is a collection of logically related data, with some meaning” (Spink, Arouca, & Teixeira, 2002). However, database management is done by a collection of software that allows clients to create and manipulate (Spink *et. al*., 2002), these managers are known by the acronym SGBD (Database Management Systems). The use of databases has become indispensable in everyday Internet and in environments where the characteristics of Atomicity, Consistency, Isolation and Durability (B. E. Soares & Boscarioli, 2013) management systems provide to applications are indispensable.

There are different technologies for storing data, based on different models, among them the most common is relational and, more recently, NoSQL, commonly used on the internet. The relational data model represents the database as a (Navathe, 2013), which after its appearance in the 1960s, began to be used by most software applications, being a model that should continue to be used for many years due to its robustness (Navathe, 2013). Due to the importance and wide use in private and government applications, it is essential to consider this model as a data source.

NoSQL databases work with denormalized data (B. E. Soares & Boscarioli, 2013). NoSQL databases are being adopted by large companies such as Google, Facebook and Twitter (B. E. Soares & Boscarioli, 2013), since this model has a differentiated architecture to facilitate the treatment of significant demand for queries, linked to the high number of information they manipulate, with high data scalability (B. E. Soares & Boscarioli, 2013). Therefore, NoSQL databases are also considered as a data source alternative in the solution proposed in this work.

Therefore, the proposed solution should collect data from any sources, be it relational DBMS, NoSQL, or CSV files, and it is necessary to add libraries and connectors that meet the application user’s need.

RELATED WORKS

The related works were identified from the search string (XBRL AND (Integration OR Mapping OR FRAMEWORK OR Middleware) AND (Relational OR Data OR Database OR NoSQL OR CSV OR JSON)), where we sought to identify all the works that had for purpose integration solutions or mapping of data for XBRL and vice versa (Bragança *et al*., 2022).

The research was carried out through a systematic literature review (RSL), due to being a method that allows a broad view of the object studied (Dermeval, Coelho, & Bittencourt, 2019). Thus, searches related to mapping and integrating data to XML can meet expectations for data mapping for XBRL, so the search included the XML language.

The 28 papers resulting from the research present solutions and proposals that are consistent with the research. The storage and validation alternatives of XBRL instances have been identified. In (Belev, 2019) the authors state that the most efficient way to store XBRL instances is by using DBMS(Database Management Systems) capabilities, which natively use the XML model, in parallel with in-memory processing and the use of proprietary platforms to generate The 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 (Schmidt *et. al*., 2001)analyzing actual and potential bottle-necks, and, naturally, comparing the pros and cons of different systems architectures have become indispensable tasks as databases management systems grow in complexity and capacity. In the course of the development of XML databases the need for a benchmark framework has become more and more evident: a great many different ways to store XML data have been suggested in the past, each with its genuine advantages, disadvantages and consequences that propagate through the layers of a complex database system and need to be carefully considered. The different storage schemes render the query characteristics of the data variably different. However, no conclusive methodology for assessing these differences is available to date. In this paper, we outline desiderata for a benchmark for XML databases drawing from our own experience of developing an XML repository, involvement in the definition of the standard query language, and experience with standard benchmarks for relational databases.”,”author”:[{“dropping-particle”:””,”family”:”Schmidt”,”given”:”Albrecht”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Waas”,”given”:”Florian”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Kersten”,”given”:”Martin”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Florescu”,”given”:”Daniela”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Carey”,”given”:”Michael J.”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Manolescu”,”given”:”Ioana”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Busse”,”given”:”Ralph”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””}],”container-title”:”SIGMOD Record (ACM Special Interest Group on Management of Data and (Yao, Özsu, & Khandelwal, 2004). In addition, no solution is proposed for mapping other data sources to XBRL.

Developing a semantic approach to integrating, processing, and querying financial information embedded in XBRL instances is proposed by (Asimadi *et. al*., 2017). This work presents an approach to integrating instantiated financial records into XBRL in a semantically identifiable format, which allows you to run queries.

A method, called X-IM, for retrieving information stored in XBRL instances is presented in (Liu, Etudo, & Yoon, 2020)a key motivation for the SEC’s XBRL mandate. To mitigate this problem, several approaches leveraging Semantic Web technologies have emerged. While some approaches are promising, their mapping accuracy in resolving semantic heterogeneity must be improved to realize the promised benefits of XBRL. Considering this limitation and following the design science research methodology (DSRM and (Bragança *et. al*., 2019). It enables data interoperability by mapping elements with different but semantically identical names on XBRL instances.

