

A Semantic Data Model: Meaning Making from Data Structures in the SQL Server

Sanjay Ramesh^{1)*}, Anthony Henderson²⁾

¹⁾³⁾ University of Sydney
Camperdown NSW 2006, Australia

¹⁾sanjay.ramesh@sydney.edu.au

²⁾Swinburne Institute of Technology
Hawthorne VIC 3122, Australia

²⁾7576277@student.swin.edu.au

Article history:

Received 29 June 2018
Revised 19 September 2018
Accepted 20 September 2018
Available online 28 October 2018

Keywords:

Semantic
Decision
Data
Business
Information

Abstract

Information systems designs are increasingly concerned with entity relationships and technical programmatic approaches to solutions architecture as opposed to semantic based, business focused information architecture that places business definitions at the centre of the information system design and implementation. The disconnect between information technology and business is perpetuated by an overly prescriptive information technology technical design method that fails to incorporate qualitative and normative aspects of business, where information is structured and delivered according to business. The paper will discuss various decision support and semantic approaches to information design and delivery and argue that the traditional modes of solution delivery do not include meaning making of the data elements which are essential to business information reporting and analytics. The meaning making aspect identified is linked to data dictionary or business data glossary that allows for the discovery of semantic meaning from the SQL Server. Using Christian Fürber's methodology on semantic programming, the analytics team developed a semantic model that enabled detailed definition of fields and the discovery of information using semantic search functionality embedded in the SQL Server. The project provided semantic data framework that provided business with the capability for semantic reconciliation and data sets that were further integrated with Tableau visualization and SQL auto processes.

I. INTRODUCTION

Traditional methods of information design and delivery are focused primarily on entity relationships and table join structures which inform the contemporary data model. These forms of relationships are premised heavily on the data schema that comes from an "out of the box" solution and influenced by standard technical interpretations, which assists the technical team in the technical architecture of the solution. Once the system is designed, built and implemented, the knowledge of the data structures, data schemas and data models that informs the data solution are embedded in the technical arena. In most cases, there is an absence of business data definitions or data dictionary or data glossary resulting in the reliance on continued support from information technology technical teams and hence tensions and conflicts between the business and the technical team ensues. These anomalies give rise to high cost of implementation because the business is disconnected from the technical platform due to a lack of the meaning making process. Solutions architect and data model design engineers determine ontological issues such as what, where, when and how but these are increasingly based on technical considerations on what can be technically tested within the pre-defined technical landscape instead of business interpretation of what ought to be. Solutions are

* Corresponding author

increasingly technically driven and business analysts are left with the painful task of semantic interpretation to explain the meaning of data elements, data dimensions and measures to non-technical management group, creating uncertainty and unease in complex information and business environments. These problems, Authors argue, can be mitigated by semantic data solutions where meaning of the data elements are embedded at the solution concept and design stages, instead of attempts to fit data definitions after the post-implementation phase of the information technology project. Semantic data modeling is not new to the data modeling literature. In fact, semantic models have been a subject of debate among information experts since the late 1980s, following the advent of relational database designs and schemas [1]. However, the adoption of semantic models has been slow because it is commercially viable for large information technology consultancies to create perpetual dependency on their technical knowledge, support and expert advice and these commercial considerations are unsustainable. This form of dependency on technical and commercial frameworks does not promote self-service, innovation or adoption of business semantic information in all aspects of information delivery. As a result, Authors argue that as far as possible decision and semantic models become the standard with solutions experts providing businesses with schemas that make discovery of the meaning of data easier and less onerous.

Traditional information modeling is based on linear, non-modular design mechanisms where entity relationships are the key drivers of the data model. These linear programming methodologies are derived from structured methods based on mathematical and information programmatic considerations where one-to-many, many-to-one and many-to-many are considered as the primer of data model design. These approaches to some aspect were aimed at bridging the gap between information database management systems and the business where operational and strategic variables informed technical design, adoption and socialization. These principles were applied to databases, data mart, enterprise data mart design and deployment. However, it was acknowledged by the business that the model did not address business requirements, hence endless debates on the inherent disconnect between business and technology.

