On Functional Requirements for Keyword-based Query over Heterogeneous Databases on the Web

Plinio S. Leitão-Junior, Fábio Nogueira de Lucena, Mariana Soller Ramada,
Leonardo Andrade Ribeiro and João Carlos da Silva
Instituto de Informática, Universidade Federal de Goiás, Goiânia Goiás Brazil

Keywords: Keyword Query Processing, Heterogeneous Databases, Functional Requirements

Abstract: **Context.** A large amount of data is made available daily on the Web, but many databases cannot be accessed by conventional search engines, as they require proper access methods and specialised knowledge through their access languages.
Focus. In the scenario of non-expert users to access databases, multiple database categories, and plural idioms, this work analyzes the functional requirements that need to be considered for keyword queries processing over data sources on the Web. The problem is still open and involves challenges such as query interpretation and access to databases.
Method. The investigation is centered on the problem itself, which is portrayed by a set of functional issues, which together represent the challenges linked to the research field.
Approach. This work introduces and systematically analyzes the functional requirements to the problem scope. Issues reported in the literature are refined and evolved to support the modeling of the problem views: functional responsibilities and their interactions by messaging between problem objects.
Conclusions and Results. This paper contributes to characterize the problem, makes clearer its understanding and promotes the development of keyword-based query processing systems. A software engineering artifact is used to model the problem and make it more formal and precise. Further studies will refine such requirements and build (specialise) artifacts tailored to the solution space.

1 INTRODUCTION

Over the years, databases (data sources) have become present in many applications and today they play a prominent role in the service of society, such as promoting the development of intelligent approaches through learning on the available data. The value of data and the popularity of databases has brought new demands and users, the latter also interested in using simple and intuitive interfaces to manipulate data.

Query languages usually comply with strict syntactic rules, which makes the validity of query statements dependent on the ability to express an information need according to such rules. Furthermore, there is also a strong dependency on the database schema, since database entities are explicitly mentioned in the queries. As a consequence, the user needs to know about the meaning of each term in the schema (i.e. entity role).

Regarding interfaces to relational databases, which are widely used in academia and industry, users face four challenges for writing queries, namely: (i) the syntax of relational database languages; (ii) the exact schema of the databases; (iii) the roles of various entities in the query; and (iv) the precise join paths to be followed. In the latter, each join path between database relations denotes a specific meaning, so a query addresses those that align with the user's intent.

Keyword search is arguably the most popular data access method for ordinary users because they do not need to know either a query language or the schema of the data. In fact, the simplicity of this method was crucial for the popularization of Web search engines. A *keyword query* is a simple sequence of words to represent the user's demand for information over one or more data sources. Thus, as database languages such as SQL and SPARQL are usually devoted to technically-skilled users, an obvious benefit of keyword queries is to allow non-expert users to access data in databases.

For data access, structured or unstructured data

S. Leitão-Junior, P., Nogueira de Lucena, F., Ramada, M., Ribeiro, L. and Carlos da Silva, J.
On Functional Requirements for Keyword-based Query over Heterogeneous Databases on the Web.
DOI: 10.5220/0010453802240231
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 224-231
ISBN: 978-989-758-509-8

sources have their own access methods, which conform to the database category and depend on their logical data structures. For example, a document-based store considers that a record (document) has a document ID and a list of attribute-value pairs (possibly recursive), so data access requires appropriate methods to deal with this data organization. In this context, query processing involves mapping the user's keyword query to access methods specific to each data source.

An important challenge faced by keyword search approaches is that many candidate answers could be retrieved in response to a user query. Each candidate answer represents a different interpretation of the query. However, users generally expect an exact answer that perfectly meets their needs, instead of wasting time choosing one of many possible answers. Then, approaches that support keyword-based queries need to deal with ambiguity, being able to reduce the number of candidate answers by removing the ones that have a very low chance of meeting the user's intention and ranking the rest of them. Despite different ways to deal with ambiguity have been proposed (Hristidis and Papakonstantinou, 2002; Bergamaschi et al., 2011; Kargar et al., 2015; Hormozi, 2019), the problem of query disambiguation is still a big challenge (Hormozi, 2019).

Besides that, after finding a reduced set of candidate answers, another challenge is to decide which extra information that exists in the data source and is semantically related to the query could help show the user more interesting data.

