

# Experimental Evaluation of a Safety Data Model to determine the Time Taken to Build, Merge and Publish XML File onto a Data Server

Ilyasu Anda<sup>1</sup>, Isah Omeiza Rabi<sup>2</sup>, Enesi Femi Aminu<sup>3</sup> & Hussaini Abubakar Zubairu<sup>4</sup>

<sup>1</sup>Department of Library and Information Technology, School of Information & Communication Technology,

<sup>2</sup>Department of Computer Engineering, School of Engineering & Engineering Technology,

<sup>3</sup>Department of Computer Science, School of Information & Communication Technology,

<sup>4</sup>Department of Information and Media Technology, School of Information & Communication Technology,

Federal University of Technology, Minna, Niger State Nigeria.

<sup>1</sup>Email: [ilyasu.anda@futminna.edu.ng](mailto:ilyasu.anda@futminna.edu.ng); <sup>2</sup>[isah.rabi@futminna.edu.ng](mailto:isah.rabi@futminna.edu.ng);

<sup>3</sup>[enesifa@futminna.edu.ng](mailto:enesifa@futminna.edu.ng); <sup>4</sup>[abu.zubairu@futminna.edu.ng](mailto:abu.zubairu@futminna.edu.ng)

---

## ABSTRACT

*Some datasets are prone to risk and hazards, inadvertently affecting the integrity of the data with attendant errors in result interpretation and usage which could sometime escalate to disastrous levels. Despite these potential errors in data leading to various mishaps, this part of the system has been ignored. This paper aims to experiment a model (Safety Data Model) to ensure safety of data used in the data analysis for decision making. It focuses on safety of data in a critical application, taking into consideration the integrity of the data, time taking to extract and publish the XML files to the data server. It, thus, represents the data in a more concise format that a consumer of such data can easily assess the sources and evaluate the integrity of the data before any decision-making. The research proposes an experimental evaluation of a safety data model that helps to prevent the possible mishaps. Twelve (12) Excel files of Safety Related Condition Reports (SRCR) data between 2002 and 2013 were used, which contain a total of 1039 rows of data. It took roughly 20.703 seconds to complete the Extract, Transform Load (ETL). Modern and sophisticated ETL software tools including Microsoft SQL Server 2012 Data Tools and Microsoft Structured Query Language (SQL) Server Management Studio were used in data manipulation. The prototype was able to filter data into safe, unsafe and hazardous data which were ready to be loaded into the Data Warehouse (DWs). The prototype was able to generate an XML document containing safe, unsafe and hazardous data. The prototype was proved to be effective because it was able to build XML Data within 0.484 seconds, merge and publish XML documents within 12.719 seconds. The purpose of this is to show the end users the actual data in order to justify whether the data is truly safe, truly unsafe or truly hazardous according to the quality summary in the metadata. The end user can also verify the whole data from the source if necessary and if the end user is satisfied with the quality of the data, then, the safe data can be extracted directly from the XML.*

**Keywords:** Safety; Big-Data; Data-Warehouse; Microsoft SQL Server; Metadata.

---

## African Journal of Computing & ICT Reference Format:

I. Anda, I. O. Rabi, E. F. Aminu & H. A. Zubairu (2018), Experimental Evaluation of a Safety Data Model to Determine the Time Taken to Build, Merge and Publish XML File onto a Data Server, *Afr. J. Comp. & ICT*, Vol.11, No.2, pp. 11 - 23.

## I. INTRODUCTION

In this era of globalization and the dynamic world of ICT, the importance of and crave for safe and error -free data cannot be overemphasized. This often posed a great challenge to the system analysts and to the end-users of these data who rely on it for serious decision making. One of the problems bedeviling data analyst in a safety critical environment is the unstructured and redundant nature of the data to be analyzed. More so, the time complexity in analyzing this type of data is always very high, consequently leading to data failure. Furthermore, some datasets are prone to risk and hazards, inadvertently affecting the integrity of the data with attendant errors in result interpretation and usage which can sometimes escalate into disasters and fatalities.

A safety-related system was developed with software and data often described as a real - world environment in which the system will function properly and play a dynamic role in ensuring accurate operations. Computer modelling with a very good engineering practice dictates that data is produced to the same integrity requirements as applicable in other similar system elements. Regrettably, existing knowledge suggests this is often not the case. This is because data processing has shifted from the role of operations support to becoming a major operation in itself, hence the need for quality management of data. However, many similarities exist between quality data model and quality data extraction.

Despite the potential errors in data processing as observed in most systems, leading to accidents or mishaps, the aspect of safety model has not been extensively explored; its application in any critical safety environment is very key in meeting end user objective in such a way that a consumer of the data can judge the integrity of the data and factor this into the decision-making aspect of their response with minimal error [1, 2, 10].

Data has always been of utmost importance in computer systems, but when it comes to safety related systems, data are used extensively in such applications in the form of constants and variables. The data in such systems can take a number of forms which are characteristics of the device, Constants for calibration, Data for plant configuration, Topological data etc.

