Automating Data Mart Construction from Semi-Structured Data Sources

Michael Scriney[1], Suzanne McCarthy[1], Andrew McCarren[1], Paolo Cappellari[2] and Mark Roantree[1]

[1]*Insight Centre for Data Analytics, School of Computing, Dublin City University, Glasnevin, Dublin 9, Ireland*
[2]*City University of New York, New York, NY, USA*
Email: [michael.scriney, suzanne.mccarthy]@insight-centre.org, [andrew.mccarren, mark.roantree]@dcu.ie, paolo.cappellari@csi.cuny.edu

The global food and agricultural industry has a total market value of USD 8 trillion in 2016, and decision makers in the Agri sector require appropriate tools and up-to-date information to make predictions across a range of products and areas. Traditionally, these requirements are met with information processed into a data warehouse and data marts constructed for analyses. Increasingly however, data is coming from outside the enterprise and often in unprocessed forms. As these sources are outside the control of companies, they are prone to change and new sources may appear. In these cases, the process of accommodating these sources can be costly and very time consuming. To automate this process, what is required is a sufficiently robust Extract-Transform-Load (ETL) process; external sources are mapped to some form of ontology, and an integration process to merge the specific data sources. In this paper, we present an approach to automating the integration of data sources in an Agri environment, where new sources are examined before an attempt to merge them with existing data marts. Our validation uses three separate case studies of real world data to demonstrate the robustness of our approach and the efficiency of materialising data marts

Keywords: Data Model Transformation, Semi structured data, ETL, Data Marts

Received 26 September 2017; revised 00 Month 2009

1. INTRODUCTION

The Agri industry is increasingly making use of data mining and analytics for decision support, predictions and preparing reports. The sector regularly has requirements to analyse trends in pricing of a specific Agri product or monitor production of various products. Increasingly, this requires integrating data from a web-based sources with their own internal databases in order to view them in a single homogeneous view for querying and analyses. The end goal is to provide Agri decision makers with a degree of certainty in their reports, rather than relying on instinct.

Apart from enterprise data, there is a significant volume of both online and third party datasets upon which to base predictions. However, correct usage can be difficult for several reasons: (i) the data is coming from multiple sources and there may be heterogeneity in the metadata structure or in the data itself; (ii) the data is often semi-structured making it unsuitable for loading to a Data Warehouse without significant manual effort on the part of data engineers; and (iii) integrating datasets often requires a domain expert to provide the context and data understanding. Solutions involve processes for cleaning, restructuring and integrating data, which are generally both expensive and time-consuming. One of the primary aims in this area of research is to reduce both cost and the time taken to make data available by automating these tasks as much as possible, generally through the help of domain experts and ontology construction.

1.1. Background and Motivation

Integrating semi-structured data sources is not new as research into integrating semi-structured data with enterprise data has been ongoing for almost 20 years eg. [1] and integration of fully unstructured sources in [2]. However, there is very little research into auto-constructing the types of multidimensional schemas needed for OLAP and data mining [3]. Traditionally, the process for constructing a warehouse (or multidimensional) schema involves matching business requirements with existing enterprise databases to specify the data marts which will generate the required datasets. An Extract-Transform-Load process [3] will be put in place to continuously harvest data from operational databases. In previous work, we used the same process to create a large Agri Warehouse model [4] although we integrated non-enterprise sources to meet specific business needs. As part of this work, a multidimensional data model was specified to map online sources to the various data marts represented in the Agri warehouse.

The problem with online Agri (in addition to many other) data sources is that they evolve much more than in-house databases. Sources can disappear or change structure, new sources can become available and this can have a widespread effect as different user needs require the construction of separate data marts. Approaches to clustering online data such as [5] can be effective when query changes require a modification to the materialised data mart but cannot process or integrate new online sources. In effect, a warehouse setup that comprises online data sources requires a methodology for analysing unknown or altered data streams to detect facts and dimension hierarchies with an ability to create data marts from multiple data sources.

1.2. Approach and Contribution

While integrating data from heterogeneous sources is often complex and domain-dependent, the data integration issues are well understood [6]. More recently, XML mapping technologies have been studied in terms of processing and transforming XML data [7] and XML sensor streams [8]. While the enhancement of data warehouses with web generated XML data was presented in [9], the schema design (facts and dimensions) was driven by enterprise databases. In this paper, we present a system which automatically creates multidimensional data marts from Agri data sources. Our approach provides a means of automatically detecting the attributes that make up a data mart from within each source stream, convert these sources to pure multidimensional graph mode and then, using an ontology, attempt to construct the required data mart by integrating the source data. This provides a significant benefit to Agri knowledge workers as it greatly reduces the time to construct data marts and can also be used to evolve data marts where the structure of source data streams has been modified.

Our previous work [10] introduced the *StarGraph* model which captures the multidimensional concepts of *facts* and *dimensions* from the online data streams.

The contribution of the work presented in this paper is as follows:

- As the StarGraph has a 1-1 relationship with external sources, a multi-source version is required to construct data marts from multiple sources. In this work, we present the*ConstellationGraph* (or *Constellation* for short) which represents integrated StarGraphs.

- We also present a methodology to automatically integrate (as far as possible) StarGraphs to complete the process of forming an integrated data mart from online data sources.

- Finally, we present an evaluation process which takes three user requirements - data marts for predicting pig prices, comparing pricing trends across Agri food products, and milk production analysis - and attempts to construct the data marts from data sources selected by our end-user partners. We run three exercises as part of the validation: firstly, the end user manually constructs the data mart; the second exercise sees the integration process take place without the use of an ontology so that we can classify exactly where semantic integration issues occur; and finally, our enriched ontology is used to drive the auto-construction of data marts which match those constructed by the end-user.

1.3. Paper Structure

This paper is structured as follows: in Section 2, we provide a discussion on the state of the art; in Section 3, we describe our StarGraph model and the methodology for creating data marts from online sources; in Section 4, we present the metadata and ontology parts to our system; as our case studies involve real end-user requirements from data streams that are currently in use, we present a brief overview of these data sources in Section 5 and a report on how they were integrated into our system; in Section 6, we provide an extensive evaluation using three user-defined case studies; and finally in Section 7, we present our conclusions.

2. RELATED RESEARCH

In [11], the authors present an ontology-based approach to constructing a data warehouse. Two different graphs are used: the datastore graph and the ontology graph. These graphs are linked by a series of formally defined mappings provided by a designer. This approach ensures semantic completeness between the datastore and the ontology. Similar to our StarGraph, the canonical model is a graph model. However, the StarGraph performs graph restructuring which attempts to define facts and dimensions *before* an ontology is required.

In [12], the authors present a system which utilises a global ontology to facilitate the ETL process and user querying, supplemented by metadata. The authors demonstrate how the system can be used to enrich user queries by providing an ad-hoc approximation search based on the global ontology. This ad-hoc querying mechanism allows for in-depth analysis which was not previously defined by a query designer. However, there is no focus on the larger ETL architecture while we provide a full methodology for importing new data sources.

An ETL process for OLAP on Linked Data is presented in [13]. This work introduces HSPOOL, a framework which provides a user with the means to perform OLAP analysis on Linked Data by extracting facts and measures and dimension hierarchies. However, a suitable OLAP schema must be constructed first in order to determine hierarchies from an ontology. In addition, while the authors use Linked data to construct the OLAP cube, we instead use an ontology to supplement the integration process for a Constellation Graph.

In [14], the authors present a unified cube, a combination of data obtained from a data warehouse and open linked data. Similar to our approach, a common data model is used to represent all data, which is then grouped into the unified cube. However, the integration process for linking cubes is controlled by a query designer, while our approach is more automated, using a global ontology to facilitate integration.

The authors in [15] present an RDF-based ETL process for Agri data. Their system utilises RDF graphs created from each Agri source and integrate by using queries with an end goal of creating a multidimensional schema. However, while we share a similar domain to the authors, our data sources are different. The authors use statistical open Agri data, presented as a series of flat spreadsheets, while we offer more flexibility by using web data which can come in a variety of data formats.