Relational database design engendered a hypostasis where entity relationship informed largely by join structures identified largely relational information systems. These relationships were developed in isolation and were influenced by technical design methodology, often based on rapid deployment due to cost and resource considerations. Whilst private sector was more circumspect about agile implementation, the public sector became “test cases” where technical expertise, business requirements and benefits converged to create a dysfunctional tri-factor where business, technical and analytical outcomes were conceptualized as distinct and potentially un-related outcomes. The problem was that the solution was already architected as “out of the box” and business requirements later became re-engineering process, and the business case based on cost and benefit analysis lacked business rigour and due diligence [1]. The problems were accentuated by lack of technical knowledge by the business subject matter experts who were unable to guide technical experts. In the end, there were two different and problematic narratives: one based on the greatness of the technical achievement and the other on workarounds due to gaps in the final solution.

To understand the mechanics of databases, it is important to trace the history of data models and its attempts to bridge the challenges of semantics and decision support. The purpose of traditional models was to set up the technical infrastructure so that it enables decision making in the organization by provisioning data extraction and collation from enterprise systems. However, before even proceeding to decision support systems, these traditional data models did not provide semantic information and decisions were left to business interpretation of information from the data itself causing misinterpretations and confusion. In a study by Terry Halpin and Tony Morgan, it was argued that the relational models and queries did not provide semantic information or were “expressed in a language that did not retain their original intent” [2]. As a result, there were lines of codes used that only made sense to the developer who developed the model.

The evolution of databases and relational models provide an interesting insight into the world of information systems and it also highlights that more recent additions to the literature emphasize decision support and semantic programs so that data elements in the databases and servers are well defined and easy to discover. Using semantic data modelling, I will demonstrate how to construct a SQL Server semantic model that allows for multiple data sources to be extracted, transformed and loaded in SQL views following semantic data labelling of fields via a SQL semantic definitions repository. The objective is to highlight the utility of a layer that empowers business to understand the data in the SQL Server database views. However before addressing the challenges of designing and implementing semantic information systems, we review the literature on relational databases and semantic models.

II. LITERATURE REVIEW

It is useful to understand the evolution of major data models as each generation of information theorists allowed greater support for databases. The first and second generations of data models led to widespread use throughout the 1960s because of their capacity to process large volumes of data. The first generation was a computer file system, mainly used on IBM mainframe systems to manage storing of data but not have defined data field relationships. File systems were inherently simple and at times inaccurate, leading to data anomalies and misinterpretations. The second generation of information systems included data hierarchy and network data models, considered to be early database management systems. Much like the file system, these database models were rather rigid and did not allow for multiple level data queries. The hierarchical model evolved into a relational system of one-to-many relationships between parent and child. The network model expanded on the hierarchical databases by allowing data elements to have more than one parent, and these types of data schemas continues to be part of the database development strategies even today. Edgar Frank Codd of IBM [3] introduced the third-generation relational database model in 1970 that extended relational databases to enable robust multiple data queries. In this model, data entities are related to each other through common data attributes and supports multi-level queries using Structured Query Language (SQL). In 1976, Peter Chen [4] introduced graphical Entity Relationship (ER) model as an enhancement to the existing relational database environment. This graphical representation of data entities and their multiple relationships was quickly adopted as the industry standard. The fourth generation of data model came about in 1985 [5] with the introduction of the object-orientated programming. The object-oriented approach supported different data types, and enabled data and their relationships to be contained in a single structure, and like the relational databases, object-oriented approach is identified by its factual content and includes information about relationships between the fact tables within the object, as well as information about its relationships with other objects. In this way, the fact tables within an object are given greater meaning which is why the object-oriented method can be conceptualized as an early form of semantic data model. In response to these developments in the mid-1980s, the relational data was extensively extended [6], adding many of the object-oriented features within the traditional relational database structures. The extended relational database model spawned new technical innovations in relational database designs encapsulated large volumes of data from multiple complex data systems [7].

The traditional relational databases and object-oriented approaches still had limitations even though they had the capability to interrogate large volumes of data from multiple systems because of the three tier processes: input, transformation and output. These approaches place a great deal of emphasis on data entities, data attributes, and the relationships among various data entities. Even the object-oriented approach is limited by its emphasis on object definitions and its relations with other objects [8].

As highlighted the traditional data models did not in any significant way facilitate semantic based modelling but did enable multiple level analysis of databases. The data was contained in tables as flat files in the early stages of the database development and then moved to a relational system where structured queries were utilized to extract information. The object-oriented programming complemented database development but lacked the semantic layer that defined data fields in business terms. However, efforts were made from the early 1980s to address these shortcomings.