Each of the above mentioned classes of data shares a characteristic that data is generated separately from the executable software. But the generation of such data happens to be a complicated procedure. For example, the data collected in describing a rail network or the surroundings of an airport always suffer from the scope of

errors and faults, and such faulty data poses severe risks in the operations of the safety related systems [3].

In order to ensure the safety of information systems, development methods involving risk-based approaches are adopted, where data related to potential risks is collected for the proposed system and then analyzed to keep identified risks at an acceptable level. One mechanism to ensure this is to stipulate development and assessment practices via Safety Integrity Levels (SIL). The level of rigor in the processes is commensurate with the safety risks associated with the system. The data collected for risks that help in defining a SIL affect the overall architecture of the proposed system a great deal. The SIL also affects the overall functioning of the system providing better safety solutions [3].

All computer-based systems, including the safety related systems require data for their operations. Those who use these systems require data for making decisions which are sometimes critical. Since data assume a great level of importance in such data driven safety related systems, carefully produced data elements leave great impacts on the overall decision-making process of the management. Decision makers generally have to face a number of challenges while analyzing the data when there are problems in the way data has been collected or generated, especially when the data happens to be safety critical [4]. Metadata is a type of data, which carries information about the data itself. The term metadata is also referred as data about data. A metadata provides information about various aspects of the data being used in the system. Some of these aspects are Format of the data, Origin of the data, Time and date of creation of data.

Qualitative attributes of data, Author of the data, Purpose for which the data was created. The metadata like any other data can be stored in a data storage system. The storage of metadata is often called as Metadata Repository [5].

The structure of the metadata needs to comply with various rules pertaining to the fields and elements of information contained in the metadata as defined in the standards. The metadata structure can be defined in a number of programming language syntaxes which provide uniformity in designing the metadata [6]. Usually the metadata is schematized by making use of markup languages like HTML or XML. XML being the industry standard these days is the most favorable mark-up language to structure metadata. Another very good example of structuring metadata is by making use of

Taxonomies which involve creating certain keywords for a particular type of data element. These key words can be applied to the data upon its creation signifying the required information for metadata [7].

For this reason, a complete data model that will accurately describe all possible relationships between files and document is essential. Such a model should also provide the foundation upon which critical data administration task can be automated for operational and business application.

## II. RELATED LITERATURE

This section investigates various literatures on safety related data extraction.

[8] This work developed two data verification approaches. The first presented an approach for automated verification of data models that extracts a formal data model from an object-relational mapping. The second approach was an unrestricted verification technique for data models which exploited the inherent modularity in Model-View-Controller (MVC) frameworks to automatically extract a formal data model from the data model specification. The two approaches were integrated and implemented in a tool called iDaVer written using the Ruby on Rails framework. The experimental results demonstrated that the developed approach is efficient in detecting and repairing errors in real-world web applications but the model did not considered safety related data.

[9] This work proposed full and generic data ware housing for complex data analysis. A generic UML model was proposed to model low-level and semantic information for complex data analysis. Then, a complex data as an XML document was integrated into Operational Data Storage (ODS). The multi-agent systems (MAS)-based ETL prototype was developed to facilitate this integration based on a flexible and progressive architecture. Furthermore, a comprehensive methodology was also proposed for future analysis of complex data. This approach was validated using a java application. This takes as input a reference multidimensional model and XML documents, and provides logical and physical models for an XML cube composed of homogeneous XML documents. The evaluation of this work showed a significant improvement in terms of response time, thus, the time complexity for the extracted files was ignored.

[10] It developed a method for building metadata through a data chain, mining this metadata and represents it for a consumer of the data to judge base on the integrity of the

data and factor this into the decision-making aspect. It proposes a design and implement for safety data model that helps to ensure integrity of data use for data analysis and decision making. More so, modern and sophisticated ETL software tools including Microsoft SQL Server 2012 Data Tools and Microsoft SQL Server Management Studio were also used for the design. The data were extracted from Safety Related Condition Reports (SRCRs) dataset and used data mining techniques to transform and filter unsafe and hazardous data from the extracted data and stored the safe data into the Data Warehouses (DWs). The prototype was able to load data into designated DWs by extracted all datasets, transform and load the data into the DWs and moved extracted files to archive folder within some few seconds. However, the prototype was unable to Build XML Metadata, Build XML Data, Write Data to XML File, Write Metadata to XML File, Merge XML Document, and Publish XML File to the server.

[11] Employed an automated system extraction structure that includes a discrete extraction rule used for extraction of items from static by way of all-encompassing of dynamic system sheets. This extraction was conducted in three different stages together with a research conducted on a Web text for extraction, position of objects of interest in a Web sheet and lastly, extracting items of interest in a page. A system of over two thousand Web pages was evaluated and 100% accuracy was achieved. Though, the system was entirely automated, but then the evaluation procedure was not computerized.

