INL/EXT-19-54294
Revision 0

Light Water Reactor Sustainability Program

Method and Application of Data Integration at a Nuclear Power Plant

June 2019

U.S. Department of Energy

Office of Nuclear Energy

INL/EXT-19-54294
Revision 0

Method and Application of Data Integration at a Nuclear Power Plant

Ahmad Al Rashdan, Cameron Krome, Shawn St. Germain, and Joel Corporan[1]
Kelly Ruppert and John Rosenlof[2]

[1] Idaho National Laboratory
[2] Knowledge Relay Inc.

June 2019

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy

ABSTRACT

The data of a nuclear power plant (NPP) are stored in silos for the various plant tools. A previous study conducted by the Light Water Reactor Sustainability (LWRS) Program identified the different data sources that can benefit online monitoring in an NPP. These data have different structures and tools, and are therefore used independently. Integrating data from various sources can result in multiple benefits to online monitoring by enhancing the ability to compare data, reduce statistical uncertainty by accessing a larger data set, increase data heterogeneity in space and time to reveal confounded information, develop intelligent methods, and share data. Data integration can result in direct cost-savings by reducing the amount of effort needed to find information. It can also enable the establishment of a data warehouse, the backbone repository of data for an NPP, which is necessary to evolve NPPs to a holistic online monitoring approach and automate labor-intensive activities.

Creating a data warehouse in a plant or a data broker for a fleet of plants requires overcoming multiple challenges, including the lack of a standard form of storing and using data and the lack of integration methods to compare and map the different data sources. This effort develops and demonstrates a data integration method and application through data fusion and data warehousing pilots. Data warehousing is focused on the application and architecture of data integration and data management. This process is commonly used by data integrators. The data fusion targets intelligently mapping or coupling data for optimized data use. The pilots were performed in collaboration with two NPPs.

The first pilot applied application-specific data integration into a data warehouse. Due to the diverse nature of NPP tools, plant staff are only familiar with tools that are used on a daily basis. If information is needed from other tools, the staff rely on other staff to access the needed information. Often, the staff are not utilizing useful information in other tools due to the inability to use the tools and access the data. This process causes work process inefficiencies that can result in several hours lost per task. Considering the large number of tasks performed at a plant, this can represent thousands or even tens of thousands of labor hours annually. For this reason, a data integration pilot was developed to search through multiple records for equipment in four data sources in a plant. The effort was performed in collaboration with a vendor that has experience in this domain and evaluated at a collaborating NPP. The developed method relied on creating a data depot to pull data from multiple structured databases into an indexable repository. The data were kept in isolated tables within the data depot because mapping of the data required a standard data integration model, which did not exist. The data were pulled in from plant tools to the data depot periodically using a custom modular interface. The pilot effort succeeded but resulted in replicated data sets that mirrored the tools' data models. Any change of these tools would require reconfiguring the data depot and interfacing modules. Additionally, the effort required extensive study of the tools used. A standard data model would have eliminated these challenges.

The second pilot was to integrate data from the work management, inventory and procurement, and preventive-maintenance planning records at a second collaborating NPP to minimize spare-parts stocking (i.e., reduce the required stock) by means of intelligent machine-learning methods with the aim to release tens of millions of United States (U.S.) dollars used for the inventory. This scope was initiated as part of this effort, but is planned to continue through future work. The data fusion part of the scope is the focus of this report. The integration of the data demonstrated that duplicate material records need to be identified and consolidated (i.e., the plant has acquired the same part under different descriptions in different tools over decades of plant operations). Considering the potentially hundreds of thousands of records, this process cannot be performed manually. Four unsupervised machine-learning clustering methods were deployed to identify repeated words (regardless of their meaning) in fields such as description and pricing. Data fusion methods can be very useful for data mapping, an anticipated challenge for the data warehouse. Machine learning can ease the data-mapping process by supervised learning methods (i.e., teach the machine to look for specific commonalities) or unsupervised learning (i.e., detecting patterns of use that look alike).

Both pilots developed as part of this effort will feed into the development of the data warehouse or the data broker standardized data integration model and ontology and to facilitate integration of plant tools into the model. The methods and key findings from this effort are described to share the gained experience with other NPPs exploring this venue as a means for cost-savings.

ACKNOWLEDGEMENTS

The authors would like to thank the U.S. Department of Energy (DOE) Light LWRS Program for funding this effort. The authors would also like to thank Nebraska Public Power District's Cooper Nuclear Station and Xcel Energy, Inc., for collaborating on this work and for providing information that led to identifying cost-savings data integration opportunities for the nuclear power industry. The authors acknowledge Dr. Ronald Boring and Dr. Thomas Ulrich from Idaho National Laboratory (INL) for designing the user-friendly human/machine interface.

CONTENTS

FIGURES

TABLES

ACRONYMS

2-D	two-dimensional
3-D	three-dimensional
API	application programming interface
CIC	component identification code
DD	data depot
DOE	U.S. Department of Energy
ETL	extract, transform, load
GUI	graphical user interface
LWRS	Light Water Reactor Sustainability
NPP	nuclear power plant
O&M	operations and maintenance
PM	preventative maintenance
REST	representational state transfer
SDM	shift duty manager
SME	subject matter expert
SSS	Solr Search Server
TF-IDF	term frequency–inverse document frequency
U.S.	United States
WBUI	web-based user interface

Method and Application of Data Integration at a Nuclear Power Plant

1. INTRODUCTION

Though nuclear energy provides several advantages over other energy sources, nuclear power plants (NPPs) have been economically challenged to compete with other sources of energy in terms of cost per megawatt-hour of power. The current fleet of NPPs in the United States (U.S.) is labor-dependent and relies on manual processes. Migration towards automation in the nuclear power industry has been slow due to its unique regulatory nature. This caused operations and maintenance (O&M) costs to remain high, while other industries leveraged technology development and were able to drop their O&M costs. The data of an NPP are stored in various systems, such as plant computers, work management systems, scheduling, systems engineering, operator logs, condition reporting, etc. A previous study by Al Rashdan and St. Germain (2019) was conducted for the U.S. Department of Energy (DOE) Light Water Reactor Sustainability (LWRS) Program and identified several different data sources in an NPP that can benefit online monitoring. These data have different structures and tools, and are therefore, used independently. Integrating data sources can result in multiple benefits that have been recognized by other industries. Curran and Hussong (2009) state the following potential benefits of data integration:

- Replication: Comparison of multiple studies can be used for verification, to perform new research on existing data, and can avoid the creation of new unneeded studies
- Increased statistical power: The effective increase in sample size that comes from integrating multiple data sets can strengthen statistical power when testing some hypotheses
- Increased sample heterogeneity: Including data from multiple separate sources makes it possible to consider population subgroups
- Increased frequencies of low base-rate behaviors: Behaviors that show up in low frequencies are more easily studied in integrated data sets because the total number of such samples is higher
- Broader assessment of constructs: Each study will use unique methods for assessing theoretical constructs and considering multiple data sets (and, therefore, multiple methods) improves the assessment strength
- Extended period of developmental study: Integrating studies from multiple different time periods can extend the total range of study
- Support of data-sharing and building a cumulative science.

In addition to these benefits, enabling NPPs to use data-driven online-monitoring methods to automate labor activities requires developing a data warehouse (Figure 1) as the backbone for these activities. A data warehouse can result in direct cost-savings by reducing the amount of effort needed for data integration, which is currently performed manually and on an as-needed basis.

Whether within a single NPP or for a fleet of NPPs, deploying a data warehouse requires overcoming multiple challenges, especially the lack of a standard form for storing and using data and a lack of integration methods to compare and map different data sources. The present effort is focused on the second challenge: Automating data integration and creating a streamlined process to combine data from all plant sources. The outcome of this effort will also benefit the first challenge (i.e., to create a data integration model and ontology for NPPs). The result of this effort will be compared to the result of using a data model (to be developed in a later effort and not included in this scope of work) to determine the benefits and applicability of adopting a standardized data integration and ontology model.

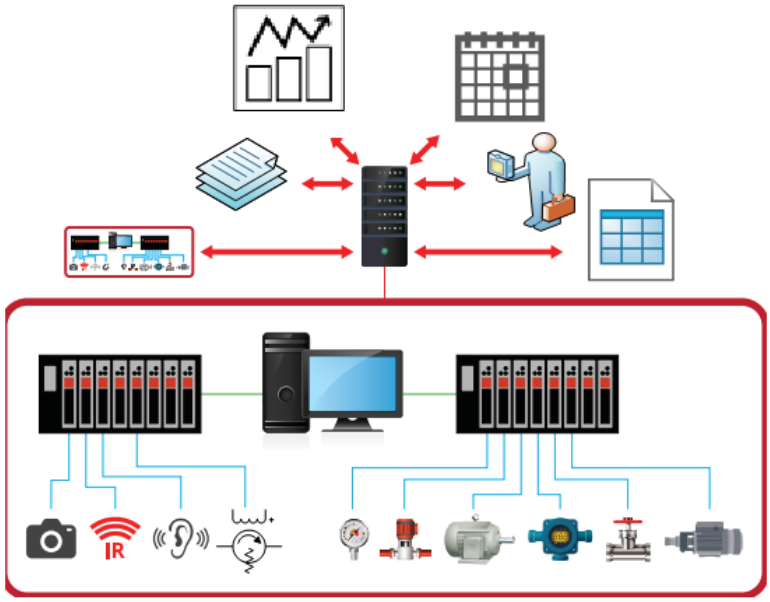


Figure 1. One of the purposes of a data warehouse in a NPP is to integrate associated data collected by all plant processes.