While previous work has proposed solutions to many aspects of the problem, a systematic study on the fundamental issues around building a keyword-based system to access heterogeneous databases is surprisingly missing. Keyword query processing addresses the mapping of the user's demand to valid and appropriate access methods to the category of the relevant databases, which are those chosen as data sources for the query. In this context, the main question for investigation is posed as follows:

Question. Which functional requirements need to be considered for keyword queries over data sources on the Web with heterogeneous idioms and database categories?

Description. The question addresses the functional responsibilities necessary to meet the user's demand, which together portray the problem related to the keyword query processing to be solved. In order to be appropriately robust, such responsibilities should involve the main issues raised in literature that refer to the functional anatomy that outlines the keyword query processing problem.

Rationale. The focus embraces non-expert users to access databases as that is a common facility over most keyword-based approaches. But the question also covers aspects that are jointly not present in literature approaches such as selection of databases pertinent to the user's demand, multiple database categories, and plural idioms. The first involves data source access availability regards its domain and content. Multiple database categories and plural idioms potentially increase the number of data sources and foment more possibilities for better answering the user query. Overall the scope makes the problem more complex related to most of the ones reported in literature but closer to real perspectives upon databases on the Web.

After this introduction, the paper is organized as follows. Section 2 introduces the paper focus as well as analyzes and describes the major functional issues of the keyword query problem. Section 3 deals with functional objects and their interactions. Related work and threats to validity are presented in Sections 4 and 5, respectively. Section 6 concludes the work and shows further developments.

2 KEYWORD QUERY PROBLEM: FUNCTIONAL ISSUES

Problem analysis is the first and primary stage in software development. It impacts on the solution engineering, because if the problem is not properly defined, possibly inappropriate software will be build for the problem. That is potentially the reason for the high rate of failures in software projects.

Thus before considering implementing a keyword-based solution, the problem itself needs to be clearly characterized and defined, which is the aim of the present work.

2.1 Focus on Problem Domain

As the investigation focus copes with the problem space rather than the solution, the definition below formalizes the paper scope:

Definition 1 (Problem Analysis of Keyword Query Processing)**.** *Problem analysis refers to the process of understanding and defining the problem to be solved – Keyword Query Processing problem – i.e. it is the software engineering stage where much of the learning on the problem needs to occur.*

The present work focuses on understanding the problem by building models and introducing definitions

from the problem statement, specifically those related to functional aspects.

In the next subsection, functional issues specific to the problem domain are introduced.

2.2 Functional Issues

Bergamaschi et al. (2016) propose an architecture for keyword search in relational databases to favor the development of scalable and effective components; the authors advocate that the proposal is general enough to be used also with other sources. The architecture is organized into layers focusing primarily on functional requirements. In the context of that paper, the notion of functional requirements conforms to (IEEE-Computer-Society et al., 2014): describe the functions that the software has to execute. This is the most recent proposal for a functional framework and represents a relevant reference for the development of new researches and the implementation of keyword query processing systems.

The research in (Bergamaschi et al., 2016) points out some issues that hinder the design and development of systems for keyword search over structured data. These issues seek to bring out the main functional points and the challenges posed to achieve them. We refine such issues by adding new ones (e.g. queries on non-structured data sources) and aggregating new perspectives on understanding the keyword query problem (e.g. how to deal with the user's intention and how to express the results from heterogeneous sources), as follows.

Interpreting the User's Intention. A simple sequence of words is often inaccurate to represent the user's demand, that is guided by the chosen words and the sequence thereof. Furthermore, the same keyword query can mean differently for distinct users, or even to the same user but at distinct times. Given these difficulties, several interpretations are eligible at capturing the user's intention. The interpretation space of the user's intention refers to the potential renderings (query interpretations) that query processing can abstract when trying to decipher the actual user's demand:

- The interpretation space is defined by analysing the potential meanings of the keyword query (query semantics). The query idiom impacts at defining the interpretation space with respect to its content and cardinality. A desirable scenario is to have a reduced space — few points in that space that means few query interpretations — and the user demand fits one or even more among all the ones in such a space. Realistic scenarios usually differ from desirable ones, as such scenarios potentially deal with conflicting objectives.