Data integration methods and their challenges using ETL are discussed in (Liu *et. al*., 2020) a key motivation for the SEC’s XBRL mandate. To mitigate this problem, several approaches leveraging Semantic Web technologies have emerged. While some approaches are promising, their mapping accuracy in resolving semantic heterogeneity must be improved to realize the promised benefits of XBRL. Considering this limitation and following the design science research methodology (DSRM. 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 so as to be applied in any BI tool (Business Intelligence).

An analysis of the advantages of using XML in data exchange is discussed (Mao & Ye, 2018), following with hierarchical XML analysis to represent a relational schema of data. This work also aims to develop a mapping algorithm that exposes the data structure of the relational database preserving and semantic integrity. A conceptual framework is presented for mapping the data contained in relational databases and data in XML institutions, preserving the structure of the data, integrity, and constraints of the relational schema.

An approach to mapping data in XML instances to relational databases was proposed in (Zhu, Yu, Fan, & Sun, 2017), called Mini-XML. According to the authors, the main advantage is to map and store the relationships of XML nodes. They consider more efficiency and m storage time and disk space consumption than the proposed S-XML[6].

An analysis and the advantages and disadvantages of different approaches, such as those with 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 is proposed in (Song, Haw, & Chua, 2019). 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.

Nassiri propose a unique syntax for querying data in XML structures, relational databases, and hybrid structures (Nassiri, Machkour, & Hachimi, 2018). A translator capable of converting SQL queries was presented to meet data recovery in relational and XML structures. The authors suggest a hybrid structure that allows you to model the data in XML and represent it in relational tables.

Chen seeks to develop a multimodel processing framework for relational data and XML, and designs a join algorithm (Chen, 2018). It presents a query algorithm titled XJOIN that, according to the proposal, outperforms traditional algorithms of XML queries and relational databases at runtime.

The work of Gamal aims to identify different approaches and techniques that map fuzzy XML schemas to relational databases or object-oriented databases (Gamal, Ahmed, Hefny, & El-Moneim, 2016). In addition, it is expected to identify different fuzzy models, XML data models and fuzzy techniques integration process in different databases. This article presents a review of fuzzy literature on XML, relational databases, and object-oriented databases.

Developing an ETL algorithm that integrates XML data from the WEB with relational databases to facilitate analysis by DW/BI systems was proposed in (Salem *et. al*., 2017). This work presents a platform capable of integrating complex data originating on the WEB in real time (real-time), e.g. semi-structured, unstructured data, chat logs, emails, images, videos.

Lyamin 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 (Lyamin & Cherepovskaya, 2018)XML data is simply understandable, it has hierarchical model, which is quite redundant. Modern systems frequently apply various methods of XML-Relational Database Management System (RDBMS. 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.

A framework for migrating relational databases to other types of databases and hierarchical structures, as XML is proposed in (Alami & Bahaj, 2017). The authors present a framework for migrating relational databases to other types of databases with semi-automatic mapping using XML.

A solution for mapping XML instances to relational databases is proposed in (Song & Haw, 2020)both on the internet and local network. However, relational database (RDB. A mapping called XML-REG is presented that acts on reading the XML instance, handling nodes, and data to load into a relational database.

6 S-XML is a mapping scheme .XML. (Subramaniam *et. al*., 2010)

Qtaish proposes mapping XML instances to relational databases. They present a mapping based on two algorithms, one reading and the other that translates xPath to SQL queries, the mapping is titled XAncestor (Qtaish & Ahmad, 2016)so it needs to be managed effectively. There are four central problems in XML data management: capture, storage, retrieval, and exchange. Even though numerous database systems are available, the relational database (RDB.

Developing a connection between XML and relational databases, retrieving the data by abstracting the syntaxes of languages, and structural models (XML or RDB) is proposed in (Nassiri, Machkour, & Hachimi, 2017). This paper presents a system that aims to extract data independently of the query language and data storage model.

A framework, called SPARQL2XQuery, that translates SPARQL queries to XQuery allowing interoperability between semantic web and XML is presented in (Bikakis, Tsinaraki, Stavrakantonakis, Gioldasis, & Christodoulakis, 2015).

A framework that identifies the relationships between the data obtained in the XML instance and the fields of the tables in relational databases is proposed in (Niewerth & Schwentick, 2018) In it the mapping between XML and relational database using a tree pattern, where it considers an XML document as a tuple, with a single root, and all edges are far from the root. This work restricts the implications on the dependencies of relationships in the database, without considering null values.

