Essential issues to consider for a manufacturing data query system based on graph

Lise KIM[1]*, Esma YAHIA[1], Frédéric SEGONDS[2], Philippe VERON[1] and Victor FAU[2]

[1] Arts et Metiers Institute of Technology, LISPEN, HESAM Université, Aix-en-Provence, France

[2] Arts et Metiers Institue of Technology, LCPI, HESAM Université, Paris, France

[3] CapGemini DEMS, Toulouse, France

* lise.kim@ensam.eu

Abstract

Manufacturing industry data are distributed, heterogeneous and numerous, resulting in different challenges including the fast, exhaustive and relevant querying of data. In order to provide an innovative answer to this challenge, the authors consider an information retrieval system based on a graph database. In this paper, the authors focus on determining the essential functions to consider in this context. The authors define a three-step methodology using root causes analysis and resolution. This methodology is then applied to a data set and queries representative of an industrial use case. As a result, the authors list four major issues to consider and discuss their potential resolutions. **Keywords:** Manufacturing data – Information Retrieval – graph database – Query system

1 Introduction

The volume of data generated by the manufacturing industry is large and increasing; it represents 3.6 EB in 2018 and will increase by 30% for 2025 **[1]**. The organization of companies in silos (justified by the need for specialization of the different business) generates data that is both distributed and heterogeneous. A part of data is managed by different information system (PDM, ERP, MES…) and generate structured data, while the other data are unstructured data (text, image, 3D …). In addition, the data can be explicitly linked to each other (like in the parent-child relations of a digital mock-up) or implicitly linked (like between the 3D of a component and its user manual).

To perform their work, employees have to query the data in order to retrieve the needed information. This task becomes complicated and time consuming due to

the increasing volume of data, which are heterogeneous and are saved in distributed resources. To solve these issues, it is necessary to define a data querying system that deliver exhaustive and relevant data as fast as possible.

To address this challenge, the authors worked to draw up the list of bare minimum issues to consider when defining the optimal framework. This paper is organized as follows: section 2 defines the main orientations chosen based on a state-of-the-art analysis. Section 3 describes the methodology used to draw up the list of issues. Section 4 describes the experimental conditions and presents the results. Section 5 concludes with discussion.

2 Graph database consideration

Querying information can be achieved through Information Retrieval Systems that need to access to data in order to provide the most relevant one. This objective is reached by managing the data in NoSQL databases rather than traditional relational databases, as the former is faster, more efficient and flexible **[2]**. The main categories of NoSQL like column database, key-value store and document-oriented database includes indexing and quick access to the information but lack expressing of the relationships between data in their schema. The graph databases answer to this issue and consequently are the more suitable in our context.

To emphasize the benefit of the graph database, different researches have shown the importance of analyzing data with strong relational nature as in [**3**], applied in different manufacturing use cases as in **[4]**. Other works define a framework to allow data querying by transforming structured data **[5]** and unstructured data **[6]** into a graph, with enrichment by data linkage **[6]** with possible using ontologies for example in **[7]**. In this article, the authors aim to define the prerequisites for a manufacturing query-data system by answering this question : “What are the minimum issues to be taken into account for a querying manufacturing data system based on the graph database?”

3 Methodology

In order to define only the bare minimum issues to consider when defining the query system, an iterative method has been implemented. This method is detailed below:

(1) **Integration of data** into a graph database. The data includes the minimum of information at initialisation (only metadata without text content). Metadata

means all the properties carried by unstructured data and all metadata carried by structured data. Thus, each data is transformed into a node and each metadata integrates the properties of this node. On the other hand, the explicit relationships of relational databases are translated into relations between nodes (see **Figure 1.**).