Selecting Relevant Databases. When submitting a query on the Web, the user expects results from available databases (data sources) whose content has affinity with the query semantics. To do this, query processing needs to find databases on the Web and then select those (hopefully, the relevant ones) that are potentially data sources for the query. We refer to private databases that are not directly accessed by traditional search engines, so-called hidden data sources. This issue involves two pertinent aspects: (i) finding databases on the Web; and (ii) having access to proper descriptions upon these databases in order to assess whether they are relevant to user's demand. As an example for both aspects, consider that the database owner makes their legacy databases visible and accessible on the Web and publishes metadata that describe customized views of these databases, respectively.

Providing Support for Multiple Idioms. Regarding idioms for processing queries, several idioms are ideally supported, both for queries and data sources. This also includes multiplicity: when a query gets results from databases whose idiom are different from the query idiom. In this context, the most plural scenario involves three distinct idioms: query idiom, database metadata idiom and data idiom.

Dealing with Heterogeneous Databases. SQL and NoSQL databases can be used as data sources for query processing by keywords — this heterogeneity increases the potential to resolve the query. In that sense, by covering all points in the interpretation space of the user's intention, it may happen that some of query interpretations may not be viable for all relevant data sources, then database-dependent queries are mapped from interpretations to those viable data sources. As query interpretations should be further expressed in database queries, there are two challenges ahead: (i) analyzing which data sources relevant to the query are feasible with respect to each query interpretation; and (ii) mapping the query interpretations to the proper access methods of the feasible data sources.

Executing Database Queries. Query processing should execute all database-dependent queries over the relevant data sources on the Web. This involves aspects such as multiple database connections, data sensitivity as well as stream scale of resulting data.

Expressing Query Results. A query result is the response obtained from a data source for an interpretation. A user query can have multiple interpretations, and an interpretation can have multiple responses, each coming from a particular data source. The results must be expressed in such a way that the user can understand and decide how useful a result is

to her/his informational demand.

Ranking the Query Results. Before being returned to the user, query results are ideally evaluated against the real user's intention, aiming to order them according to their relevance. So, this issue refers to the estimation of how much each query result may fit the actual information needs of the user. In this sense, a ranking model is applied to estimate how valuable a result is to the user. This model can evolve by learning about a particular user over the time. In the best scenario, the model promotes that the user has the most relevant results highlighted in the presentation interface. Nevertheless, such a model is an open challenge in today's systems.

Visualising Query Results. This issue is referred to the presentation of the query results to the user. It is related to how the results are displayed so that the user can better understand them and judge their value with respect to the expected demand.

Overall the functional issues explained above outline the major challenges that define the problem scope and promote proper understanding for further design of keyword query systems.

3 FUNCTIONAL OBJECTS AND THEIR INTERACTIONS

Functional responsibilities are fulfilled by objects identified from the problem, which interact by exchanging messages to carry out their functional duties: the characterization of the keyword query problem is materialized by objects whose interactions aid to make clearer the functional requirements of the problem.

Table 1 presents the objects that abstract the main functional responsibilities: Column 1 identifies the problem objects; and Column 2 describes the functional responsibility of each object.

Figure 1 presents a Sequence Diagram of the UML (Unified Modeling Language) — Sequence Diagram, for short, which represents the interaction through messages passed between objects, in the perspective of the problem itself. Two aspects that base the instrument in the figure are:

- The Objects that represent functional responsibilities (rectangles at the figure top) and interact through the exchange of messages are actually objects of the problem, rather than objects of a solution such as a particular keyword query system.
- Interactions between objects promote understanding to the problem, not necessarily establishing a strict order of message exchanges. That means, they depict consistent and robust interactions to represent the problem, but other alternatives to message sequences are also permissible.

The following are focused on how the functional issues addressed in Section 2 are carried out in the sequence diagram.

3.1 Interpreting the User's Intention

Interpreting the user's intention includes defining the interpretation space — a set of interpretations for the meaning of the keyword query. Definition 2 formally determines what an interpretation is.