In [16], the authors present an automatic ETL system which uses an ontology to facilitate semantic data integration. The authors use a combination of a thesaurus derived from a starting data warehouse schema and a lexicon (e.g. WordNET) and clustering to determine similarity between attributes. However, in our approach, we use abstract types defined by our meta model to determine candidates for integration and propose a suitable integration strategy.

The authors in [17] present a semi automatic approach for combining business requirements and data sources to create a multidimensional schema and a corresponding conceptual ETL process. All data sources are captured in an OWL ontology with corresponding mappings while the business requirements are stored as a series of structured XML files. When the system is presented with a new business requirement, the requirement is first validated, then compared to the ontology of data sources to generate a multidimensional schema and ETL process. Similar to our approach, an ontology is used to facilitate semantic integration. However, the authors require an ontology detailing the mappings for each data source, while our process automatically generates mappings during the StarGraph creation phase.

In [18], the authors present a method for creating a data warehouse from heterogeneous data. Each data source is provided with a corresponding ontology, which are subsequently connected through a global domain ontology. The system uses structured requirements to extract suitable DW schemas from the global ontology based on requirements provided by a user. However, the ontologies for each data source identify the mappings and instance data provided is stored in a relational format. Instead, our approach adopts a data lake approach which stores instance data in its raw format and thus, provides us with a means of quickly detecting changes to the structure of schemas.

The authors in [19] present a system which constructs a data warehouse based on user requirements. The first step in this approach is to present the system with a domain ontology. The system parses this ontology to derive facts. It then constructs facts based on each concept detailed in the ontology. These facts are then presented to the user and the user selects the facts they wish to use. Similar to the authors' approach, we attempt to identify facts, dimensions and measures from a data source prior to user interaction. However, while the authors' approach requires a domain ontology, our approach identifies facts, measures and dimensions per source automatically.

In [20], the authors present an automatic ETL approach facilitated by domain specific modelling, where they use their own language (DSL) to represent different stages of the ETL process. A user (typically a domain expert), using DSL, outlines the concepts and operations for the conceptual ETL process. This process is then automatically deployed as a domain-specific ETL process. The authors use Domain Modelling to encapsulate the data sources and their interactions while our approach utilises an ontology to facilitate this semantic integration. Our approach extends this work by identifying facts, dimensions and measures automatically, requiring limited human intervention when integrating sources to a data mart.

The authors in [21] present an ETL process which utilises RDF/OWL ontologies to create OLAP data cubes from heterogeneous data sources. Each data source is provided with a corresponding ontology which is used to create RDF representations of the source data. The data is then extracted using RDF queries obtained from a cube definition represented as OWL. The authors use RDF queries to extract the data from each source, whereas our approach uses the data source's native format (XPath, JSON etc.) to extract the data. Additionally, our system automatically generates these mappings without the need for a pre-defined ontology bound to a data source as our ontology is used purely for assisting only in some integration scenarios.

3. BUILDING INTEGRATED DATA MARTS

Let us start by presenting a high level overview of our step-by-step approach to analysing and transforming data streams before integration into existing data marts or the creation of new ones. Throughout the rest of this paper, our usage of the terms *data mart* and *cube* represent: a multidimensional construct with data originating from separate sources. Our approach uses a *StarGraph* to model each data source and a *Constellation* to model the integrated data mart. For the data transformation shown in figure 1, the methodology comprises four main processes to manage different aspects of the overall transformation.

- **Stream Introduction (P1).** This process adds a data source to the Data Lake.
- **StarGraph Construction (P2).** This process creates a StarGraph from a single data source.
- **Constellation Construction (P3).** This process creates a Constellation from two or more StarGraphs. The process has three sub-processes, `Term-mapping`, `Type-mapping` and `Integration`. Term-mapping is used to assign canonical terms to StarGraphs by consulting the ontology; Type-mapping is used to assign canonical types and analyse a StarGraph with respect to the metamodel; the `Integration` process combines two StarGraphs to produce a Constellation; or adds a StarGraph to an existing Constellation.
- **Materialise (P4).** This process populates a data mart (StarGraph or Constellation).

We now provide a description of our system in terms of workflow and the transformations that are necessary to deliver an integrated data mart from non-enterprise data sources.

3.1. Stream Introduction

The stream introduction process (P1) is used to add new data streams with a *Stream Service process* to update a data lake with new instances of each data source at their specified update interval. A user is necessary to provide the URL to the data stream, an update interval and (if available) a schema of the stream data. A Metabase is used to capture stream metadata and a tree representation of the schema is constructed and stored in the Metabase, together with the source (URL) and update interval. This structure will be used as the basis for constructing a StarGraph. When appropriate, the process `P1a Stream Service` will populate the data lake with a new instance of the data stream at each update interval. The system Metabase is described in detail in section 4.

3.2. StarGraph Construction

This process (P2) constructs a StarGraph from a single data source. A StarGraph is a tree-based construct, which is annotated to capture fact, measure and dimensional data detected in a data source (should this information exist). As part of the construction of the StarGraph, mappings between StarGraph and data source are stored in the Metabase.

While the StarGraph construct is described in detail in earlier work [10], we present an overview here to provide the reader with an understanding of how the multidimensional data mart is represented. Both a StarGraph and Constellation are comprised of a set of nodes N and a set of edges E. Each edge is a three-tuple $E = \langle X, Y, REL \rangle$ where: $X, Y \in N$; REL is a type denoting the relationship which exists between the nodes X and Y. The possible values for REL are:

- 1-1, denoting a one-to-one relationship.
- 1-m, denoting a one-to-many relationship.
- m-m, denoting a many-to-many relationship.

The relationship type is obtained from examining the cardinally between attributes such as `maxOccurs` and `minOccurs` for an XSD schema.

Each node $n \in N$ is a four-tuple node such that $n = (name, class, source, dType)$. $name$ is the name of the node, $dType$ is the datatype of the defined node. $source$ is an indicator of where the particular data item is to be found in the schema. This attribute can take the form of an XPath query (for XML/HTML data) or dot-notation for JSON data.

$class$ indicates the type of node. There are six possible types:

- `Dimension` marks a node which is the beginning of a dimension.
- `dimension_attribute` is a marker denoting that the node in question is an attribute of the parent dimension node.
- `container` indicates the node is an instance containing other nodes.
- `measure` indicates that this node is a measure.
- `key` indicates that this node is a primary key
- `key-ref` indicates that this node is a foreign key, and as such depends on a node which is classified as a primary key.

3.3. Constellation Construction

The *TermMapping* process resolves naming differences between data sources to prepare data sources for integration. The process uses a set of canonical terms stored in an ontology. For example, consider two nodes which denote the country `Ireland` where the first node has the name `IRL` and the second has the name `Rep. of Ireland`. The term $IRELAND$ and all of its synonyms are captured in our ontology in the collection of terms associated with the *GeoLocation* dimensional data.

The *TypeMapping* process overcomes semantic differences between StarGraphs. Consider two nodes, one named `IRELAND` and the other named `FRANCE` which

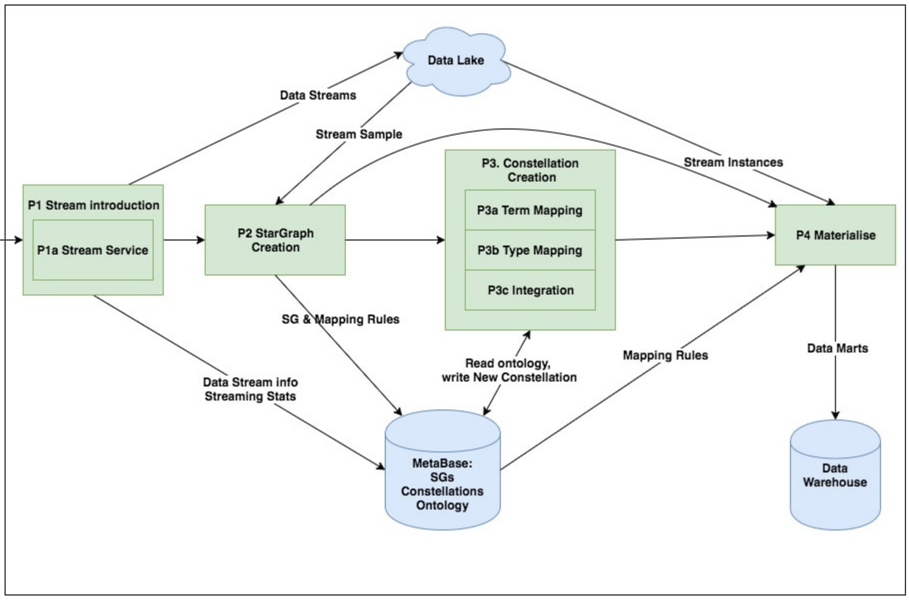

FIGURE 1: A StarGraph-based ETL System

both represent dimensional information. A simple comparison would determine that these two nodes are not equal. However, they are both classified as countries, and as such are members of the same `Geo` dimension. This process is also necessary to ensure compliance with the ontology's metamodel described later in section 4. Each detected measure is assigned the type `Value`. However, each measure must also be provided with the attributes `Item`, `Metric` and `Unit`. These types are assigned to a StarGraph during the type mapping phase. The *TypeMapping* process examines each node and can either apply a pre-defined type or a user defined type.