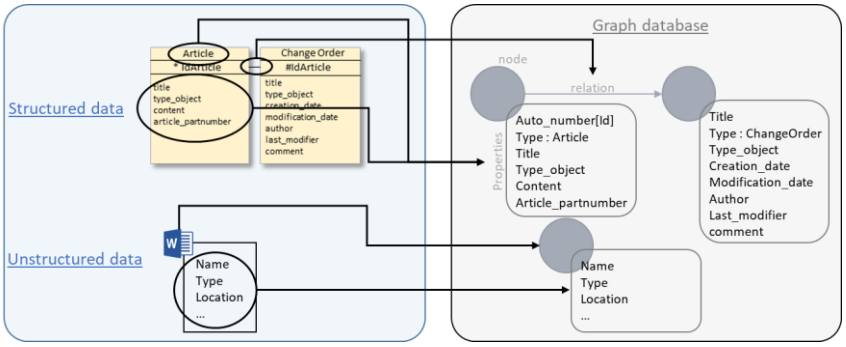

Figure 1 - Data set transformation into the graph at initialization

(2) **Application of queries** to the graph database, here refers to the translation of the user query adapted to the graph. Query transformation includes, in particular, the path of relations between data (e.g. : query = *employees related to 'additive manufacturing'* become *finds the nodes mentioning 'additive manufacturing' and linked to nodes carrying employee information and return the employee information*) and the search for either a list of data (e.g.: query = *battery*) or a specific element within a data (e.g.: query = *price of battery*). The notion of an element is translated by the search for the associated property (value of the property 'price') and the sentences identifying the element ("the price of battery is […]"). Natural Language Processing (NLP)**[8]** tools will be used here to find the sentences.

(3) **The evaluation** of the proposal is conducted based on three requirements that are the response time (between the submission of the query and the result display), the completeness and the relevance of result (using precision[1] and recall[2] measures). The latter are calculated based on the expected results that are manually defined. When the results are below the accepted limits, the analysis of each error is then made (excess or missing data and too long execution data) in order to detect the root causes based on the Ishikawa diagram method[3]**[9]**). A score is then established by root cause according to its impact on the results (calculated with the number of errors associated with this root

[1] The precision is the number of relevant documents found compared to the total number of documents proposed in the result for a given request.

[2] The recall is defined by the number of relevant documents found with regard to the number of relevant documents in the database.

[3] Method of analysis used to research and to represent the different possible causes of a problem.

cause over the total number of errors). Once, this list of root causes is identified, it allows to define the main issues to be treated.

4 Experimentation conditions and results

The study was based on a dataset composed of 686 elements, representing data from a drone manufacturing company, and distributed as following: 47% unstructured data including spreadsheets, videos, photos and textual documents; 22% tree structure data; 17% of data from relational databases and 15% of geometrical data. All these elements represent the data necessary for the development of a mechanical system (from design to prototyping through logistics, purchasing and project management). 19 queries have been written in response to innovative use cases characterized by Capgemini[4] (e.g.: a designer is looking to identify the product requirements or the justification for a product, a manager looking for identify an available team with the right skills, a salesperson looking for a customer's usage parameters, etc.). The expected performance thresholds are less than 1 second for time; this was fixed according to study conclusions on the impact of response latency in web search **[10]**. The tools used are *Neo4J*[5] for storage and querying in a graph database and *Standford CoreNLP*[6] for the exploitation of natural language. These tools are open source and relatively well documented.

After the first cycle of the methodology, the results are insufficient (see **Table1**). Analysis of the results of this first cycle has shown that more than half of the anomalies are caused by the lack of textual content of the data in the graph database. For example, the search for the battery reference does not give any result because the information is carried by the content of an excel named "Bill of Materials". In order to treat this issue, a second cycle was therefore launched to integrate the text content of unstructured data. The text content is extracted using *Apache Tika*[7] as a parsing tool (to extract text from a document) and *Tesseract*[8] as an Optical Character Recognition Tool (to extract text from an image). Then the extracted text is integrated into the graph by adding a property named 'content' to each node.

The results of this second cycle are visible in **Table 1** and the list of root causes is listed in **Table 2**. It is possible to remove the cause (6), only cause of which at least one element of the initial architecture is supposed to be resolved (an optimization of OCR is necessary). The remaining root causes therefore provide the list of the bare minimum issues to be resolved.