The following section describes the vision of the data broker that this work will enable for an NPP or fleet of NPPs. Two data integration pilots were conducted in this effort. The first is related to data warehousing and is described in Section 2. An integration application was developed to search through multiple records for equipment in four data sources of an NPP. The second method, described in Section 3, is a machine-clustering method deployed to identify duplicate records as part of an initiated effort that is planned to continue through future work. This is to integrate data from inventory records, work orders, and planned maintenance information at a second NPP to minimize spare-parts stocking (i.e., reduce required stock minimum-hold requirements).

1.1 Data Warehouse and Data Broker

NPPs have multiple data sources in various forms. These data are not typically used beyond the intended target for collecting that information. As NPPs move from manual and labor-intensive activities to online monitoring, the data generated needs to be consolidated, associated, and analyzed with respect to historic baselines from multiple sources in a systematic manner. Whether on a local or fleet basis, a data warehouse may be challenged by:

- Lack of standard form to share the data
- Lack of automated data-mapping and preparation methods
- Security (including cybersecurity)
- Legal (for the fleet scope of the data warehouse)
- Lack of infrastructure to share data
- Effort and cost needed to explain, modify, or clean the data
- Cost for enabling data-sharing
- Uncertainty due to issues not foreseen by the NPP
- No clear big-picture potential yet.

On the plant local level, the data warehouse will:

- Act as a hub between NPP organizations
- Host data generated by the data sources
- Host a library of proven methods created to process the data
- Host a powerful and scalable data infrastructure for projected needs
- Use a standard data model and ontology for data repository and use
- Create the data integration and mapping methods to prepare and convert the data from the NPP to a useful form
- Interface with a fleet data warehouse, if needed
- Take responsibility for ensuring the data are protected (from a physical and cybersecurity perspective)
- Control the access and use of data according to user need and granted rights
- Operate based on a value proposition and business model with a proven path for cost-savings.

A data warehouse for a fleet of NPPs can perform as a data broker. Data-sharing among multiple plants results in several benefits (described in Section 1), mainly related to establishing a broader baseline to share experiences and validate the performance of equipment or processes in an NPP against other NPPs. For example, a specific bearing failure of a pump might result in the identification of a specific vibration signature that precedes that failure. Unless the NPP has experienced such a failure, that signature has never before been documented; therefore, the failure might take years to occur before that signature might be captured. Looking at the NPP fleet, this signature will probably be known or the failure can be expected to occur sooner so that the signature might be obtained.

On the fleet level, the data broker will:

- Act as a hub between NPPs, vendors, and research organizations, including universities. Vendors, and research organizations will be referred to as data users.
- Host data generated by the NPPs and methods created by the data users.
- Host a powerful and scalable data infrastructure for projected needs.
- Be located in one or multiple strategic locations (i.e., distributed data warehouse) that reduces the total cost of ownership of a large-scale data center.
- Have dedicated contracts with major internet/data service providers to enable establishing dedicated data links from the NPPs to the data broker.
- Adopt and use a standard data model for repository and use and distribute or adopt standard guidelines for the NPPs to follow.
- Create the data integration and mapping methods that transform the data from the NPPs to a useful and standard form.
- Enable data use on the data broker servers, including allowing the data to be transferred to the user after specific authorization is issued by the NPP or the data are sanitized (i.e., once the context is removed).
- Host a library of proven methods (created by data users), make this accessible to NPPs to benefit from, and present the methods to NPPs to use.
- Enable NPPs to validate user-developed methods on the data broker servers, then download the algorithms from these methods to use locally in the NPP.

- Create a standard for application programming interface (API) development.
- Provide consultancy services to NPPs to enable its role.
- Take responsibility for ensuring the data are protected (from a physical and cybersecurity perspective).
- Control access to and use of data according to user need and NPP-granted rights.
- Establish all legal agreements needed with NPPs and data users to enable data use within the scope of agreed upon use.
- Ensure all data-use activities are compliant with laws and regulations.
- Develop a value proposition and business model to sustain its operation.

The following sections describes two pilots aimed at enabling data integration for a data warehouse or data broker.

2. DATA WAREHOUSING

The data warehousing scope of this effort is focused on the application and architecture of data integration and data management. This effort is common among data integrators in non-nuclear industries seeking to achieve cost-savings. For example, Moore and Starr (2006) used data integration to identify which pieces of equipment to repair, and in what order, by making use of data from a variety of sources: production schedules, condition-monitoring, maintenance-management, financial records, and health and safety regulations. Maintenance activities are prioritized using data pulled from multiple sources.

2.1 Search Engine

Due to the diverse nature of tools within an NPP, the staff rely on other staff to access needed information. Often, staff do not utilize useful information in other tools due to their inability to use those tools to access the data. This process causes work process inefficiencies that can result in several lost hours per task. Considering the large number of tasks performed at a plant, this can represent thousands or even tens of thousands of labor hours annually. For example, it is not uncommon for an NPP to call in a staff member after hours to provide insight on a piece of equipment that shows an anomaly or to conduct meetings to exchange data and information that could be acquired electronically. An autonomous work process necessitates breaking data boundaries and enabling the use of data regardless of tools knowledge by means of data integration. This work aims to pilot such an effort. The scope of this pilot resulted from a data integration need study. An example of the outcome is shown as a part of Table 1. This table is exploratory and plant or person-specific and is, therefore, shown here for demonstration purposes. Table 1 lists a sample of plant users and data sources and is broken into three categories:

- Data source is available to the user and is being extensively used: green
- Data source is not available to the user (not used) but is needed: red
- Data source is not available to the user, but is not needed: yellow.

The table's red cells require action. In addition to defining the columns on an organization basis, it is possible to apply the same approach on specific-use cases. In this pilot, the shift duty manager (SDM) was chosen as the targeted-use case. The SDM needs quick access to several data sources to support emergent issue resolution and to quickly catch up on recent changes to plant status. The pilot was developed to search data for an SDM through four different data sources in a NPP. These are work management data from a new system and a previous system, condition-reporting data, and operations clearance orders (though already used by operators for ease of use).

The preliminary resulting search tool process and graphical user interface (GUI) are shown in Figure 2, Figure 3, and Figure 4. As the tool is evaluated in the plant following this report, the GUI may

evolve to other forms that are human factor optimized for the targeted use. The current approach is based on inserting text that is expected to be a part of the functional location or component identification code (CIC) to search for in a text box. The functional location and CIC are two forms of unique identifiers used to refer to a component or specific piece of equipment in the plant. Either one or both of the two identifiers are used in plant tools. The search will result in all records that have the text be listed. Once one record is selected, side tabs will present different data sources to explore. Every tab will have its own data extracted from that data source.

Table 1. Identifying data access gaps in a NPP (subset for demonstration).

	Operations	Maintenance	Systems Engineering	Planning	Scheduling	Projects	SDM Use Case
Operator Rounds	Used	Not used, but needed	Not used, but needed	Not used and not needed	Not used and not needed	Not used and not needed	Used
Control Room Logs	Used	Not used, but needed	Not used, but needed	Not used, but needed	Not used, but needed	Not used, but needed	Used
Clearance Orders	Used	Not used, but needed	Not used, but needed	Used	Used	Not used and not needed	Used
Work Orders	Not used, but needed	Used	Used	Used	Used	Not used, but needed	Not used, but needed
Condition Reports	Not used, but needed	Used	Used	Not used, but needed	Not used, but needed	Used	Not used, but needed
Process Information	Used	Not used, but needed	Used	Not used and not needed	Not used and not needed	Not used and not needed	Used
Materials & Inventory	Not used, but needed	Used	Not used, but needed	Used	Used	Not used, but needed	Not used, but needed
Schedule	Used	Used	Not used, but needed	Used	Used	Used	Used

Several assumptions were made with the developed tool:

- The tool does not detect or handle all typographic errors in the source data. For instance, when performing search functions, an extra space or misspelled word in the plant data will not be returned by a given query. The tools used can mitigate typos in the search terms, to a certain degree (but not the source data). When the tool tokenizes the words in a field to index, it can transform it to word roots and perform spell-checking for the search terms.
- To integrate data, records among source systems must relate using a common identifier. For this pilot, the records in each system was assumed to relate using the functional location as this is the identifier all records have in common.