Theories on semantic data models emerged in the late 1980s when information experts started deep analysis of existing databases and object-oriented models and further deliberated on ways of enhancing business rules, metadata and key performance indicators to create a more business focused, business friendly and agile data schemas based on the utilization of basic language of interpretation (semantics) and meaning making (semiotics). These approaches were aimed at building on the traditional relational models to embed business definitions and quality measures which would allow for easy discovery business critical information by enabling business analytics and reporting. The aim of this unique approach was to move information system design from purely entity relational and object-oriented type solutions to one where business meanings are incorporated and embedded at each stage of the design and the delivery of the system and data solutions.

According to Joan Peckham and Fred Maryanski [9] a semantic data model provides more semantic data content based on business definitions as opposed to an information model based on technical relationships on fact tables. This semantic content captures business processes, business rules, and performance metrics in a single semantic repository that is reusable by the business for insights into their information. The work of Peckham and Maryanski highlighted that there was a business requirement to build on the existing databases and business warehouses by constructing a descriptive semantic model that explains business meaning of data. Descriptive definitions of business information were to be in plain English or in simple vernacular that allowed non-technical business users to easily understand and interpret data fields, including business processes, rules and performance. Peckham and Maryanski argued for a fundamental shift in the thinking on traditional forms of information modelling and

recommended semantic approaches where meaning making and embedding these meanings into the data models were two fundamental objectives.

Building on the discussions already started Maryanski, Barbara von Halle [10] highlighted that new ways of thinking on semantic data maximized value to the organization. She argued that there should be a move away from traditional entity relationship style databases to a model that utilizes new design methodologies to produce quality semantic measures which provided detailed meaning and purpose to the business information. von Halle detailed her semantic model as a kind of decision model which brought to the world of business and business rules a well-defined conceptual structure based on semantic logic extended with integrity and normalization principles of modern information management [11]. For Halle, data ought to be categorized, interpreted, normalized and definitions of data elements derived automatically from the system and data models, instead of classification taxonomies at the end of the data classification process. A semantic model, according to von Halle is a template for organizing and managing business data meaning and logic [11].

The semantic model resembles and to an extent enables the decision model, which is based on the premise that business logic has its own existence, independent of how it is executed, where in the business it is executed, and whether or not its execution is implemented in automated systems. As an output of von Halle's analysis, the decision model has a 'recognizable structure that is not the same as the structure of other kinds of models' for it has multiple semantic layers [11]. It is important to note that the semantic model is pre-requisite to a decision model in the sense that the semantic information contained in the semantic layer informs the final form of the decision model which is used to make decisions after the meaning of the information is already established and agreed to by the business. The semantic and the decision models collectively forms the backbone of the business intelligence system.

Other studies on semantic model compliment von Halle's thesis. In Popovič et al study [12] of decision support systems, the main objective was to provide a comprehensive understanding of the interrelationships between business intelligence success factors and special attention was given to semantic information quality, the use of information in business processes and the factors affecting the level of use of information in business processes that lead to the creation of business value. Moreover, the study was aimed at establishing clear links among databases, semantic layers and decision support. The study provides insights on the foundations of technical and business discourses, including critical success factors for business intelligence implementation projects. The work of Popovič et al [12] expands the focus from data warehousing to semantic data quality measures and extends the study of business intelligence by adding semantic analytical capabilities of the information management system. This approach suggests the use of semantic information in organization's processes as a dependent variable and establishes links between success dimensions. The authors encourage business intelligence practitioners to focus more on knowledge workers' actual needs (providing content quality), and not merely on providing the rapid processing and delivery of information (access quality). In addition, semantic data will ensure the comprehensiveness and conciseness of the information. Popovič et al further state "a lower understanding of business processes, supporting information systems, legacy systems, and even hardware infrastructure will be reflected in lower business intelligence maturity, semantic quality and, consequently, more robust information use. Although the improved information quality impacts the level of information use, limits may be expected on the quantity of information an organization can absorb and the related dominant impact of organizational culture on semantic modelling and decision-making, specifically the attitude to the use of information in decision-making processes." [12].

Extending on the work of Popovič [12], von Halle [10] [11], and Krysinski et al [13] present their case for creating a convenient semantic environment for ordinary users so that meaning of data fields are easily discovered as well as maintaining rich functionality available for professionals for semantic engagement with multiple information systems. The authors pay special attention to semantic standards and data cooperation issues, which makes their system available to other applications implementing semantic data and keeps it open for data integration projects. [13] The authors argue for the creation of a semantic data repository in subject, predicate and object format which is a natural and widely used representation of semantic data. A schema describing the structure of data is expressed as Ontology Wide Language document unlike in the case of relational based systems in which a structure of data is reflected by the structure of a database [13].