The *Integration* process begins with an analysis of the two StarGraphs selected for integration. The role of the ontology is described in section 4. We outline the process here.

- When evaluating two StarGraphs, the first attributes checked for are the `Date` and `Geo` attributes as at least one of these attributes must be present for integration to take place. If only the `Date` attribute is present, the StarGraphs will be merged on `Date`. If `Geo` is present, they will be merged on their respective `Geo` attributes. However, if both attributes are present, they will be merged on both, combining rows where their values match. Even if neither attribute is found, integration can take place if the user provides location details for these attributes.

- The next stage is to examine the Item, Metric and Unit attributes for the measures found. If all three values are the same for both products then both sources can be directly integrated without the need for transformation functions. If the Item attribute is identical for both but the Metric and Unit values are different, then the measures will be joined based on sharing the same product. Finally if the Item, Metric and Unit attributes are different, they are joined on the `Date` and `Geo` dimensions which have been previously evaluated.

There are two possible integration strategies. The first is the `row-append` strategy. This strategy is used when two data sources present semantically identical information, and as such can be directly integrated. The second approach is the `column-append` strategy. This strategy is used when data must be integrated on a number of attributes. The column-append strategy seeks out common dimension values between data from two sources and joins them based on these values.

The fallback approach for the `column-append` strategy is to join data simply based on the values present in the `Date` and `Geo` dimensions.

When two StarGraphs are set to be integrated, the presence of both a Date and Geo dimension are examined. These dimensions will be used to join data from two independent sources. If both a Date and Geo dimension are present, both will be used to join the data.

An example of missing data for these attributes can be seen in the Bord Bia data [22]. Where there is no specified Geo dimension, it assumes a user knows that all information provided is from Ireland and as such, the value for the Geo dimension is `'Ireland'` for all values.

4. METADATA AND STORAGE

As depicted in Fig. 1, our approach leverages three repositories: the Data Lake, the Metabase, and the

TABLE 1: Sources used in the Pig Prediction Data Mart

Name	ID	Format	Nodes	Edges	Dims	Measures	Materialize (ms)	Ins.
Aimis_1	aim_1	HTML	12	12	10	32	27	30
Aimis_2	aim_2	HTML	12	12	10	7	32	30
Bord Bia_1	b_1	HTML	2	2	1	1	70	Varies
Bord Bia_2	b_2	HTML	17	13	5	7	70	1-4
AHDB_1	p_1	HTML	88	88	11	8	122	8
AHDB_2	p_2	HTML	2	2	1	1	91	42
AHDB_3	p_3	HTML	4	2	3	1	150	156
AHDB_Bpex_1	bp_1	CSV	5	4	2	3	320	4
AHDB_Bpex_2	bp_2	CSV	5	4	2	3	320	3
cme_1	c_1	HTML	8	8	1	6	128	8
cme_2	c_2	HTML	8	8	1	6	131	12
imf	imf	XML	6	3	2	1	256	1122
usda	usda	CSV	39	38	33	6	166	84

Data Warehouse. The Data Lake is where raw data retrieved from the data sources is stored. The Metabase contains descriptive and administrative information regarding the data sources, the raw data, their mapping to the respective StarGraphs and Constellations, and the rules to automate the data integration. The Data Warehouse holds the materialisation of the data cubes resulting from our automated integration process. The rest of this section details these three repositories.

4.1. Data Lake

The Data Lake is the repository where raw data retrieved from the data sources is staged before being analysed and integrated. Raw data is retrieved (independently) from each known source at specified intervals, and "appended" to existing data already collected from the same source. Data is stored in its native format in simple flat structures, e.g. JSON, XML, or CSV. The Data Lake does not enforce a rigid structure on data pulled from the same data source: as the data structure from a source evolves, the data lake approach offers us the flexibility to seamlessly accommodate any change without having to undergo a revision of our repository. In fact, data belonging to the same source may be stored in multiple files, where each file may differ in both structure and format. Information about the origin, format, and location of data is captured as metadata and maintained in the Metabase as described in the next section.

4.2. The Metabase

The Metabase contains descriptive and administrative information about the input data, our integration graphs and processes, and data marts.

4.2.1. Data Source and Cube Metadata

In our system, data sources and their instance data are described independently, so as to keep separate the origin of the data from the actual format that the raw data has when retrieved. Data sources and associated instances are described by the following two properties: *Data Source* and *Data Instance*.

DEFINITION 4.1 (Data Source). *A data source $DS = \langle U, N, W, I \rangle$ is a four element tuple where U is the unique identifier the system generates for each new data source, N is the mnemonic associated with the data source, W is the URL at which the raw data is available for retrieval from the remote data source, and I is the interval of time specifying the pace at which to pull data from the provided URL.*

The Data Source property characterises the source of the data so that our system knows when and from where to retrieve (new) data. The Data Instance property, on the other hand, characterises the actual data as it is stored in the Data Lake, so that the system knows how old the last batch of retrieved data is and in what format it should be accessed locally. The Data Instance property is defined as follows:

DEFINITION 4.2 (Data Instance). *A data instance $DI = \langle I, U, D, T \rangle$ is a four element tuple where I is the unique identifier the system generate each time new data is retrieved from a specific source, U is the identifier of the source from which data is retrieved, D is the UTC date-time of data retrieval, and T is the format used to store the raw data into the data lake.*

Similar metadata is maintained for the data marts and cubes materialised in the Data Warehouse. Every time data from the input sources is integrated to update an existing cube, or to generate a new one, the system keeps track of the new data mart instance by the property *Cube Instance*, that is defined as follows:

DEFINITION 4.3 (Cube Instance). *A cube instance $CI = \langle C, D, L \rangle$ is a three element tuple where C is the unique identifier the system generates for each new data mart created, D is the UTC date-time of the cube creation/update, and L is number of times the cube has been loaded since its (first) creation.*

4.2.2. Integration Ontology

Information and processes on how to populate the data mart starting from the raw input data is described in the StarGraph, in the Constellation and in their associated mappings, as described in Sec. 3. All this information is stored in the Metabase. While StarGraph information can be automatically deduced given the input raw dataset, the Constellation and its mapping are based on an integration ontology, which is a separate component of the Metabase. This ontology is composed of two main parts: a metamodel for further describing each StarGraph from a semantic point of view, and a set of rules that, acting in a cascade fashion, automate the integration process.

Metamodel. The metamodel augments each attribute in the StarGraph with annotations that drive the integration process by helping to identify compatible attributes across different StarGraphs. We group these annotations under the property called *Attribute Semantic*. When attributes from two (or more) StarGraphs are semantically compatible, the StarGraphs can be integrated to form a Constellation. Instances of compatibility are described as mappings that are then executed to import actual data from the data lake into a data mart.

The Attribute Semantic property is described as follows.