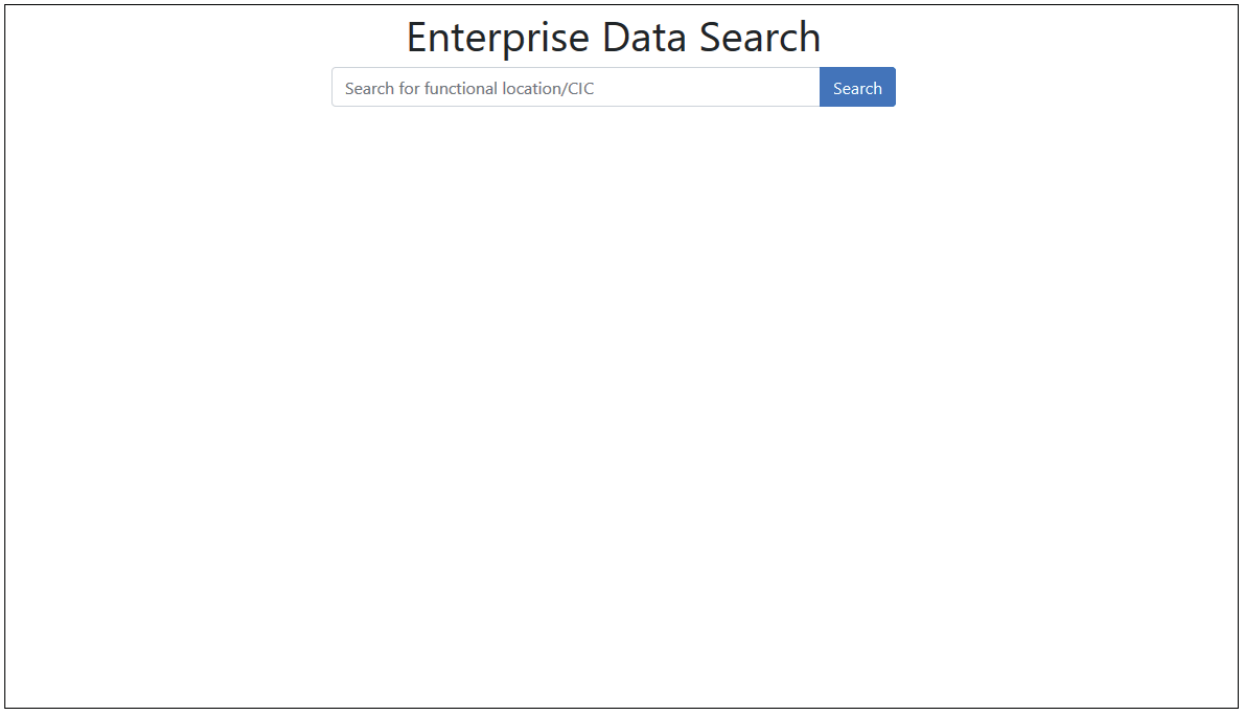


Figure 2. Step 1: Search for text in functional location or CIC.

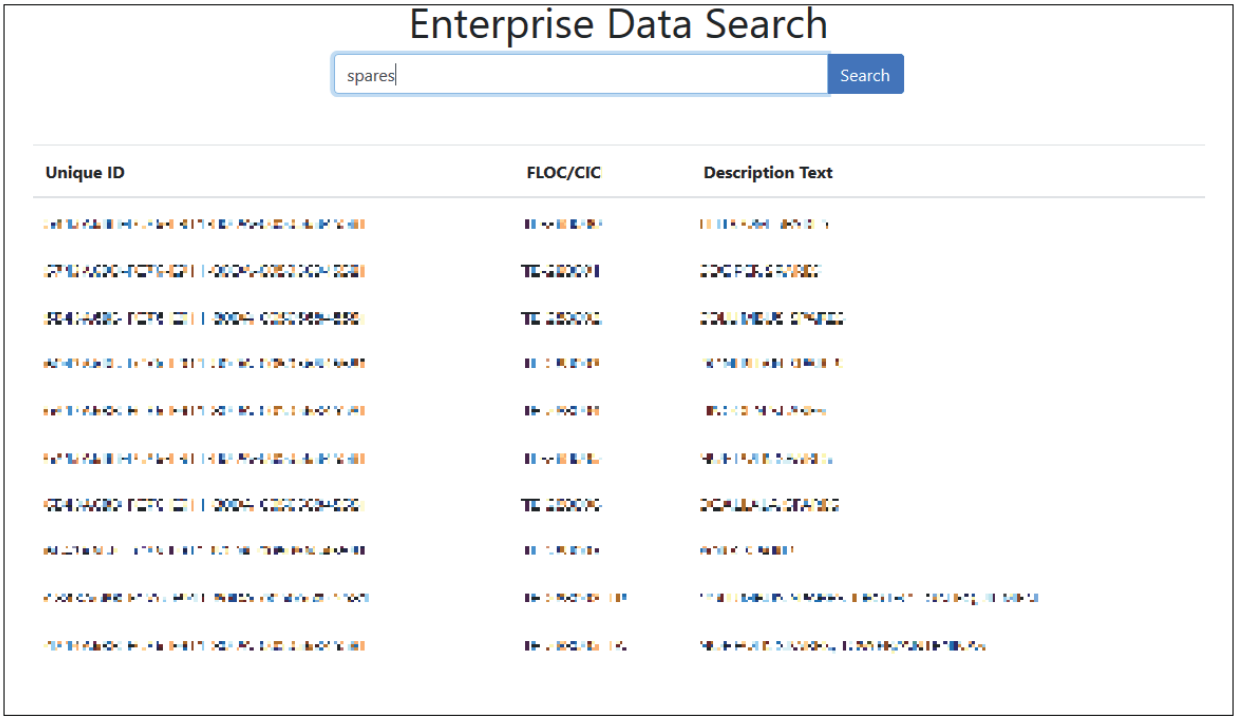


Figure 3. Step 2: Select the targeted functional location or CIC (plant data blurred).

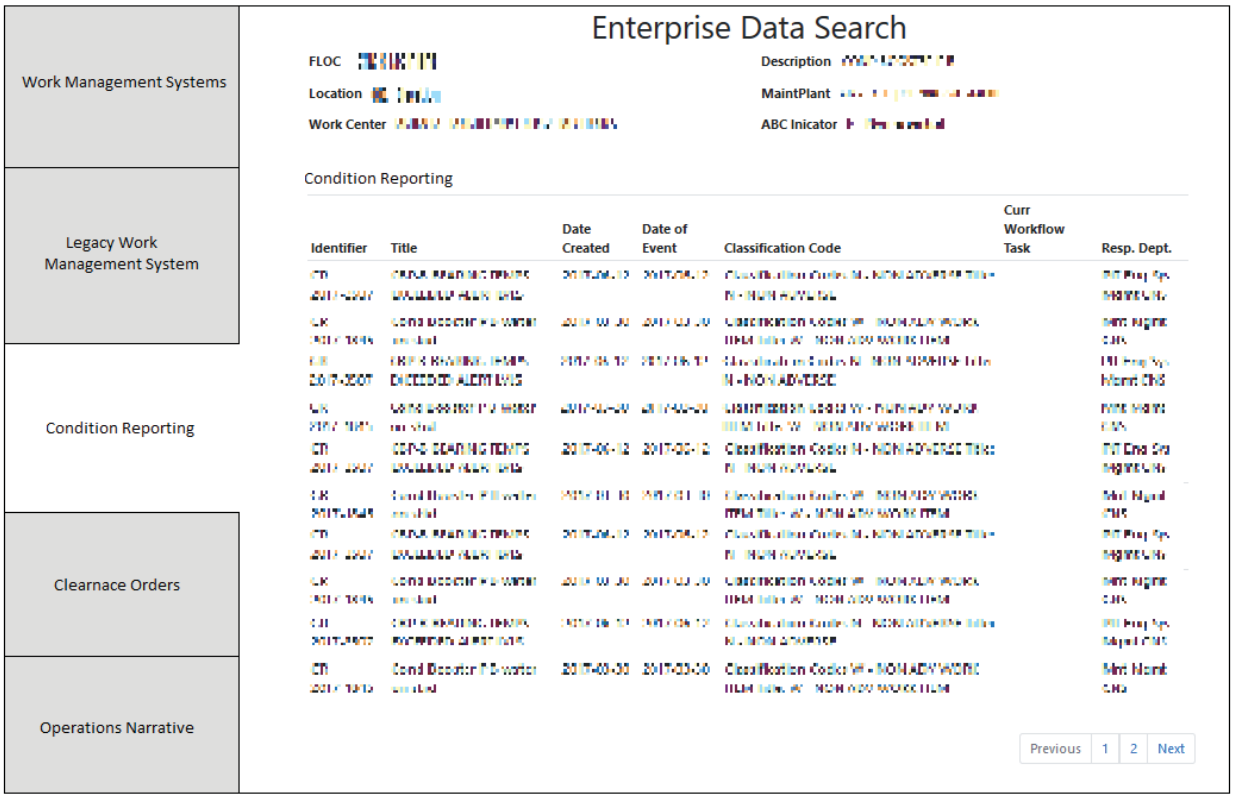


Figure 4. Step 3: Select the data source of interest from a side tab once a record is selected (plant data blurred).

2.2 Method

The search solution comprises three basic parts for collection, processing, and presentation of data, as shown in Figure 5:

- A data depot (DD) to stage the data
- A Solr Search Server (SSS) to index and search data
- A web-based user interface (WBUI) to take search queries submitted by the user, submit it to the SSS, and display the results in a user-friendly format.

The DD is a set of staging tables populated with data from as many sources as is necessary to support a particular task, as shown in Figure 6. Due to the lack of a standard data model, it was deemed more efficient to put all data into one set of staging tables capable of replicating some of the source's columns. Because the data are in different source databases, coding all of the application logic in a single program was deemed undesirable. An extract, transform, load (ETL) application was used to copy data from each database to the DD. It is a command-line application that queries source databases and inserts data into the DD. All queries are compiled into the application. This application encapsulates all data access logic, the location and schema of all source tables, and how they map to destination tables in the DD. The application was developed using C#. Targeting the .NET Core framework made it platform- (e.g., Windows, Linux, or Mac) independent, as long as the platform is supported by .NET Core. Users can run the ETL application to populate the DD at any frequency desired.