The semantic model defines data models by using business logic and definitions and as a result, it is independent of technology and dependent on business definitions of data fields. The essence of a semantic model is to give easy to discover and understand the meaning to technical fields that are embedded in data models. Just as in a decision model, the semantic model allows for managing business logic and business rules via business meaning making and this form of data modelling enables better decision and business outcomes and evidence-based approaches at all levels of the organization.

It is therefore important to the role of meaning making in the semantic model to define business logic, present definitions in a way that is easily understood and interpreted by business, use semantics that are easily implementable and platform agnostic, ensure semantic models are embedded in the semantic layer of the data or the decision system, address problems associated with business definitions including how to effectively develop, test and implement business meanings that provide comprehensive understanding of business in layman's terms.

Whilst the Decision Model is an intellectual template for perceiving, organizing, and managing the business logic behind a business decision a semantic model is a definition template for defining, embedding and managing business meaning of data. As a template, the semantic model is a logical definition of business meaning and is platform agnostic because it attempts to define key data elements so that information is easily discovered, and analytics becomes a less onerous process.

Similar to a Decision Model that can be translated into one or more target database and information technologies through appropriate design methods, a semantic model propagates a similar approach where business meaning takes the centre stage in the concept, design and implementation phases of the data models. It should be noted that a semantic model follows from a decision model, which is aimed at exposing business logic. Once this business definitions are exposed, we must define the meaning of the data fields in the data model that come out of business decisions. Moreover, the semantic model also fosters collaboration. Knoll [14] argue that a semantic model makes it easier to define a problem, search for solutions, and generate evaluate and implement solutions through context and process awareness. According to the authors, semantic context awareness "is to achieve automatic self-configuration, where types of systems often employ adaptation mechanisms based on rule systems. Once an event is detected and a condition satisfied, the system executes actions, which represent foreseen adaptations." [14]. Process awareness, distinct from context awareness, "aims at coordinating users to accomplish a goal. In this type of system, the process logic is separated from the application code to keep system flexibility if the process changes." [14].

In their analysis of a semantic model and decision models, Richard Wang and Diane Strong [15] argued that the semantic model should be accessible to the data analysts so that business definitions can be easily retrieved from the model itself and the business measures created in the semantic model must be interpretable so that accurate conclusions could be arrived by the business. Wang and Strong recommended a hierarchical semantic model structure that is adaptive to the changing information requirements. A semantic hierarchy model, like a non-hierarchical semantic model, include improved classification and detection of business information by business analysts, who make further meanings out of data because the definitions of data elements are embedded in the model itself. The semantic hierarchy algorithm starts by constructing a minimal feature hierarchy and proceeds by adding semantically equivalent representatives to each business measure. The purpose of the semantic model for Wang and Strong was to embed meaning in the data model itself and to ensure that these data meanings are easily discoverable by analysts.

Daniel Moody and Graeme Shanks [16] in their analysis of data and decision models supported a hierarchical semantic model but cautioned semantic model enthusiasts because their research indicated that there was a risk of over-emphasizing the classification and interpretability of a business measure instead of accurately constructing the measure based on the business requirement and business context. Moddy and Shanks concerns are legitimate since a lot of time is spent on classification of information at multiple times and its leads to long development time. Sharon Bjeletich [17]agree that semantic data models can be very complex and until semantic databases are commonly available, the challenge remains to find the optimal balance between the semantic model that is easily to develop and deploy in an increasingly agile development environment. The key to success of a semantic data model is to understand the issues of data definitions, make the necessary mitigations for those issues, and then embark on semantic design. While there are different theoretical positions on semantic models, there is a common theme across all semantic model analyses: that the semantic model is an improvement to the information model where business definitions of the data fields are embedded in the system to provide robust information delivery and discovery.

Better delivery and discovery of information can lead to a more informed and quicker analysis and reporting. Carl Adams [18] argued that a defined semantic model will not only assist in robust information management but also assist organizations to evaluate all relevant business dimensions and measures and assist in formulating strategic enhancements to their current information assets, leading to strategic business planning and better decision making.

A brief case study below provides information on developing and implementing a semantic solution that enables business meaning discovery from data fields using SQL Server. The model in the case study is aimed at addressing specific business issue and bridging the disconnect between business and technology.

