Towards a Knowledge Graph-specific Definition of Digital Transformation: An Account Networking View for Auditing

Florina Livia Covaci[1] [a], Robert Andrei Buchmann[1] [b] and Radu Dragos[2]
[1]*Business Information Systems Department, "Babeș-Bolyai" University, Cluj-Napoca, Romania*
[2]*Computer Science Department, "Babeș-Bolyai" University, Cluj-Napoca, Romania*

Keywords: Knowledge Graph, Accounting Auditing, Digital Transformation, RDF.

Abstract: This paper reports on an experimental Digital Transformation project where RDF graphs are adopted in an organization's accounting and document management system as a novel approach to accounting digitization, going beyond traditional ERP systems to enable account-centric network analysis and more insightful master data management - having accounting contextualized in a relationship-rich Knowledge Graph that captures some of the tacit knowledge that accountants and auditors apply during their common tasks. Legacy ERP systems that are based on relational databases face challenges when aggregating information regarding the transactions that an account was involved in, which sometimes involve multihop JOINs, links to contextual documents that may reside elsewhere, or rules that mimic (at least partially) an auditor's reasoning. The paper reports on a knowledge capture effort for mapping accounting information into an RDF graph in order to overcome limitations of legacy systems with auditing support, currently implemented in a feasibility demonstrator of low Technological Readiness Level. As theoretical implications, we also derive from this experience a novel, specialized definition of Digital Transformation.

1 INTRODUCTION

Every organization maintains systematic records of financial transactions, typically employing an ERP system to support accounting activities. Records of financial transactions are carried out based on accounting principles which primarily treat data as table structures and reports, although implied semantics are always present in the form of an accountant's tacit knowledge about how accounts participate together in double-entry records in various situations and how those entries are co-dependent based on either practical patterns or accounting-specific regulations. Moreover, occurrence of errors in financial statements is a problem to be dealt with on a monthly basis and legacy accounting systems offer limited capabilities sometimes requiring SQL skills or the support of an IT department to look for non-compliant patterns in multi/self-JOIN chains. The objective of this paper is to report on a Design Science project that adopted RDF graphs (Bizer C., 2009) as a treatment to a master data management problem in a legacy accounting system (and not only, but the current paper's scope is limited to the work of the accounting digitization team).

An accounting information system should hold not only financial data records, it should also accumulate and manipulate knowledge (i.e. semantic links and domain-specific rules), and then use that to mimic the reasoning and data navigation patterns of an accounting professional. Knowledge representation has gained importance in recent years because of the rise of semantic technology advertised as "Knowledge Graphs" (KG) which are indicated in recent Gartner reports as being both an Artificial Intelligence hype ingredient (Gartner, 2021a) and a data analytics trend (Gartner, 2021b), with possible ramifications in various Knowledge Management aspects.

The work at hand is an effort to translate some of those qualities to accounting data analysis that is relevant for catching accounting errors or investigating accounting patterns that an auditor would also look for, typically by visual scrutiny. Therefore, the problem statement is hereby formulated in terms of the Design Science problem template (Wieringa, 2014):

Improve auditing capabilities with an existing accounting information system (**problem context**)

[a] https://orcid.org/0000-0003-1184-5992
[b] https://orcid.org/0000-0002-7385-1610

Covaci, F., Buchmann, R. and Dragos, R.
Towards a Knowledge Graph-specific Definition of Digital Transformation: An Account Networking View for Auditing.
DOI: 10.5220/0010875000003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 637-644
ISBN: 978-989-758-547-0; ISSN: 2184-433X

...by treating it with a Knowledge Graph layer over existing ledger data (**artifact**)

...to enable quasi-social views focusing on financial records connectedness (**requirement**)

...in order to enable auditing reasoning patterns and contextualized accounting data management (**goals**)

Design science is often used to build new systems in order to evaluate whether their prescriptions are feasible and useful, and to gain deeper insights into the problem being investigated. The field of accounting information systems can benefit greatly from this methodology as domain-specific information, tacit knowledge and implicit reasoning patterns can be captured in innovative design decisions.

