Towards a Framework Embedding Formalisms for Data Warehouse Specification and Design

Isaac N. Mbala [1] and John A. van der Poll [2]
[1] School of Computing, Florida, University of South Africa
[2]Graduate School of Business Leadership (SBL), Midrand, University of South Africa
[1] isaacnkongolo64@gmail.com, [2] vdpolja@unisa.ac.za

ABSTRACT

The Data Warehousing use implies the establishment of Data Warehouse and business intelligence systems. A Data Warehouse is a subject oriented, integrated, time variant and non-volatile data collection which helps decision makers with strategic information. Many researches have proved that most of data warehouse projects miss to meet the business purposes and business requirements because of the carelessness of the significance of entire requirements elicitation and subsequent specification stages. Most of the time, the requirements performed in the course of these definition stages are inadequate or inconsistent, steering to erroneous specifications. In this paper we introduce the challenge of data warehousing projects failure and propose a framework that would facilitate the formalisms for requirements elicitation for data warehouse systems design. Our approach leans towards a hybrid-driven technique for requirements analysis for data warehouse systems. We propose some formalisms for requirements elicitation, followed by formal specification in Z.

KEYWORDS

Data warehouse design, Requirements engineering, Formal methods, Formal specification, Z.

1 INTRODUCTION

Nowadays, data warehousing (DWH) is the technology more used in large industries [1], [2]. The usage of data warehousing involves the creation of a data warehouse (DW) made up of data marts (DMs) or operational databases with business intelligence (BI) embedded in the resultant structure. This paper aims at showing to what extent may formal specifications facilitate data warehouse systems design using a proposed framework based on hybrid-driven technique for requirements analysis. Numerous authors have researched requirements analysis for DW systems, notably [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], and [11] to name but a few, and rather few research works have attempted to address the requirements engineering aspect. [5], [7] and [12] have defined a data warehouse as "a subject oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decisions". A data warehouse has been recognized as one of the most complex information systems and its maintenance and design are described by numerous complexity coefficients [2], [5], [7]. A data warehouse is considered as the cornerstone of business intelligence systems [12], [13].

As observed by [2], the data warehouse purpose has always been to produce concise analysis for assisting decision makers with strategic information and also enhance the corporeal organizational performance. Building a conventional operational system requires to take into account not only requirements upon how to automatically execute the company operations, yet for developing a data warehouse system, the analytical requirements carrying the decision making process require to be taken [11]. The data warehouse development and maintenance is a complex and tedious task which requires solving many different problem types [14]. The design of DW systems is instead different from the design of traditional operational systems that provide data to the warehouse since the objective of data warehouse projects is fundamentally relied on the support of the process of decision making of the enterprise in order to facilitate the analysis processes [2], [5], [8], and [11].

It is well documented that a prominent reason why many DW projects have failed in the past is not only because they attempted to supply strategic information from operational systems while those operational systems were not intended to provide strategic information [5], but also because the requirements analysis phase were often overlooked [1], [2], and [3] during the design process and because of these reasons, [2] and [14] have declared that over 80% of DW projects miss to meet with the users and stakeholders' requirements. The analysis phase of requirements can be executed informally based on simple requirements glossaries instead of formal diagrams but such an informal (or maybe semi-formal) approach may be inappropriate for a requirements-driven framework that requires more organized and comprehensible techniques [3].

Data warehouse projects are similar in several phases to any software development project and claims a definition of different activities which ought to be executed related to demands collection, design and implementation within an operational platform, amongst other activities [1], [14]. Despite the similarity to general software development, the effective development of a DW relies upon the quality of its models (design and specification) [15]. However, the system success under the development may be strongly affected by the discovering process of involved stakeholders demands and sustaining those demands while transforming and documenting them under a form which will be analyzable and communicable. This process is generally known as Requirements Engineering (RE) [11], which is a necessary and vital phase in the software development life cycle (SDLC) [16].

According to [16] and [17], RE is the process which is intended to collect, document, analyze and manage requirements for systems and software product throughout the SDLC. According to [5], RE in the data warehouse arena has acquired increased importance and it has the goal of identifying the information demands of the decision makers. However, researchers are actually attempting to utilize numerous techniques of requirements engineering to analyze the specification of data warehouse systems in order to avoid the risk of failure. Several techniques and methods are used in the requirements engineering activities and in this paper we are more interested on formal methods [16], [18], and [19] for analyzing system behavior, factors of risk and problems related to its implementation [20] during the design of the system.

The use of Formal Methods (FMs) in the construction of reliable software has been controversial for a number of decades. Advocates of such techniques point to the advantages to be gained in constructing provably correct systems, especially in the arena of mission/safety-critical systems, e.g.

nuclear power plants and aviation systems. Critics of FMs object to the steep learning curve involved in mastering the underlying discrete mathematics and formal logic needed for the effective use of the methodology. Yet, the literature suggest that using FMs in the design of data warehouse systems ought to be useful in improving the reliability of such systems and other functionality [19].

Formal methods are mathematical approaches sustained by tools and techniques for the verification of the desired and necessary properties of software or hardware systems. FMs are necessary for the control of the quality parameters such as completeness; correctness; and consistency and verification of requirements of a system [13] and they are based on (often discrete) mathematical notations and logic to clearly and accurately express requirements specification [21].