DEFINITION 4.4 (Attribute Semantic). *An attribute semantic* $AS = \langle D, G, I, M, U, V \rangle$ *is a five element tuple where* D *is a Boolean describing whether the attribute represents the Date dimension,* G *is a Boolean describing if the attribute is a Geo dimension,* I *indicates the name of the measure in case the attribute is a measure of interest,* M *describes the metric for the measure,* U *specifies the units for the metric, and* V *is an actual value from the domain.*

In the Attribute Semantic property, the set of elements $\{D, G\}, and \{I, M, U\}$ are mutually exclusive because an attribute can either be a quantitative measure or a dimension (either date-time or geo-location). If the attribute is a measure, then all the elements $\{I, M, U\}$ must be specified in order to identify the name of the measure, the metric it is derived from, and the unit of reference for the attribute's values. These pieces of information are crucial to ascertain whether two dimensions or measures (e.g. metric) from different StarGraph are semantically compatible, and whether some transformation is needed in order to achieve homogeneous data in the cube (e.g. units for the metric).

Integration Rules. The metadata described in the Attribute Semantic is stored along with the *Integration Rules*. The latter are used to determine the integration strategy to construct the Constellation. Each rule describes an integration technique. Which rule to apply depends on the result of the semantic compatibility of the attributes in the StarGraph to integrate. Integration Rules are defined as follows:

DEFINITION 4.5 (Integration Rules). *An integration rule* $IR = \langle R, C, T, F \rangle$ *is a four element tuple where* R *is the identifier of the rule,* C *is the condition that determines whether the rule is applicable to the case under analysis,* T *and* F *are the the actions to execute in case the condition is satisfied or not, respectively.*

Actions T and F can be of three types:

- The first type is a rule cascade, where the action returns a set of further actions to trigger (via the action identifier R). This strategy allows us to have a set of cascading rule sets that can easily be reconfigured.
- The second type is the user-prompt action: when there is not enough metadata to understand how to integrate the StarGraphs, users are prompted to provide additional information.
- Finally, the third type is the integration strategy recommendation: the system can suggest which strategy is more effective. The integration strategy provides a recommended structure for the integrated data and tells the integration process specifically which attributes will be used to join the datasets.

4.3. Data Warehouse

The Data Warehouse stores populated data marts created from either a StarGraph or Constellation. When a data mart population process triggers, the relevant source data is obtained from the Data Lake and the facts, dimensions and measures required for a star schema are extracted using the mapping rules from the Metabase.

5. DATA SOURCE ANALYSIS AND PROCESSING

In this section, we present how our method fared with 120 unknown Agri data sources. We first briefly discuss the features and the type of data published in the considered data sources, then we analyse how these are data sources are imported into our StarGraph and Constellation constructs. An examination of the results allowed us to create a set of classifiers on the benefit and usage of each data source.

5.1. Agri Datasets

Here, we briefly describe the data sources captured in one of the sample Constellations and, using a small subset of the ontology, explain how a simple normalisation process can create usable data marts. An approach advocated in [23] is to use lightweight dynamic semantics - which our system adopts - in order to import and integrate new source data. Table 1 in Section 3.3 provides details of how this process worked for the 13 Agri sources which comprise the data mart we describe here.

- **Agriculture and Agri-Food Canada** [24], in their Aimis webpage, publish weekly data reports on multiple measures from which we have extracted data on hog slaughters at packing plants in Canada (Aimis_1 and Aimis_2 in Table 1).
- **USDA** [25] provide the **Quickstats** API for easy access to several datasets such as production, price and slaughters. This data mart imports pig crop data from this source (usda in Table 1).
- **Bord Bia** [22] is the Irish food board, founded to promote sales of Irish food and horticulture as well as providing certification of Irish products. This data mart is extracting: pig prices (Bord Bia_1 in Table 1); dairy prices (also Bord Bia_1 in Table 1); and pig slaughters (Bord Bia_2 in Table 1).
- The **Agriculture and Horticulture Development Board** (AHDB)[26] is a statutory levy board who provide research and development programs and market information to farmers and businesses in the Agri sector. This data mart is extracting: annual per capita consumption (AHDB_1 in Table 1); annual pig slaughters (AHDB_2); pig prices (AHDB_3); and weekly pig slaughters (AHDB_Bpex_1 and AHDB_Bpex_2 in Table 1).
- **CME Group** [27] is a Designated Contract Market who conduct and publish economic research in addition to publishing quotes for various commodities. Live pig data is used in this data mart (cme_1 and cme_2 in Table 1).
- **International Monetary Fund** [28] publishes daily exchange rates based on the unit of account SDR (Special Drawing Rights) (imf in Table 1).

5.2. StarGraph Analysis

We attempted to import 120 unseen Agri-data sources and the construction of a StarGraph for each of them. However, it is important to note that, unlike traditional data streaming sources, where there is one StarGraph per source, multiple StarGraphs can be produced from a single data source. This could be the result (for example) of multiple `<table>` elements in the HTML document from where data is extracted. In this case, the 120 streaming sources provided 120 StarGraphs. However, not all sources are usable, or have dimensional data, or are in the format that we assume for data sources (a basic schema). By using 120 different data sources, there is enough empirical evidence for us to classify data sources in terms of their usability. The classifications are as follows:

- **Full.** This indicates the construction of a "perfect" StarGraph, with all attributes of the source being correctly captured and stored.
- **Partial.** This indicates that after StarGraph construction, not all metadata has been processed correctly but nevertheless, it still functions as a working data mart.
- **Descriptive.** This indicates that a dimensional structure was found within the StarGraph, but no measure could be identified. However, the dimensional structure can be used in integration to enhance another StarGraph.
- **Missing.** These were constructed StarGraphs where dimensions were detected, but no facts could be identified and the dimensions found were not of sufficient depth to prove useful for integration.
- **Unusable.** In this case, the data source did not allow a StarGraph to be constructed.

From table 2, we can see that of the 120 potential sources, our approach automatically imported and integrated 84 (70%) usable Agri sources of which 41 (34%) were imported with 100% accuracy. Each column illustrates how a classification was achieved: `Facts` indicates that one or more facts were identified; `Dimensions` indicates that one or more dimensions were found; `Holistic` confirms that no attributes were discarded during the creation of the StarGraph; `Integrated` indicates that the StarGraph can be used in a subsequent integration by providing dimension hierarchies and more details to a `usable` StarGraph; `Non-Spurious` indicates that data is the constructed StarGraph is valid and usable; and finally, `Count` denotes exactly the number of constructed StarGraphs for each classification. Note that the classification `Unusable` means that a StarGraph cannot be constructed while the `Non-Spurious` metric being false means that a StarGraph was constructed, but the data was usable.

The sources which were classified as *unusable* were mainly due to the format of the source data (e.g. a PDF) and for others, it was due to issues with the file structure (e.g. a file having a .CSV extension, but the content was not a CSV file). Some sources could be correctly parsed yet they still proved unusable. All of these sources were HTML documents which displayed data in a tabular fashion, but this was accomplished using CSS, while the underlying HTML failed to use tags in the correct fashion (e.g. using `<div>` elements to represent a table). Additionally, some sources which were classified as *unusable* were, in fact, fully constructed data cubes. However, creating a StarGraph

TABLE 2: Results of Analysing 120 Agri Data Sources

Classification	Facts	Dimensions	Holistic	Integrated	Non-Spurious	Count	Sources(%)
Full	✓	✓	✓	✓	✓	41	34%
Partial	✓	✓	✓	✓	✗	29	24%
Descriptive	✗	✓	✓	✓	✗	14	12%
Missing	✗	✓	✗	✗	✓	2	1%
Unusable	✗	✗	✗	✗	✗	34	29%

from fully constructed data cubes is not currently possible using our mapping process.

The two sources classified as *Missing*, found dimensions but these proved to be unusable without context. In both instances, the only dimension found was a date dimension, and in the absence of a fact, measure or other dimensions to relate to, the constructed StarGraphs were of little use.

For those classified as *Descriptive*, no facts were found in the source data. However, a large degree of dimensions and hierarchies were discovered. These can be integrated with other StarGraphs in order that their dimensional data is reusable.