Definition 2 (Query Interpretation)**.** *A query interpretation of the user's demand is a set of term sequences, such that they have the same meaning to each other. The simple directed graph G_R represents the Interpretation R — G_R is a directed graph with no loops and no multiple arrows with same source and target nodes — and it is defined as $G_R = (N, E, s, f)$: N is the set of nodes and each node $n_i \in N$ represents a term or a fork; E is the set of edges and each edge $e_i \in E$ has a source and a target nodes; s and f are the input and the output nodes of G_R, respectively; and each path from s to f denotes a sequence of terms.*

For instance, given the keyword query 'which 2020 award-winning films', the interpretation shown in Figure 2 is a set of four sequences: "2020 awarded films", "2020 awarded movies", "2020 award-winning films" and "2020 award-winning movies".

Definition 3 (Query Interpretation Space)**.** *The query interpretation space (or interpretation space, for short) is the set of interpretations related to the keyword query. The cardinality of such a set impacts the number of query results.*

The *Query Interpreter* object typically applies Information Retrieval (IR) and Natural Language Processing (NLP) techniques during keyword query analysis to abstract the interpretation space of the user's demand. Each interpretation in this space has its term sequences enriched from the support of Domain and Idiom Services: *dealWithRestrictions* and *getDomainExtension* messages to *IdiomService* and *DomainService* objects, respectively.

3.2 Selecting Relevant Databases

Relevant databases are selected by the *Data Source Selector* object, which uses the interpretation space defined for the user's query (*selectDataSource* message sent by the *Controller* object). There is interaction with the *Data Source Catalog Provider* object (*getDataSourceCatalog* message sent by the *Data*

Table 1: Functional responsibilities of the keyword query problem.

Problem Object	Functional responsibility
Query Interpreter	Defines the interpretation space of the user's query. Each point in that space refers to a possible interpretation in relation to the user's intention that is represented by the keyword query.
Domain Service	Adds value to the interpretation process by providing knowledge about potential domains linked to user demand, for example, by using ontologies.
Idiom Service	Handles support for idiom-related extensions, such as providing synonyms and terms with related meaning, as well as translation between idioms.
Data Source Catalog Provider	Serves the description of the data sources visible and accessible on the Web, that is, provides the information on databases necessary to select the relevant ones to the user's query.
Data Source Selector	Selects the databases that can serve as a source of data for user's demand from those available on the Web.
Database Query Generator	Generates, for each point in the interpretation space, the access methods (database-dependent queries) for each data source selected as relevant to the user's demand.
Data Source Proxy	Represents the service to access data sources, i.e. database accesses in the solution scope. For instance, to obtain information upon data sources and to perform queries over the relevant databases.
Database Query Runner	Handles the execution of each database query, which refers to an interpretation applied to a relevant data source.
Ranking Model	Refers to the ordering of the results to the user's query, aiming to rank them when presenting them to the user.
Search Controller	Deals with the interaction with the user and the control of the sequence of messages to the other responsibilities.

Source Selector object), aiming to get metadata from data sources: refers to hidden databases on the Web (in relation to the documents accessible by traditional browsers), such as legacy databases but with views available for access according to their published metadata. Regarding relevant data sources, two definitions are introduced as follows:

Definition 4 (Relevant Data Source)**.** *A data source relevant to a query interpretation is characterized if the content of that source covers data from which a potential response to that interpretation can be obtained.*

Definition 5 (Set of Relevant Data Sources)**.** *The set of relevant data sources are those that are relevant to at least one interpretation within interpretation space for the user's demand.*

3.3 Supporting Multiple Idioms

The idiom of the user's query is the one used in the abstraction process of its interpretations. So the interpretation space idiom is the same as the keyword query. However, in order to reach data sources in the context of plural idioms, it is pertinent to know and understand the idiom of each data source in order to decide on those that are relevant.

Thus, the interaction with the *Idiom Service* object (*getIdiomSupport* message sent by the *Data Source Selector* object) is justified because the catalog of relevant data sources and space for interpretations may have different idioms. That is necessary to allow idiom-normalised analysis for searching relevant data sources.