In the research work published by [16], Formal methods are more likely to be used at the levels of design and verification of the software development. As observed by [13], formal methods are attached with the three techniques which are formal specification, formal checking (discharging proof obligations), and refinements. Formal specifications aim to provide for an unambiguous and coherent complement to natural language descriptions [21], [22] and are rigorously validated and verified conducting to the early detection of specification errors [22]. One of the broadly used formal specification languages amongst many different formal specification languages is the Z, selected in this paper owing to its simplicity and widely-usedness in the formal methods arena [22].

The layout of the paper follows: Following our research questions below, we introduce in Section 2 the fundamental concepts of data warehouse systems design by discussing various design approaches. Section 3 presents a Z specification of a data warehouse star schema and illustrates the utility of discharging a proof obligation arising from the specification. Section 4 discusses related work in this area. In Section 5 we address our methodology and finally, conclusions and directions for future work are presented in Section 6.

1.1 Research questions

In this article we aim to find answers to the following questions:

RQ1: What are the requirements elicitation approaches for data warehouse development?

RQ2: To what extent may formal specification facilitate data warehousing?

RQ3: How may formal proofs increase confidence in a formal specification?

RQ4: How may the two (2) prominent elicitation techniques be combined?

2 DATA WAREHOUSE SYSTEMS DESIGN

Building a DW system is unlike transactional systems with respect to the development; the structures are not only ones to be thought of as in those kind of source systems, but cognizance should also be given about the purposes and strategies of the organization [8]. Data warehouse systems have the purpose of supporting the process of decision making of an enterprise. The development of a DW requires that the analytical requirements supporting the decision making process be captured and such requirements are usually not easy to extract and specify

[11]. A DW is generally defined as the *linking* of a number of operational databases with the aforementioned intelligence (e.g. decision-making) added to the resultant structure. Since a DM is viewed as a subset of a DW, we view a data mart as being one of the operational databases in the DW.

Subsequently, we formalize a DW as follows:

$$\underset{i=1}{\overset{n}{Link}}\ DB_i \text{, where}$$

$$(\forall i)(\forall j)(1 \leq i, j \leq n \bullet\ i \neq j \Rightarrow DB_i \cap DB_j = \emptyset)$$

The above definition assumes that different databases, when correctly normalized do not contain common elements, except of course, for foreign key matches. The following example illustrates the above definition.

<u>Example 1</u>

Consider a data warehouse (DW) consisting of four (4) linked databases (DB_1, …, DB_4). Since we view a DW as a partition of individual databases, we can indicate it diagrammatically as follows:

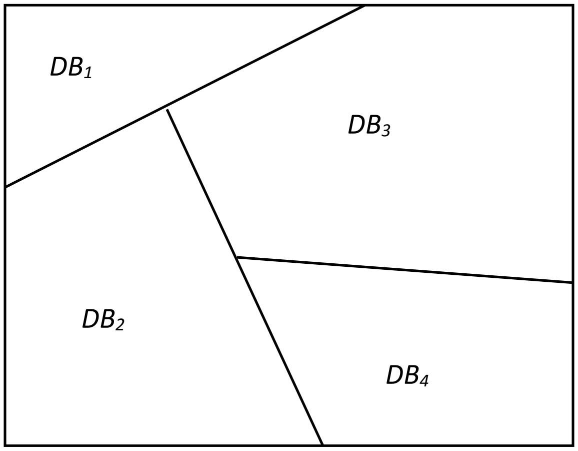

Figure 1: *Data warehouse partitioned by four databases*

The design of data warehouse systems is unlike the design of transactional systems that provide data to the warehouse [8]. There are two well-known authors in the world of data warehousing; Bill Inmon and Ralph Kimball, advocating complementary, yet different techniques to the design of data warehouses. The technique applied by Bill Inmon is the familiar *top-down* design which begins with the Extraction-Transformation-Loading (ETL) process working from external data sources in order to build a data warehouse, whilst Ralph Kimball applies the equally well-established bottom-up technique which begins with an ETL process for one or more data marts separately. Most of proponents of data warehouse design subscribe to either of the two techniques [12].

2.1 Data warehouse systems design Approaches

The DW systems design is based on two approaches that are alternative and inverses of each other viz: the Data-driven approach also known as the Supply-driven approach and the Requirement-driven approach also known as the Demand-driven approach [1], [3], [7], and [10]. The process of development of a DW starts with the identification and collection of requirements. The design of the multidimensional model is next, followed then by testing and maintenance [9]. To develop a data warehouse requires a set of steps to accomplish throughout the process, namely, the requirements analysis phase, conceptual phase, logical phase and physical phase [6], [8]. According to [9], the design stage is the most significant operation in the successful construction of a DW.

The following sections elucidate the context of requirements analysis and conceptual design which are considered as the two main phases within the data warehouse systems design process [8].

2.1.1. Requirements Analysis

Requirements analysis has as its aim detecting which knowledge is useful for decision making by investigating the user's demands and expectations in user-driven and goal-driven approaches, or by verifying the validity of operational data sources in a data-driven approach [8]. Requirements analysis of users plays a crucial role in data warehouse systems design. It has a major influence upon the taking of decisions throughout the data warehouse systems implementation [2], [23]. The requirements analysis phase leads the designer to unveil the multidimensional schema necessary elements (facts, measures and dimensions) which are claimed to assist future data manipulations and calculations. The multidimensional schema has a significant impact on the success of DW projects [2], [3], and [14].