The larger context for the problem tackled here is an experimental Digital Transformation project towards a data-centric IT architecture, from an application-centric mindset that has lead over time to financial data silos requiring scheduled synchronization, as well as time consuming manual consolidation and verification. To obtain a contextualized account network, a GraphDB instance (Ontotext, b) is populated through its OntoRefine plug-in for lifting legacy tabular data to RDF graphs (Ontotext, c). Semantic queries and SPARQL-based reasoning patterns are then employed in an account analytics workbench to build a network of how accounts interact with each other depending on their co-occurrences in a double-entry ledger system, and to navigate those relationships according to some patterns of domain-specific interest.

The remainder of the paper is structured as follows: Section 2 summarizes the Accounting Cycle, afterwards Section 3 describes the knowledge capture process and derived semantic patterns. Section 4 discusses evaluation challenges and Section 5 presents a SWOT analysis.

2 THE FINANCIAL ACCOUNTING CYCLE

In a business, every transaction affects at least two accounts - as a debit (increase in assets) and as a credit (increase in liability, equity, income). The debit and credit entries must always be equal. After the transactions are recorded in the general journal the financial statements are prepared starting with a trial balance sheet. A balance sheet is a financial snapshot of a company's financial position at a specific point in time. It includes a list of assets, liabilities, and the difference between the two, known as net worth. The balance sheet is built on the accounting equation (assets = liabilities + owner's equity).

The accounting cycle is a series of steps that transform a company's basic financial data into financial statements. The accounting cycle ensures that the company's financial statements are consistent, accurate, and in compliance with official accounting standards.

The accounting cycle consists of the following six steps:

Step 1: Gather and analyze documentation for the current accounting period, such as receipts, invoices, and bank statements.

Step 2: The ledger is made up of journal entries, which are a chronological list of all of a company's transactions, written down according to double-entry accounting procedures. To ensure that the company's bookkeeping is always up to date, journal entries are recorded to the ledger on a continual basis, as soon as business transactions occur.

Step 3: Prepare a trial balance: this is analyzed mainly for the accounts that involve expenses and income accounts in correlation with the result of assets and liabilities accounts. Based on this correlation possible errors in the recorded transactions are investigated by analyzing the account statement.

The error detecting process requires a significant amount of time because it implies the human analysis of multiple inter-related account statements that reflect an economic event. The following types of errors may occurs in the process of recording of transactions:(Renu G., 2013)

A. Clerical Errors. Clerical errors are errors in recording, posting, totalling, and balancing. Clerical errors are further separated into two types: (I) omission errors and (ii) commission errors (e.g. posting in wrong account, error in totaling and balancing; errors in carry forward totals to trial balance and so on). Clerical errors may or may not have an impact on trial balance.

B. Errors of Principle. When commonly accepted accounting principles are not followed when recording the transactions in the books of account, it is called a principle error. For example, selecting the incorrect account head or declaring capital expenditure as revenue. A trial balance or routine inspection will not reveal such an inaccuracy. It can only be discovered through searching or independent verification.

C. Compensating Errors. Compensating errors are ones that occur as a result of previous faults being compensated for. This is difficult to detect because the net effect is zero. The totals and posts can all be checked for these inaccuracies, which have no effect on the trial balance.

D. Errors of Duplication. Duplication errors arise when the identical transaction is entered twice in the books of original entry and hence twice in the ledger accounts. These have no impact on trial balance.

Step 4: Adjust the identified errors: based on the identified errors in the previous step the recorded transactions are adjusted in order to correctly reflect the economic event.

Step 5: Prepare an adjusted trial balance, with the adjusted transactions and amounts. This balance sheet will be the starting point for preparing the company financial statements.

Step 6: Prepare financial statements that will offer a view about the performance of the company.

