From Data to Knowledge: A Graph-Based Reliability Approach to Assess System Health

C. Wang[a], D. Mandelli[a], C. M. Godbole[a], V. Agarwal[a], M. Movassat[b], B. Mori[b], D. Liang[b], E. Nur[b], A. Birjandi[b], B Lobo[b], N. Jacome[b]

[a] *Idaho National Laboratory, Idaho Falls, USA*

[b] *Ontario Power Generation, Toronto, Canada*

Abstract: With the goal of maximizing plant reliability and availability, complex systems such as nuclear power plants continuously monitor and record the performance and health status of many components, assets, and systems. Such data may take the form of online monitoring data, condition reports, and maintenance reports, and they can provide system engineers with insights into anomalous behaviors or degradation trends as well as the possible causes behind them and predict their direct consequences. The analysis of such data however poses a few challenges. While some of these challenges are technical in nature (i.e., data are often distributed over several physical servers or databases), others are conceptual (i.e., data elements come in different formats, numeric or textual) and measured values have different scales (e.g., vibration spectra and oil temperature). This paper directly tackles these challenges and focuses on integrating all these data elements to assist plant system engineers in analyzing component, assets, and systems performances and optimizing maintenance activities. This is performed by extracting knowledge from textual data via technical language processing methods and quantifying system, asset, and component health from numeric condition-based data. We rely on model-based system engineering (MBSE) models of systems and assets to identify their architecture and functional (i.e., cause and effect) relations. Numeric and textual data elements are then associated with an MBSE graph element, based on their nature. This bonding of MBSE models and data elements constitutes a first-of-its-kind knowledge graph of a nuclear power plant system, with data elements being organized in a structured manner that enables system engineers to identify cause-effect trends in data elements and carry out appropriate actions in response.

Keywords: Natural language processing, condition monitoring, MBSE.

1 INTRODUCTION

The rapid development and deployment of advanced condition-based monitors and data analytics techniques (e.g., anomaly detection, diagnostic, prognostic methods) is considerably helping system engineers and plant operators to monitor the performance of several assets that constitute complex systems. Similarly, digitizing operation and maintenance activities allows them to keep track of events at the system or plant level (e.g., plant planned shutdown or system taken out of service) and, more important, observed asset abnormal conditions and operations that have been performed on such assets (Coble, 2015; Xingang, 2021). As a drawback, engineers and operators are now facing the challenge of processing the amount of equipment reliability (ER) data being continuously generated, which is not only extremely large but also appears in different forms: textual and numeric.

This paper addresses this challenge by presenting methods to assist engineers and operators in extracting knowledge from ER data. The first point we claim here is that all the ER data elements described earlier equally provide indications about asset and system performance, and, hence, they cannot be analyzed separately. The second claim is that generating knowledge from data requires the ability to put data into "context." Here, context is the additional piece of information needed by ER data analysis tools to understand what these data elements are referring to.

Here, we employ model-based system engineering (MBSE) models of systems and assets to capture their architectural and functional (i.e., cause-effect) relations. With that, ER data elements (both textual and numeric) are processed by identifying first which elements of the developed MBSE elements they are referring to. For numeric ER data, this task is fairly easy while we employ technical language processing (TLP) methods to "extract knowledge" from textual elements. Filtering abnormal behaviors can then be performed from numeric (through anomaly detections and diagnostic and prognostic methods) and textual elements (by understanding their semantic nature). Such abnormal instances, which are associated to a specific MBSE element, are then stored in a relational database. Such a database takes the form of a graph where the main skeleton is the actual system MBSE model and abnormal instances are "linked" to such skeleton. At this point, both numeric and textual data elements are integrated and put into context. From here, graph-based analysis methods can be employed to perform "machine reasoning," which include identifying abnormal patterns and the root cause behind such patterns.

Since this work was performed in collaboration with a nuclear utility, the data elements and corresponding figures reported in this paper have been intentionally altered to hide proprietary information.