Generating XML schemas, based on XML Schema, with data coming from relational databases is the proposal of (Frozza & Mello, 2020). This article presents a solution for translating relational databases into XML Schema, validating schema structure, data semantics, and integrity constraints.

A meteorological data modeling methodology called space-temporal fuzzy in XML and the transformation of space-time l fuzzy data from XML to relational database is the proposal of (Bai *et al.* 2015).

The ansi sql json investigan features of the different relational database systems (DBMS's) integrate them are discussed in (Dimou *et al*., 2014). This work shows insert codes and query codes (select) in JSON format. The author states that Oracle DBMS natively implements the ANSI SQL JSON concepts, while Microsoft SQL Server partially and PostgreSQL does not implement any ANSI SQL JSON concept.

A way to detect errors in data by prechecking the JSON schema is proposed in (Roohani, Furusho, & Koizumi, 2009). An application that validates JSON documents in a JSON schema is developed and distributed in the github repository with code in the Python programming language.

Yaghmazadeh developed an approach to the integration of tree structures, e.g. JSON and XML, for relational databases (C. S. Soares, Mallone, & Andrade, 2021). for this was developed a system entitled MITRA as acronym for "Migrating Information from Trees to Relations". the article visually shows the steps of the algorithm to perform data migration, however access to the source code is not available through the link provided in the article.

The advantages and disadvantages of integrating JSON and Database Management Systems and suggestions for troubleshooting identified issues are described in (Subramaniam, Haw, & Kuan Hoong, 2010). At work, the advantages in storing JSON in DBMS are identified, such as: (i) semi-structured data storage; (ii) reduced data management costs; (iii) Increased productivity for developers. It has also been identified that the most relevant problem is the lack of DBMS with native json integration capability.

In (Frozza & Mello, 2020) is proposed to represent geographic data using JSON, so that JSON Schema is integrated with DBMS's NoSQL. An extension of the JSON (schema) for Spatial Data compatible with DBMS's NoSQL, called JS4Geo, is suggested. The proposed standard facilitates integration between JSON documents with geographic data and DBMS's NoSQL.

The use of the Remote Table Access Framework (RTA) to make relational database tables available on the internet for consultation is discussed in (Doi & Toyama, 2019)." The authors describe the functioning of the RTA Framework and for cases where data originates in Comma Separated Values (CSV) presents the framework extension called Table on Top (ToT).

The investigation developed in this work identified only two studies (Bragança *et. al*. 2019) and (Liu *et. al*. 2020) that address the issue. It is salutary to observe that the proposals found in the research, have their focus on solving specific problems such as meeting the legislation or identifying semantic similarity in XBRL instances, that is, we can still see that the works are limited to dealing with a single path of information, being mostly in the sense of generating the XBRL instance. Due to the scarcity of papers dealing with XBRL and the consortium responsible for XBRL admitting XML, JSON and CSV files, it was decided to investigate mappings related to XML, JSON and CSV in search of works that satisfied this question. In this perspective we identify the works (Jayashree and Priya 2020), (Zhu *et. al*. 2017), (Song *et. al*. 2019), (Nassiri *et. al*. 2018), (Chen 2018), (Salem *et. al*. 2017), (Lyamin and Cherepovskaya 2018), (Alami and Bahaj 2017), (Song and Haw 2020), (Qtaish and Ahmad 2016), (Nassiri *et. al*. 2017), (Liu *et. al*. 2020), (Bikakis *et. al*. 2015), (Niewerth and Schwentick 2018), (Maatuk *et. al*. 2015), (Asimadi *et. al*. 2017), (Bai *et. al*. 2015), (Petković 2017a), (Yaghmazadeh *et. al*. 2018), (Petković 2017b) and (Doi and Toyama 2019) which deal with mapping the data from XML or JSON instances and CSV files to databases.(Bragança *et. al*., 2019)(Liu *et. al*., 2020)a key motivation for the SEC's XBRL mandate. To mitigate this problem, several approaches leveraging Semantic Web technologies have emerged. While some approaches are promising, their mapping accuracy in resolving semantic heterogeneity must be improved to realize the promised benefits of XBRL. Considering this limitation and following the design science research methodology (DSRM(Jayashree & Priya, 2020)integration of data from various sources forms the vital role. Many of the vital business initiatives for an organization requires the data to integrate. Such initiatives performed by creating a customized solution across every data source. In such given architecture, the complexity to integrate data source is proportionate to the increase in data sources. Use of eXtensible Markup Language (XML(Zhu *et. al*., 2017)(Song *et. al*., 2019)(Nassiri *et. al*., 2018)(Chen, 2018)(Salem *et. al*., 2017)(Lyamin & Cherepovskaya, 2018)XML data is simply understandable, it has hierarchical model, which is quite redundant. Modern systems frequently apply various methods of XML-Relational Database Management System (RDBMS(Alami & Bahaj, 2017)(Song & Haw, 2020)both on the internet and local network. However, relational database (RDB(Qtaish & Ahmad, 2016)so it needs to be managed effectively. There are four central problems in XML data management: capture, storage, retrieval, and exchange. Even though numerous database systems are available, the relational database (RDB(Nassiri *et. al*., 2017)(Liu *et. al*., 2020)a key motivation for the SEC's XBRL mandate. To mitigate this problem, several approaches leveraging Semantic Web technologies have emerged. While some approaches are promising, their mapping accuracy in resolving semantic heterogeneity must be improved to realize the promised benefits of XBRL. Considering this limitation and following the design science research methodology (DSRM(Bikakis *et. al*., 2015)(Niewerth & Schwentick, 2018)(Maatuk, Ali, & Aljawarneh, 2015)(Asimadi *et. al*., 2017)(Bai, Yan, Ma, & Xu, 2015)we study the methodology of modeling fuzzy spatiotemporal data in XML and transforming fuzzy spatiotemporal data from XML to relational databases as well. To accomplish this, we devise a