Knowledge on accounting patterns can be a basis for designing accounting intelligence systems and this knowledge often relates to aggregating information across multi-hop relationships that exist over the network of interacting accounts. Knowledge Graphs are a natural machine-readable information structure that can capture this network.

3 FROM TACIT KNOWLEDGE TO KNOWLEDGE GRAPH

The data-centric mindset (DCManifesto, 2021) is in line with an auditor's default mindset, one which reasons in accounting principles while navigating a data context. The navigable "data context" is formed of all the direct and indirect relationships to which a data point participates, limited by a relevance threshold - e.g., the number of "hops" in indirect relationships after which the semantic connectedness becomes too weak to be of interest for the operational purpose). This highly synthetic notion of "context" was recently formulated in (Cagle, 2021) and it is the asset primarily harnessed during KG-based Digital Transformation.

In the host organization, such context navigation is not typically supported by the existing ERP's rigid tabular views, leaving it to a human auditor to manually browse printed account cards (as suggested in Figure 1) in a manner that resembles Knowledge Graph navigation but is performed on paper, a strenuous exploration - at best assisted by search-and-filter features on tabular views in the ERP system.

The legacy accounting system has limited capabilities (mostly basic form validation) for providing auditing support in identifying errors. The identification of errors is usually made based on the account statement - a periodic summary of that account's activity. The individual verification is a time consuming process since the number of accounting transactions can be in a range of 20000-25000/month, but it is fundamentally a navigation of the graph of account associations, not unlike the KG browsing that open knowledge repositories like DBPedia provide in a browser (DBpedia, 2021).

In order to build the Knowledge Graph for our Design Science project we followed a knowledge capture approach that involved three sources of knowledge:

- Database Engineer – staff that contributed to the development and implementation of the existing ERP software, thus having a good understanding the boundaries between legacy data silos and the improvised conventions for synchronizing them;
- Domain-specific Operators - staff with experience in financial-accounting field, the actual users of the legacy systems able to demonstrate how work is performed;
- National legislation - certain accounting rules are prescribed as double-entry associations in natural language form - available to domain-specific operators as professional knowledge but not embedded in ERP systems functionality. Although some changes in legislation are occasionally observed, they are rather rare and should be easily edited as association rules that an information system can automatically confront with ledger records.

The knowledge capture process comprised the following steps:

1. Identification of the main entities (in ER sense) and corresponding tables of the existing ERP system. The structure of the tables together with a data samples were isolated in spreadsheets and subjected to discussions to clarify the (sometimes cryptic) meaning of data fields;
2. Person-to-person interactions with think aloud operations recorded to capture the data browsing experience and reasoning assumptions of the domain-specific operators, isolating the hints they look for during their monthly verifications;
3. Discussions between the Knowledge Engineers and the Database Engineers to identify gaps in the accounting cycle that are not explicitly captured in current data silos schemas;
4. Retrieval of accounting rules from the national legislation, those that can take the form of double-entry association rules - i.e. which accounts are allowed to credit a given account;
5. Several iterations of this process were necessary to eliminate understanding gaps between three

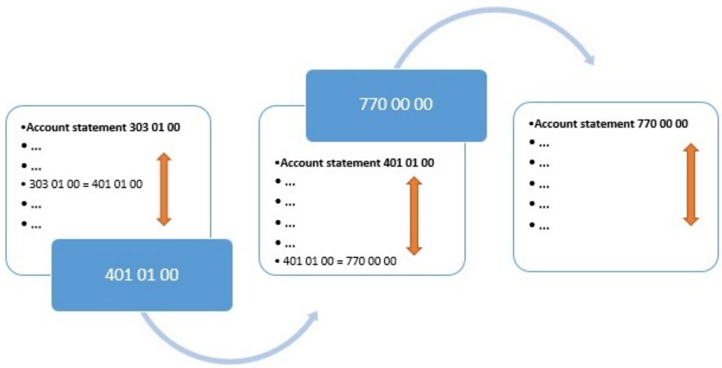

Figure 1: Manual Account Statement Analysis Process.

categories of stakeholders (legacy database engineer, operators and knowledge engineer).