For those sources classified as *Partial*, StarGraphs were constructed but were unusable. An example of a *Partial* StarGraph is where a data source uses `<table>` elements to model both tabular data and to dictate layout of content on a webpage. When elements are misused in this manner, it is difficult for our system to determine whether or not the data located inside the `<table>` is dimension or fact values, or neither. In all of these cases, user interaction can be used to convert these StarGraphs to *Full* StarGraphs. In effect, this means that 58% of the unknown Agri sources are usable as data marts with a further 12% ready for integration into existing data marts.

6. EVALUATION AND DISCUSSION

For a robust evaluation of our methodology, we engaged with industry partners who specified three business requirements, each necessitating the construction of a new data mart. For each case, we perform three types of integration: user defined, non-assisted, and ontology assisted. In the user-defined integration, the data marts were created manually by editing the structure of a StarGraph and forcing the generation of mappings. This provided our ground truth version for each data mart. In the non-assisted integration, we tested the automated construction of data marts with a limited usage of our ontology to understand where issues with automatic *integration* of data lie. Here, the ontology was used to normalise terms before StarGraph construction but was not involved in the integration process. This was to evaluate the degree to which a fully automatic process could be used for integration and the degree to which an ontology is required. This is useful for domains for which no ontology is readily available. In the ontology-assisted integration, we tested the automated construction of data marts but with a semi-automatic integration approach utilising the ontology to facilitate semantic integration. This approach is fully automated for those streams classified as *Full* in our earlier analysis, but require some level of user intervention for streams classified as *Partial*. In these cases, the appropriate semantic attributes defined in our metamodel could not be detected.

For the remainder of this section, we will present our evaluation using three separate case studies for each of our partner's business requirements.

6.1. Case Study 1: Predicting Pig Prices

This first case study analyses pig market trends and prices in order to predict future pricing. This case study uses all 13 data sources outlined in Table 1 to construct a data mart to be used to predict the price of pigs on the global market. It contains the number of pig slaughters and prices per date and location. The dataset `imf` is required in order to resolve different currencies. Of the 13 data sources specified by the end user, 8 sources were classified as `Full` while the remaining 5 were classified as `Partial`.

User Defined Integration. The manual data mart was created by editing the mapping files of each generated StarGraph to influence the integration process. Four measures were identified: the number of pigs slaughtered `slaughter`, the price of pigs `price`, the milk futures quotes `milk-future` and the corn futures `corn-future` quotes. In addition there were three main dimensions, `Date`, `Geo` and `Currency`.

The dimensions `Date` and `Geo` are dimensional hierarchies containing various levels of granularity. For the `Date` dimension, the data sources provided were either monthly or weekly. For the `Geo` dimension, the hierarchies were based on area. For example, the `USDA` source provided a breakdown of slaughtering by state, while the `Bord Bia` source lists the number of slaughters as a whole.

A high-level overview of the Constellation can be seen in Fig 2, with details of dimensions and facts and the links between them. Solid nodes indicate a dimension while dotted nodes indicate a fact. Solid edges detail the links between dimensions and facts, while dotted edges indicate hierarchical relationships within a dimension.

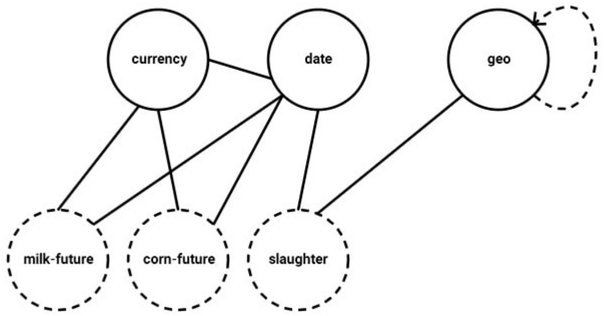

FIGURE 2: User defined Strategy (Case Study 1)

Non-Assisted Integration. The integration process is outlined in Table 3 and contains all the three approaches. `Source_1` and `Source_2` indicate the sources used at different stages of the integration process and for `Source_1`, the value `Cons` represents the current Constellation. `Issues` highlights problems which appeared during the non-assisted integration and `OntologyAssist` indicates the rule or process utilised by the ontology to resolve the issue. The `User Supplied` column indicates the metamodel values a user was required to supply in order for the integration process to complete. For example, `GEO` indicates that the user was prompted to provide a value for the GEO attribute, while `NONE` indicates that no user intervention was required.

The initial Constellation (Step 1) was created from the datasets `aimis_1` and `aimis_2`. From here, the integration process adds data sources (Table 1) in step-wise fashion to the Constellation. There was a high degree of integration at Step 1. This is because structurally `aim_1` and `aim_2` were very similar. However, granularity for integrated data proved a problem. Both of these sources contained a measure called `%` which denotes the percentage change between two dates. However, for one source the percentage meant the percentage change from the previous *week*, and for the second source, it meant the percentage change for a *year*. Despite these both being correctly identified as measures, the ontology is needed to resolve these issues of granularity.

Step 2 integrated the `b_1` data. Here, there was a single node to be integrated with the previous data, based on the `date` dimension. However, as there was no matching node in the Constellation for the measure, it was not combined with an existing one and instead occupies its own column in the fact table. This measure should have integrated with the previous two sources in addition to the Date dimension as they all relate to the sales and production of pigs. However, without a form of abstraction (supplied by an ontology) to semantically link the two measures they remain separate. Step 3 included the StarGraph created from `b_2` into the Constellation. Similar to `b_1`, this source was integrated based on the `date` dimension as no suitable candidate was found for measure integration. In other words, the dimensional hierarchy was enriched but no new facts were added.

Step 4 integrated `p_1` into the Constellation. Once again, a matching `date` candidate was found which saw a large reduction in the graph (as this source is largely time-series data). However, other dimensions which failed to integrate were country identifiers (e.g. "Austria"). Again, an extension to the ontology to indicate a type hierarchy would see a large degree of integration produced (and subsequently a lower number of nodes & edges). As this data was initially modelled as a matrix, a large number of rows were produced in the fact table associated with the measure which could not be integrated with the existing Constellation. Steps 5 & 6 integrates `p_2` and `p_3`. These sources were simple tables matching years to a measure. As such the data was integrated based on the date dimension and the measure was added to the fact table.

Step 7 integrated `bp_1` with the Graph partially integrated on the countries listed in `p_1` as one dimension specified a country. As expected, without the ontology, which contains a full dimensional hierarchy, some countries failed to be integrated due to differing tags (e.g."Great Britain" and "Northern Ireland" combined would be synonymous with the tag "United Kingdom"). Step 8 integrated the `bp_2` data source. This data source was identical in structure to `bp_1`. As such, there was a 1-1 integration between this source and the Constellation with all measures additionally being merged with those provided by `bp_1`.

Step 9 integrated `c_1` with date providing the only common attribute for integration. This is due to the fact that the data source `c_1` refers to future prices. However, a large number of measures were found within this data source, and as such have been added to the fact data joined on the date dimension. Step 10 integrated `c_2` and, similar to Step 9, the structural similarity between `c_1` and `c_2` facilitated a 1-1 mapping between all nodes. Step 11 sought to integrate currency conversion data from the `imf` data source. The data was successfully integrated on the date dimension, while the new currencies occupied new dimensions and the rates were included as measures within the fact table. Finally, Step 12 integrated the `usda` data, once again using the date dimension. The data was highly dimensional, adding in 30 previously unseen dimensions and 8 measures.

Ontology-Assisted Integration In this section, we describe how a close-to-automatic process for data mart constructed was achieved. However, at various points in the process, it was necessary for the user to update the ontology (through a system prompt) so that integration could complete and to ensure a fully automated integration for the same sources in future data marts.

For the initial Constellation, both a Date and Geo dimension were found for both data sources. Two measures were found for each source, but there was no defined Item, Metric or Units attributes. As both of these sources were the same structurally and semantically, they were fully integrated. However, the process prompted the user for input in both cases.

The next source b_1, was very sparse, containing only two attributes: date and measure. In this instance, the integration processes stopped again to prompt a user for input for the dimensions Geo and the attributes Item, Metric and Units, as the ontology again could not provide the precise level of detail. Once supplied, integration was completed using the Date and Geo dimensions, and along the measure by the item dimension. This is because both measures are of the product Pig but have different metric and unit attributes.