fuzzy spatiotemporal data model in XML to capture the semantics of fuzzy spatiotemporal features. To allow for better and platform independent sharing of fuzzy spatiotemporal data stored in a relational format, we propose a temporal edge approach to transform fuzzy spatiotemporal XML data into relational databases. The unique feature of our approach is that no schema information is required for transformation of fuzzy spatiotemporal data. Moreover, temporal, spatial, and fuzzy features of fuzzy spatiotemporal data in XML documents are taken into consideration. Finally, the experimental results demonstrate the performance advantages of our approach. Such approach of transformation could provide a significant consolidation of the interoperability of fuzzy spatiotemporal data from XML to relational databases.”,”author”:[{“dropping-particle”:””,”family”:”Bai”,”given”:”Luyi”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Yan”,”given”:”Li”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Ma”,”given”:”Z. M.”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””},{“dropping-particle”:””,”family”:”Xu”,”given”:”Changming”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””}],”container-title”:”Applied Intelligence”,”id”:”ITEM-1”,”issue”:”4”,”issued”:{“date-parts”:[[“2015”,”12”,”1”]]},”note”:”Este artigo traz na introdução um panorama das pesquisas e visões da integração do XML para o Relacional\n\nCandidato a snowboling”,”page”:”707-721”,”publisher”:”Springer New York LLC”,”title”:”Incorporating fuzziness in spatiotemporal XML and transforming fuzzy spatiotemporal data from XML to relational databases”,”type”:”article-journal”,”volume”:”43”},”uris”:[“http://www.mendeley.com/documents/?uuid=5205c95d-402f-3504-b960-99917350f373”]}],”mendeley”:{“formattedCitation”:”(Bai, Yan, Ma, & Xu, 2015(Petković, 2017a)(Yaghmazadeh, Wang, & Dillig, 2018)(Petković, 2017b)(Doi & Toyama, 2019)

Therefore, there are works that deal with the mapping of XBRL data and mappings related to the languages admitted by the consortium responsible for XBRL.It is observed in the investigated studies the effort to overcome the difficulties related to the exchange of data between data source in the different models (Relational, NoSQL, CSV) and XBRL or XML documents.

DATA MAPPING PROCESS TO XBRL

The proposed model, represented in Figure 1, aims to connect relational or NoSQL databases and enable the joining of the data extracted from these databases with the taxonomy to be used generating the XBRL instance.

The proposal of the XBRL instance generation process was divided into two stages: (A) connection to the database and (B) generation of the XBRL instance from the informed taxonomy.

Dividing the solution into two steps will allow better maintenance in the code and development of possible improvements, with this each step has its functional context, increasing cohesion and decreasing coupling. In software, modularization allows a change in one component to have minimal impact on others.(Lazzari & Farias, 2022)

Connection to databases

The process of connecting to the database begins with the choice of the DBMS, as we can see in Figure 1(A-3), the list of supported DBMS’s will be limited only by the available *drivers*, however, any driver can be added, e.g. driver for PostgreSQL, Oracle, MySQL, SQL Server, MongoDB, or other that is necessary for the extraction of data for the generation of the XBRL instance.