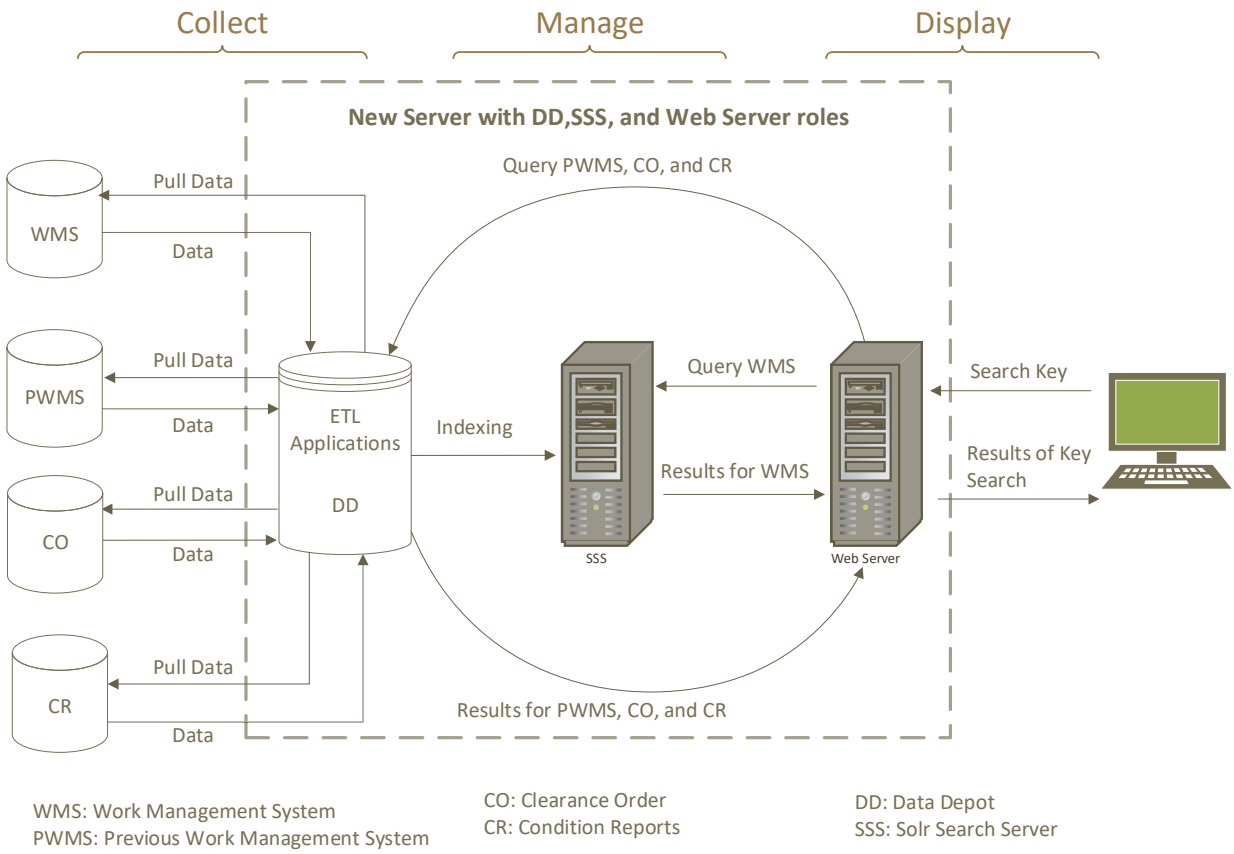


Figure 5. The process of application data integration used for the pilot.

The source data is accessed using views. Instead of getting access directly to the tables, the views query the tables and present the data. Views are sufficient because they return data without requiring direct access. Queries written for views can eliminate or transform data as needed. Direct source access is, overall, predicted to perform better than views because views add in an additional step, creating minor latency issues and introduce the risk that complete data sets are not presented. Views were used by this pilot due to security constraints imposed by the NPP.

Once data is copied into the DD, the ETL application triggers the SSS to retrieve data from the DD and index it. The population of the DD is based on a pull of the data. This implementation is favored over a push of data from the source to the DD when changes occur in the source because the amount of time it takes to refresh the data meets performance requirements, and the system does not experience degradation of performance, which is the typical reason a push is normally deployed. Additionally, a push of data requires use of change data-capture technology, which can be less reliable.

The SSS creates an index file on the server, searches the indexed DD data for relevant records, and supplies them to a presentation layer. It encapsulates all search logic—indexing the searchable data, processing queries, and returning search results. Using the defined schemas, the used tool, Solr, has a utility that converts DD data into a searchable document. The use of the Solr offered a rich set of search features that are all accessible from a representational-state transfer (REST)-ful interface. A RESTful interface is an API, the architectural style of which is based on the REST model. This makes it easy for a variety of applications, from web to desktop applications, to interact with the server, increasing overall flexibility. Solr was selected for this particular project due to the factors described above, in addition to existence at the pilot site and staff familiarity with the product. Other tools, such as ElasticSearch, were

not evaluated; a survey of the best tool to use was not conducted for this effort. After the data are indexed using the generated document, the SSS accepts search queries and provides results. For this specific case, the indexing step was only applied to the site's work management system (i.e., Solr only needed to index the functional-location data from the work management system) because the returned records can be used by the WBUI to retrieve related data from the other sources (i.e., previous work management system, condition-reporting system, and clearance orders tool), as shown in Figure 5. The results of the search are captured directly from the other data sets in the DD.

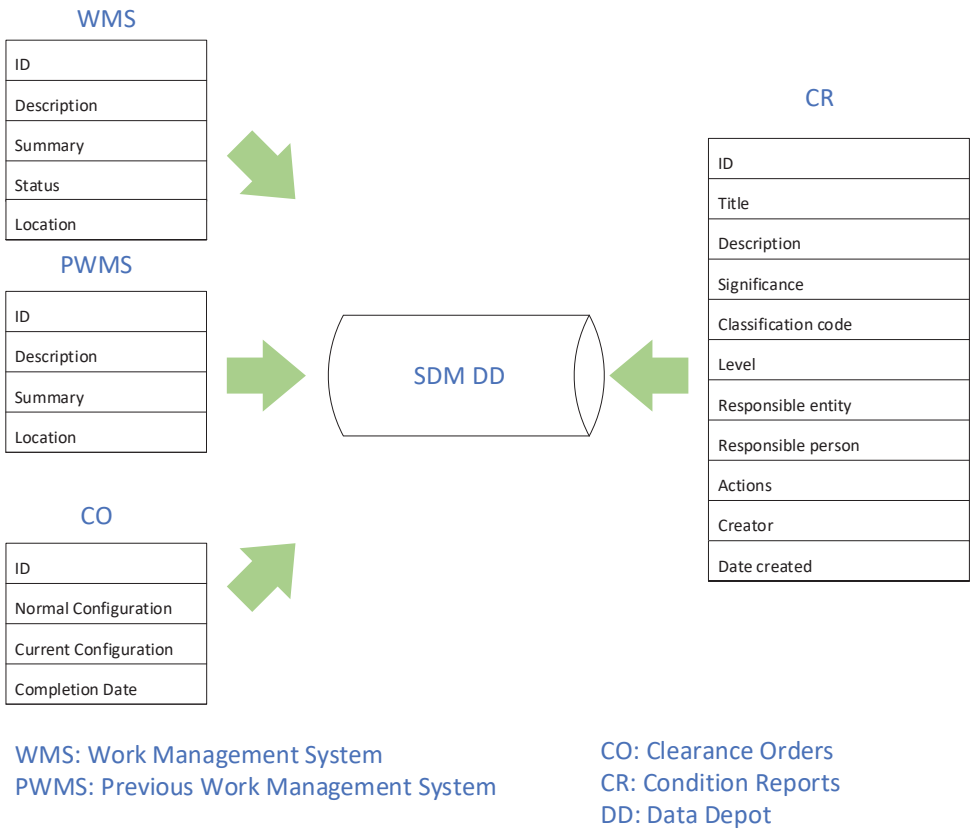


Figure 6. The data integrated into a DD for SDM.

Adding additional data to the application is relatively easy. Instead of reprogramming a search engine to tap into additional sources, data are added to the DD, allowing it to become searchable. Only minor adjustments to the SSS are needed. Additionally, if a data source changes, the data directed into those particular DD fields can be changed. In this case, no changes are needed to the SSS. For example, if the data source schema changes such that the functional location is now stored in a different column or table, or even a combination of columns across tables, the ETL application can be modified to retrieve the field from one or more of the new columns and place it into the DD column that has always contained the functional location. When SSS retrieves data from the DD, it does not need to be aware of any changes to the source schema.

To enable the user to search and view results, WBUI was used. WBUI takes search terms from the user, formats them into a search query, submits it to the SSS via an API provided by the search tool, and presents the search results to the user in an easy-to-read format. WBUI was used for consistency purposes in that the SSS was already web-based. It also increased usability because users do not have to install an application on their machine and configure settings or permissions. The web framework, Aurelia, was

chosen for this effort because it is based on newer web technology and standards, so it is not expected to become outdated soon, and it uses pure HTML and JavaScript standards. It does not require special syntax like other frameworks. Though Aurelia was selected for this particular effort due to the factors described, other tools were not evaluated. A survey of best tools was not conducted for this effort.