Several research works have published on the various approaches used during the requirements analysis phase of DW systems design, leaning on the two techniques mentioned above – the Top-down technique and Bottom-up technique. Implementations of these are: Data-driven approach, Goal-driven approach, User-driven approach, and Mixed-driven approach [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], and [11]:

- The Data-driven approach also known as the supply-driven approach. It utilizes the bottom-up technique and yields subject-oriented business data schemas by only leaning on the operational data sources and ignoring business goals and stakeholder needs.

- The Goal-driven approach applies the top-down technique rather than the bottom-up technique. It allows for the generation of information like Key Performance Indicators (KPIs) of principal business areas by relying on business objectives and granted business processes only, and essentially overlooks data sources and user demands.

- The User-driven approach is similar to the goal-driven approach, and it applies the top-down technique. It permits to produce analytical requirements interpreted by the dimensions and measures of each subject by ignoring business goals and data sources.

A user-driven approach starts with a detailed agreement of the needs and expectations of the users and this brings about numerous advantages like increasing of the production, enhancing of the work quality, support and training costs reductions and improvement of general user satisfaction [24]. User requirements analysis does not prescribe to a standard approach which designers may rely on for designing their data warehousing projects [16]. As declared by [6] the data-driven approach yields a conceptual schema through a re-engineering process of the data sources by ignoring the end users and stakeholders' contribution, whilst the requirements-driven approach aims at generating the conceptual schema by only considering the needs expressed by end users and stakeholders. The correct elicitation of user requirements remains a fine challenge and many techniques, e.g. the use of JAD (Joint Application Design) sessions [25] have been put forward.

These three primary aforementioned approaches have their merits and demerits. However, in an attempt at overcoming this problem, numerous authors suggested the mixed-driven approach which consists of combining two or even all three the primary approaches (either user-driven and data-driven approaches or goal-driven and data-driven approaches or a combination of user-driven, goal-driven and data-driven approaches) as detailed in [2], all aimed at getting a "best result" that will meet users and stakeholders' demands and expectations. According to [14] and [26], the requirements-driven approach is also called the analysis-driven approach; the supply-driven approach is also called the source-driven approach and the requirement/supply-driven approach is known as an analysis/source-driven approach, but is also known by the name of a hybrid-driven approach.

The various approached discussed above are elicited in **Figure 2** below.

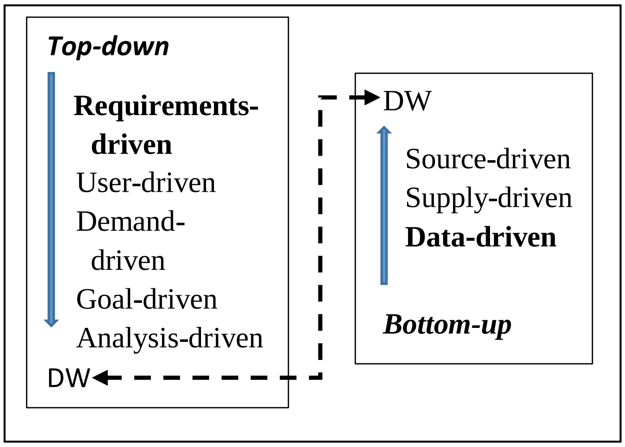

Figure 2: *Complementary top-down & bottom-up*

Table 1 below presents the advantages and disadvantages of approaches grouped by technique.

Table 1: Advantages and disadvantages of techniques.

Technique	Approach	Advantages	Disadvantages
Top-down	User-driven Goal-driven Demand-driven Analysis-driven Requirements-driven	A coherent data dimensional views through data marts is provided from the DW. It's easy from a data warehouse to reproduce a data mart.	It's not flexible to the requirements change during the implement-ation. It's highly expose to the risk of failure.
Bottom-up	Data-driven Source-driven Supply-driven	It's less expose to the risk of failure. It facilitates the return on investment and leads to concrete results in short time.	The data view for each data mart is narrowed. The penetration of redundant data within each data mart.

The above discussion presents an answer to our **RQ1** above.

2.1.1.1. Requirements-driven Approach

The conceptual schema development within the requirements-driven approach is based on business- or user requirements. The organizational purposes and demands that the DW systems are expected to present, support the process of decision making which include the requirements needed for the conceptual schema. The information collected serves as a foundation for the initial data warehouse schema development [14], [26].

Figure 3 below depicts the analysis-driven approach framework with all the steps considered:

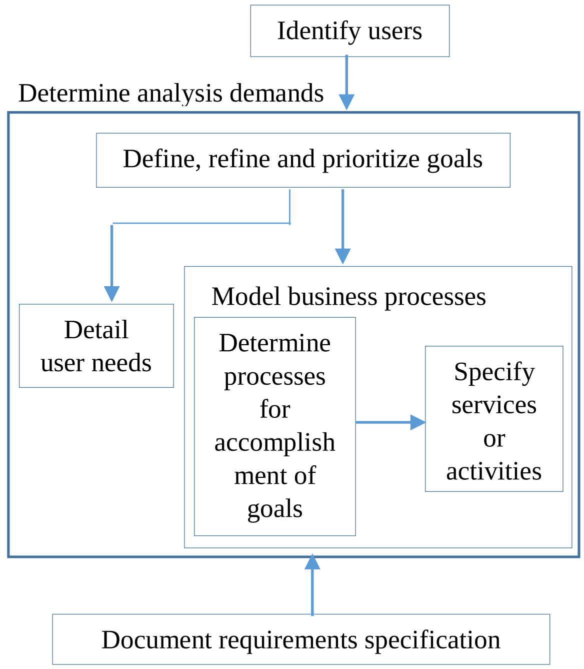
Identify users
Determine analysis demands
Define, refine and prioritize goals
Detail user needs
Model business processes
Determine processes for accomplishment of goals
Specify services or activities
Document requirements specification