[12] An innovative procedure called EXALG was presented for the extraction of organized data from a sequence of web pages originating from a common model. The model was revealed by implementing two designs such as, correspondence classes and differentiating parts. Investigation reveals that EXALG on the assembly of web pages disclosed that it is efficient for data extraction from web pages. This investigation has therefore shown that the whole web page was not considered and no special attention for the safety of the extracted data.

[13] This paper developed an automated method of extracting structured data from web pages. These pages comprise several groups of structured data records. Firstly, a tag free based on visual information was built. Furthermore, a tree edit distance method and a visual cue were used to carry out post-order traversal of the tree and subtrees were matched in the process. At the end of the process, data items found in data items were aligned and

extracted. The method was found to be able to effectively extract data from both flat and nested records. Experimental evaluation showed that data extraction could be carried out accurately.

[14] Examined a wrapper induction, a method that constructs wrappers repeatedly, to address traditional methods of data extraction, which is tedious and error-prone. In this work, the ability of this new technique to handle actual internet resources was considered and the time it required to learn wrappers in each class. These were achieved using six wrapper classes. The results obtained showed that most of the wrapper classes considered were very useful. However, a limited number of the classes were very slow and safety of data model was generally not used. This made the extracted data vulnerable to attack.

[15] Information Extraction Based on Pattern Discovery (IEPAD) was proposed. This system can automatically determine extraction rules from web pages. Firstly, repeated pattern mining and multiple sequence alignment automatically identified a record boundary. A data structure called PAT tree was used to realize identification of the repeated patterns. In addition, repeated patterns were further extended by pattern alignment to realize all record instances. This new method is fully automated and experimental results showed that the system outperformed fourteen popular search engines.

[16] A computer program was developed to assist in the assessment process. A field test of the Safety Management Assessment System (SMAS) was conducted at a marine terminal in California. Two teams, with members from the terminal and the regulatory agency, along with a facilitator, were selected and trained. These two independent teams conducted separate evaluations on the same marine terminal. A comparison of their assessments showed that the use of ranges and comments were very helpful. An analysis of field test data shows that SMAS can produce results more consistent than randomness. SMAS shows promise as an efficient and practical method to assess humans and organizations.

An analysis of field test data showed that SMAS can produce results more consistently than randomness with a conclusion that SMAS computer program is critical in conducting the assessment and SMAS also shows promise as an efficient and practical method to assess humans and organizations but in overall results, the safety analysis of the data used was not considered.

[17] Safety related tasks are modelled using the Structured Analysis and Design Technique (SADT). This shows the inputs, resources and criteria/constraints necessary to produce the required outputs; risks are modelled as deviations from normal or desired processes. The framework emphasizes the dynamics of safety management as a process. It aims to provide an abstract ordering of the field which can clarify and specify research and policy needs for the future but a dynamic pattern for safety data extraction was ignore.

[18] Data-driven systems use data from a number of sources including data extracted (and possibly processed) from existing external information systems and data produced specifically for the required system. This data is used to describe the system environment using configuration data (which is largely static or slowly moving) and status data (which is dynamic and will sometimes change rapidly). However, upon all the data extracted from numerous sources, attention was not given for data integrity, thus, safety data was not considered.

### III. RESEARCH METHODOLOGY

This chapter describes the methodology that is followed in the development of the Safety Data Model design for Metadata and the use of navigational data. These developments are carried out in five steps and presented a conceptual Logical Design for Data Model layers that includes Data Source Layer, ETL Layer, Physical Layer, and Logical Layer to examine the necessary transformation in each layer. [10]

This method is reliable as it offered opportunity for navigational data, which is highly useful in shifting towards an acceptance of electronic data on a large scale through the use of several sources to affect data safety. The big data also needs to be verified and validated to ensure that the data is highly acceptable for further usage in a safety of critical applications.

#### Tools used for Data Analysis

SQL Server 2012 Management Studio.

According to Brian Larson, SQL Server Management Studio (SSMS) is a suite of user interface tools used to manage databases for SQL Server Integrated Services (SSIS) project. SSMS allows users to connect to Instance database servers, create databases and tables, and other database objects.

Furthermore, SSIS creates structures known as packages and the packages are used to move data from one system to the other [19].

### Data Extraction process

The SQL Server Integrated Services (SSIS) package executes using the Debugger provided in Visual Studio 2012. The Debugger helps in carrying out unit testing, evaluation for each component in SSIS package and thus makes it easy to identify errors and fix bugs. The green ticks show that the control flow executed successfully without any error.

Figure 1 shows the control flow of the data extraction process. The entire process begins with the creation of SSIS project and to create an SSIS project, the following procedure is followed:

Click on File, then click on New Project, on the drop down provided by the New Project, choose Business Intelligence, then Template and choose Integration Services on the right-hand. The name of the project was entered as HealthAndSafetyETL. Then click OK to create an SSIS project.