III. METHODS

An analytics team spend 30-40% of their time to collect, cleanse, manipulate, store and interrogate data to achieve the objectives, strategies and targets of a large organisation. Data is extracted from multiple sources on daily, weekly and monthly intervals and stored in Microsoft (MS) Office formats on network drives. Data is then cleansed, normalised, standardised and manipulated using MS Access databases. The volume of data often exceeds the conventional limits of MS Excel spreadsheets and MS Access databases.

Collection and manipulation of data in this manner has a number of negative impacts. It is time consuming to prepare the data and bring it to a point where it is fit for use, leading to data service delivery failures. The process is fairly cumbersome because data is extracted from multiple databases and systems and then imported into multiple database tables and then combined using Access queries. Data analysts time is not used effectively and efficiently because they are constantly preparing the data and the risk is that different analysts use their own method and create their own databases which then leads to multiple versions of the data. This can lead to the consumption of inconsistent and possibly inaccurate reports.

A business case was developed to acquire and implement a SQL Server 2012 solution that would efficiently and effectively capture and deliver quality information to the respective stakeholders in a consistent and timely manner. The purpose was to enable single source of truth and semantic analytics which were absent from the legacy approach. A great number of benefits were realised from this solution. It was a relatively inexpensive solution that allowed semantic information to be embedded following the importation of data fields from multiple sources and enabling semantic SQL coding.

Using Christian Fürber [19] method on semantic programming, the aim of the SQL semantic model was to understand entities and relationships in a machine-readable way and in a manner that facilitated knowledge of data elements that was transparent to the analysts and the business user. The solution would be scalable to incorporate new data elements that would be easily understood within the business context and there would be an opportunity for some form of semantic reconciliation where users of the data can precisely define the meaning of the data at various stages of data interpretation.

Fürber [19] argues for semantic representation of data as essential for getting the model right. Following extensive analysis of data sets and data models, Fürber came up with the Semantic data management framework.

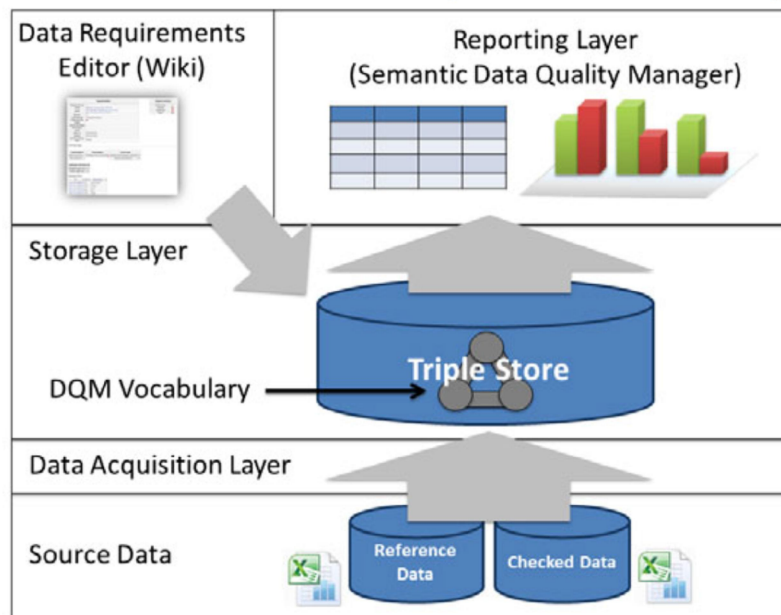


Figure 1: Christian Fürber's Semantic Data Management Framework in C.Fürber, *Data Quality Management with Semantic Technologies*, Wiesbaden: Springer, 2016, p.87.

Using Fürber as a reference, the analytics team replicated the semantic data modelling framework to develop a semantic model using SQL Server. The objective was to build semantic search capabilities within the SQL server which allows the user to query the meaning of the document due to automatic tag extraction, content discovery, and navigation across similar content.

Query of Semantic Search using SQL:

```
SET @Title = 'Sample Semantic Document.docx'  
  
SELECT @DocID = DocumentID  
FROM Documents  
WHERE DocumentTitle = @Title  
  
SELECT @Title AS Title, keyphrase, score  
FROM SEMANTICKEYPHRASETABLE(Documents, *, @DocID)  
ORDER BY score DESC
```

Data extracted from multiple systems was stored in the SQL Server and semantic methods were used to transform data. For example, Supplier Name. Name was standardised to Supplier Name and the meaning of Supplier Name was extracted using semantic search functions. This process was repeated for all the data fields.