Using the described approach, the three systems (e.g., DD, SSS, WBUI) that excel at their intended purpose can be used for their optimal use—one is optimized for gathering data, another is optimized for searching for data, and a third is optimized for displaying results. Dividing the method into separate applications limits the impact of any changes made to each part. The advantage of this approach is that it is less likely to introduce errors to other parts of the process by making a particular change. The limitation is that changes in multiple places are needed to enhance the product. Through the development process, other approaches were considered, but researchers reached the conclusion that they were less desirable for a variety of reasons. The approaches and conclusions are:

1. Access data directly from the source applications using an API: APIs are slow to retrieve data and cause the source system to slow down to the detriment of the user community. It also requires expertise to understand each source system's API, which is challenging and may also introduce additional points of failure (i.e., if the source system's API had bugs). If an API was used, it would be to populate the DD such that the only latency experienced was from the source data to DD, instead of throughout the entire search-solution process. In order to mitigate latency while loading data into the DD, it is possible to employ table swapping such that temporary tables are utilized to temporarily store the refreshed data as they populate. Once the refreshed data are completely ready, they would replace the now-outdated data in the DD. This approach emulates how cloud-based data, also subject to latency issues, could be handled. The use of an API generally drives up licensing and maintenance costs.
2. Direct access to the source data, using WBUI, instead of to a populated DD: From an architectural standpoint, the web application would need to be much more sophisticated because it would contain all the logic that was included in the ETL. It would need to know how to combine the data as users submitted search criteria, likely causing performance problems. When new data sources are added or existing sources are changed, developers would need to understand the data and alter the web application.
3. Incorporate a solution similar to the one currently deployed, but integrating the tables in the DD: This only makes sense for data that have similar schemas or if a data repository with a standardized form exists. The data from the tables have some similarities, but not enough to consider merging all of them into the same set of tables.

Whether using Approach 1 or 2, if some of the source data are stagnant or rarely in need of refresh, each time a request was created, the API or WBUI would unnecessarily perform a fresh search on the source. The use of an API or WBUI will cause a loss of the ability to sanitize and control the data input to the SSS. Using the DD approach, if users update the source data frequently, the application could be run to populate the DD frequently. If the source data were rarely (or never) changed, the application to populate the DD would run less frequently (or only once). Possible future enhancements could include configuring the frequency of data updates for certain tables or databases and having a microservice on the search server notified of a DD refresh, triggering a re-indexing of the search data.

2.3 Lessons Learned

The lessons learned from this effort can be summarized as:

- A subject matter expert (SME) needs to take ownership from the plant. The SME should know the plant processes and the means by which data are stored in all source systems. Typical users only understand data in one to two systems. When the goal is to integrate data, getting answers on how to assimilate data is challenging.

- Some source data are extremely complex. In this pilot, one data source managed many enterprise processes, some of which were disparate. The number of tables and fields deployed was quite high and often not easily reconciled between processes. This emphasized the need of having SMEs readily available.
- Some data in the source systems were found to need preparation. For example, duplicate entries in a single source system were found during the initial phase of the pilot.
- For this pilot, the source data were presented in a uniform fashion using views accessing data stored in a specific server database. In the future, additional data may be stored in other databases or even Excel spreadsheets. With the current search solution, in which a DD is used to gather and stage data, this would not be an issue. The ETL application would gather data from these diverse sources and assimilate them into the standardized DD.
- All source data were structured data. If, in the future, additional data were unstructured from heterogeneous data sources, the search approach could handle this using ETL applications. They would gather data from these diverse sources and assimilate them into the standardized DD using standard unstructured-data search methodologies in utilities such as HADOOP. This can be done without affecting the other system-design elements.
- Because virtually all organizations will have quasi-unique toolsets with variations in data types and forms, the DD model allows for ETL level manipulation of source data such that they can be transformed into a common set, usable even in a bidirectional fashion (if required).

3. DATA FUSION

Data fusion efforts are focused towards novel ways to combine heterogeneous data. In 1991, the Joint Directors of Laboratories, known as the Data Fusion Group, defined data fusion as: “process dealing with the association, correlation, and combination of data and information from single and multiple sources.” (White 1991). The concept of data fusion has been around for a while, but has been recently increasingly applied to all science and engineering fields. For example, Hall and Llinas (1997) applied data fusion for different areas at the U.S. Department of Defense, from surveillance to target recognition using Bayesian inference to combine data from multiple sensors and related information from associated databases. The method is applied at the feature level, as well as at the decision level. Sorber et al. (2015) discuss a framework for fusing data from multiple data sets that assumes any of the sets may be sparsely populated, and that each is represented as a multi-dimensional tensor. Each data set is factorized with matrix/tensor decomposition (i.e., its dimensionality is reduced), and the common subfactors between sets are used to join them. Correa et al. (2010) discuss the use of canonical correlation analysis or its multiset variant to achieve a similar objective. In the case of multi-modal data sets, Correa et al. (2010) reduce each modality to a lower dimension representation (or features) first. Variations in these features are then investigated and used to find connections, enabling the fusion of data. Smilde et al. (2005) makes use of two data sets, each with tens of measurements and many thousands of variables. The data fusion process is broken into several levels. The first level of fusion is described as simple concatenation of the measurements where the samples are the shared mode. The midlevel fusion also aims at concatenation of the data, but this is performed after selecting a finite number of variables (e.g., those with the highest covariance to the response). High-level data fusion is performed by creating separate models for each set and then combining the predicted values. Wilderjans et al. (2009) discuss techniques for analyzing data blocks of different size. Using simulated datasets that emulate a defined criterion, the paper tested two different ways to apply weights to fuse data for optimization. The first method was to consider entries themselves during analysis (so that each entry provided equal influence) and the second was to consider each data block (so that each block provided equal influence). Van Mechelen and Smilde (2010) use the term ‘coupled’ data sets in reference to those that contain multiple types of data about the same system. The paper defines a data block as one piece in a coupled data set (i.e., data from just one data source) and describes a generic model for fusing data blocks together. Each unique feature present across all data

blocks is called a *mode*; thus, coupled data are created by connecting data blocks together wherever there is a shared mode. The authors describe a mathematical model for coupled data sets, present algorithms for construction and optimization, and describe analytical issues that may be encountered. These are all examples of different methods of data fusion. The next section explains other methods that were used for this scope of work due to their suitability, simplicity, and effectiveness.

3.1 Clustering of Inventory

The goal of this effort is to minimize the inventory size of an NPP without impacting maintenance activities by causing a shortage of parts. The motivation behind this is purely financial; funds invested in excess spare parts could be utilized elsewhere, but not having a spare part available when needed could disturb plant activities. The cost of having work delayed in a facility far outweighs the benefits of minimizing stock, so, to date, plants have focused on having significantly more parts than needed. The current approach for determining when to order parts and how many of each to order is manual. A minimum threshold for each part is decided, and anytime the quantity on hand drops below the minimum, a fixed number of units is ordered. If the demand for spare parts can be predicted with reasonable accuracy, it could be used to determine more intelligently when to order spare parts and potentially reduce the amount of excess inventory. Making these predictions as accurately as possible will prevent them from negatively impacting maintenance.

In order to achieve the inventory optimization goal, data sets need to be integrated. The data sets used in this study contain information about maintenance work performed at nuclear facilities. The data were acquired from several different systems, as shown in Figure 7; therefore, the sets do not perfectly align. Each set contains inventory data (the list of all parts used by the facility and their prices), work order details (what work was performed, when the work was performed, why it was performed, and what materials were used), and planned preventative maintenance (PM) information (e.g., what tasks need to be performed as PM and how often will they be scheduled). The initial challenge that was faced lies in the variation of product descriptions for the same product. Because the data are acquired from different systems, a unique common key to map the products together does not exist. The only features the sets have in common are the product description, which are manually entered text strings, and price. However, the same or similar material may exist in different data sets with different product descriptions and a slightly different price. Additionally, even within a single data set, there are cases where the same product exists in multiple different rows, usually because it is made by several different manufacturers or because it was re-entered by staff. Finding and eliminating duplicate rows can present a challenge, as can mapping sets that have different features. Even within common features, differences in the units of measurement or quality of data can be problematic. Often individual differences can be resolved manually, but supervising the integration of large data sets—potentially hundreds of thousands of records—makes this approach impractical.

3.2 Method

The strategy applied to detect and consolidate duplicate records is to use unsupervised machine-learning techniques. To that end, this study explores clustering methods. For the past few decades, clustering algorithms have been used for a variety of applications, from image segmentation algorithms (Veksler 2000) to a rainfall forecasting model (Lin, Wu, and Tsay 2017). A clustering method is used to divide or partition the data into groups (clusters) where every object in that group presents more similarities than other objects placed in another group. The clustering of inventory was performed in three steps:

1. Preparing the records data.
2. Converting the records to vectors.
3. Clustering the records and evaluating the results.