3.4 Dealing with Heterogeneous Databases

To handle heterogeneous data sources such as SQL and NoSQL databases, queries dependent on the database category are applied to the relevant data sources, as defined below:

Definition 6 (Database-dependent Query)**.** *A database-dependent query (or database query, for short) is the representation of a query interpretation*

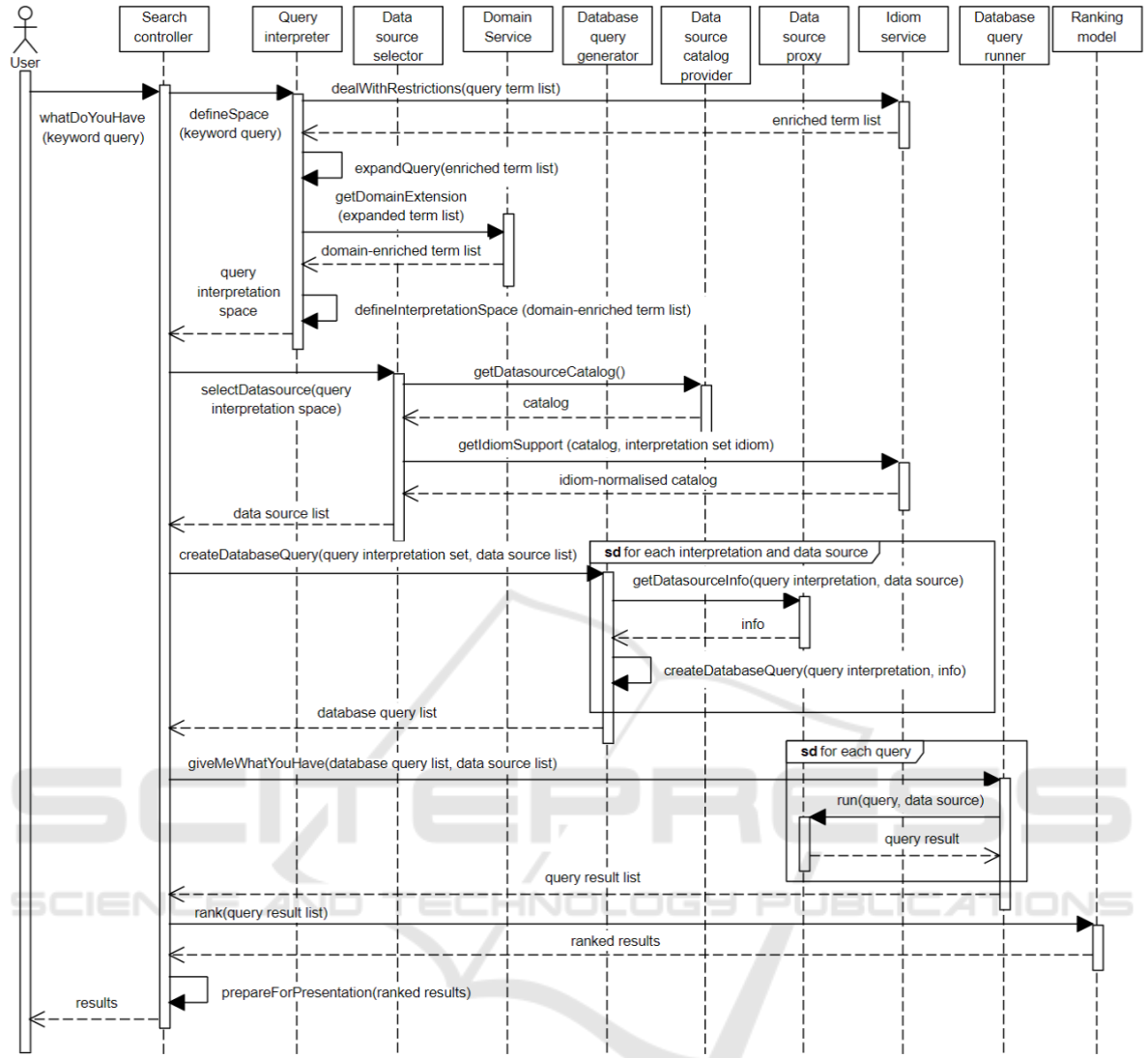

Figure 1: UML sequence diagram for the keyword query problem.