2 CONSIDERED SYSTEM

The system under consideration in this paper is the circulating water system (CWS) of an existing nuclear power plant. Typically, this system is used in many types of power plants (e.g., coal, gas, oil) and is designed to remove the residual heat from the turbine-condenser system and release it into the environment.

In our case, water is collected from a body of water (e.g., lake or a river) through service gates. Then, using traveling screens, the water is cleaned of debris, water life, and foreign bodies that might damage CWS components. Screen wash pumps provide spray water to remove accumulated debris on the screens. The CWS also contains a vacuum priming system that removes any air from the system. Then, water is pumped through heat exchangers located in the plant secondary loop and removes heat from the turbine-condenser system. Lastly, the same water is then released downstream of the same body of water. Depending on the environmental conditions, a portion of this water is released back into the service gates to avoid any ice formation that would block water flow. Several systems support the CWS, such as alternating current systems (4,160 and 480 V) and water-cooling systems.

From an operational standpoint, even though the CWS does not directly support a plant safety function, any performance degradation or abnormal behaviors may directly affect power generation (either in power derate or power shutdown) and, consequently, plant economic revenues.

3 ER DATA

Based on the considerations presented in Section 2, CWS operational conditions are continuously monitored in order to detect early signs of degradation and proactively perform maintenance to restore system operations and guarantee system availability. In this respect, Table 1 provides a list of the available monitoring variables collected over the past decade; note these variables not only provide indications of performance of the CWS pumps and condenser but also of systems interfacing with the CWS. Note that plant environment variables are also available (water body and air temperature); Section 6 provides considerations about the importance of environment variables to remove seasonal (i.e., periodic) trends from CWS plant monitoring variables when performing anomaly detection.

Table 1. List of CWS monitoring variables.

Variable IDs	Description
$x_1^{pump,unit}, \dots, x_4^{pump,unit}$	Monitoring variables associated with CWS pumps of a specific plant unit
$x_5^{cond,unit}, \dots, x_9^{cond,unit}$	Monitoring variables associated with the condenser of a specific plant unit
$x_{10}^{unit}, \dots, x_{12}^{unit}$	Monitoring variables associated with systems interfacing with the CWS
T_{water} and T_{air}	Plant environment variables

In addition to the numeric data described in Table 1, the considered nuclear power plant has also recorded in its databases all operational events as follows:

- *Reactor operator shift logs*: major events that have caused deviations from normal plant operations;

- *CWS condition reports*: abnormal events that occurred in the CWS;
- *CWS work orders*: maintenance operations performed to the CWS;
- *Plant outage data*: time instances where the plant was shut down for either planned or unplanned outages.

Note that all these events recorded in textual form (while the data elements described in Table 1 are in numeric form) provide indications not only about the historical reliability performance of the CWS but also precise information about the nature of the recorded abnormal events and corresponding operations performed to restore CWS operations.

Lastly, the provided design documents gave us precise information about the architecture and functional relations between the CWS, the rest of plant, and the assets that are part of the CWS.

4 ANALYSIS OF ER DATA

Figure 1 shows our approach to process and analyze CWS historical performances provided the ER data elements (numeric and textual) described in Section 3. Constructing the knowledge graph starts by performing four different workflows:

- *Step 1: MBSE workflow.* System architecture information provided by plant and CWS design documents is translated into MBSE models (see Section 5).
- *Step 2: Numeric workflow.* CWS anomalies are inferred from CWS numeric monitoring data (see Section 6).
- *Step 3: Textual workflow.* CWS related events reported from operator shift logs, conditions, or maintenance reports are processed using TLP methods (see Section 7).
- *Step 4: Event to time series correlation analysis.* Based on the temporal occurrence of the inferred anomalies (see Step 2) and reported events (see Step 3), we test whether the occurrence of these events had a cause-effect relation with observed monitoring data.
- *Step 5: Knowledge graph construction.* The construction of the knowledge graph starts by translating the system MBSE model into a graph structure where each node of this graph is a physical entity of the CWS (e.g., pump, traveling screen). Each edge in such a graph represents a physical connection between two entities where the nature of such connection can be of different types (mechanical, electrical, hydraulic, digital). Then, the set of anomalies derived from Step 2 and the events processed in Step 3 are digitally associated with one (or more) node of the graph derived from the MBSE model (see Section 5).