In the sequence Figure 1(A-4), the connection data is requested in order to allow the connection to the DBMS. Data such as TCP port, login and password are indispensable information for the connection. When the connection data is entered, a connection test is run.

After completing the connection process, the execution of the insertion of the code of the query Figure 1(A-5), e.g.SQL or the code pertinent to the DBMS selected in the previous activities, begins. The possibility of inserting the direct code in the application, allows a flexibility in the query of the data in order to facilitate adjustments and addition of columns with static data, e.g. the information related to the financial values being in Brazilian currency (BRL). After the code is executed, it can be seen in Figure 1(A-6) that the next activity is the display of the query result for conferencing.

The connection to the database was planned so that all the data and parameterizations can be reused in a later execution, for this the file with the settings is reused in the subsequent executions Figure 1(A-2).

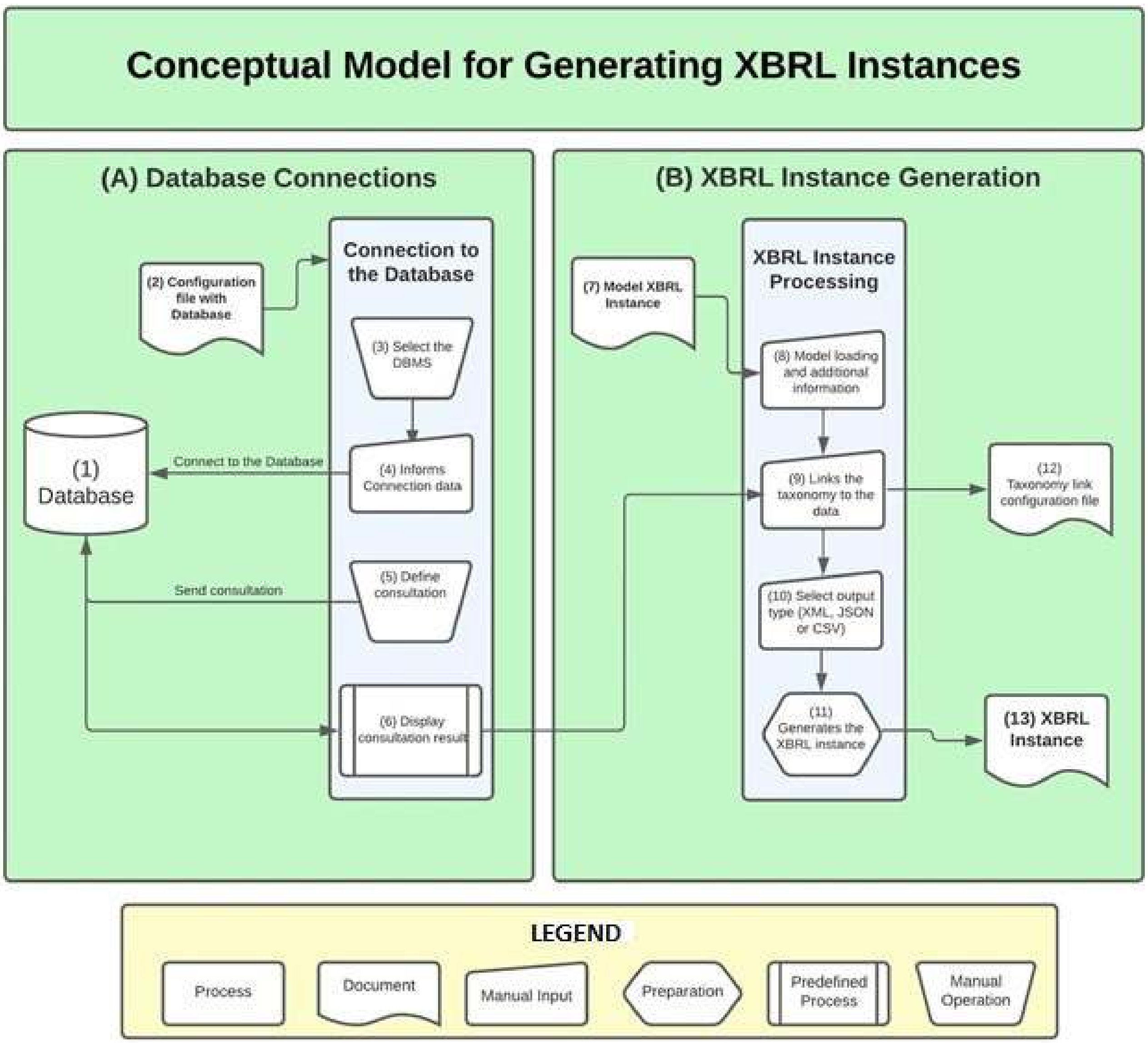

Figure 1. Conceptual model for XBRL instance generation

XBRL instance generation

The process of linking taxonomy to data is not an ordinary process, Asimad pointed out in his work that the XBRL integration process remains complex, so the beginning of XBRL instance processing, Figure 1(B-7), brings the possibility of using a model XBRL instance to retrieve as many links as possible between the taxonomy and the data. The model instance can be provided by the creator of the taxonomy to which the data will be sent. (Asimadi *et. al*., 2017)