After creating the SSIS project, a default package was created and made ready for use with ETL. Next, the variables were created and named as FileName, Archive, XML Data and XML Metadata for storing the filename of the current excel file, storing the path to the archive folder where extracted XML data and XML metadata were stored.

The next step is to create a Foreach Loop Container that loops through each excel file inside HealthAndSafetyDataset folder and extract, transform and load data into DWs. Then the Excel files were moved into an archive folder.

The Foreach Loop Container was dragged from the SSIS Toolbox on the left under the Containers section. Then the Data Flow Task and File System Task were dragged into the Foreach Loop Container. The purpose of the Data Flow Task is to design how data will be extracted from each excel file, transforming the extracted data and loading it into DWs. The main aim of the File System Task is to help in moving the extracted dataset file into the specified archive folder.

This paper is an improvement of an existing literature [10]. However, in comparing the existing model with this

work, the following differences and similarity were observed:

Existing Literature was able to extract files from the datasets, transform the data and move the extracted files into archive folder. However, the current work was able to make the following improvement:

- Build XML Metadata
- Build XML Data
- Write Data to XML File,
- Write Metadata to XML File
- Merge XML Document
- And Publish XML File to the server

Furthermore, a modern sophisticated ETL software tools including Microsoft SQL Server 2012 Data Tools and Microsoft SQL Server Management Studio were also used for the design in both work. The data were extracted from SRCRs in both researches as well.

### Main Data Source

Figure 7 shows the interface of the Main data source in Excel sheet, renamed as HealthAndSafetyDataset in a new created folder. A complete form of the datasets sources is available at:

<http://catalog.data.gov/dataset?q=SRCRs>.

Twelve (12) Excel files of SRCR data between 2002 and 2013 was used, which contain a total of 1039 rows of data. It took roughly 20.703 seconds to complete the ETL as at the time the experiment was conducted

### Safety Data Model

The Safety Data Model was developed based on the requirements analysis as showing in Figure 2.

### Data Source

Data source layer represents Big Data source datasets where many datasets are available for decision makers to analyze. The datasets can be in the structured or unstructured format. Structured data such as XML, which is made available via APIs call, is more preferable as a data source. However, unstructured data in a text, HTML or excel format can also be used as source data as the case may be.

### ETL Layer

ETL Layer represents Extract, Transform, and Loading layer. This layer extracts data from the datasets, transforming the data into a format required by the DWs,

filtering the unsafe and hazardous data out into a separate database before loading the safe data into the DWs.

#### **Physical Layer**

The Physical Layer is the central data repository of a Safety Data Model where data will be stored in DWs.

#### **Presentation Layer**

The application layer is referred as the presentation layer of the Safety Data Model as shows in Figure 3 where the metadata will be published and rendered to end-users in XML format. We used visual studio 2012 to extract data and metadata from the DWs in XML format. The XML file was then made accessible to the end user via a remote server. Figure 3 bellow shows the Presentation Layer Design. [10]

#### **Logical Layer**

The logical layer of the Safety Data Model represents the logical view of the entire data that controls and manages errors that could have emanated from DWs for safety control and build the XML metadata and XML data before XML document are generated.

### **IV. IMPLEMENTATION AND RESULTS**

This section presents an evaluation and the results of the prototype implementation. The prototype was evaluated using a self-evaluation approach with the Visual Studio Debugger and also discussed other possible evaluation methods relevant to our implementation.

#### **Logical and Presentation Layer Implementation**

We implemented a logical and presentation layer together using Visual Studio 2012 Data Tools. These tools make it easy to query DWs and generate XML documents on the fly, merging XML documents and sending XML documents to a remote server.

#### **Metadata Implementation**

We implement metadata to provide summary information about the quality of SRCR datasets and schema. Figure 4 bellow shows the metadata structure in XML format.

The XML metadata document contains as shown in Figure 4. Quality Summary, Data source and Schema Description tags. The end user will use Quality Summary in the metadata to understand if the data is safe for analysis and decision making. Data source provides information about the source of the data while Schema

Description gives detailed information about column name, data type and description of the column.

The data contain Safe, Unsafe and Hazardous data in different tags for easy access as depicted in Figure 5. We used Execute SQL Task to run a query that generates XML, Metadata, XML Data and Script Task write the XML Metadata and XML Data to the file. We also used XML Task and FTP Task to merge the two XML documents and publish it to a remote server to allow easy accessibility for the end user.