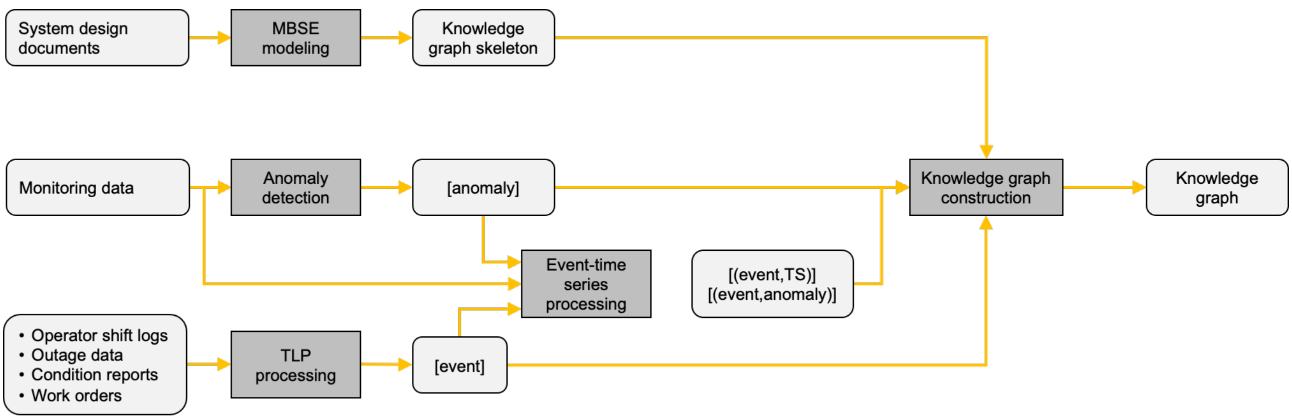

Figure 1. From ER data to knowledge graph: a graphical description of the workflows. Analysis methods are highlighted in dark grey while generated and input data is highlighted in light grey.

Table 2. Functional description of workflows designed to construct a knowledge graph from ER data.

#	Workflow	Input	Output
1	MBSE modeling	Design documents describing system architecture (form and functional description)	MBSE models for the considered system and derived graph structure
2	Anomaly detection	System monitoring data (labeled or unlabeled)	List of inferred anomalies
3	TLP processing	Textual event data (e.g., operator shift logs, condition reports, maintenance reports)	Graph representation of event
4	Event to time series correlation analysis	System monitoring data, anomalies identified in Workflow 2, events processed in Workflow 3	List of events correlated to time series variations; list of events correlated to identified anomalies
5	Knowledge graph construction	Data elements generated in Workflows 1, 2, 3, and 4	Graph structure

5 SYSTEM DIGITAL REPRESENTATION

From a reliability standpoint, it is vital to identify the causal relationships among ER data, maintenance activities, and failure modes. This is typically neglected in the state of practice in current ER data analysis methods based on machine learning (ML) methods. This limitation is due to the fact that data are not enough to identify such causal relationships. Instead, system models are needed to perform such a discovery.

In this respect, MBSE (Borky, 2018) methods afford several solutions for modeling systems, assets, and components from both a *form* (i.e., which elements are part of the structures, systems, and components) and a *functional* (i.e., how systems and assets interact with each other, and which functions they support) standpoint. These solutions are based on MBSE languages that represent system and asset form and functional elements via a set of diagrams. The most commonly used languages are Object-Process Methodology (OPM) (Dori, 2002), Lifecycle Modeling Language (LML) (LML, 2022), and Systems Modeling Language (SysML) (Friedenthal, 2008).