In the next step, complementary data can be added Figure 1(B-8), it is understood as complementary information those that are unique to the organization that is generating the XBRL instance, e.g. registration code of the organization with the agency that will receive the data.

After the first adjustments in the insertion of the model taxonomy and complementary information, the link of the data received from the database with the taxonomy Figure 1(B-9) is created, information that is used repeatedly can be preserved, if necessary for later retrieval, in the pertinent configuration file Figure 1(B-12). (Dimou *et. al*., 2014)

The XBRL consortium after the year 2021 defined new ways to use the technology, defining as a recommendation beyond the extended XBRL of XML, the extended XBRL of JSON and CSV files. As seen in Chapter 2, XBRL in JSON and CSV are limited to instance, and the file containing the taxonomy is not allowed in these formats. Also, it was not clarified by the consortium responsible for XBRL what the XBRL-JSON and XBRL-CSV instances will look like when generated from the GL taxonomy. However, to be aligned with the recommendations of the XBRL consortium it is necessary to allow the possibility of generating the instance in the three formats admitted by the XBRL consortium. This prediction exists at this stage of the process, as can be seen in Figure 1 (B-10).(Cerqueira & Silva, 2016)

Finally, the data processing is performed to generate the XBRL instance Figure 1 (B-11), having as product the instance with all the data and its links with the taxonomy, as well as the complementary data of unit and references through the links, Figure 1 (B-13).

CONCLUSION AND FUTURE WORK

The solutions to generate XBRL instances available on the market have two problems that prevent them from being considered suitable for expressive data volumes, (i) unable to connect in DBMS's for data collection, (ii) necessary to enter all data that will be instantiated in XBRL, e.g. Fujitsu's XWand software (Roohani *et. al*., 2009). Consistent with the problems pointed out is the fact that they are not free, so little open source.

The proposal presented has as a basic premise to be open source, providing the extension of the code to meet the most diverse scenarios, so we idealize it to be a solution that can connect a wide range of databases.

In future works it is intended to code this model presented in order to validate it. It is also necessary to solve the problem in the delivery of the Accounting Balance Matrix - MSC of the Federated States of Brazil (C. S. Soares *et. al*., 2021) with the parameterized solution for the taxonomy of the National Treasury Secretariat - STN.

REFERENCES

Alami, A. El, & Bahaj, M. (2017). 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* (pp. 1–7). https://doi.org/10.1109/AICCSA.2016.7945763

Asimadi, E., Reiff-Marganiec, S., Donnelly, B., Baker, J., & Fang, D. (2017). Semantic approach to financial data integration for enabling new insights. *CEUR Workshop Proceedings*, *1890*, 1–15.

Bai, L., Yan, L., Ma, Z. M., & Xu, C. (2015). Incorporating fuzziness in spatiotemporal XML and transforming fuzzy spatiotemporal data from XML to relational databases. *Applied Intelligence*, *43*(4), 707–721. https://doi.org/10.1007/s10489-015-0677-7

Beelitz, C. (2017). The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information. *CEUR Workshop Proceedings*, *1890*, 1–11.

Belev, I. (2019). Alternatives for Storing and Validating XBRL Data. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *60*(1), 191–201. Retrieved from: https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/5289

Bikakis, N., Tsinaraki, C., Stavrakantonakis, I., Gioldasis, N., & Christodoulakis, S. (2015). The SPARQL2XQuery interoperability framework: Utilizing Schema Mapping, Schema Transformation and Query Translation to Integrate XML and the Semantic Web. *World Wide Web*, *18*(2), 403–490. https://doi.org/10.1007/s11280-013-0257-x

Bragança, H. A., Caetano, P., & Bernadino, N. (2022). Data Mapping for XBRL : A Systematic Literature Review. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, *90*, 124–143. Retrieved from http://asrjetsjournal.org/