#### **The Safe Data schema**

Figure 6 depicts Safe Data schema, which shows the Time Taken to Confirm and Report the status of the data. These two variables were added to get day differences between when a condition was discovered and when it was actually confirmed. The two added times help in the process of determining the quality of safe data and thereby allow us to filter out hazardous data. If the Time Taken to Confirm Existence or Time Taken to Report has a positive value. It indicates that the condition is truly discovered before confirming the existence and the data can be relied upon. However, if either time has a negative value, it indicates that the date value has been wrongly recorded and is not reliable. Therefore, decisions made from such data could lead to loss of life and properties; hence, we filter this as hazardous data. [10]

Figure 7 shows Safety Data Model Implementation execution result. It depicts the time taken for each task to be completed. The package with filename Package.dtsx represents an organized collection of connections, control and data flow elements, event, variables, parameters, etc., using either the graphical design tools that SSIS provides. The dts settings files are stored in the DTSX format and are appended with the dtsx extension. The green tick at the base of Figure 7 shows the overall success of the execution process

Table 1 represents an execution result summary. It shows automated self-evaluation results with indication of various tasks, the time taken and the success or failure of the evaluation as observed from the experiment. The overall results prove that the implementation was successfully carried out as follows: Loop Through Dataset, ETL & Move Extracted Files into Archive folder, Build XML Metadata and XML Data, Write Data and Metadata to XML File, Merge XML Document, and Published XML File to the server.

#### **Possible Method of Evaluation**

The plan for user evaluation includes evaluation documents, which consist of explanation of the user interface, user manual and tasks to be performed by the participants. The target participants are data analysts that make use of SRCRs data for decision-making. This document needs to be well prepared and explained to the participants. In addition, a questionnaire asking relevant questions would be presented to the participants immediately after the tasks to capture their observations, experiences and feedbacks. This questionnaire will be analysed for a greater understanding of the strengths and the weaknesses of the current Safety Data Model implementation.

## V. CONCLUSIONS

This research presents an experimental evaluation of safety data model to ensure the integrity of the data used in the data analysis and decision-making. It however, takes into consideration the time taken to Build, Merge and Publish XML File to a Data Server so that a consumer of such data can verify and judge if the data is safe or free from errors. Software tools such as Microsoft SQL Server 2012 Data Tools and Microsoft SQL Server Management Studio were employed to implement the developed model. The overall results of the model appeared in 00:00:20.703 microseconds, which proved that the results are effective, seamless and the integrity of the data extracted, are largely free from error.

## ACKNOWLEDGEMENT

All praises are due to Allah, Omnipotent, Omniscience who has in all time given us the strength and energy to carry out this research. The proffered gratitude goes to some of our colleagues whom in one way or the other offered useful suggestions for the success of the research.

## REFERENCES

- [1] Kumar, S and Patanka, S. (2012), Easy, Real-Time Big Data Analysis Using Storm, last retrieved from <http://www.drdoobsom/open-source/easy-real-time-big-data-analysis-using-s/240143874> on 18th September 2013.
- [2] Agile Alliance, Agile principles, 2001. Available: <http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software/> 2001: [Accessed: 7 July 2013].
- [3] Dahl, G, The use of Bayesian belief nets in safety assessment of software-based system, HWP-527, Halden Project, 1998, pp.98-104
- [4] Caspi, P, Formal Design of Distributed Control Systems with Lustre, SAFECOMP99, LNCS 1698, Springer-Verlag, Berlin, 1999, pp. 396-409
- [5] Pataricza, A, Checking General Safety Criteria on UML State charts", Computer Safety, Reliability and Security, Proc. 20th Int. Conf., SAFECOMP-2001, Springer, vol. 2187 of LNCS, 2001, pp. 46–55
- [6] Egyed, A. Value-Based Requirements Traceability: Lessons Learned. In Proceedings of the 15th IEEE International Conference on Requirements Engineering, RE'07, 2007, pp. 115-118.
- [7] Politi, M, Modelling Reactive Systems with State charts: The Stalemate approach, iLogix, 1999, pp.90
- [8] Jaideep, Nijjar (2014) Analysis and Verification of Web Application Data Models, University of California, 1-159
- [9] Boussaïd, O., Darmont, J., Bentayeb, F. and Loudcher, S. (2017) 'Warehousing complex data from the Web', Int. J. Web Engineering and Technology, pp. 1–27.
- [10] Anda, Ilyasu; Isah, Omeiza Rabi; Enesi, Femi Amin, "A Safety Data Model for data analysis and decision making", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.9, No.4, pp.21-30, 2017. DOI: 10.5815/ijieeb.2017.04.04
- [11] Buttler, D., Liu, L., Pu, C. A fully automated extraction system for the World Wide Web. IEEE ICDCS-21, 2001.
- [12] Arasu, .A. and Garcia-Molina, H. Extracting Structured Data from Web Pages. SIGMOD-03, 2003.pp. 1-25. A-5
- [13] Liu, Bing, and Yanhong, Zhai. "NET—a system for extracting web data from flat and nested data records." In International Conference on Web Information Systems Engineering, Springer Berlin Heidelberg, pp.487-495. 2005.
- [14] Nicholas Kushmerick, Wrapper induction: efficiency and expressiveness. Artificial Intelligence, 118:15-68, 2000.
- [15] Chang, C. and Lui, S-L. IEPAD: Information extraction based on pattern discovery. <https://www.semanticscholar.org/paper/IEPAD%3A-information-extraction-based-on-pattern/10,2001> : [Accessed: 5 Feb 2014].
- [16] D.D. Hee, B.D. Pickrel, R.G. Bea, K.H. Roberts, R.B. Williamson Reliability Engineering and System Safety, safety Management Assessment System (SMAS): 65 (1999) 125–140.
- [17] A.R. Hale. B. H.J. Heming a, J. Catfhey b, B. Kirwan b "modelling of safety management systems", Safety Science, Elsevier Science Ltd. Holland 1997, V26, 121- 140
- [18] A. Faulkner, N. Story. "Data: an often-ignored component of safety-related systems," International Conference on Computer Safety, Reliability, and Security 86-95, 2000. 6, 2000
- [19] L. Brain, Delivering Business Intelligence. New York, MC, Graw-Hill: 2012.pp. 4-770