Figure 3: *Requirement-driven approach framework [25]*

2.1.1.2. Supply-driven Approach

In the supply-driven approach, the development of the conceptual schema leans on the data available in the operational systems. The objective of this approach is to identify multidimensional schemas which may be conveniently implemented over the legacy operational databases (data marts – see above). An exhaustive analysis is made over these databases to elicit the essential elements which can depict facts with attached measures, dimensions and hierarchies and the discovering of these elements induced to an initial DW schema which can correspond to many different analysis goals [14], [26].

Figure 4 below represents the supply-driven approach framework with the all considered steps.

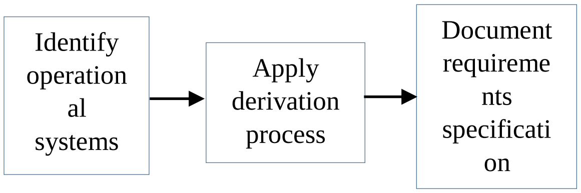
Identify operational systems
Apply derivation process
Document requirements specification

Figure 4: *Supply-driven approach framework [25]*

2.1.1.3. Hybrid-driven approach

The hybrid-driven approach combines the two abovementioned approaches that can be used in parallel in order to obtain a best design [14], [26]. The operation of requirements mapping takes place while fact tables, measures and dimensions are identified during the decisional modeling, and are mapped over entities into the source schema [3].

The hybrid-driven approach framework is illustrated below in **Figure 5**.

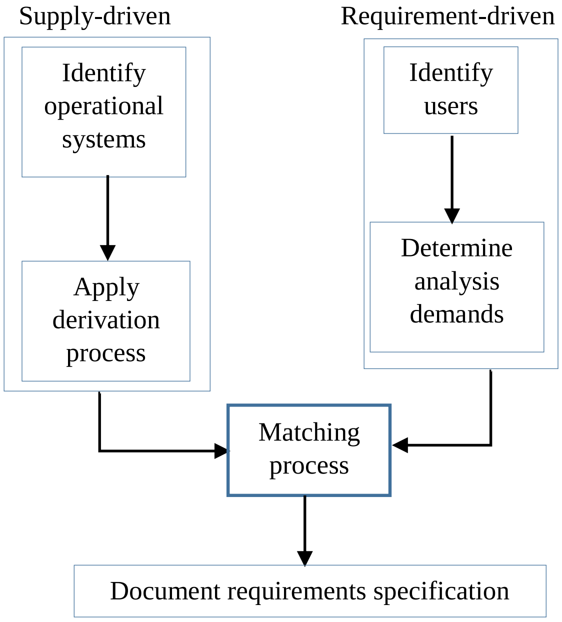
Supply-driven
Requirement-driven
Identify operational systems
Apply derivation process
Identify users
Determine analysis demands
Matching process
Document requirements specification

Figure 5: *Hybrid-driven approach framework*

According to [14] and [26] for all the previously described approaches, showing the different iterations which can be claimed before the final DW schema is developed, the essential step in all the aforementioned approaches that we concentrate on in this paper, is the document requirements specification step. This enables the documentation of all the information obtained from the previous step. This step will contain the business requirements and business objectives expressed in more detail, e.g. what operations may be done, who has the right to access to the data, what measures and dimensions are represented, etc.

2.1.2. Conceptual design

Although most of the research on the design of DWs concerns their logical and physical models, the essential foundation to build a data warehouse on is an accurate conceptual design that is well-documented and thoroughly fulfills the requirements [27]. We addressed some works that discussed the conceptual design of data warehouse systems above.

In [28] the authors proposed a hybrid technique which has as objective to summarize techniques while defining user requirements for DWs in order to get a multidimensional conceptual model using an ensemble of rules Query-View-Transformation (QVT) based on the multidimensional normal forms for the preciseness. In the paper published by [15] an approach of the multidimensional star schema validation supported through reparation solutions has been proposed while assisting designers within the detection of constraint violations and by suggesting reparation solutions based on a number of mistake-based rules formalized in Prolog. However, the main purpose of this stage is to develop a conceptual diagram that will meet the functional requirements from the requirements analysis stage (requirement-driven approach) and the data model designed from the legacy operational systems [8] in order to fulfill the users and stakeholders' demands and expectations. This phase would be useful for the representation of the necessary elements into the multidimensional schema after the specification of requirements.

According to [29] a schema is defined by the relation between facts and dimensions. Fact is the subject of analysis or the focus of interest in the process of decision making [27] and dimensions are different perspectives used for the analysis of facts. Fact contains numerical attributes commonly called measures [29].

Figure 6 below depicts a multidimensional schema for data warehouse systems, following the suggestions in [15].

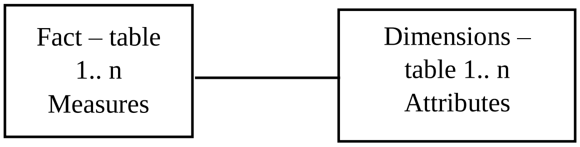

Figure 6: *A multidimensional schema*

Next we present a formal specification of the star-based structure in **Figure 6** in Z.