---

4 Company of digital services in the manufacturing industry.

5 https://neo4j.com

6 https://standfordnlp.github.io/CoreNLP/

7 https://tika.apache.org/

8 https://opensource.google/projects/tesseract

Table 1 - First and second cycles results

	Cycle 1	*Cycle 2*
Response time (ms)	5790	16978
Precision	0.50	0.44
Recall	0.01	0.31

Table 2 - Second cycle root causes

Root causes	*Anomalies distribution*
(1) No highlighting performed on the most relevant result	26%
(2) The information carried by the table format is not used	24%
(3) Information is carried by a term close to the keyword	19%
(4) The property carrying the information is close to the keyword	13%
(5) Searching through the implicit relationships is impossible	9%
(6) The OCR algorithm didn't extract the correct characters	7%
(7) Extracting bulleted lists extraction is not correct	2%

5 Discussion

At the end of the second experimental cycle, 7 root causes remain present. The authors propose to classify them into 4 large families. Each one has a potential action plan in order to enhance the response time, recall and precision. The authors propose to prioritise, at first, the actions affecting both precision and recall:

(a) **Extracting text without format is not enough.** The cause (1) indicates that it is necessary to translate the information carried by the table format (rows and column) in order to use it in query. For example, to detect a reference contained in a specific cell of a bill of materials. Cause (7) indicates that bulleted lists processing is necessary for the performance of the chosen NLP tools. The table format and bulleted lists must be transformed to be used.

(b) **Searching for exact keyword or exact property is not enough.** Causes (2) and (4) indicate that reconciliation between different terms is necessary. For example, if the term ‘reference’ is used in the query, the term ‘Part Number’ must also to be searched. The use of a semantic network as an ontology could resolve part of the errors **[11]**.

(c) **There is no order by relevance in the results**. Cause (3) indicates that unexpected results (but potentially relevant) are displayed in the same way as expected results. For example, searching for the battery reference provides many results with the terms "reference" and "battery" in the content, but these results are far from the information being searched for. pre-labelling of data or additional filtering can be a solution.

(d) **The implicit links between data must be exploitable.** In some cases, related elements such as an element's functional reference and its supplier reference are disjoined in the different enterprise systems. In order to resolve the cause (5), the integration of the implicit relationships between data must be integrated into a graph.

The defined actions have a direct impact on increasing the precision and recall requirement and probably have a negative impact on the response time.

In conclusion, the above action list allows considering the essential functions for a query system construction, based on data graph and adapted to manufacturing data. This list was obtained according to the methodology described in section 3, with a heterogeneous, distributed and relational data set and by applying queries in response to expected uses in the manufacturing industry.

References

1. Reinsel D., Gantz J. and Rydning J. The Digitization of the World From Edge to Core. IDC White Paper, 2018.
2. Nayak A., Poriya A. and Poojary D. Type of NOSQL Databases and its Comparison with Relational Databases. International Journal of Applied Information Systems (UAIS), 2013, 5(14), 16-19.
3. Miller J. Graph database applications and concepts with Neo4j. Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, March 2013.
4. Schalbus S. and Scholz J. Spatially-linked manufacturing data to support data analysis. Journal for Geographic Information Science, 2017, 1(15), 126-140.
5. Paradies M., Lehner W. and Bornhövd C. GRAPHITE: An extensible graph traversal Framework for relational database management systems. In Proceedings of the 27th International Conference on Scientific and Statistical Database Management, SSDBM ’15, La Jolla, June 2015, pp. 1-12.
6. Groger C., Schwarz H. and Mitschang B. The Deep Data Warehouse: Link-based Integration and Enrichment of Warehouse Data and Unstructured Content. In Proceedings of the 18th IEEE International Enterprise Distributed Object Computing Conference, EDOC’14, Ulm, September 2014, pp. 210-217.
7. Mordinyi R., Schindler P. and Biffle S. Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments. 20th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA’15, 2015, pp. 1-8.
8. Chowdhury G. G. Natural language processing. Annual Review of Information Science and Technology, 2003, 37(1), 51-89.
9. Barsalou M. A. Root Cause Analysis: A Step-By-Step Guide to Using the Right Tool at the Right Time. 2014 (CRC Press Taylor & Francis Group)
10. Arapakis I., Bai X. and Cambazoglu B. Impact of Response Latency on User Behavior in Web Search. In Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’2014, New York, 2014, pp. 103-112.
11. Yan L., Manoj A. T. and Kweku-Muata O.-B. Ontology-based data mining model management for self-service knowledge discovery. Information Systems Frontier, 19(4), 2017, 925-943.