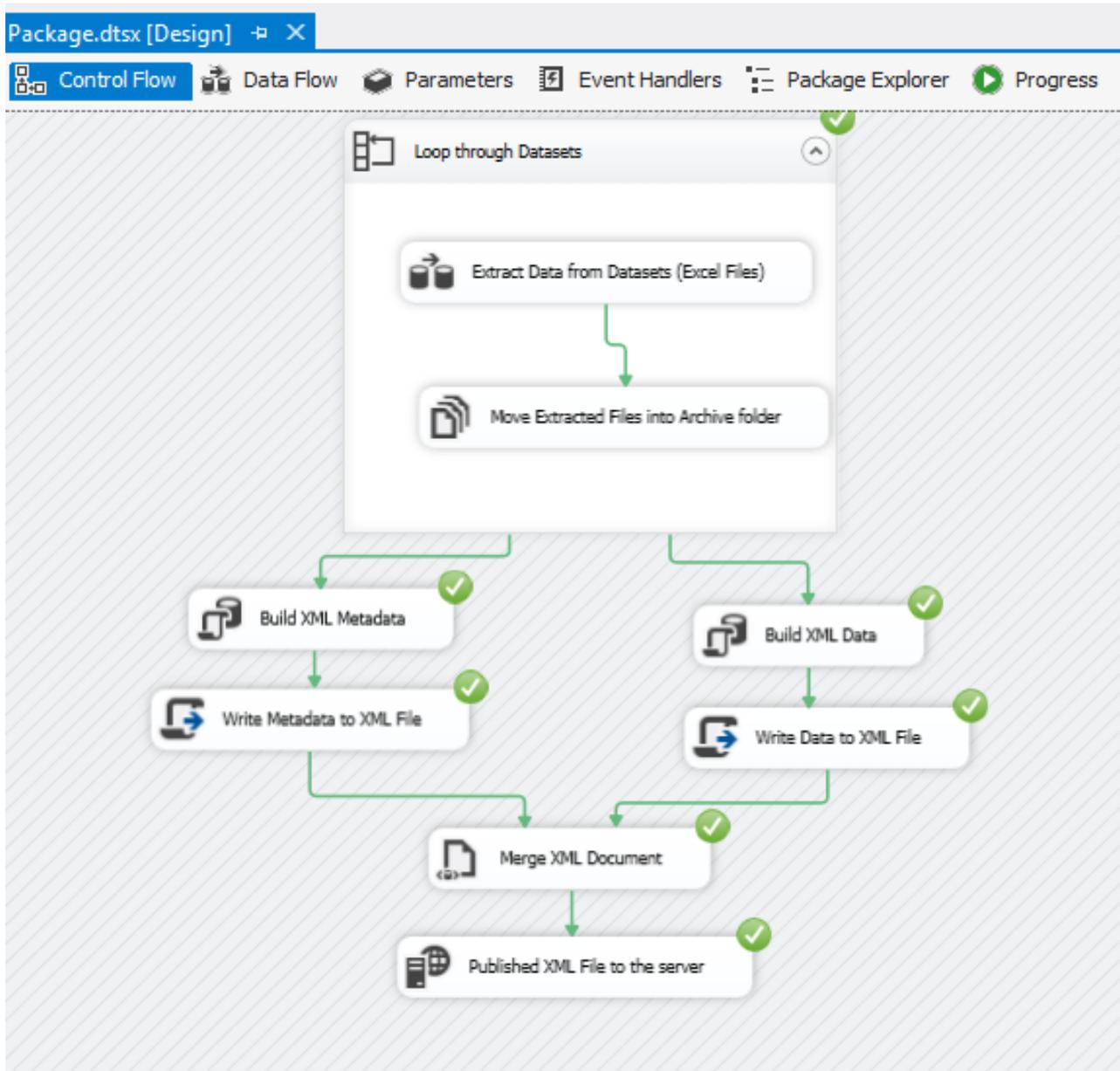


Figure 1: Control Flow for Data Extraction

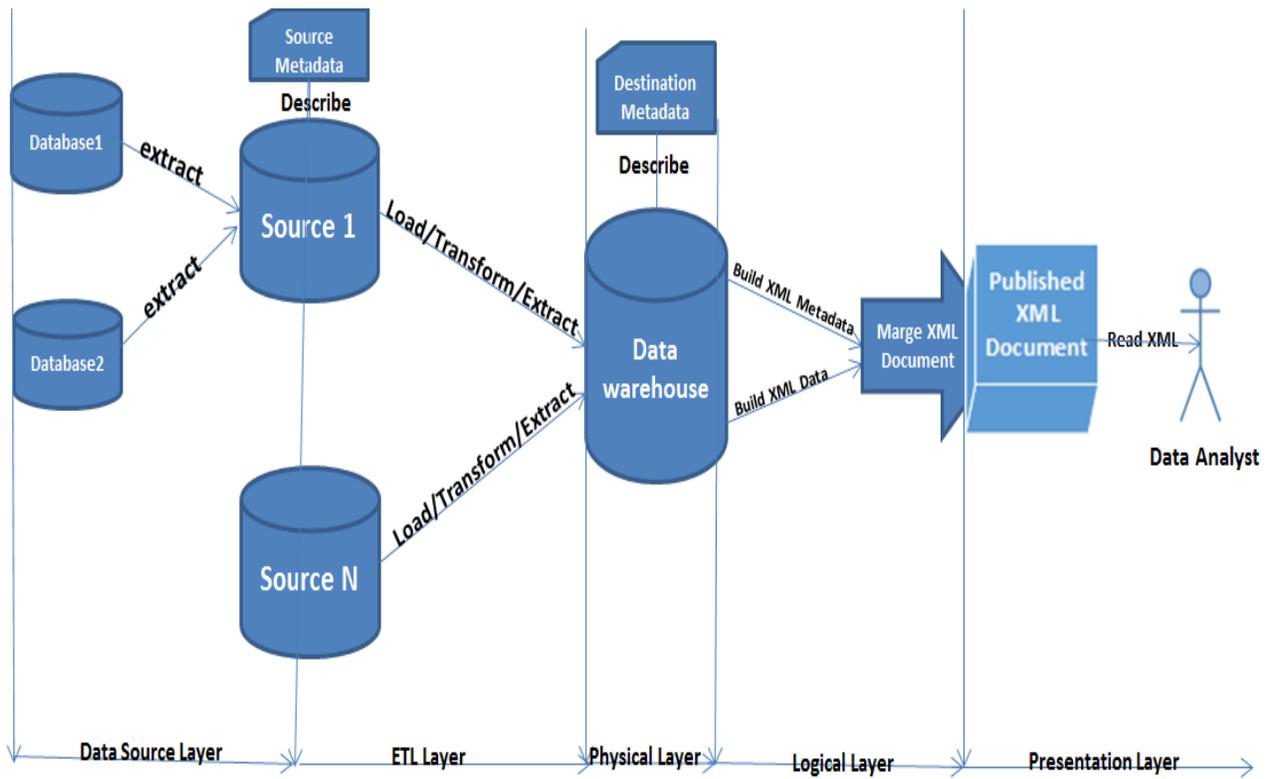


Figure 2: Safety Data Model [3]

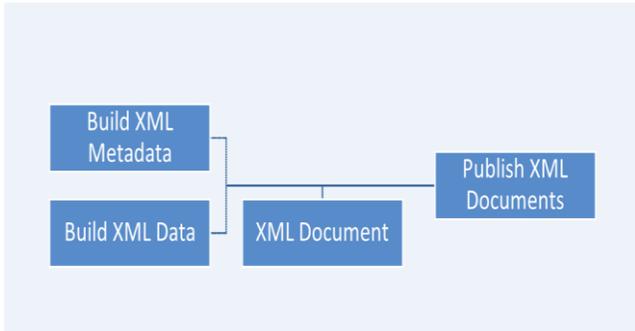


Figure 3: Presentation Layer Design

```
<Metadata>  
  <QualitySummary>...</QualitySummary>  
  <DataSource>...</DataSource>  
  <SchemaDescription>...</SchemaDescription>  
</Metadata>
```

Figure:4 Metadata Structure

```
<Data>  
  <Unsafe>...</Unsafe>  
  <Safe>...</Safe>  
  <Hazardous>...</Hazardous>  
</Data>
```