The benefits of the project were clearly articulated, and they were: Data analysts were able to cut time spent on information discovery. A semantic data model allowed for easy to search information from the SQL Semantic repository and there was high level of accuracy in interpreting the data fields. Post production validation highlighted that the data was accurate and was fit for use after extraction and easy for the clients to work with after the data was shared. Once semantic information was embedded, it was easy for other analysts outside the business unit to understand data, especially those liaising with senior managers and executives and automation of data cleaning, normalising, categorisation resulted in savings in time and resources. The semantic description embedded in the SQL auto process ensured that data cleaning, normalising and categorisation processes were repeatable, and the technical process was understood by the business. Furthermore, the semantic data model enabled the team to integrate Tableau visual analytics to the SQL semantic server. Since the semantic layer was well defined during the design stage, the visualisation layer was very easy to understand and interpret and there were a number of positive feedback from customers on dashboards deployed in production.

The solution also provided the opportunity to the business to predict and forecast on spend trends and scope savings opportunities through more strategic projects. Using the SQL Semantic Server as a baseline, other data projects are currently underway including cleaning and standardising and defining other more complicated unstructured data sets with NoSQL properties.

IV. RESULTS

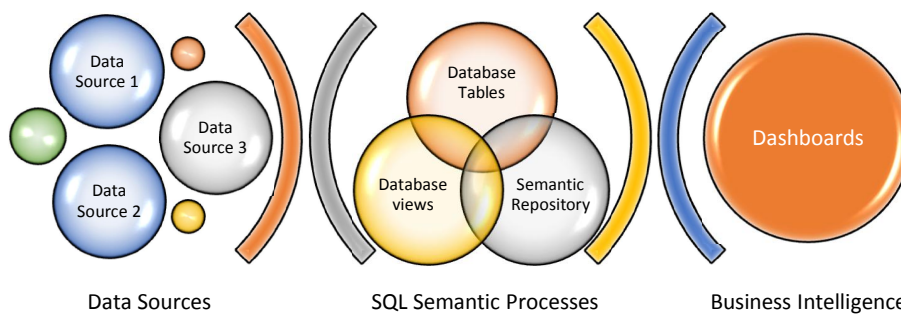


Figure 2. SQL Semantic Model

The above SQL semantic model enabled automation of data cleansing, categorisation, normalisation and standardization which assisted analysts to spend a greater proportion of time on activities such as providing business intelligence through data visualization. Information has a high level of accuracy and is meaningful to clients because business process information and meanings are embedded in semantic processes. The interpretation of data is not left to the data analyst, but rather meaning making is enhanced in consultation with subject matter experts and clients. In this way, the logic that makes meaning from underlying data is stored in a single location and controlled in a formal data governance rule. Put this all together and the result has enhanced data analytics team effectiveness in managing business expectations and enables a platform for the team to undertake more strategic data analytics projects.

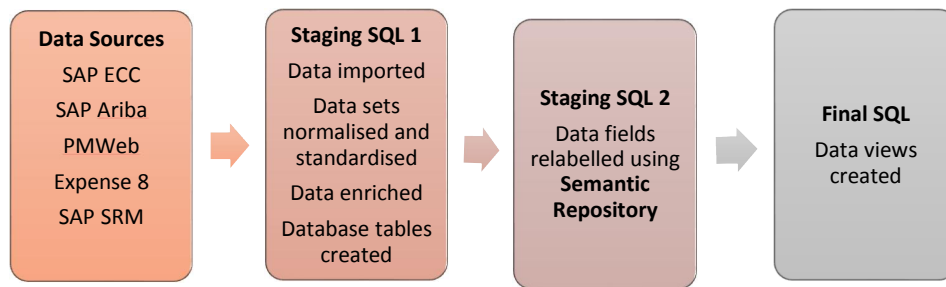


Figure 3. SQL Semantic Model-Detail

In detail, after confirmation of the technologies available to the analytics team, a thorough examination of the business processes to be covered by the semantic data repository was conducted. For each business group, we identified existing reporting requirements, key performance indicators and the source applications. The data analytics team sit outside of the core IT team responsible for general support of the source applications, therefore, direct access to the tables was not possible within the corporate security and governance model. In addition, multiple source applications were cloud-based products where direct database queries against database objects were not supported. The reporting functionality within each of the source applications was used to extract the required data in text or Excel formats so that it may be loaded into the SQL Server environment. SQL auto-processes were established to import the extracted report data into the SQL staging environment