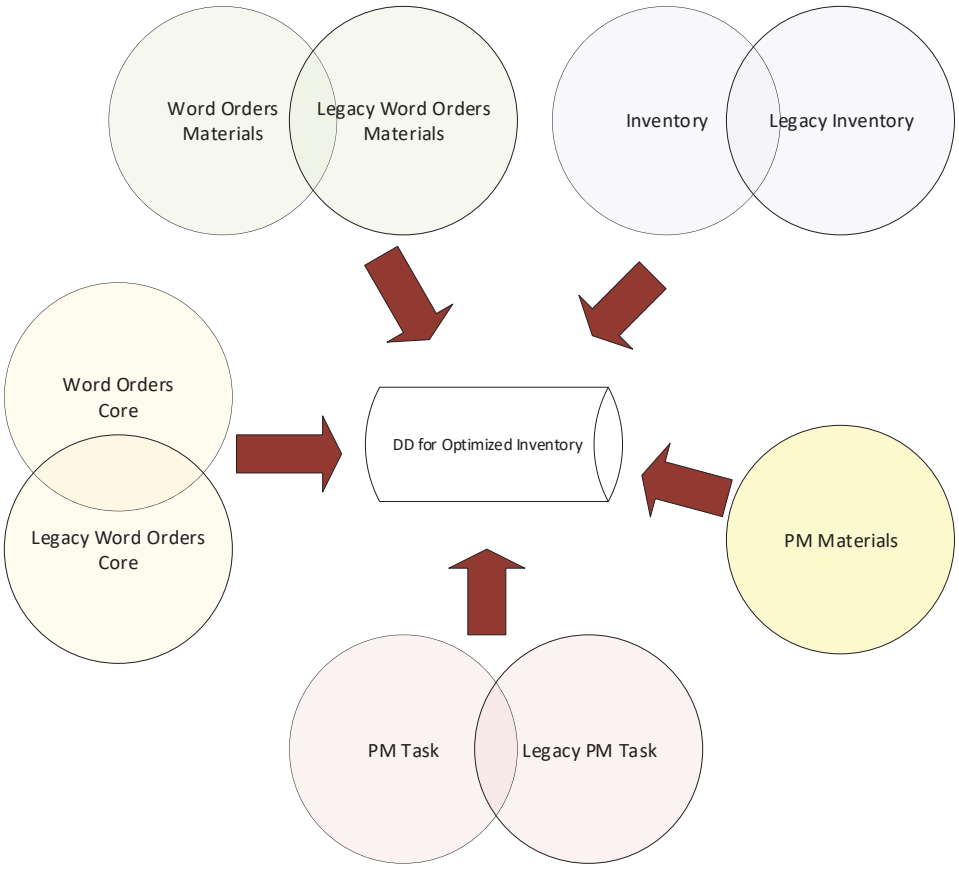


Figure 7. The data integrated to achieve optimal inventory strategy.

3.2.1 Preparing the Data

Manually entered, product descriptions are problematic; they are not full sentences, have no consistent structure, and regularly contain spelling errors and shorthand. Thus, before machine-learning can be performed, some pre-processing and re-structuring of data need be done:

- Records where no description was provided were discarded, as they provide no real purpose.
- All letters in the descriptions were changed to lower case. Since most of the descriptions were entirely in capital letters or entirely in lower-case letters already, very little information was lost.
- Manual bulk cleaning and preparation, including auto-correction of text, was performed by using common methods that rely on finding the root of a word using the Porter stemming algorithm (Porter 1980) (e.g., agreed → agree or falling → fall).

3.2.2 Converting Records to Vectors

To perform clustering on the records (discussed next), the records are transformed into vectors. The first decision made was on the level of clustering to apply. The text can be converted to clusters on the letter level, the word level, or the sentence level (see Figure 8). Because sentence clustering was deemed to be more suitable to compare paragraphs to paragraphs, and the letter level eliminates the impact of letter sequencing, the words level clustering was selected for this effort.

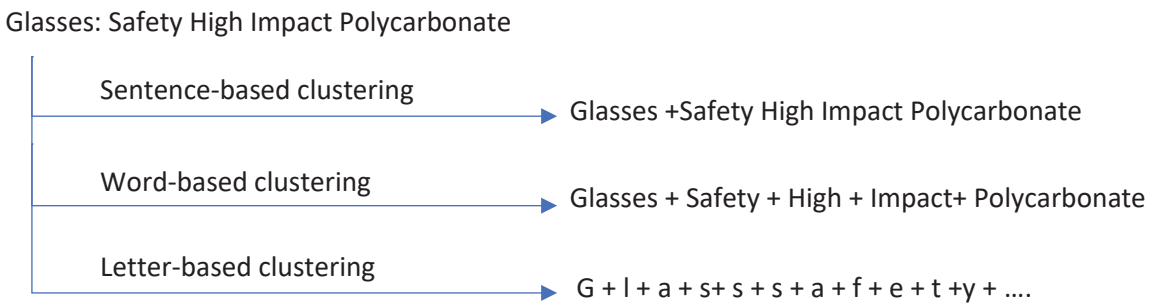


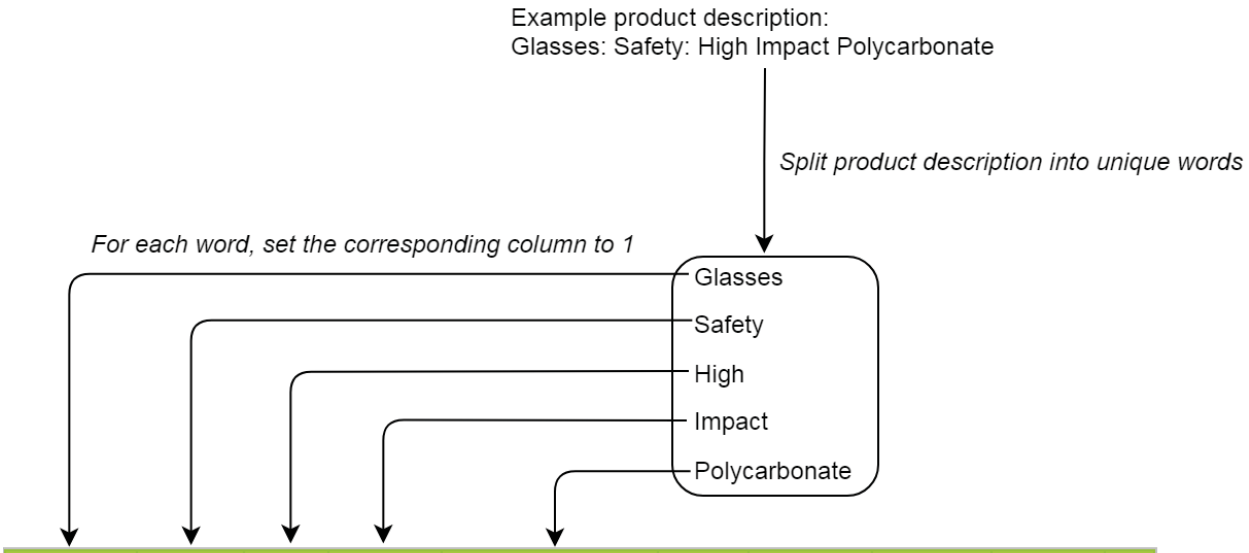
Figure 8. An example of clustering using sentences, words, and letters.

Two different vectorization approaches were tested: term frequency–inverse document frequency (TF-IDF) and count vectorization. TF-IDF is a popular method that gives each word an importance score based on the number of times it appears in a document and how many documents in total contain that word. The count vectorization method involves creating a vector for each record. The length of the vector is equal to the number of unique words in the combined data set, while each entry represents a specific word (see Figure 9). For a given record, the corresponding vector will have a zero in a position where the word is absent and a positive integer indicating the number of times the word appears otherwise. With the record vectors in hand, several methods of clustering the data could be tested. The choice of vectorization method had a significant impact on the quality of the results regardless of which clustering method was used. In general, the vectors formed using count vectorization appeared to perform better than those formed using TF-IDF because product descriptions are not sentences (i.e., they typically have no filler words and TF-IDF aims to increase the value of words that are not present in many other records and likewise decrease the value for common words).

3.2.3 Clustering Records and Evaluate

Clustering is an unsupervised machine-learning method of grouping records that are alike. The aim is that, by grouping the spare-part records, duplicate products are considered as one consolidated record. Data objects within each cluster should exhibit a significant degree of similarity, while the similarity among the different clusters should be minimized. Figure 10 shows an example of a clustering data set that has two dimensions, where the clustering is demonstrated by a color representing a cluster. While it is visually possible to cluster 2-D or three-dimensional (3-D) data, high-dimensional data (i.e., beyond 3-D) cannot be visually represented and are therefore grouped using clustering methods.

For evaluating the clustering methods performance, three techniques can be used to validate the clustering results: (1) external clustering validation; (2) internal clustering validation; and (3) relative clustering validation. Though details can be found in references such as Theodoridis and Koutroubas (2008) and Halkidi et al. (2002), a summary of these techniques are described. External validation is used when comparing the clusters to known external information. Entropy (Shannon et al. 1948) is one of the validation measures that can be used.



Glasses	Safety	High	Impact	Polycarbonate	Wrap	Around	Starlite	Amber	...
1	1	1	1	1	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...

Figure 9. Example of count vectorization illustrating a vector indicating a positive count for the number of times the word appears and zero in the position where the word was absent.

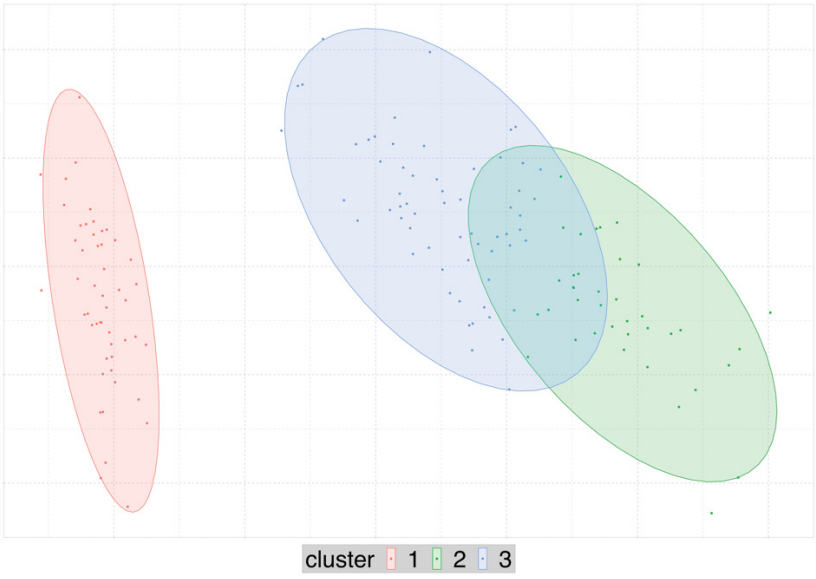


Figure 10. An example of clustering of two-dimensional data.