The next source - b_2 - was again missing a Geo dimension and Item, Metric and Units attributes. Once provided, the system proceeded to integrate this source on the Date and Geo dimensions, and integrated the measure with the aimis_1 and aimis_2 sources, as they were identical. The source p_1 found Date and Geo dimensions, but failed to find the attributes Item, Metric. However a Units attribute was found (kg per head). The source p_2 found a Date and Units attributes, but failed to find a Geo and Item attributes. Once again, a user prompt updated the ontology and integration was completed.

The source p_3 found a Date attribute and two Unit attributes. However, there was no Metric or Geo dimension. The source bp_1 found all required attributes except an Item and a Unit attribute. Once provided by a user, it was integrated into an existing Metric and Unit dimension.

The source bp_2 was also missing the Item and Unit attributes and, after user prompting, integration with bp_1 was successful.

The source c_1 provided several new Metric attributes. However, a Unit was not listed and the Geo dimension was not found. c_2 was also missing Unit attributes and a Geo dimension. However, with a user prompt and ontology update, the integration was completed. The final integration was different as the data source provided quotes about corn, and every Item thus far referred to pigs. Thus, the system integrated on the Date and Geo dimensions, which was correct.

The source imf found a series of Item attributes and a Date attribute. However, it failed to find a Metric or Unit attribute for each measure. Once again, this data was of an entirely new domain, currency conversion rates, and as such was integrated on the Date and Geo dimensions, after the ontology was updated. The final source usda found all required attributes and was integrated automatically.

Summary. Case study 1 required the integration of 13 data sources to construct the data mart. In terms of the user-defined approach, not only do they have to have an in depth understanding of the data they are capturing, it was also necessary to manually design and create the Data Mart (e.g. construct the schema, add constraints, write queries to insert data). This raised the same issues as were seen during both non-assisted and ontology-assisted integration strategies, in terms of having to determine the correct grain in hierarchy dimensions and in detecting the correct attribute for integration. Manual integration was estimated to take between 8 and 10 hours for what was a reasonably complex data mart (13 separate sources), while the automated approach required 118ms and populated with a batch update in 1.1 seconds. The manual data mart had a faster population time of 0.9 seconds.

It is worth highlighting the reasons as to why manual construction takes such a long time as these are generally the same issues that are resolved during ontology-assisted integration. This process also highlights how the ontology is used to aid integration and how rules are updated when issues with the structure of data sources are uncovered.

Step 1 integrates two sources into a Constellation (Cons) and from there, proceeds to integrate more sources into this constellation. At a high level, there are three main issues. The first is GRAIN_MISMATCH, which occurs when two data source are of differing levels of granularity. For example, the Date dimension has one source that provides weekly data and the other providing monthly data. Similarly for the Geo dimension, some data sources indicate individual countries while others provide statistics globally. The approach is to create a separate fact for each level of granularity. Later, a ROLLUP operation can be employed to join facts.

The second issue is labelled generically as MISSING_ATTR. This indicates that the data source did not contain enough information to correctly (semantically) integrate facts. In short, it means that the data source did not contain all of the attributes specified in the metamodel. The approach in this instance is for the ontology to prompt a user for input on these values. The most common reason for this issue to occur is that the data is sparse in terms of dimensions and attributes. When this occurs, it is impossible to infer these values so the system defers to a user to provide the missing contextual information.

The final issue is labelled as TERM-TYPE_MISMATCH and revolves around two problems. The first being different terminologies used across data sources, particularly across the Geo dimension (for example 'US' vs 'America'). These issues are resolved by the ontology during the term mapping phase, where both possibilities are resolved into a single canonical term. The second issue arises from a lack of type information in both sources. This issue arises when attempting to

TABLE 3: Integration Issues for Case Study 1

Step	Source_1	Source_2	Join on	Issues	OntologyAssist	User Supplied
1	{aimis_1}	{aimis_2}	DATE, GEO	GRAIN_MISMATCH	GRAIN_CHECK	ITEM, METRIC, UNIT
2	Cons	{b_1}	DATE	MISSING_ATTR	METAMODEL_CHECK	GEO, ITEM, METRIC, UNIT
3	Cons	{b_2}	ALL	MISSING_ATTR	METAMODEL_CHECK	GEO, ITEM, METRIC, UNIT
4	Cons	{p_1}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	ITEM, METRIC
5	Cons	{p_2}	DATE	MISSING_ATTR	METAMODEL_CHECK	GEO, ITEM
6	Cons	{p_3}	DATE	MISSING_ATTR	METAMODEL_CHECK	GEO, METRIC
7	Cons	{bp_1}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	ITEM, UNIT
8	Cons	{bp_2}	ALL	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	ITEM, UNIT
9	Cons	{c_1}	DATE	MISSING_ATTR	METAMODEL_CHECK	GEO, UNIT
10	Cons	{c_2}	ALL	MISSING_ATTR	METAMODEL_CHECK	GEO, UNIT
11	Cons	{imf}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	METRIC, UNIT
12	Cons	{usda}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	NONE

resolve different attributes and dimensions which are of the same abstract type. For example one source may have a dimension called 'France' and another 'Australia'. Without an ontology linking these two concepts under the common theme 'Country', they cannot be integrated.

For this case study, term and type mapping resolved integration problems for the Geo and product dimensions, while the metamodel was used to enforce semantic integration by prompting a user for the `metric` and `units` attributes.

For the remaining two case studies, we will provide a more abbreviated discussion as the issues are identical across case studies. However, it is important to demonstrate the generic nature of our work and the wider applicability.

6.2. Case Study 2: Price Trend Comparison

This case study compares the trend in the price of butter with the price of vegetable oil. It requires the 9 sources shown in Table 4 to construct the appropriate data mart. Of the 9 sources, 8 were classified as `Full` StarGraphs and `GlobalDairyTrade` being classified as `Partial`. Data sources such as `PPOIL_USD` provide historical data up to the current week.

User Defined Integration and Non-Assisted integration. A manual design of a fact table satisfying this requirement is shown in Fig. 3. All of the data sources provided for this case study provide the same information, a product and a price at datetime t.

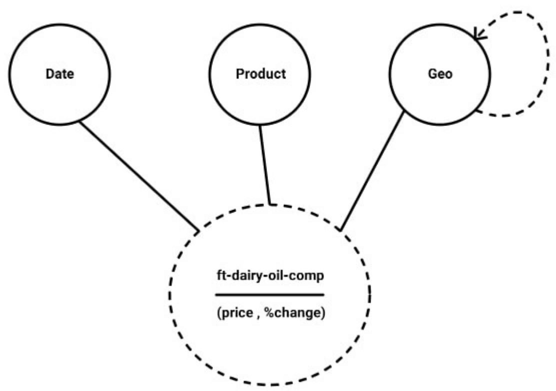

FIGURE 3: User defined Strategy (Case Study 2)

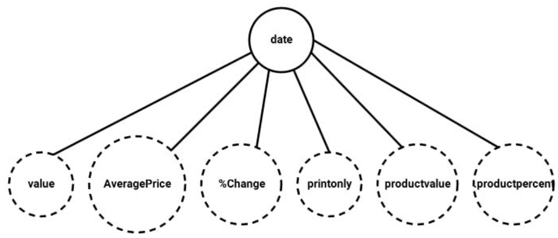

FIGURE 4: Non-assisted Strategy (Case Study 2)

TABLE 4: Sources for Case Study 2.

Name	ID	type	instances	Description
PPOIL_USD	pp	CSV	447	Historical Palm oil prices for Africa
PROIL_USD	pr	CSV	447	Historical Rapeseed oil prices for Africa
PSOIL_USD	ps	CSV	447	Historical Soy Bean oil prices for Africa
PSUNO_USD	psu	CSV	447	Sunflower oil prices for Africa
REN_SU	ren	CSV	143	Rennet Casein price (Milk production) from Global Dairy Trade
WLD_COCONUT_OIL	cn	CSV	687	Prices for World coconut oil production
WLD_PALM_OIL	po	CSV	687	Prices for world palm oil production
WLD_SOYBEAN_OIL	so	CSV	687	Prices for world soybean oil production
GlobalDairyTrade	gdt	HTML	24	Prices for dairy products

However, some attributes such as product name are taken from the name of the source (e.g. `PPOIL_USD` refers to the product Palm Oil in USD). The Non-Assisted Constellation is shown in Fig. 4 illustrating again the missing information (the grain in the geo dimension). As the issues involved in this case study were described in the previous case study, we provide a summary of the integration strategies in Table 5.