Graph building was partly manual (the language of the national legislation does not have reasonable natural language support and creating such support is out of the project's scope) and partly based on the OntoRefine ETL-like approach provided by GraphDB, capable of semantic lifting and reconciling tabular data sources. In the following we will reveal some of the semantic patterns resulting from this process.

In a traditional ERP each transaction is recorded in a double-entry ledger based on the raw information that characterizes it, with attributes such as Debit Account, Credit Account, Value of transaction, Date of transaction. Depending on the complexity of the organization, the recording of the transactions may require additional information like funding sources or detailed subtypes of expense/income.

To illustrate the design decisions underlying the Knowledge Graph approach, in the following we showcase the required accounting transactions related to the economic event of acquisition of inventory items. In the debit of account 303 01 00 "Inventory items in the warehouse" a record registered the value at registration price of the inventory objects purchased from third parties based on the invoice issued. The credit account will be 401 01 00 "Suppliers"(1). At the same time in the debit of the account 442 60 00 "Deductible value added tax" and in the credit of account 401 01 00 there is need to be recorded the payed added value tax for the purchased inventory items (2). When the inventory item is put into use in the debit of account 303 02 00 "Inventory items in use" and in the credit of account 303 01 00 is registered the price of the inventory items (3). The payment of the invoice is registered in the debit of the account 401 00 00 and the credit of account 770 00 00 "Available funds" (4). When the inventory item is out of use the value at registration price of the inventory objects is registered in the debit of account 603 00 00 "Expenditure on inventory items" and the credit of account 302 00 00 (5).

Table 1 provides a summary of the transactions described above in a similar manner that they are recorded in a table in a relational database.

Such tables were converted into a network of accounts whose pairing can be weighted and enriched by a number of aggregate properties. The work does not currently employ a full fledged ontology (like e.g. FIBO) as it focuses on SPARQL-based RDF-star reasoning that can enrich the graph or aggregate information over relationship-rich data. Figure 2 shows some of these records in a linked data form revealing chains of how accounts influence each other by participating in the same ledger double-entry.

Regulations on how accounts can be associated are available in the accounting law and belong to the profession's domain knowledge - they form a dependency graph that can support certain levels of error checking. Figure 3 shows a graph fragment with accounting rules related to the recorded transactions in Figure 2, also suggesting on the left side the national law (in textual form) from which these rules have been derived.

The knowledge graph can be exploited for identifying errors using SPARQL-based reasoning rules which are deterministic. For compliance with the national law, the graph fragment with accounting rules in Figure 3 becomes a reasoning premise to detect principle accounting errors:

```
INSERT
{?transaction a :principleError}
WHERE
{
?transaction :debitAccount ?cd;
:creditAccount ?cc.
FILTER NOT EXISTS
        {GRAPH :AccountingRules {?cd
        :mayHaveCreditEntry ?cc}}
}
```

Table 1: Accounting transactions related to a purchase of inventory items.

Transaction ID	Debit Account	Credit Account	Transaction Amount	Transaction Date	Other relevant attrs (e.g. doc links)
1	303 01 00	401 01 00	750	03-10-2021	...
2	442 60 00	401 01 00	142.5	03-10-2021	...
3	303 02 00	303 01 00	750	03-17-2021	...
4	401 00 00	770 00 00	892.5	04-10-2021	...
5	603 00 00	302 00 00	750	11-10-2021	...