3 A FORMAL SPECIFICATION

As per the Established Strategy for constructing a Z specification [18], we define the basic types of the system. Our state consists of measures and additional attributes, therefore we have as basic types:

$$[Measure, Attribute]$$

A Fact table consists of attributes, but given the structure of the star schema [15] we distinguish between measures and ordinary attributes as follows.

Fact_table

$measures : \mathbb{P}\ Measure$

$attributes : \mathbb{P}\ Attribute$

$measures \cap attributes = \varnothing$

While measures of fact tables in a star schema are also attributes, we give them special status to reflect the structure defined in [15].

A Dimension as per **Figure 6** consists of a number of fact tables.

Dimension

$dimension : \mathbb{P}\ Fact\_table$

As per the Established Strategy for constructing a Z specification, a specifier has to define an initial state, and discharge a proof obligation (PO) arises, namely, show that such an initial state may be realized (i.e. it exists).

Subsequently, the initial state of the star state is given by:

Init_Dimension

$Dimension'$

$dimension' = \varnothing$

The PO that arises is:

$$\vdash \exists\ Dimension' \bullet Init\_Dimension$$

Therefore, $\vdash \exists\ dimension' \mid dimension' = \varnothing$ from which it follows that the initial state of the system contains no fact tables. Subsequently, $measures = \emptyset \land attributes = \emptyset$. Therefore the initial state *Init_Dimension* above can be realized.

The above Z specification provides an answer to our **RQ2** above.

3.1 The utility of proof

Returning to the partition of the data warehouse in **Example 1** above, we consider the operation of linking a new database to the warehouse. Recall that the databases partition the data warehouse. Consequently we would expect the following proof obligation (PO) to arise from linking (adding) a new database to obtain a new data warehouse, *New_DW*, from the existing one, *DW* (# indicates set-theoretic cardinality):

$$\#New\_DW = \#DW + 1.$$

The use of an automated reasoner, e.g. the late William McCune's classic open source, resolution-based theorem prover OTTER (http://www.cs.unm.edu/~mccune/otter/McCune) is a lucrative way for discharging the above PO.

Suppose we use the following 1^{st}-order logic definitions of cardinality in the proof attempt,

$$(\forall A)(\#A = 0 \Leftrightarrow A = \varnothing).$$

$$(\forall A)(\forall n)\ (\#A = (n + 1) \Leftrightarrow (\exists x)\ (x \in A \land \#(A - \{x\})) = n).$$

then the reasoner finds no proof in a reasonable time (runs out of memory after 20 minutes).

Closer investigation of the 2^{nd} formula above reveals that the definition generates nested *functors* (function symbols) which something to be avoided with resolution-based reasoners. If we, therefore, remove the offending functors by rewriting the 2nd formula as

$$(\forall A)(\forall n)\ (\#A = (n + 1) \Leftrightarrow\ (\exists B)(\exists x)\ (x \in A \\ \wedge\ \#B = n \wedge\ (\forall y)(y \in B\ \Leftrightarrow\ y \in A \wedge \\ y \notin \{x\})))$$

which alleviates the functor problem, then then the reasoner finds a quick proof in the order of *0.2 seconds*.

The above experiment illustrates the usefulness of proving properties of a formal specification, thereby assisting in developing provably correct software systems.

Successfully discharging the above proof obligation provides an answer to our **RQ3** above.

Figure 6 suggests that in the multidimensional schema, we can have several facts link to various dimensions. We choose to design the DW conceptual schema with the model of star.

In **Figure 7** we show how a multidimensional star schema may appear for a specific case.

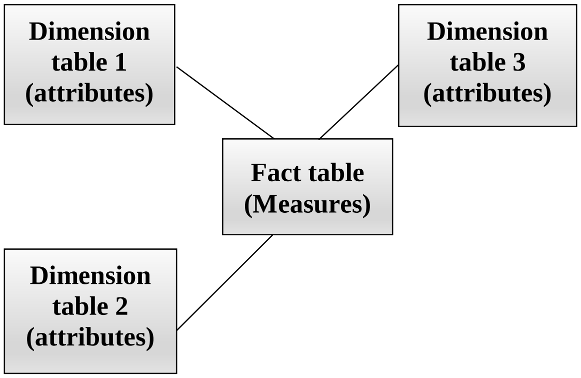
Dimension table 1 (attributes)
Dimension table 3 (attributes)
Fact table (Measures)
Dimension table 2 (attributes)

Figure 7: *A multidimensional star schema [12]*

Example A

Consider an example of a car rental company represented in **Figure 8** to illustrate the generic multidimensional star schema concepts in **Figure 7**.

An example where a decision maker is interested in analyzing the fact Renting in terms of amount measure is described in [15].