Figure 5: XML Data Structure

Column Name	Data Type	Allow Nulls
SRCR_Report_ID	bigint	<input type="checkbox"/>
[Status_Open-Closed]	nvarchar(255)	<input checked="" type="checkbox"/>
Operator_ID_Number	bigint	<input checked="" type="checkbox"/>
Operator_Name	nvarchar(255)	<input checked="" type="checkbox"/>
Date_SRCR_Received	datetime	<input checked="" type="checkbox"/>
[Interstate-Intrastate]	nvarchar(255)	<input checked="" type="checkbox"/>
Commodity	nvarchar(255)	<input checked="" type="checkbox"/>
[Onshore-Offshore]	nvarchar(255)	<input checked="" type="checkbox"/>
[Shutdown]	nvarchar(255)	<input checked="" type="checkbox"/>
Conditions	nvarchar(255)	<input checked="" type="checkbox"/>
Date_Condition_Discove...	datetime	<input checked="" type="checkbox"/>
Date_Determined_To_Exist	datetime	<input checked="" type="checkbox"/>
Condition_City	nvarchar(255)	<input checked="" type="checkbox"/>
Condition_County	nvarchar(255)	<input checked="" type="checkbox"/>
Condition_State	nvarchar(255)	<input checked="" type="checkbox"/>
TimeTakenToConfirmed...	bigint	<input checked="" type="checkbox"/>
TimeTakenToReport	bigint	<input checked="" type="checkbox"/>
ETLDate	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties

Figure 6: Safe Data Schema

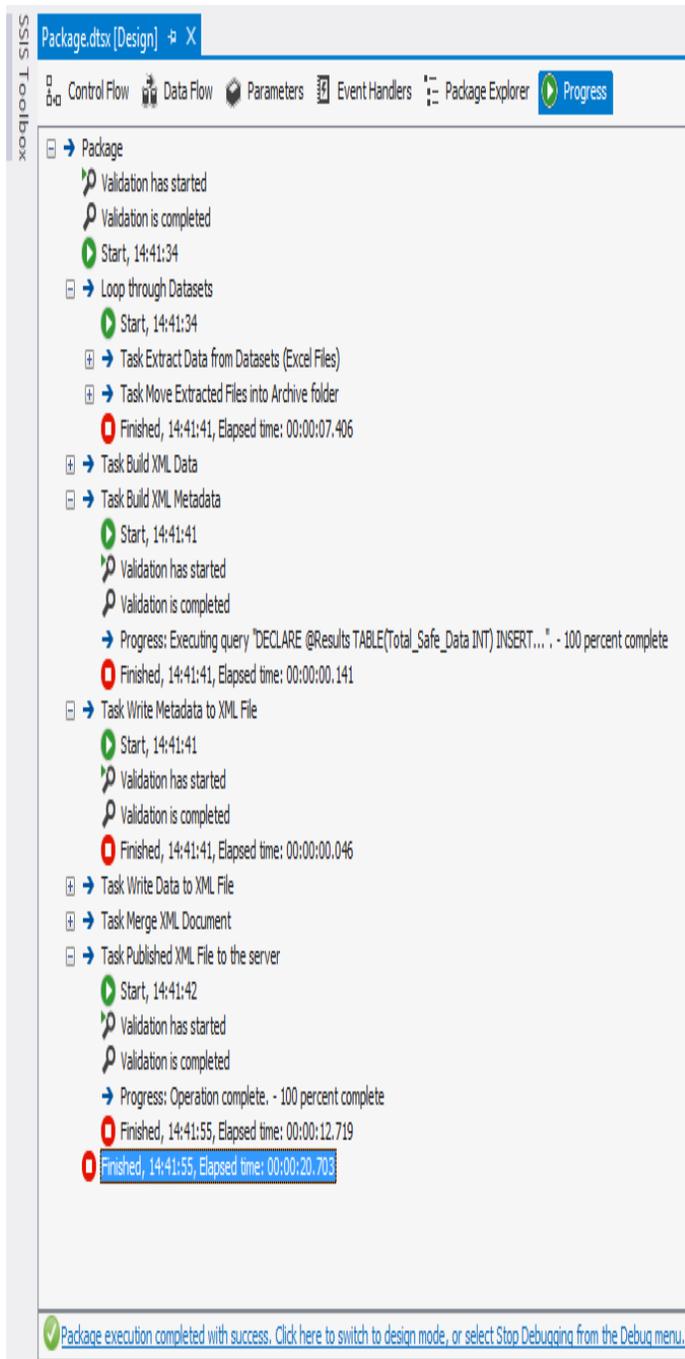


Figure 7: Safety Data Model Implementation Execution Result

**Table 1: Execution Result Summary**

○ S/N	○ Task	○ Time Taken	○ Success / Failure
○ 1	○ Loop Through Dataset, ETL & Move Extracted Files into Archive folder	○ 00:00:07.406	○ Success
○ 2	○ Build XML Metadata	○ 00:00:00.141	○ Success
○ 3	○ Build XML Data	○ 00:00:00.484	○ Success
○ 4	○ Write Metadata to XML File	○ 00:00:00.046	○ Success
○ 5	○ Write Data to XML File	○ 00:00:00.016	○ Success
○ 6	○ Merge XML Document	○ 00:00:00.078	○ Success
○ 7	○ Published XML File to the server	○ 00:00:12.719	○ Success
○	○ Total	○ 00:00:20.703	○