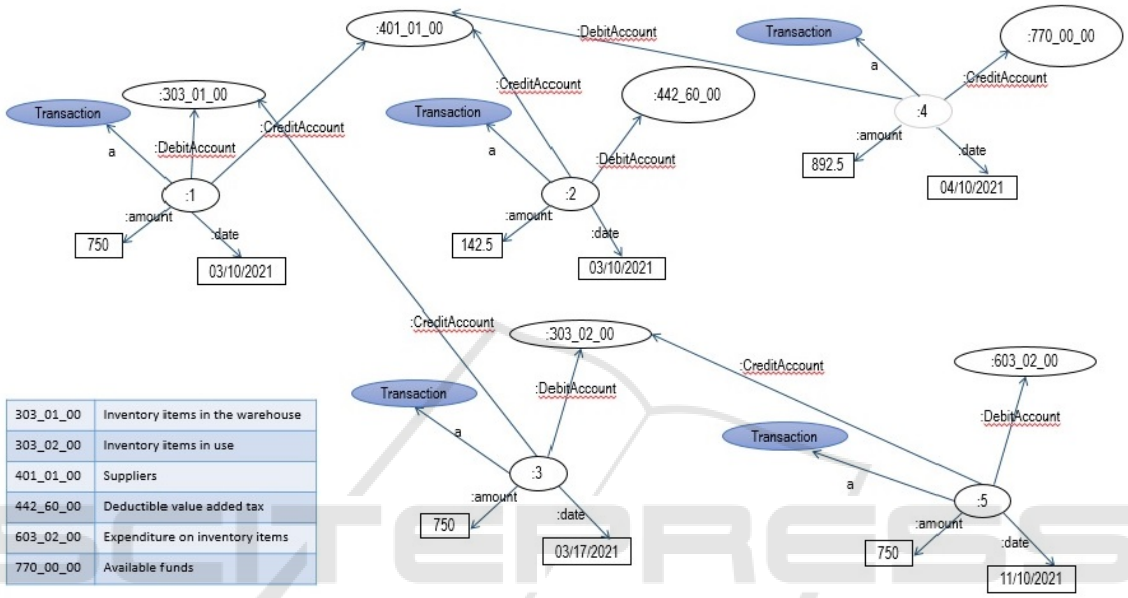

Figure 2: Knowledge Graph fragment lifted from legacy transaction data.

We go beyond error-checking to employ the RDF approach for richer, relationship-focused analysis. A network of accounts is generated by the following rule based on two accounts participating in the same double-entry transaction:

```
INSERT { << ?x :hasCreditEntry ?y >>
:totalMonetaryAmount ?z }
WHERE
{
   SELECT ?x ?y (SUM(DISTINCT ?amount)
   AS ?z)
   WHERE {?transaction :debitAccount ?x;
   :creditAccount ?y; :amount ?amount}
   GROUP BY ?x ?y
}
```

In addition each co-participation is marked using the RDF-star extension that allows RDF statements to have their own properties - i.e., properties of property instances aimed to close the functional gap between RDF graphs and labelled property graphs (Hartig, 2021). The rule example above computes the total amount involved by all interactions between the connected accounts. Similarly other valuable aggregations may be obtained as suggested in Figure 4 (e.g. lists of partner organizations, lists of links to primary documents).

In order to query the graph about all the accounts that have influenced directly or indirectly on account 401 01 00 we can use the following SPARQL query:

```
SELECT ?x
WHERE
{:401_01_00 (^:hasCreditEntry|
:hasCreditEntry)+ ?x}
```

In contrast to a relational database, a query about all the accounts that 401 00 00 was involved with in the same operation, needs to perform n inner join operations on the accounting transactions table. The challenge of performing such a query has to do among others with the indefinite value of n (depending on the account that we query on), which makes it an ideal case for the navigation of the accounts' network. Since the network is not separated, but kept together with the legacy system from which it was derived, navigation through the network may also collect relevant data attached to every operational pair of accounts - amounts, document links etc. Detection of