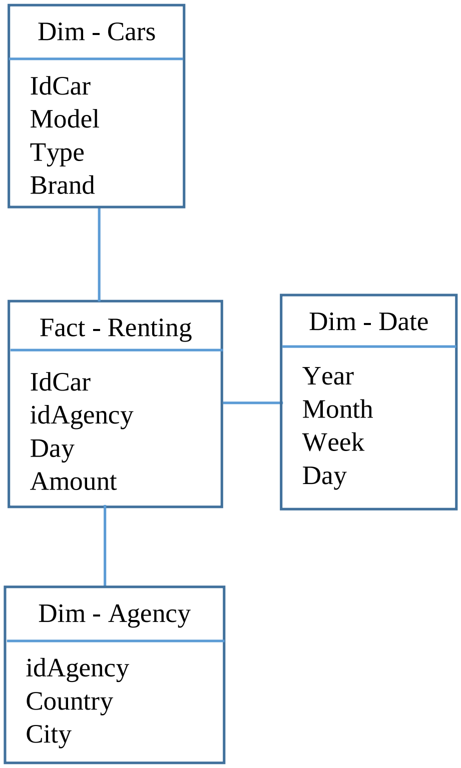

Figure 8: *A multidimensional star schema*

4 RELATED WORK

Several works in the literature have addressed requirements analysis for data warehouse systems and just few works try to link requirements engineering to DW systems [30]. Requirements analysis is thought of as one of the significant tasks to facilitate the success of a data warehouse project [2]. However, we discuss below some work that examine the requirements analysis phase in more detail within the data warehouse systems design arena.

[6] Proposed a hybrid approach based on the two main approaches used for data warehouse systems design (requirements-driven approach and supply-driven approach) which helps to produce the conceptual schema based on a graph-oriented representation, using in turn an automatic operation of data sources reengineering, based on a set of constraints derived from requirements of users.

In [5] the authors introduced various requirements elicitation methodologies by addressing advantages of each of them and they investigated the role of requirements engineering within the DW development and made a comparative study of different requirements elicitation methodologies (GDI model, AGDI model, Use case approach, DFM, CADWA, Tropos) for the DW development.

[10] Discussed requirements analysis approaches for the design of data warehouse systems and suggested a requirements analysis framework based on business objects for data warehouse systems to identify the analytical requirements and further refined those requirements to map onto the conceptual level of the model of data warehouse design, using either requirements-driven or supply-driven approach for DW requirements analysis. The author declared that the multidimensional data model of the conceptual level is the primary deliverable of the data warehouse requirements analysis.

In [9] the author proposed an object oriented framework for DW systems conceptual design and used UML (Unified Modeling Language) within the design process of software system development as this latter has become a standard for object oriented modelling during design and analysis phases. He also made a comparative study of various data warehouse design approaches and schemas used, according to different authors. Various authors, e.g. [14], [28] have discussed different approaches and methodologies used for the DW systems design. Following that, we have chosen to use the hybrid approach to create our framework as a goal of this paper.

5 OUR METHODOLOGY

Our methodology is depicted **Figure 9** below. It is organized as a sequence of steps where each step follows in depth the application abstraction level, as the requirements of the project are collected to form the requirements basis. Such requirements are ultimately formally specified, using e.g. Z.

As is evident from the diagram, we base our approach on both the hybrid-driven approach as well as requirements-driven approach, aimed at assisting designers of data warehouse systems to design a conceptual schema that will meet the users and stakeholders' needs and expectations. Therefore, we are not concerned with all these steps of requirements elicitation for documenting requirements – details may be observed in [14].

The two sets of requirements stem from bottom-up and top-down analysis respectively. Furthermore, for the requirements analysis, requirements are collected from dissimilar users and on the supply side from different legacy operational systems. Finally the requirements are combined as shown and discussed below.

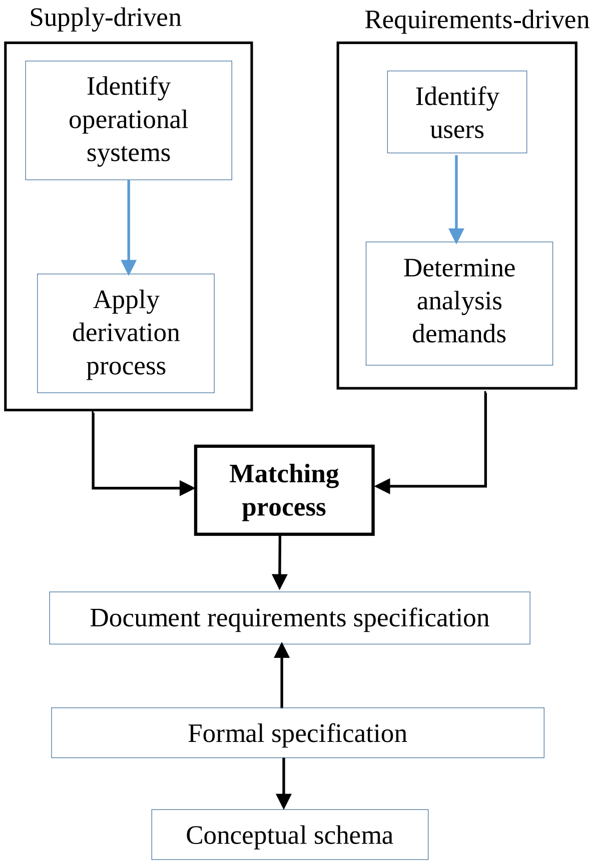

Figure 9: *Proposed framework*

The crucial step in **Figure 9** above is the matching of the two sets of requirements obtained through the top-down processes and the bottom-up processes. It is plausible that these requirements sets may be non-homogenous and in different formats, e.g. one may contain structured data (bottom-up data), while the other set may contain unstructured data obtained through incomplete and often inconsistent user requirements.