For the scope of this project, we have chosen LML and OPM since they provide the basic modeling elements we sought and because—more importantly—digital data structures (i.e., graphs) can be automatically generated from LML and OPM diagrams. Each element of an OPM and LML diagram can be either a *function* (e.g., an action or a transformation) or *form* (e.g., a physical entity) element. In addition, function and form elements in an OPM diagram are connected to each other through a set of *links* designed to convey precise meanings (Dori, 2002).

Figure 2 shows the LML diagram of the considered CWS. Note that each asset included in the LML diagram of the CWS may be further described by its own separate LML or OPM diagram. In other words, a network of LML and OPM diagrams can be constructed to refine and further detail the architecture of the considered system. For example, in the CWS LML diagram in Figure 2, the centrifugal pumps are indicated as pertaining to a different OPM diagram that represents the pump architecture in greater detail.

6 ANOMALY DETECTION METHODS

The amount of anomaly detections (applied to any scientific or technological context) available in the open literature is vast, and it is not within the scope of this paper to provide an exhaustive overview of such methods to compare performances for the considered system. Such methods can rely on classical statistical, ML, or deep learning methods with different pros, cons, and ranges of operability. The main requirements for the choice of anomaly detection methods were fast computation, ability to deal with period patterns and missing data, ability to identify anomalies defined over time instance or time intervals, scalability, and interpretability.

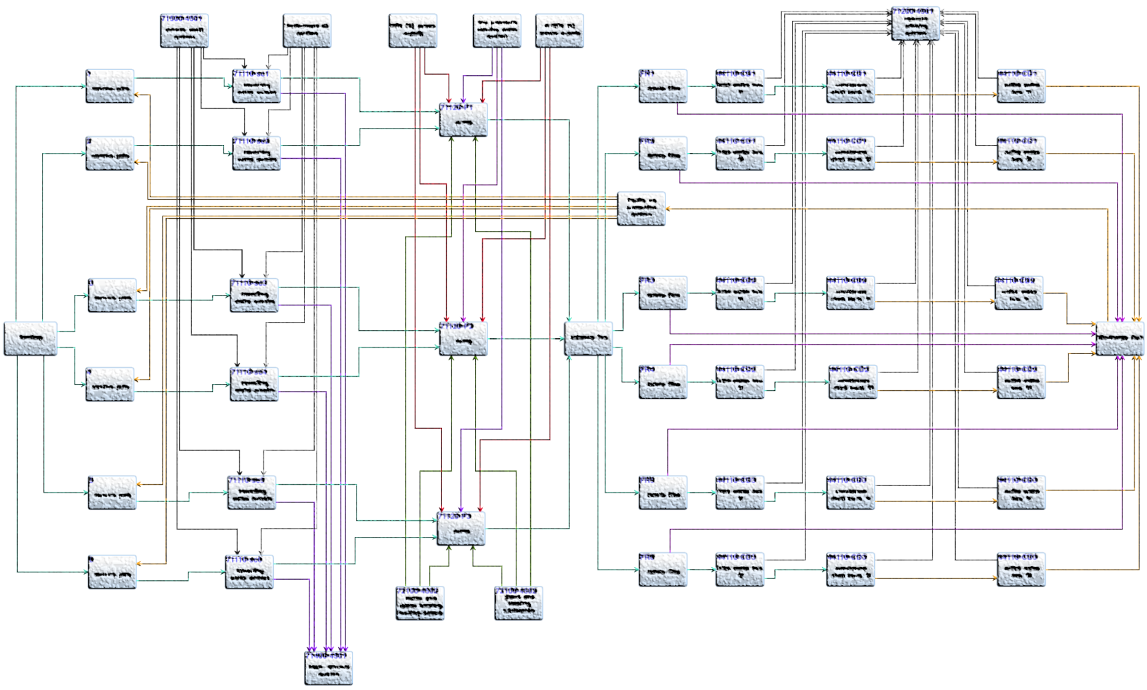

Figure 2. LML model of the considered CWS. The figure has been intentionally edited graphically to hide any proprietary information.