Ontology-assisted integration. Term mapping and type mapping were correctly able to identify all attributes necessary for the GTD data source. However, due to the sparse nature of the CSV data sources, user input was required to determine the Type, Metric and Units for these data sources. Once provided, all data sources shared a date dimension and all CSV sources shared the same metric and units dimensions

Comparison Table 5 details the issues found in the non-assisted integration, the ontology rules applied to mitigate these issues for the ontology-assisted integration, and a marker denoting whether user intervention was needed at a particular integration step for Case Study 2. Most of the sources for this data mart were structurally identical, containing only a Date and a measure. The non-assisted integration approach integrated on both attributes. This was the incorrect approach, as despite the fact that these sources are structurally identical, they are semantically different.

What was different in this case study, and was due to the sparse nature of sources, was the lack of information required to deliver proper (semantic) integration. These issues are overcome through a combination of the ontology (`TERM-TYPE_MAP`) and user intervention through the ontology's `METAMODEL_CHECK` function. The user supplies the necessary metamodel values for most of these sources and in general, these contained only a date and measure.

Summary. Case study 2 used nine data sources in the construction of the data mart. The user defined approach and the ontology-assisted approach suffered from the same problem: the sparseness of the majority of the data sets. However, as most of the datasets were identical in structure, there was a significant saving in development time when manually constructing the data mart. We estimate that this manual approach completed in about 5 hours and took 17 seconds to perform a population. A large amount of time was spent understanding the data due to its sparse nature. Conversely, the automatic approach had a time of 102ms and populated the data mart in 23 seconds.

6.3. Case Study 3: Analysing Milk Production for Major Producers

This case study examines year on year changes for milk production and milk deliveries and uses the 5 data sources outlined in Table 6. Two were classified as `Full` and three as `Partial` StarGraphs. For this case study, we again use a summary table (Table 7) to provide a brief overview of the different integration strategies and the issues that arose.

User Defined and Non-Assisted Integration. Similar to the first case study, this data mart requires use of information which is not visible to the StarGraph. For instance, terms with names such as `Kg/$` imply that this attribute is a measure; it is created from two units. Additionally, the knowledge a designer has such as `New Zealand, Germany == Country` allow the schema designer to create generic dimensions such as Type, Country and Units. The reason for the units dimension is that unless one explicitly identifies those units used for a specific measure, they cannot be directly compared.

Due to that fact that some sources were simply csv files with two attributes `date` and `value`, this provided a direct mapping for integration. One other source also provided a date dimension and as such the attributes named `butter`, for example, integrated on

TABLE 5: Integration Issues for Case Study 2

Step	Source_1	Source_2	Join on	Issues	OntologyAssist	User Supplied
1	{pp}	{pr}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR	TERM-TYPE_MAP, METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
2	Cons	{ps}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR	TERM-TYPE_MAP, METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
3	Cons	{psu}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR	TERM-TYPE_MAP, METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
4	Cons	{ren}	DATE	MISSING_ATTR	METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
5	Cons	{cn}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR, GRAIN_MISMATCH	TERM-TYPE_MAP, META-MODEL_CHECK, GRAIN_CHECK	GEO, METRIC, ITEM, UNITS
6	Cons	{po}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR	TERM-TYPE_MAP, METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
7	Cons	{so}	DATE, MEASURE	TERM-TYPE_MISMATCH, MISSING_ATTR	TERM-TYPE_MAP, METAMODEL_CHECK	GEO, METRIC, ITEM, UNITS
8	Cons	{gdt}	NONE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	GEO, METRIC, UNITS

TABLE 6: Sources for Case Study 3.

Name	ID	Type	Instances	Description
Argentina_Milk_Deliveries	`amd`	HTML	12	Deliveries of Cow's Milk in Argentina
Cows' milk collection and products obtained	`wdp`	CSV	18450	Production of world dairy products
USDA	`usda_2`	CSV	1231	US Dairy production
Milk Production Germany	`mpg`	HTML	4	Dairy Germany
NZ Milk Production	`nzmp`	HTML	12	Milk production for New Zealand

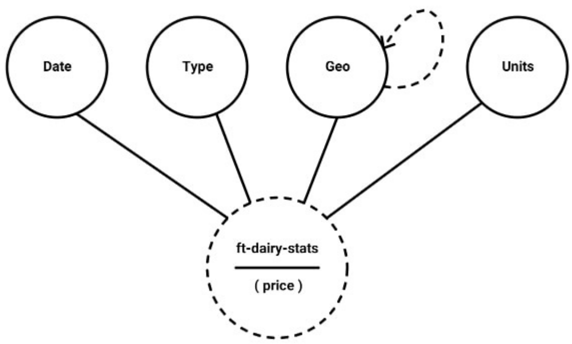

FIGURE 5: User defined Strategy (Case Study 3)

this dimension. However, for the remaining source, the only attribute to integrate on was the name of the measure. Despite the fact that it provided usable facts, the data content was incorrect in data mart usage. This required an entry in the ontology to *prevent* this aspect to the integration.

The differences between user-defined (Fig. 5) and non-assisted (Fig. 6) are primarily due to abstractions of which the user-defined approach has knowledge. In general, all facts present in the automatic approach are combined into a single fact entity. This is accomplished through the use of a `Type` dimension. However, without a suitable ontology to inform the automatic approach that these facts can be combined, they will remain separated.

Ontology assisted integration All attributes required for an integration approach were found within both csv files. However, there was no `geo` dimension found for the Argentina and NZ milk production tables. Finally, for the Milk production in Germany data, no `date` or `geo` dimensions were located. This required the user prompt for dimensions and an ontology update before the integration process could complete.

TABLE 7: Integration Issues for Case Study 3

Step	Source_1	Source_2	Join on	Issues	OntologyAssist	User Supplied
1	{amd}	{wdp}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	GEO, ITEM
2	Cons	{usda_2}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	NONE
3	Cons	{mpg}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	GEO
4	Cons	{nzmp}	DATE	TERM-TYPE_MISMATCH	TERM-TYPE_MAP	GEO, ITEM

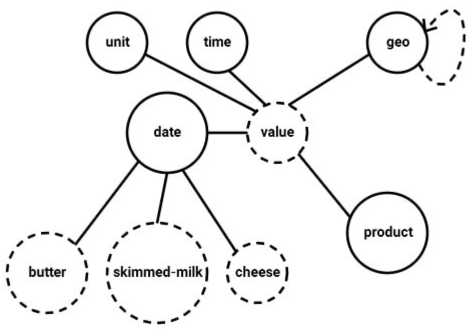

FIGURE 6: Non-assisted Strategy (Case Study 3)

Comparison. **Summary.** While case study 3 constructed a data mart from just 5 sources, the levels of missing metadata and heterogeneity in the data, resulted in the most difficult integration effort of all three case studies. Regarding the user-defined approach, in addition to the time taken to understand the data and design/implement a data mart, additional time was spent individually examining the markup specific to each HTML source so that the correct data could be extracted. We estimate this process took in the region of 10 hours while the automatic approach completed in approx. 110ms.

With regards to population time, the automatic approach was 11 seconds slower than the manual approach (74s to 63s). Table 7 outlines the issues found during the non-assisted approach, the ontology rule applied to overcome this issue, and whether or not user intervention was required at an integration step. For all sources the only issues found were those of term and type mapping. Once again, the ontology uses these to assign canonical terms to each data source, and provides a layer of abstraction between the data sources so that they can be semantically linked.