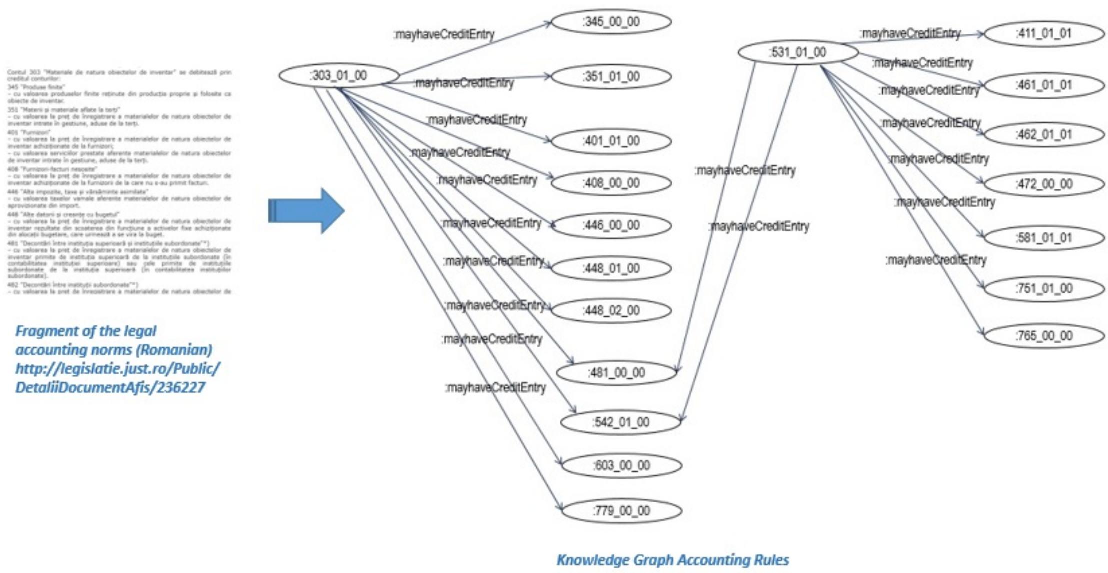

Figure 3: Transformation of legal accounting norms to Knowledge Graph.

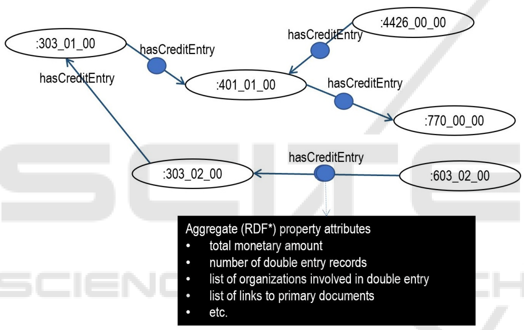

Figure 4: Networked View on the Account Interactions.

arbitrary length paths has been recently made available as a plug-in service on GraphDB (Ontotext, a) - due to weak default support for path detection in the SPARQL standard, forcing our early stage attempts to involve an overhead of network analysis libraries. Multihop directed chains of relationships can now be detected and highlighted - e.g. the shortest path connecting two recorded transactions through a chain of accounts:

```
SELECT ?start ?property ?end ?index
WHERE{
SERVICE
<http://www.ontotext.com/path#search>
 <urn:path> {
path:findPath path:shortestPath;
path:sourceNode ?pathSource;
path:destinationNode ?pathDestination;
path:startNode ?start;
path:propertyBinding ?property;
path:endNode ?end;
path:resultBindingIndex ?index. }
:TransactionX:DebitAccount?pathSource.
:TransactionY:CreditAccount
?pathDestination. }
```

Traditional tabular analytics are not excluded - the following builds a time series of all transactions using some reference account as debit account:

```
SELECT (SUM(?amount) AS ?dailyAmount) ?date
WHERE
{?transaction :debitAccount :603_02_00;
:amount ?amount; :date ?date}
GROUP BY ?date
ORDER BY ?date
```

These are some relevant patterns prescribed for the analysis workbench which makes it possible to gain a networking view on accounts interaction without making use of any network analysis tools. A rudimentary interface for graph browsing/path highlighting, CONSTRUCT-ing subgraphs and running SPARQL* queries was build on top to showcase key operations (see Fig. 5).