Given these requirements, we have chosen to build the anomaly detection methods based on the matrix profile algorithm (Yeh, 2016). In simple terms, this algorithm is a distance-based approach over a sliding window; here, the considered time series is progressively scanned by identifying the smallest distance between the portion of the time series limited within the considered time window and the set of time windows previously processed. An example of anomalies detected is shown in Figure 3 where the matrix profile algorithm has been applied to the time series of two monitored variables of the CWS. Here, two time series are considered (shown in blue in Figure 3), T_{water} and X_1^{pump}, and the corresponding temporal matrix profiles are shown in red in Figure 3. Anomalies are identified by looking at the regions characterized by high values of the matrix profiles.

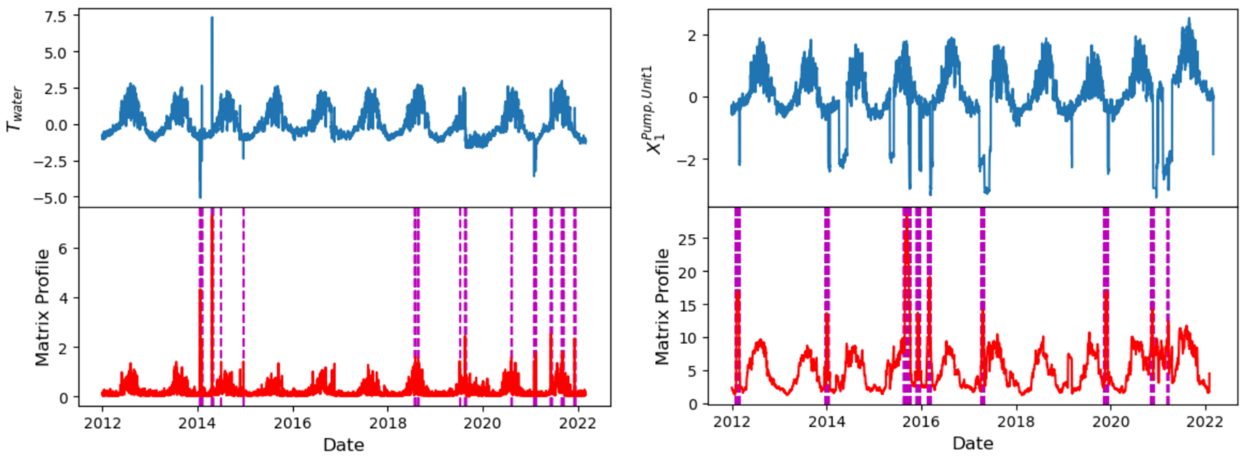

Figure 3. Example of anomalies detected by matrix profile algorithm when applied to the time series of the monitored variable X_N of the CWS.

Once an anomaly is detected, it is digitally recorded by observing the time interval under which it is detected and the set of variables employed to detect it. More precisely, a generic anomaly `An` is defined as a specific entity, which is defined as `An =([vars],` $\mathtt{t}_{\mathtt{in}}$, $\mathtt{t}_{\mathtt{fin}}$`)` where `[vars]` corresponds to the list of variables under which the anomaly was observed, while $\mathtt{t}_{\mathtt{in}}$ and $\mathtt{t}_{\mathtt{fin}}$ define the temporal duration of such an anomaly.