To merge the two data sets we enhance an SRE (Software Requirements Elicitation) algorithm defined in [25] as follows:

```
Algorithm 1: Enhanced SRE – Merge structured and unstructured data sets obtained from bottom-up and top-down requirements elicitation.
Begin
(* Elicit unstructured- & structured data through requirements- and supply driven elicitations. *)
(* Assume m structured data sets *)
for i := 1 to m step 1 do
   $S_i$ := Structured data set i of supply-driven requirements;
(* Assume n unstructured data sets *)
for j := 1 to n step 1 do
   $U_j$ := Unstructured data set j of user requirements;

(* Amalgamate the two data sets into two separate sets S and U. ⋃ denotes set-theoretic arbitrary union. *)
   $S := \bigcup_{i:=1}^{m} S_i;$      $U := \bigcup_{j:=1}^{n} U_j$
(*Combine outcome of elicitations as per Figure 8 into set C.*)
   C := S ∩ U
End.
```

Following the construction of set *C* containing both structured and unstructured data in **Algorithm 1**, the next step is to construct a requirements definition [31], [17] as per **Figure 9**. From the definition a formal specification [22] would be constructed in line with the ideas presented in Section 3.

Algorithm 1 provides an answer to our **RQ4** above.

6 CONCLUSIONS AND FUTURE WORK

In this paper we presented numerous works which discuss about approaches used for data warehouse systems design. Various authors in the literature review [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and [32] have proposed different techniques and/or approaches and frameworks to address the matter of DW systems design. We also investigated the extent to which the use of formal methods – formal specification and proof – for data warehouse systems may alleviate such failures within the design process.

We proposed a requirements engineering methodology which represents our proposed framework which is created from one of the existing requirements analysis frameworks for the specification of system requirements to requirements analysis for the design of DW systems at the conceptual level. The use of a formal specification as a last step in the process is proposed.

We anticipate our proposed framework to facilitate with removing vagueness or inconsistencies, at least to some extent. Such ambiguity is often present in natural language specifications used to specify data warehouses.

As for future work, we intend to test our methodology on a case study using the CZT toolset, aimed at facilitating the building of a formal specification.

The combined set C in **`Algorithm 1`** above contains both structured and unstructured data, hence the set is non-homogeneous. Z, however, is a strongly typed language, containing only “homogeneous” sets. Therefore specifying the above processes in Z will require us to devise a mechanism to deal with structuredness as well as unstructuredness in the same set.

REFERENCES

[1] N. Jukic and J. Nicholas. (2010). A Framework for Collecting and Defining Requirements for Data Warehousing Projects. *Journal of Computing and Information Technology*, *18*(4), pp. 377–384. https://doi.org/doi:10.2498

[2] N. H. Z. Abai, J. H. Yahaya and A. Deraman. (2013). User Requirement Analysis in Data Warehouse Design: A Review. *Procedia Technology*, *11*, pp. 801–806. https://doi.org/10.1016/j.protcy.2013.12.261

[3] P. Giorgini, S. Rizzi and M. Garzetti. (2008). GRAnD: A goal-oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, *45*(1), pp. 4–21. https://doi.org/10.1016/j.dss.2006.12.001

[4] D. T. A. Hoang. (2011). Impact Analysis for On-Demand Data Warehousing Evolution. In *ADBIS (2)* (pp. 280–285). Retrieved from https://pdfs.semanticscholar.org/ae5c/a847a8afc046951e34653fcbd3ade06322cb.pdf

[5] S. Mathur, G. Sharma and A. K. Soni. (2012). Requirement elicitation techniques for data warehouse review paper. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), pp. 456–459. Retrieved from https://www.researchgate.net/profile/Girish_Sharma4/publication/259827907_IJETAE_1212_84/links/02e7e52e0dc392211f000000.pdf

[6] F. Di Tria, E. Lefons and F. Tangorra. (2011). GrHyMM: A graph-oriented hybrid multidimensional model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6999 LNCS, pp. 86–97). Springer. https://doi.org/10.1007/978-3-642-24574-9_12

[7] M. Golfarelli. (2010). From User Requirements to Conceptual Design in Data Warehouse Design. *Data Warehousing Design and Advanced Engineering …*, 15. https://doi.org/10.4018/978-1-60566-756-

0.ch001

[8] M. El Mohajir and I. Jellouli. (2014). Towards a Framework Incorporating Functional and Non-Functional Requirements for Datawarehouse Conceptual Design. *IADIS International Journal on Computer Science and Information Systems*, *9*(1). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.640.5590&rep=rep1&type=pdf

[9] R. Jindal. (2012). Comparative Study of Data Warehouse Design Approaches : A Survey. *International Journal of Database Management Systems*, *4*(1), pp. 33–45. https://doi.org/10.5121/ijdms.2012.4104

[10] A. Sarkar. (2012). Data Warehouse Requirements Analysis Framework: Business-Object Based Approach. *International Journal of Advanced Computer Science and Applications*, *3*(1), pp. 25–34. https://doi.org/10.14569/IJACSA.2012.030104

[11] A. Nasiri, E. Zimányi and R. Wrembel. (2015). Requirements Engineering for Data Warehouses. Retrieved from http://code.ulb.ac.be/dbfiles/NasZimWre2015incollection.pdf

[12] T. Oketunji and O. Omodara. (2011). Design of Data Warehouse and Business Intelligence System. *Master Thesis*. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:831050