4 TENTATIVE EVALUATION

Design Science artifacts can be evaluated according to a large variety of criteria which have been organized in a taxonomy by (Prat-N., 2014). Limited by the current Technological Readiness Level of the proposed solution, we are currently focusing on:
Consistency with Organization: the proposed artifact satisfies the set of competency questions derived from interviews with operators (accountants) of the legacy ERP systems. The competency questions are

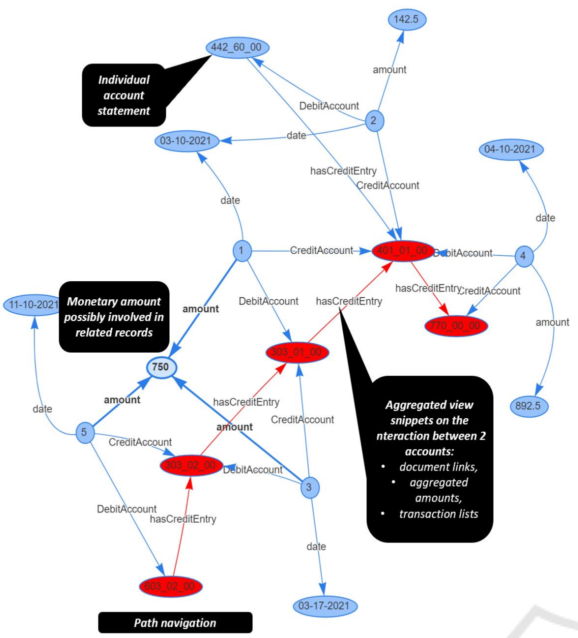

Figure 5: Account Graph Navigator.

answered by the queries and reasoning patterns detailed in the previous section. *Consistency with technology*: a Knowledge Graph can be populated with legacy data in several ways - either an ETL strategy lifts legacy data to a new form of data warehouse; or, a virtual graph layer is deployed to unify access to existing data - the hereby reported project employs for now the first approach with the help of GraphDB's OntoRefine RDFizer (Ontotext, c). The legacy ERP system is however very limited in terms of interoperability channels and a future phase of the research aims to build demonstrators of accounting operations that can advocate replacement of the legacy systems; obstacles in this respect are not only technological, also having to do with extensive reskilling and change management.

A second tier of evaluation priorities is postponed for future work - i.e, Consistency with people: no data is currently available on technology acceptance, as the proposed artifact still requires some technical understanding of graph technology and query-level interaction that should be obfuscated entirely behind a graph-aware user experience. However, current iterations focused on feasibility as a prerequisite for acceptance. Digitally savvy human capital is needed in any Digital Transformation project - however KG savviness is a major challenge due to lack of educational content - less than ten percent of the project team had prior awareness about KG and a major effort of upskilling developers was needed; this effort should not be further pushed to end-users, for which a graph-driven user experience must be designed - a distinct challenge that will be tackled by follow-up projects.

We follow the practice of Design Science research of trying to inform design-based theorizing, by proposing the following key characteristic for a KG-based Digital Transformation: *it is a form of information system transformation supported by digital means that specifically provide - both to relationship-rich data governance and to relationship-aware digital assets (services, agents etc.) - an ability to navigate data context, as it was defined at the beginning of Section 3.*

5 CONCLUSIONS AND FUTURE WORK

Knowledge representation and reasoning are inspired by human problem solving, and can empower intelligent systems to mimic deterministic pattern seeking. Considering the current Technological Readiness Level of the proposal, we summarize a SWOT evaluation to inform future iterations of the proposal, as the project is open-ended and will evolve towards further data integrability with heterogeneous sources that can enrich the discussed patterns:

Strengths: We defined a method of mapping existing accounting transactions and accountant's knowledge to semantic query patterns based on RDF graphs and the RDF-star extension. They exploit a networked view of accounts interactions based on their co-occurrence in the same ledger double-entry or check them against regulatory constraints that are domain-specific and only available in textual regulations.