Internal validation occurs when information of the clustering process is only used to evaluate the integrity of the clustering, such as using the clusters distances as a measure for clustering performance. Dunn Index (Dunn 1974) aims to identify 'compact and well-separated clusters,' with a low variance

between elements of the cluster. If the dataset contains compact and well-separated groups, the distance between the clusters is expected to be large, and the diameter of the clusters is expected to be small. Similar to the Dunn index, the Davies-Bouldin (Davies and Bouldin 1979) index is based on the separability or ratio within and between clusters.

Relative clustering validation is comparing the quality of the clustering by using the external or internal validation; however, the validation process is performed using a criterion that is defined and available for validation. For example, by using K-Means with different cluster instantiations and comparing the clusters number to a predefined expected number, it is possible to use the resulting cluster number as a measure to determine the validity of the clustering process.

The clustering methods tested as part of this effort included K-Means (MacQueen 1967), density-based spatial clustering of applications with noise (DBSCAN, Ester et al. 1996), ordering points to identify the clustering structure (OPTICS, Ankerst et al. 1999), and agglomerative clustering, a subset of hierarchical clustering (Ward 1963). The limitation of the K-Means approach is the requirement that the number of clusters be specified for a data set. For example, if 100,000 records are being considered, K-Means requires estimating how many unique records are going to result from clustering. If exactly four duplicates existed for each record in the set, the number of clusters specified would be 25,000. The challenge is that the number of unique records is not known at the start, and therefore, neither is the number of clusters to use. To assist in selecting the number of clusters, a common technique called 'the elbow method' was used. This is performed by running K-Means over a large range of cluster sizes and then calculating the total within-cluster sum of squares of distance from the center for each run. The resulting values were plotted and the point at which the values stopped decreasing consistently (the elbow of the graph) was used for the final run. The final value found in this case was 1150 clusters. This is clearly inadequate to cover the number of unique spare parts in the set. The algorithm takes significantly increasing time to run as the number of clusters grows, so capturing parts with no duplicates in their own clusters (i.e., large total number of clusters) is not computationally feasible.

The results shown in Table 2 demonstrate the differences between clustering methods. A single product was selected—one with the description "GLASSES:SAFTY:HGH IMP POLYCRBNT"—for demonstration. K-Means clustering was moderately successful, but as shown in Table 2, the clusters typically contained spare parts of the same type (not necessarily those that were actually identical).

DBSCAN is another clustering technique, but unlike K-Means, it does not require a predefined number of clusters. Instead, clusters are determined as the algorithm runs. The method starts by selecting a random record that is not yet part of a cluster. It then checks how many other records are within some distance, ε, of the start. If the number of records nearby is less than the required minimum, specified when the algorithm is started, the record is marked as singular. Otherwise, the starting record and each of the nearby records is placed into a cluster together, and the process is repeated for each of the newly added records. Once the cluster is complete, the process is restarted, using another not-yet-considered record until all records have been placed into clusters or marked singular. Given the type of clusters targeted as part of this effort (i.e., duplicate products), any cluster size is acceptable. That is, products could be unique (a cluster size of one) or a cluster could be any number of duplicates (a cluster size of two or more). Singular records are captured by the algorithm through non-clustering, so the minimum number of records used for the algorithm to continue with a cluster was selected to be two. The distance allowed between samples, for them to be considered grouped, is more difficult to select. An elbow test like the one used for K-Means was used to select the ε value although, in this case, the test was based on a histogram of distances to the nearest neighbor, as shown in Figure 11. The results of the clustering were more promising than those obtained using K-Means: approximately 80% of the records ended up in their own clusters, implying that they may be unique products, and the remaining clusters had between two and ten records.

Table 2. Example of clustering results for the record “GLASSES:SAFTY:HGH IMP POLYCRBNT” using four methods.

K-Means

Material Description
GLASSES:SAFTY:POLYCRBNT,WRAP ARND
GLASSES:SAFTY:POLYCRBNT,WRAP ARND
SAFTY GLASSES:POLYCRBNT:HGH IMP
GLASSES:SAFTY:HGH IMP POLYCRBNT
GLASSES:SAFTY:HGH IMP POLYCRBNT
GLASSES:SAFTY:POLYCRBNT,CLR,VAPOR BLU
GLASSES:SAFTY:POLYCRBNT,SPLSH GOGLS,CLR
GLASSES:SAFTY:POLYCRBNT,CLR,STARLITE
GLASSES:SAFTY:POLYCRBNT,AG MIR,GRA
GLASSES:SAFTY:ESPRESSO POLYCRBNT
GLASSES:SAFTY:POLYCRBNT,CLR,RED,WHT,BLU
GLASSES:SAFTY:POLYCRBNT,CLR
GLASSES:SAFTY:POLYCRBNT,DIR VNT DST GOGL
GLASSES:SAFTY:POLYCRBNT,IMP GOGLS,NYL
GLASSES:SAFTY:POLYCRBNT,CLR,ANTI-FOG
GLASSES:SAFTY:POLYCRBNT,GRA,GRA,STARLITE
GLASSES:SAFTY:ESPRESSO POLYCRBNT

Agglomerative

Material Description
GLASSES:SAFTY:POLYCRBNT,WRAP ARND
GLASSES:SAFTY:POLYCRBNT,WRAP ARND
SAFTY GLASSES:POLYCRBNT:HGH IMP
GLASSES:SAFTY:HGH IMP POLYCRBNT
GLASSES:SAFTY:HGH IMP POLYCRBNT
GLASSES:SAFTY:POLYCRBNT,CLR,STARLITE
GLASSES:SAFTY:ESPRESSO POLYCRBNT
GLASSES:SAFTY:POLYCRBNT,CLR
GLASSES:SAFTY:POLYCRBNT,IMP GOGLS,NYL
GLASSES:SAFTY:ESPRESSO POLYCRBNT

DBSCAN and OPTICS

Material Description
SAFTY GLASSES:POLYCRBNT:HGH IMP
GLASSES:SAFTY:HGH IMP POLYCRBNT
GLASSES:SAFTY:HGH IMP POLYCRBNT

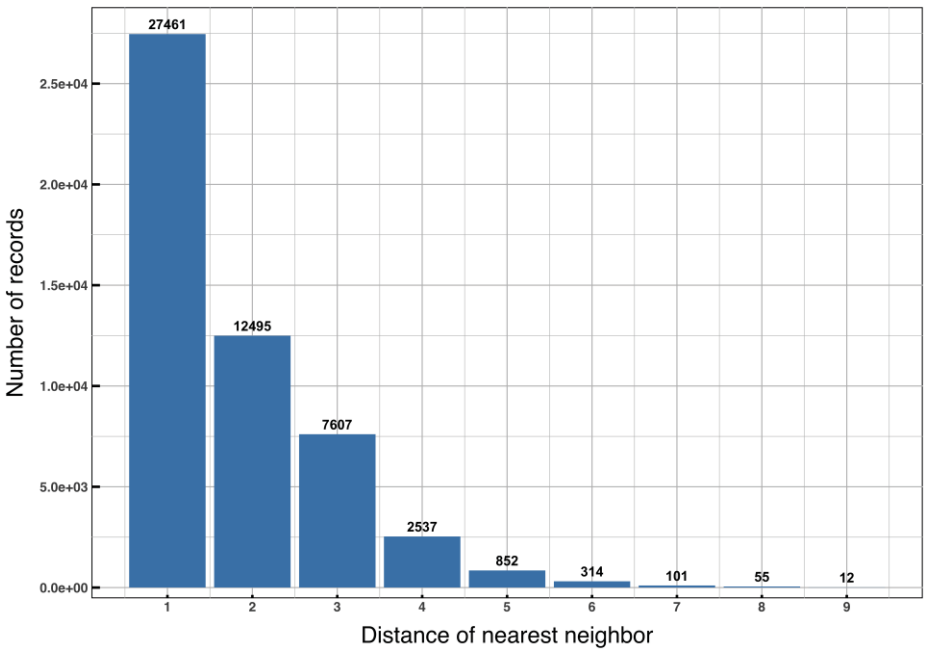


Figure 11. Elbow test for DBSCAN clustering.

The OPTICS method is similar to DBSCAN and uses the same parameters, ε, and a minimum number of records. The method calculates a reachability distance between each record and every other record that is within ε distance. A reachability distance value from a start record to another record is the larger of either the distance between the two or the radius of a circle surrounding the first record that would capture the minimum number of records specified. These reachability distances are then plotted and used to identify cluster boundaries. The practical upshot is that OPTICS better factors in the density of records when determining whether or not a record should be added to a cluster. Choosing parameters is much simpler with OPTICS. In general, the ε value serves only to speed up the calculation. Optimal results can be obtained by leaving ε as a very large number (i.e., the maximum possible value) as long as running time is not an issue. The strategy for picking a minimum number of records in a cluster is identical to DBSCAN. The results of the clustering were similar to DBSCAN, with about 76% of the products ending up clustered singularly. The cluster containing the product, "GLASSES:SAFTY:HGH IMP POLYCRBNT," was identical between this method and DBSCAN, as shown in Table 2.