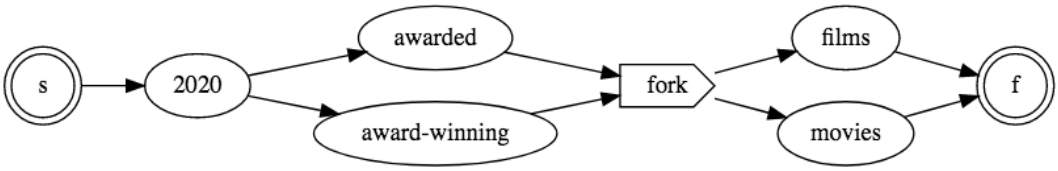

Figure 2: An example of interpretation.

when applied to a particular data source. It considers the data source category (e.g. column family, relational tables, key-value stores, among others) and its access methods as well as the entities that describe the data source.

The *Database Query Generator* object receives the *createDatabaseQuery* message (sent by the *Controller* object) with two parameters: query interpretation space and relevant data sources. For each interpretation and data source, there is interaction with the *Data Source Proxy* object through the *getDataSourceDescription* message to obtain information on the data source entities. Then the *Database Query Generator* object analyzes the feasibility of that query interpretation being applied to the data source. If so, it translates the query interpretation into a proper database query to that data source.

3.5 Executing Database Queries

The *Database Query Runner* object receives from the *Controller* object the message *giveMeWhatYouHave*, whose parameters are 'database query list' and 'data source list'. For each query database and data source, the *Database Query Runner* object interacts with the *Data Source Proxy* object to obtain results from that interpretation of user demand.

3.6 Expressing Query Results

The query results need to be expressed in such a way the user can decide how useful they are.

Definition 7 (Query Result)**.** *A query result is the triple* $\langle \alpha, \beta, \gamma \rangle$ *such that:* α *is a query interpretation;* β *is a data source; and* γ *is data obtained from* β *related to* α*. The former supports user upon data reading and assessing how the interpretation is close to her/his informational demand; the second addresses data origin (e.g. url, database description); the later refers to data themselves.*

The *Database Query Runner* object builds the result triples and sends them to the *Controller* object in the response of *giveMeWhatYouHave* message. Together the triple components make query results more evaluable as well as improve user experience.

3.7 Ranking the Query Results

The *Ranking Model* object receives the query result set from the *Controller* object and applies a ranking model to that set, as defined:

Definition 8 (Ranking Model)**.** *A ranking model is the function* $r(f(\alpha), g(\beta), h(\gamma))$ *that returns how appropriate a result is to the user's demand, such that:* $f(\alpha)$ *measures the adequacy of the interpretation* α *to the query; function* $g(\beta)$ *scales the fitness of data source* β *to the query; and function* $h(\gamma)$ *scores how proper data obtained from* α *in* β *meet user's demand.*

As a consequence the *Controller* object receives scored query results, i.e. a rank of query results.

3.8 Visualising Query Results

The *Controller* object prepares the ranked query results in order to present them to the user. For instance, initially the descriptions of the interpretation and the data source are presented, ordered according to the rank of the results. Then the user 'clicks' on one or more of the results to view the data obtained.

The definitions introduced above seek to converge the understanding of the problem terminology and to more precisely outline functional responsibilities and their interactions.

4 RELATED WORK

Regarding the focus of this research, *Selection and ranking of relevant data sources* with respect to a keyword query were addressed in (Sayyadian et al., 2007; Li et al., 2008; Ramada et al., 2020). Pu and Yu proposed in (Pu and Yu, 2008; Pu and Yu, 2009) support for *keyword query cleaning*, which involves spelling corrections and segmentation of neighbouring keywords so that each segment corresponds to a high quality data term. Various *ranking functions* have been proposed to deal with the inherently ambiguity of keyword searches, providing a way to measure the relevance of each result and order query results (Hristidis and Papakonstantinou, 2002; He et al., 2007; Luo et al., 2008; Hormozi, 2019). Despite the many contributions from the community to addressing issues related to the keyword search problem, to the best of our knowledge, the research field has mainly addressed efforts from the solution perspective. A systematic study of the fundamental issues surrounding the construction of a keyword-based query system, from the problem perspective, has yet to be developed.

5 THREATS TO VALIDITY

Functional requirements are illustrated by using a software engineering artifact, a UML Sequence Diagram. That is a more formal way to introduce the functional responsibilities and interactions thereof related to the keyword query processing problem instead of including figures whose notation is usually not known.