Weaknesses: The current proof-of-concept is on a low Technological Readiness Level. An ontology is still to be designed to further enrich the graph - we're currently considering FIBO (FIBO, 2021) although it is currently considered overkill relative to the project objectives. For the work hereby reported we are currently not inclined towards developing a novel ontology as data was lifted from legacy SQL-based silos and inherits concepts and properties from the source schemas. The focus of the reported project iteration was on fulfilment of certain key use cases under prototypical feasibility conditions - to theorize on the Digital Transformation characteristics that were distilled in Section 5.

Opportunities: Complex fraudulent patterns may be devised and prescribed as graph-driven features. The accounting ecosystem is fundamentally link-oriented, something already demonstrated by the XBRL standard (XBRLInternational,) which heavily relies on XLink - we conclude that XBRL was the XML-era compromise for a tentative accounting-specific knowledge representationn. Financial authorities being preoccupied with cross-checking business activities that are fundamentally federated across partner businesses, Knowledge Graphs may be a key enabler for cross-organization auditing if such technology is adopted at large.

Threats: The uptake of Knowledge Graphs is still limited and the data analytics methods practiced by auditors are still fundamentally table-biased. Promoting a mindset where semantic networks (in terms of the accounting domain) meets the obstacle that most accountants have no awareness of the existence of non-tabular data models, even if they are closer to how accounting principles are applied by a human operator.

As a future work we propose to extend our work in order to identify possible benefits with respect to fraud detection.

ACKNOWLEDGEMENTS

The current work was supported by the project POC/398/1/1/124155 - co-financed by the European Regional Development Fund (ERDF) through the Competitiveness Operational Programme for Romania 2014-2020.

REFERENCES

Bizer C., Health T., B.-L. T. (2009). Linked data – the story so far. In *International Journal on Semantic Web and Information*.

Cagle, K. (Accessed June 20th, 2021). The role of context in data. In *Available at https://www.linkedin.com/pulse/role-context-data-kurt-cagle/*.

DBpedia (Accessed April 1st, 2021). In *https://dbpedia.org*.

DCManifesto (Accessed April 1st, 2021). In *The Data-centric Manifesto http://www.datacentricmanifesto.org*.

FIBO (Accessed April 1st, 2021). Enterprise data management council. fibo the financial industry business ontology. In *https://spec.edmcouncil.org/fibo/*.

Gartner (Accessed April 1st, 2021a). Gartner: 2 megatrends dominate the gartner hype cycle for artificial intelligence. In *https://www.gartner.com/smarterwithgartner/2-megatrends-dominate-the-gartner-hype-cycle-for-artificial-intelligence-2020/*.

Gartner (Accessed April 1st, 2021b). Gartner: Top 10 data and analytics trends for 2021. In *https://www.gartner.com/smarterwithgartner/gartner-top-10-data-and-analytics-trends-for-2021/*.

Hartig, O. (Accessed April 1st, 2021). Position statement: Rdf* and sparql*. In *https://www.w3.org/Data/events/data-ws-2019/assets/position/Olaf/Hartig.pdf*.

Ontotext. Graph path search service on graphdb. In *https://graphdb.ontotext.com/documentation/9.9/enterprise/graph-path-search.html*.

Ontotext. Graphdb. In *https://graphdb.ontotext.com/*.

Ontotext. Ontorefine. In *https://graphdb.ontotext.com/documentation/free/loading-data-using-ontorefine.html*.

Prat-N., Comyn-Wattiau I., A. J. (2014). Artifact evaluation in information systems design science research - a holistic view. In *PACIS 2014 Proceedings*.

Renu G., B. M. (2013). Errors and frauds in financial transactions: Auditors opinion. In *The Global eLearning Journal*.

Wieringa, R. (2014). Design science methodology for information systems and software engineering. Springer.

XBRLInternational. The xbrl standard. In *https://www.xbrl.org/*.