For every report that is extracted from one of the source systems, there are two database tables in the staging environment: the first table to accept the raw imported data and the second table to accept a selection of the raw data that had an identical structure to the production object including unique constraints and naming conventions. The second table containing the new data is then used to update or insert into the production object where relationships with other SQL tables are defined, usually one-to-many relationships. Views are then developed to bring together data drawn from different reports and systems. The data is also enriched with third party data sets such as business registers and commodity coding schemas. The results from these views are written into the second staging area at which point column names are relabelled according to the data definitions in the semantic repository. The repository is manually maintained and includes three key attributes: the original label, the semantic table and the associated semantic meaning. After the semantic labelling is complete, the final views are created, forming baseline semantic layer for all data reporting and analytics from the SQL Server. These views are then linked to Tableau visualisation for visual analytics and data sets refreshed monthly.

More importantly, the SQL Server semantic solution allowed for semantic based development of SQL codes and the field names and labelling of information using business definitions and not technical ones which were directly inherited from the system. The process that was enabled also allowed for semantic based transformation and load of data and the process was repeatable and any changes to the data field definitions were updated using semantic master data field definitions table.

V. DISCUSSION

As mentioned in the generic case study, a semantic data model would provide for a well-defined business measure and support enhancement to the existing information model to capture new types of semantic information produced by inclusion of new or disparate data sources. A semantic data model will be simple to use and will not have the complexities of the information model and it will facilitate integration with resources outside the file store and support exporting and interactions with the business metadata.

Recent studies have extended the benefits of the semantic model to include data extract, transform and load processes which is highlighted in the case study. According to Rudra Pratap et al [20], in order to create better decisions for business analytics, organisations increasingly use structured, semi-structured, and unstructured data in addition to the (mostly structured) internal data. The current Extract-Transform-Load (ETL) tools are not suitable for this 'open world scenario' because they do not consider semantic issues in the integration processing. They argue that current ETL tools neither support processing semantic data nor create a semantic Data Warehouse (DW), a repository of semantically integrated data. The purpose of a semantic ETL in its final form is to embed meaning at each stage of the extract, transform and load process so that you get a data in the final stage with inherited semantic

meanings. The data model then is a semantic model by design and assists in business decision support as highlighted by Barbara von Halle and Larry Goldberg.

The advantages of the approach suggested by Rudra Pratap et al is that the model attempts to embed semantic layers at each stage of the ETL processes and these eventually will take time. Defining each stage of the ETL using semantic meaning making will be labour intensive which will prohibit agile development of solutions. However, the SQL Server semantic model that we developed embedded semantic meaning at the database table view, considering that the team was working in an agile environment. Data was extracted from multiple sources, combined using ETL processes in the SQL Server and auto processes developed that called on the semantic repository to enhance data field definitions before finalising the SQL data views. Another advantage of applying semantic layer before finalising the views was that the SQL views were ready to be plugged onto Tableau visualisation. Nevertheless, we should not dismiss semantic ETL programming because it is time consuming, the challenge is to develop all layers of semantic data transformations using agile methods because it encourages easy to understand data transformation processes, including detailed traceability of data definitions from source to target.

The agile semantic mission is taken up by Ralph Bergman and Yolanda Gil who goes beyond the embedding semantic layers in the ETL processes and focuses on the agile semantic workflows and “the need to enable non-IT staff to create workflows and to control their execution” [28]. Semantic workflows are currently not central to the discussion on the semantic model because the focus is on semantic outputs: easy data interpretation and decision support. However, similar to the discussions on semantic ETL, semantic workflows are important to the overall success of any semantic model. For Bergman and Gil, the semantic workflows enrich data workflows by adding semantic metadata that “captures the semantics of tasks, data items, and control flow items” [21]. The approach of Bergman and Gil is most robust but it requires complex algorithms to capture sometimes complicated system workflows that go beyond sequential data retrievals and labelling. In support of simplicity, we utilised Christian Fürber’s model develop the SQL Server semantic model because it was easy to adopt and enabled semantic tagging of data fields and content discovery post ETL.