6.4. Overall Summary

The goal of our evaluation is to examine the differences between a manual integration approach and our automated approach, both to determine the value of our methodology and to identify potential improvements to our process. Table 8 outlines the results of all three use cases under both a manual and automatic approach. Column `Name` relates to the case study and the approach used: the columns `Sources` and `Instances` refer to the number of data sources involved in the integration, and the number of instances for each; `Metadata` refers to the number of attributes found in the multidimensional schema once integration has been completed; `Time` is the time taken to construct the final data mart; Structure relates to the usability of the data (yes/no); and finally `Semantics` refers to the correctness of data (yes/no).

The most important column, `Time`, illustrates the savings in time using our approach. The 3 manual approaches required between 5 and 10 hours approximately while the automated approach was between 3 and 11 minutes. In the earlier discussion on case studies, the automated time was reported as between 110ms and 7s. However, here we include 1 minute for each user prompt and response (60 seconds was the longest time recorded). This clearly shows the benefit of the automated approach.

In terms of Metadata, the manual approach detected and removed more redundancies than the automatically generated approach. This is due in a large part to the domain expert's knowledge of the dataset compared to the automatic approach. While the semi-automatic approach had similar results to the automatic approach, this is undoubtedly due to the semi-automatic approach requesting information from the user as needed. This results in marginally slower times for batch updates as recorded in the case study discussions. The degree of redundant data determines the difference in data loading times and is our current area of research in terms of improving the ontology.

7. CONCLUSIONS

There are many websites generating information that covers a wide range of activities in the Agri sector. When properly processed, synchronised and aggregated, these sources can provide vital input for Agri decision makers. The main issues are that these data streams come and go, are prone to change, and can be costly to process for many Agri sector workers.

TABLE 8: Analysis of all case study approaches

Name	Sources	Metadata	Instances	Time	Structure	Semantics
cs_1_user_defined	13	24	262	8hrs	✓	✓
cs_1_non_assisted	13	143	262	N/A	✓	✗
cs_1_ontology_assisted	13	24	262	11m	✓	✓
cs_2_user_defined	9	16	4016	5hrs	✓	✓
cs_2_non_assisted	9	28	4016	N/A	✓	✓
cs_2_ontology_assisted	9	16	4016	8m	✓	✓
cs_3_user_defined	5	17	19709	10hrs	✓	✓
cs_3_non_assisted	5	60	19709	N/A	✗	✗
cs_3_ontology_assisted	5	17	19709	3m	✓	✓

In earlier work [10], we presented a method to convert smart city web streams into a StarGraph construct to enable the construction of a *single* source data mart from unseen source data. In this paper, we extend this work by integrating StarGraphs to create multi-source data marts which we call Constellations. For our evaluation, we used 120 unseen Agri data sources to first determine how many sources were usable by our system. This analysis showed that 70% of the sources were successfully transformed to our StarGraph model. We then worked with industry partners who provided three different requirements and a list of data sources with which to create data marts. Our automated approach was shown to deliver considerable benefits, firstly by eliminating the need for manually constructing data marts (the smallest data mart required 5 hours in construction); requiring a minimal effort by the data mart designer in terms of fully understanding the separate sources (user prompts for ontology updates refer only to specific parts of the schema); and finally, new information learnt about data sources, is maintained in the ontology for future data mart construction.

FUNDING

This work is supported by Science Foundation Ireland under grant number [SFI/12/RC/2289]

REFERENCES

[1] Bergamaschi, S., Castano, S., and Vincini, M. (1999) Semantic integration of semistructured and structured data sources. *SIGMOD Record*, **28**, 54–59.

[2] Kittivoravitkul, S. and McBrien, P. (2005) Integrating unnormalised semi-structured data sources. *Proceedings of Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, Proceedings*, pp. 460–474. Springer-Verlag, Berlin.

[3] Inmon, W. H. (2002) *Building the Data Warehouse,3rd Edition*, 3rd edition. John Wiley & Sons, Inc., New York, NY, USA.

[4] McCarren, A., McCarthy, S., Sullivan, C. O., and Roantree, M. (2017) Anomaly detection in agri warehouse construction. *Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2017,Australia, January 31 - February 3, 2017*, pp. 17:1–17:10.

[5] Roantree, M. and Liu, J. (2014) A heuristic approach to selecting views for materialization. *Softw., Pract. Exper.*, **44**, 1157–1179.

[6] Batini, C., Lenzerini, M., and Navathe, S. B. (1986) A comparative analysis of methodologies for database schema integration. *ACM computing surveys (CSUR)*, **18**, 323–364.

[7] Roth, M., Hernández, M. A., Coulthard, P., Yan, L., Popa, L., Ho, H.-T., and Salter, C. (2006) Xml mapping technology: Making connections in an xml-centric world. *IBM Systems Journal*, **45**, 389–409.

[8] Roantree, M., Shi, J., Cappellari, P., O'Connor, M. F., Whelan, M., and Moyna, N. (2012) Data Transformation and Query Management in Personal Health Sensor Networks. *J. Network and Computer Applications*, **35**, 1191–1202.

[9] Martinez, J. M. P., Berlanga, R., Aramburu, M. J., and Pedersen, T. B. (2008) Integrating data warehouses with web data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, **20**, 940–955.

[10] Scriney, M., O'Connor, M. F., and Roantree, M. (2017) Generating cubes from smart city web data. *Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2017, Australia, January 31 - February 3, 2017*, pp. 49:1–49:8.

[11] Skoutas, D. and Simitsis, A. (2007) Ontology-based conceptual design of etl processes for both structured and semi-structured data. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **3**, 1–24.

[12] Priebe, T. and Pernul, G. (2003) Ontology-based integration of olap and information retrieval. *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pp. 610–614. IEEE.

[13] Komamizu, T., Komamizu, T., Amagasa, T., Amagasa, T., Kitagawa, H., and Kitagawa, H. (2016) H-spool: A sparql-based etl framework for olap over linked data with dimension hierarchy extraction. *International Journal of Web Information Systems*, **12**, 359–378.

[14] Ravat, F., Song, J., and Teste, O. (2016) Designing multidimensional cubes from warehoused data and linked open data. *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pp. 1–12. IEEE.

[15] Berro, A., Megdiche, I., and Teste, O. (2015) Graph-based ETL processes for warehousing statistical

open data. *ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April*, pp. 271–278. Spri.

[16] Bergamaschi, S., Guerra, F., Orsini, M., Sartori, C., and Vincini, M. (2011) A semantic approach to ETL technologies. *Data Knowl. Eng.*, **70**, 717–731.

[17] Romero, O., Simitsis, A., and Abelló, A. (2011) Gem: requirement-driven generation of etl and multidimensional conceptual designs. *International Conference on Data Warehousing and Knowledge Discovery*, pp. 80–95. Springer-Verlag, Berlin.

[18] Selma, K., IlyèS, B., Ladjel, B., Eric, S., Stéphane, J., and Michael, B. (2012) Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool. *Computers in Industry*, **63**, 799–812.

[19] Romero, O. and Abelló, A. (2010) A framework for multidimensional design of data warehouses from ontologies. *Data & Knowledge Engineering*, **69**, 1138–1157.

[20] Petrović, M., Vučković, M., Turajlić, N., Babarogić, S., Aničić, N., and Marjanović, Z. (2017) Automating etl processes using the domain-specific modeling approach. *Information Systems and e-Business Management*, **15**, 425–460.

[21] Niinimaki, M. and Niemi, T. (2010) An etl process for olap using rdf/owl ontologies. *Journal on Data Semantics XIII*, **5530**, 97.

[22] Bord Bia. http://www.bordbia.ie/Pages/Default.aspx.

[23] Barnaghi, P. M., Bermúdez-Edo, M., and Tönjes, R. (2015) Challenges for quality of data in smart cities. *J. Data and Information Quality*, **6**, 6:1–6:4.

[24] Agriculture and Agri-Food Canada. http://www.agr.gc.ca/eng/home/?id=1395690825741.

[25] USDA. https://quickstats.nass.usda.gov/.

[26] Agriculture and Horticulture Development Board. http://pork.ahdb.org.uk/.

[27] CME Group. https://www.cmegroup.com/.

[28] International Monetary Fund.[Online]. http://www.imf.org/external/index.htm.