[13] T. Pandey and S. Srivastava. (2015). Comparative Analysis of Formal Specification Languages Z, {VDM} and B. *International Journal of Current Engineering and Technology*, *5*(3), pp. 2277–4106. Retrieved from http://inpressco.com/wp-content/uploads/2015/06/Paper1082086-2091.pdf

[14] E. Malinowski and E. Zimányi. (2008). Advanced data warehouse design: from conventional to spatial and temporal applications. Retrieved from https://books.google.co.za/books?id=XPMVW3PtGtEC&pg=PA259&lpg=PA259&dq=incomplete+requirements+for+data+warehouse&source=bl&ots=pbxOHI231W&sig=ZufrX0byf75gvHMUeMOg3-9z_v4&hl=en&sa=X&ved=0ahUKEwjF9df6-KbWAhUrJsAKHateCJQQ6AEIOTAF#v=onepage&q=incomplete%20requirements%20for%20data%20warehouse&f=false

[15] A. Salem and H. Ben-Abdallah (2015). The design of valid multidimensional star schemas assisted by repair solutions. *Vietnam Journal of Computer Science*, *2*(3), pp. 169–179. https://doi.org/10.1007/s40595-015-0041-1

[16] M. Dos Santos Soares and D. S. Cioquetta. (2012). Analysis of techniques for documenting user requirements. In *International Conference on Computational Science and Its Applications* (pp. 16–28). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-31128-4_2

[17] I. Sommerville. (2015). Software Engineering (10th ed). Pearson.

[18] B. Potter, J. Sinclair and D. Till. (1996). An Introduction to Formal Specification and Z, 2nd edition, Prentice Hall International Series.

[19] J. Q. Zhao. (2007). Formal Design of Data Warehouse and OLAP Systems.

[20] S. A. Han and H. Jamshed. (2016). Analysis of Formal Methods for Specification of E-Commerce Applications, *35*(1), pp. 19–28.

[21] S. H. Bakri, H. Harun, A. Alzoubi and R. Ibrahim. (2013). the Formal Specification for the Inventory System Using Z Language. *The 4th International Conference on Cloud Computing and Informatics*, (64), pp. 419–425.

[22] M. Gulati and M. Singh. (2012). Analysis of Three Formal Methods-Z, B and VDM. *International Journal of Engineering*, *1*(4), pp. 1–5. Retrieved from http://www.ijert.org/browse/june-2012-edition?download=297:analysis-of-three-formal-methods-z-b-and-vdm&start=120

[23] J. Schiefer, B. List and R. Bruckner. (2002). A holistic approach for managing requirements of data warehouse systems, 13. Retrieved from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1372&context=amcis2002

[24] M. Maguire and N. Bevan. (2002). User requirements analysis. In *Usability* (pp. 133–148). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-35610-5_9

[25] W. F. Friedrich and J.A. van der Poll. (2007). Towards a Methodology to Elicit Tacit Domain Knowledge from Users. *Interdisciplinary Journal of Information, Knowledge and Management (IJIKM)*, Volume 2, 2007, pp. 179 - 193. ISSN: Print 1555-1229. URL: www.ijikm.org

[26] E. Zimanyi. (2006). Requirements Specification and Conceptual Modeling for Spatial Data Warehouses, *4278*(October 2017). https://doi.org/10.1007/11915072

[27] M. Golfarelli, D. Maio and S. Rizzi. (1998). Conceptual design of data warehouses from E/R schemes. *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, *7*, pp. 334–343. https://doi.org/10.1109/HICSS.1998.649228

[28] J. N. Mazón, J. Trujillo and J. Lechtenbörger. (2007). Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data and Knowledge Engineering*, *63*(3), pp. 699–725. https://doi.org/10.1016/j.datak.2007.04.005

[29] A. Vaisman and E. Zimányi. (2014). *Data Warehouse Systems*. https://doi.org/10.1007/978-3-642-54655-6

[30] F. Rilston, S. Paim, A. E. Carvalho and J. B. De. Castro. (2002). Towards a Methodology for Requirements Analysis of Data Warehouse Systems, pp. 146–161.

[31] I. Sommerville. (2004). Software Engineering. A Brief History of Computing (9th ed). Pearson. https://doi.org/10.1111/j.1365-2362.2005.01463.x

[32] W. H. Inmon. (2005). *Building the Data Warehouse* (4th ed). Indianapolis, Ind: Wiley.

[33] W. H. Inmon. (2005). *Building the Data Warehouse* (4th ed). Indianapolis, Ind: Wiley.

[34] W. McCune. (1994) *Otter 3.0 reference manual and guide (Vol. 9700)*, Argonne, IL: Argonne National Laboratory.

[35] J.A. van de Poll and P. Kotze. (2005). Enhancing the Established Strategy for Constructing a Z Specification. *South African Computer Journal*, Number 35, pp. 118 - 131, December 2005.

Biographies

Isaac N. Mbala holds an Honours BSc Computer science degree and is completing his MSc Computer science degree. His current focus area is to determine the extent to which the use of formal methods for data warehouse systems may alleviate such failures within the design process. Looking forward to do his PhD in Computer science.

John A. van der Poll holds a PhD in Computer science and is a full time professor in the Graduate School of Business Leadership (SBL) at the University of South Africa (Unisa). He teaches numerous business ICT courses and his research interests are in formal specification techniques for business ICTs and subsequent automated reasoning.