Regarding the generalization of the findings, they are grounded on functional issues, in which some of them were initially introduced by the research field. Such issues represent what the literature 'thinks' at the moment about functional challenges related to the keyword query processing problem.

The contributions were initially based on evidence from papers related to the problem and evolved in a systematic way: the authors were randomly divided into pairs; each pair analysed every issue described in Section 2 to produce a partial artifact per issue; the partials were then evaluated and evolved by at least another pair of authors; several meetings with all authors integrated the partial artifacts to consolidate the final artifact of the complete problem.

6 CONCLUSIONS

The present work analyzed functional issues that are related to the keyword query processing problem. Such issues represent important aspects that make such a problem challenging and complex.

The problem involves submitting keyword-based queries — using an appropriate interface for non-expert users — to access the content of heterogeneous data sources with respect to the database category and the plurality of idioms. Unlike other sources of data accessible on the Internet, such as documents of public content, the problem reaches sensitive databases, which are usually hidden on the Web.

The main contribution of the work is to advance the characterization and formalization of the problem in functional terms, such that it promotes a greater understanding of its functional requirements and a better perception of its complexity. Some specific contributions of this work include:

- evolution of functional issues present in the literature, by bringing more details and perspectives for understanding as well as adding new issues not yet addressed;
- abstraction of problem objects that are actually necessary responsibilities for modeling the functional requirements;
- construction of a software engineering artifact - UML Sequence Diagram - describing the problem domain objects and the interactions between them; such a diagram also eases keyword-based query comprehension;
- introduction of definitions in the problem domain to add more formalism, extend existing terminology and promote new perspectives on the functional issues.

Threats to validity are addressed by: (i) building a software engineering artifact, as it is a systematic representation; (ii) grounding the results on functional issues introduced mainly by the research area; and (iii) producing partial artifacts by random pairs of authors, and integrating them to consolidate the final artifact of the complete problem by all authors.

Further studies will refine the functional requirements and build (specialise) artifacts tailored to the solution domain.

REFERENCES

Bergamaschi, S., Domnori, E., Guerra, F., Lado, R. T., and Velegrakis, Y. (2011). Keyword Search over Relational Databases: A Metadata Approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 565–576. ACM.

Bergamaschi, S., Ferro, N., Guerra, F., and Silvello, G. (2016). *Keyword-Based Search Over Databases: A Roadmap for a Reference Architecture Paired with an Evaluation Framework*, pages 1–20. Springer Berlin Heidelberg.

He, H., Wang, H., Yang, J., and Yu, P. S. (2007). Blinks: Ranked keyword searches on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 305–316. ACM.

Hormozi, N. (2019). Disambiguation and Result Expansion in Keyword Search Over Relational Databases. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 2101–2105. IEEE Computer Society.

Hristidis, V. and Papakonstantinou, Y. (2002). DISCOVER: Keyword Search in Relational Databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 670–681. Morgan Kaufmann.

IEEE-Computer-Society, Bourque, P., and Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge SWEBOK Version 3.0*. IEEE Computer Society Press, Los Alamitos, CA, USA.

Kargar, M., An, A., Cercone, N., Godfrey, P., Szlichta, J., and Yu, X. (2015). Meaningful Keyword Search in Relational Databases with Large and Complex Schema. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 411–422. IEEE Computer Society.

Li, G., Ooi, B. C., Feng, J., Wang, J., and Zhou, L. (2008). Ease: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, page 903–914. ACM.

Luo, Y., Wang, W., and Lin, X. (2008). SPARK: A keyword search engine on relational databases. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 1552–1555. IEEE Computer Society.

Pu, K. Q. and Yu, X. (2008). Keyword query cleaning. *Proceedings of the International Conference on Very Large Data Bases*, page 909–920.

Pu, K. Q. and Yu, X. (2009). FRISK: keyword query cleaning and processing in action. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 1531–1534. IEEE Computer Society.

Ramada, M. S., da Silva, J. C., and de Sá Leitão-Júnior, P. (2020). From keywords to relational database content: A semantic mapping method. *Information Systems*, 88:101460.

Sayyadian, M., LeKhac, H., Doan, A., and Gravano, L. (2007). Efficient keyword search across heterogeneous relational databases. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 346–355. IEEE.