VI. CONCLUSIONS

A semantic model assisted the organisation to leverage existing business glossary and available reporting and analytics tools much more efficiently and effectively because it provided the semantic infrastructure for information management. The team utilised Fürber’s method to develop an SQL Semantic Model that provided the capability for search of meanings of data fields and allowed developers to code semantic information as part of the coding process. Once the semantic model was in production, Tableau was used to deliver visual reporting and analysis to the business.

REFERENCES

- [1] R. Batra, "A History of SQL and Relational Databases," in *SQL Primer: An Accelerated Introduction to Basic SQL*.: Apress, 2018, pp. 183-187.
- [2] Terry Halpin and Tony Morgan, *Information Modeling and Relational Databases*. Burlington: Elsevier, 2008.
- [3] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377-387, 1970. <https://doi.org/10.1145/362384.362685>.
- [4] Peter Pin-Shan Chen, "The entity-relationship model—toward a unified view of data," *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9-36, 1975. <https://doi.org/10.1145/320434.320440>.
- [5] Frank Manola and Umeshwar Dayal, "PDM: an object-oriented data model," in *OODS '86 Proceedings on the 1986 international workshop on Object-oriented database systems*, California, USA, 1986, pp. 18-25.
- [6] E. F. Codd, *The Relational Model for Database Management: Version 2*. Redwood: Addison-Wesley, 1990.
- [7] Carlos Coronel, Steven Morris, and Peter Rob, *Database Systems: Design, Implementation and Management, 11th ed*. Stamford: Cengage Learning, 2014.
- [8] John W. Satzinger, Robert B. Jackson, and Stephen D. Burd, *Systems Analysis and Design in a Changing World*. Boston: Course Technology, 2009.
- [9] Joan Peckham and Fred Maryanski, "Semantic data models," *ACM Computing Surveys*, vol. 20, no. 3, pp. 153-189, 1988. <https://doi.org/10.1145/62061.62062>.
- [10] von Halle B, "Data: asset or liability," *Database Programming and Design*, vol. 4, no. 7, pp. 13-15, 1991.
- [11] von Halle B and L. Goldberg, *The Decision Model: A Business Logic Framework*. Florida: Auerbach Publications, 2010.
- [12] Aleš Popovič, Ray Hackney, Pedro Simões Coelho, and Jurij Jakliča, "Towards business intelligence systems success: Effects of maturity and culture on analytical decision making," *Decision Support Systems*, vol. 54, no. 1, pp. 729-739, 2012. <https://doi.org/10.1016/j.dss.2012.08.017>.

- [13] M.Krysiński et al., "Semantic Data Sharing and Presentation in Integrated Knowledge System," in *Intelligent Tools for Building a Scientific Information Platform*. New York: Springer, 2013, pp. 67-83.
- [14] Stefan W. Knoll, Jordan Janeiro, Stephan G. Lukosch, and Gwendolyn L. Kolfshoten, "A Semantic Model for Adaptive Collaboration Support Systems," in *Semantic Models for Adaptive Interactive Systems*. New York: Springer, 2013, pp. 59-81.
- [15] Richard Y. Wang and Diane M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5-33, 1996. <https://doi.org/10.1080/07421222.1996.11518099>.
- [16] Daniel L. Moody and Graeme G. Shanks, "Improving the quality of data models: empirical validation of a quality management framework," *Journal Information Systems*, vol. 28, no. 6, pp. 619-650, 2003.
- [17] S. Bjeletich, "SQL Server 2007: Best Practices for Semantic Data Modelling for Performance and Scalability," 2008. [Online]. http://codesmartinc.com/codesmartinc/wp-content/uploads/2012/08/semantic_db_modeling.docx
- [18] C. Adams, "Lessons for Information System Development," in *Business Information Technology Management: Alternative and Adaptive Futures*. London: Macmillan, 2000, pp. 25-39.
- [19] C.Fürber, *Data Quality Management with Semantic Technologies*. Wiesbaden: Springer, 2016.
- [20] Rudra Pratap Deb Nath, Katja Hose, Torben Bach Pedersen, and Oscar Romero, "SETL: A programmable semantic extract-transform-load framework for semantic data warehouses," *Information Systems*, vol. 68, pp. 17-43, 2017. <https://doi.org/10.1016/j.is.2017.01.005>.
- [21] R. Bergmann and Y. Gil, "Similarity assessment and efficient retrieval of semantic workflows," *Information Systems*, vol. 40, pp. 115-127, 2014. <https://doi.org/10.1016/j.is.2012.07.005>.