Bragança, H. A., Ladislau, S. P., da Silva, M. A. P., & da Silva, P. C. (2019). XBRL-ETL ENGINE: A DATA TRANSFORMATION TOOL FOR XBRL-SICONFI TAXONOMY Motor XBRL-ETL: Uma ferramenta para transformação de dados baseada na taxonomia XBRL-SICONFI, (1), 1–19. https://doi.org/10.5748/16CONTECSI/XBR

CEBS. (2023). European Committee of Banking Supervisors. Retrieved from: https://www.bankingsupervision.europa.eu/home/html/index.en.html

Cerqueira, M. G. De, & Silva, P. C. Da. (2021). A survey of XBRL adoption impact on financial software development processes and software quality. *International Journal of Business Information Systems*, *37*(2), 263–286. https://doi.org/10.1504/IJBIS.2021.115366

Cerqueira, M. G., & Silva, P. C. da. (2016). 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*, *13*, 3185–3209. https://doi.org/10.5748/9788599693124-13contecsi/ps-4103

Chen, Y. (2018). Worst case optimal joins on relational and XML data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 1833–1835). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3183713.3183721

Dermeval, D., Coelho, J. A. P. de M., & Bittencourt, I. I. (2019). Mapeamento Sistemático e Revisão Sistemática da Literatura em Informática na Educação. *Metodologia de Pesquisa Em Informática Na Educação: Abordagem Quantitativa de Pesquisa (Volume 2)*, (2), 1–26. Retrieved from: https://metodologia.ceie-br.org/livro-2

Dimou, A., Sande, M. Vander, Colpaert, P., Verborgh, R., Mannens, E., & Van De Walle, R. (2014). RML: A generic language for integrated RDF mappings of heterogeneous data. *CEUR Workshop Proceedings*, *1184*.

Doi, Y., & Toyama, M. (2019). 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* (pp. 423–429). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3366030.3366130

Dunce, M. M. M., Silva, P. C. da, & Viana, S. (2013). Similarity Evaluation Between Concepts Represented By Xbrl, 3933–3963. https://doi.org/10.5748/9788599693094-10contecsi/ps-457

FERC. (2023). Federal Energy Regulatory Commission. Retrieved from https://www.ferc.gov/

Frozza, A. A., & Mello, R. dos S. (2020). JS4Geo: a canonical JSON Schema for geographic data suitable to NoSQL databases. *GeoInformatica*, *24*(4), 987–1019. https://doi.org/10.1007/s10707-020-00415-w

Gamal, M. M., Ahmed, A. E. A., Hefny, H. A., & El-Moneim, M. A. (2016). 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* (pp. 1–6). https://doi.org/10.1109/ICoCS.2015.7483293

Gray, G. L., & Miller, D. W. (2009). XBRL: Solving real-world problems. *International Journal of Disclosure and Governance*, *6*(3), 207–223. https://doi.org/10.1057/jdg.2009.8

Jayashree, G., & Priya, C. (2020). Data Integration with XML ETL Processing. *2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020*, (March). https://doi.org/10.1109/ICCSEA49143.2020.9132936

Lazzari, L., & Farias, K. (2022). An exploratory study on the effects of event-driven architecture on software modularity. .https://doi.org/10.48550/arXiv.2110.14699

Liu, D., Etudo, U., & Yoon, V. (2020). X-IM framework to overcome semantic heterogeneity across XBRL filings. *Journal of the Association for Information Systems*, *21*(4), 971–1000. https://doi.org/10.17705/1jais.00626

Lyamin, A. V., & Cherepovskaya, E. N. (2018). XML-Relational mapping using production rule system. In *2017 Intelligent Systems Conference, IntelliSys 2017* (Vol. 2018-Janua, pp. 422–429). https://doi.org/10.1109/IntelliSys.2017.8324328

Maatuk, A. M., Ali, M. A., & Aljawarneh, S. (2015). An algorithm for constructing XML Schema documents from relational databases. In *ACM International Conference Proceeding Series* (Vol. 24-26-Sept). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2832987.2833007

Mao, J., & Ye, X. (2018). 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* (Vol. 2018-Janua, pp. 2380–2383). https://doi.org/10.1109/CompComm.2017.8322961

Nassiri, H., Machkour, M., & Hachimi, M. (2017). Integrating XML and Relational Data. *Procedia Computer Science*, *110*, 422–427. https://doi.org/10.1016/j.procs.2017.06.107

Nassiri, H., Machkour, M., & Hachimi, M. (2018). One query to retrieve XML and Relational Data. *Procedia Computer Science*, *134*, 340–345. https://doi.org/10.1016/j.procs.2018.07.201

Navathe, E. &. (2013). *Sistemas de Banco de Dados*. *Journal of Chemical Information and Modeling* (Vol. 6ed).