Agglomerative clustering requires each record forming its own cluster, and then merging nearby clusters until either a specified maximum distance between clusters is met or a maximum number of clusters is reached. Just as with K-Means, selecting a maximum number of clusters is not feasible for this data set; instead, priority was placed on identifying a valuable maximum distance. This is identical to the ε value for DBSCAN, so the previously performed elbow test result was reused. This method performed slightly better than K-Means, but still fell short when compared to DBSCAN and OPTICS, as shown in Table 2.

Given the strict similarity requirements of consolidated records, clustering using DBSCAN and OPTICS was deemed more suitable than using K-Means and Agglomerative clustering, as seen from Table 2, but visual examination showed that some of the larger clusters still occasionally contained different products of the same type. This is an expected outcome because it is practically impossible to achieve 100% accuracy. Table 3 shows an example of a case where similar products with different specifications were incorrectly grouped together. Though not presented in this effort, a supervised machine learning method can be added on top of the unsupervised clustering to identify misclustering. This work remains for future efforts.

Table 3. Example of misclustering using DBSCAN.

Material Description
INDICATOR:TEMP:0 TO 752DEG F
INDICATOR:TEMP:0 TO 300DEG F
INDICATOR:TEMP:0 TO 200DEG F
INDICATOR:TEMP:0 TO 400DEG F
INDICATOR:TEMP:0 TO 1200DEG F
INDICATOR:TEMP:0 TO 1400DEG F
INDICATOR:TEMP:0 TO 600DEG F

4. CONCLUSIONS FOR A DATA WAREHOUSE OR BROKER

From the search engine pilot, it was concluded that creating a DD, as part of a data warehouse, can represent one means of creating a data hub for multiple tools to use. The process highlighted the need for SMEs to walk the data integrator through the process. Each SME was an expert in a subset of systems and the process was manual and time-consuming, despite this being a simple pilot. A standard data model for a data warehouse or data broker would help streamline the process. Some specific findings are:

- A standard data model would likely have the fields that are most critical to users already identified. In order to conform to the standards, the data would likely already be transformed and cleansed, eliminating some of the issues described above. The DD part of the search tool would no longer be needed as the staging tables. The SSS part of the tool would access the standard data model tables. The developed search approach can be easily installed at any site that utilizes a standard data model with minimal efforts. With a standard data model, the heavy lifting is all done populating those tables.
- While the standard data model would contain the bulk of the critical fields, minor additions or adjustments would be required. Users often identify some of these critical fields after utilizing a product. Having a standard data model simplifies the new additions.
- The different tools will still likely have a small amount of critical data outside of the standard data model. The standard data model would be interfaced to the plant-specific data objects by deploying a parallel DD model, where ETL would be used to create the needed associations to the data model.
- The developed search tool serves as a launch point for identifying fields critical to business users. Given each user and each NPP has varying needs, the approach described is one way to uncover the variety of fields NPPs need in a standard data model. Deploying this search solution at one test site and allowing users to interact with the system result in the implementation of refinements, including the removal and addition of fields and data sources. Once the test users are satisfied with the solution, the same solution is deployed at several other sites and implements the same refinement process. A comparison of the final solutions at each site should reveal similarities and differences, allowing conclusions to be drawn regarding which fields are needed in a standard data model.

From the data fusion pilot, it was concluded that accumulating data in a warehouse requires intelligent mapping methods to make use of the data. Whether to map or couple data from different sources or to prepare data (such as clean up duplicates from the same data set), machine learning can improve the data integration by fusing data into optimized sets that are more meaningful for using the data. Though not presented, the approach developed in the duplicate inventory removal pilot can also be used for the search tool pilot. Mapping hundreds of tables was deemed to be challenging in the

development of the search tool and resulted in high reliance on SMEs. An unsupervised machine-learning method could have identified similar table headers and automatically mapped them.

5. SUMMARY

Integrating data can benefit virtually every aspect of NPP activities on the plant and fleet levels. The integration can result in direct cost-savings by enabling comparison of multiple studies for verification, increasing statistical confidence, increasing data heterogeneity in space and time to reveal confounded information, increasing frequencies of rare occurrences, better methods development, and enabling data-sharing. Data integration can also enable the creation of a data repository (i.e., a data warehouse for an NPP or a data broker for a fleet), a necessary backbone for all processes of an automated plant.

Data integration can be achieved from an application, architecture, and data management perspective or from a data fusion perspective. A pilot was performed targeting each type in this effort. The first pilot integrated data from four data sets into a DD that was accessed through a search engine and a webserver. The resulting pilot was a search tool that spanned over the four data sets to enable an SDM to access needed data that are currently inaccessible. This use case was chosen out of a set of potential use cases that can facilitate the exchange of information among plant staff.

The second pilot was performed to enable NPPs to reduce inventory minimum requirements by predicting inventory use using machine pattern recognition. The pilot was focused on using machine-clustering methods to detect inventory duplicates (often listed under different descriptions) and to consolidate them. This is a necessary step before analyzing inventory use patterns because duplicate parts are often used interchangeably, and skipping this step would result in inaccurate inventory use prediction methods. Four methods were tested to achieve this objective: K-Means, DBSCAN, OPTICS, and Agglomerative. DBSCAN was deemed to be suitable.

Both pilots developed as part of this effort will feed into the development of the data warehouse or data broker through: (1) use of data integration methods to automate and streamline the data warehousing and data mapping; and (2) provide additional insight and use-cases for establishing a standardized data integration model and ontology.

6. REFERENCES

Al Rashdan, A., and S. St. Germain, 2019. "Methods of data collection in nuclear power plants," *Nuclear Technology*, 1–13, 31 May 2019.

Ankerst, M., M. M. Breunig, H. P. Kriegel, and J. Sander, 1999. "OPTICS: Ordering points to identify the clustering structure," in *ACM SIGMOD Record*, 28(2) 49–60.

Correa, N. M., T. Adali, Y. O. Li, and V. D. Calhoun, 2010. "Canonical correlation analysis for data fusion and group inferences," *IEEE Signal Processing Magazine*, 27(4) 39–50.

Curran, P. J., and A. M. Hussong, 2009. "Integrative data analysis: The simultaneous analysis of multiple data sets," *Psychological Methods*, 14(2) 81.

Davies, D. L., and D. W. Bouldin, 1979. "A *Cluster Separation Measure*," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2) 224–227.

Dunn, J. C., 1974. "Well-separated clusters and optimal fuzzy partitions." *Journal of Cybernetics*, 4(1) 95–104.

Ester, M., H. P. Kriegel, J. Sander, and X. Xu, 1996. "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, 96(34) 226–231.

Halkidi, M., Y. Batistakis, and M. Vazirgiannis, 2002. "Cluster validity methods: Part I," *ACM SIGMOD Record*, 31(2) 40–45.

Hall, D., and J. Llinas, 1997. "An introduction to multisensor data fusion," *Proceedings of the IEEE*, 85(1) 6–23.

Lin F., N. Wu, and T. Tsay, 2017. "Applications of cluster analysis and pattern recognition for typhoon hourly rainfall forecast," *Advances in Meteorology*, 2017, Article ID 5019646, 17 pages.

MacQueen, J., 1967. "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14) 281–297.

Moore, W. J., and A. G. Starr, 2006. "An intelligent maintenance system for continuous cost-based prioritisation of maintenance activities," *Computers in Industry*, 57(6) 595–606.

Porter, M. F., 1980. "An algorithm for suffix stripping," *Program*, 14(3) 130−137.

Shannon, C. E., 1948. "A mathematical theory of communication," *The Bell System Technical Journal*, 27(4) 379–423.

Smilde, A. K., M. J. van der Werf, S. Bijlsma, B. J. van der Werff-van der Vat, and R. H. Jellema, 2005. "Fusion of mass spectrometry-based metabolomics data," *Analytical Chemistry*, 77(20), 6729–6736.

Sorber, L., M. Van Barel, and L. De Lathauwer, 2015. "Structured Data Fusion," *IEEE Journal of Selected Topics in Signal Processing*, 9(4) 586–600.

Theodoridis, S, and K. Koutroubas, 2008. *Pattern Recognition*. 4th edition. Academic Press: New York, NY, USA.

Van Mechelen, I., and A. K. Smilde, 2010. "A generic linked-mode decomposition model for data fusion," *Chemometrics and Intelligent Laboratory Systems*, 104(1), 83–94.

Veksler, O., 2000. "Image segmentation by nested cuts," in *Proceedings of the IEEE CS Conf. Computer Vision and Pattern Recognition*, (1) 339–344.

Ward, Jr, J. H., 1963. “Hierarchical grouping to optimize an objective function,” *Journal of the American Statistical Association*, 58(301) 236–244.

White, F. E., 1991. Data Fusion Lexicon, Joint Directors of Laboratories, *Technical Panel for C3, Data Fusion Sub-Panel*. Naval Ocean Systems Center, San Diego, CA, USA.

Wilderjans, T., E. Ceulemans, and I. Van Mechelen, 2009. “Simultaneous analysis of coupled data blocks differing in size: A comparison of two weighting schemes,” *Computational Statistics & Data Analysis*, 53(4), 1086–1098.