Niewerth, M., & Schwentick, T. (2018). Reasoning About XML Constraints Based on XML-to-Relational Mappings. *Theory of Computing Systems*, *62*(8), 1826–1879. https://doi.org/10.1007/s00224-018-9846-5

Petković, D. (2017a). JSON Integration in Relational Database Systems. *International Journal of Computer Applications*, *168*(5), 14–19. https://doi.org/10.5120/ijca2017914389

Petković, D. (2017b). SQL/JSON Standard: Properties and Deficiencies. *Datenbank-Spektrum*, *17*(3), 277–287. https://doi.org/10.1007/s13222-017-0267-4

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

Roohani, S., Furusho, Y., & Koizumi, M. (2009). XBRL: Improving transparency and monitoring functions of corporate governance. *International Journal of Disclosure and Governance*, *6*(4), 355–369. https://doi.org/10.1057/jdg.2009.17

Salem, R., Darmont, J., Boussaid, O., Salem, R., Darmont, J., Boussaid, O., … Boussa, O. (2017). Active XML-based Web data integration To cite this version : HAL Id : hal-01433718 Active XML-based Web Data Integration, *15*(3).

Schmidt, A., Waas, F., Kersten, M., Florescu, D., Carey, M. J., Manolescu, I., & Busse, R. (2001). Why and how to benchmark XML databases. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *30*(3), 27–32. https://doi.org/10.1145/603867.603872

SICONFI. (2023). Secretaria do Tesouro Nacional. Retrieved from https://siconfi.tesouro.gov.br/siconfi/index.jsf

Silva, P. C., Silva, L., Santos, A., & Cruz, M. (2008). O Framework Xbrl. *International Conference on Information Systems and Technology Management 5th*, 4343–4365.

Soares, B. E., & Boscarioli, C. (2013). Modelo de Banco de Dados Colunar: Características, Aplicações e Exemplos de Sistemas. *Escola Regional de Banco de Dados–Sociedade Brasileira de Computação (IX ERBD–SBC)*. Retrieved from https://turing.pro.br/anais/ERBD-2013/artigos/pesquisa/111410.pdf

Soares, C. S., Mallone, V., & Andrade, N. De. (2021). Gestão pública municipal e os processos internos determinantes para o envio da matriz de saldos contábeis.

Song, E., & Haw, S.-C. (2020). XML-REG: Transforming XML Into Relational Using Hybrid-Based Mapping Approach. *IEEE Access*, *8*, 177623–177639. https://doi.org/10.1109/ACCESS.2020.3026006

Song, E., Haw, S. C., & Chua, F. F. (2019). Handling XML to relational database transformation using model-based mapping approaches. In *2018 IEEE Conference on Open Systems, ICOS 2018* (pp. 65–70). https://doi.org/10.1109/ICOS.2018.8632805

Spink, P., Arouca, F. L., & Teixeira, M. A. (2002). O Banco de Dados. *Cadernos Gestão Pública e Cidadania*, *7*(22). https://doi.org/10.12660/cgpc.v7n22.52342

Subramaniam, S., Haw, S. C., & Kuan Hoong, P. (2010). 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* (Vol. 2, pp. V2-149-V2-153). https://doi.org/10.1109/ICACTE.2010.5579277

US-SEC. (2023). US-SEC. Retrieved from https://www.sec.gov/

W3C. (2023). W3C. Retrieved from https://www.w3.org/

XBRL. (2023). XBRL The Business Reporting Standard. Retrieved from https://xbrl.org/

Yaghmazadeh, N., Wang, X., & Dillig, I. (2018). Automated Migration of Hierarchical Data to Relational

Tables Using Programming-by-Example. *Proc. VLDB Endow.*, *11*(5), 580–593. https://doi.org/10.1145/3177732.3177735

Yao, B. Bin, Özsu, M. T., & Khandelwal, N. (2004). XBench benchmark and performance testing of XML DBMSs. *Proceedings - International Conference on Data Engineering*, *20*, 621–632. https://doi.org/10.1109/ICDE.2004.1320032

Zhu, H., Yu, H., Fan, G., & Sun, H. (2017). 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* (pp. 839–843). https://doi.org/10.1109/ICIS.2017.7960109

Editor-in-chief: Edson Luiz Riccio

Data Availability Statement: All data generated or analysed during this study are included in this published article.