7 TLP KNOWDLEDGE EXTRACTION METHODS

Issue reports (IRs) and work orders (WOs) are valuable data sources for tracking asset health histories, identifying health trends, and performing root-cause analyses. These data sources, typically obtained in text form, are usually available in digital repositories. Natural language processing methods (Lane, 2019) have been developed over the past two decades to enable ML models to analyze textual data and classify textual elements based on their nature (e.g., safety-related vs. non-safety-related). In the context of the present work, we are not interested in solving any type of classification problem but rather in extracting actual knowledge from textual data. This is a harder task, as it requires the development of context-dependent models and vocabularies. The medical field is leading the way in this area by developing methods to extract knowledge from textual data (e.g., for diagnostic purposes or to estimate the performance of specific treatments). When applied to the nuclear field, knowledge extraction consists of several tasks, including identifying:

- Plant-specific entities, such as systems, assets, and components (e.g., centrifugal pump, accumulator system, and pump shaft)
- Temporal attributes that characterize events (e.g., the occurrence, duration, and order of events)
- Measured quantities (i.e., a numeric value followed by unit of measure)
- Phenomena (e.g., material degradation or asset functional failure)
- Causal relations between events.

This process of knowledge extraction is enabled by a series of data, models, and methods. The developed series of TLP methods was designed to identify all elements listed above, using a mixture of rule-based and ML algorithms. These methods (Wang, 2024) heavily rely on data dictionaries and plant, system, and asset models. Data dictionaries containing a large number of keywords related to the nuclear field were partitioned into several classes (e.g., materials, chemical elements and compounds, degradation phenomena, and electrical, hydraulic, and mechanical components).

The ability of system engineers to analyze textual data is enabled by their knowledge of the architectural scheme of the components and assets that comprise the system. In simpler terms, they know what physical elements comprise a given asset or system, along with their functional relations and dependencies. Without such information, knowledge extraction from textual data is very difficult, as putting the text into context becomes much harder. For the present study, our methods were designed to check whether OPM entities (see Section 3) are mentioned in ER textual data elements.

Figure 4 provides an example of knowledge extraction from an ER textual data element. Based on the developed libraries, the asset (i.e., pump) and reactions (i.e., cracking and failure) mentioned in the text are identified, along with a specific pump MBSE entity (i.e., shaft). Furthermore, additional elements are captured: the existence of a conjecture and the temporal attribute associated with pump failure.

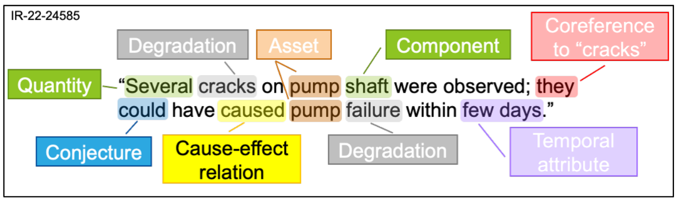

Figure 4. Example of natural language processing knowledge extraction from an ER textual data element.

8 EVENT TO TIME SERIES CORRELATION ANALYSIS

Sections 6 and 7 have presented methods of analyzing numeric and textual ER data elements, and we explained how MBSE diagrams can be employed to identify possible causal relationships between ER data elements. The word "possible" is intended to indicate that two events sharing an OPM-based direct relation may in fact exist independently from each other. The first step in testing such dependence is to observe their temporal correlation. Our work extends that presented by Luo (2014), in which the temporal correlation between time series and events is formulated in terms of a two-sample problem (Gretton, 2006). Our extension includes three relevant items: a modification to the testing process structure, a different two-sample testing algorithm, and the handling of events defined over an interval (as opposed to a time instant).

In its original formulation by Luo (2014), the temporal correlation was measured between a set of identical events and the time series. In the scope of the present work, we often deal with single events (e.g., abnormal behavior of an asset) rather than sets of events. The algorithm presented by Luo (2014) was based on measuring the statistical difference between the portions of the time series pertaining to both before and after (indicated as l_E^{front} and l_E^{rear}, respectively [see the left-hand plot in Figure 5]) an event (as defined over a temporal instant). Our extension, which enables dealing with events defined over a temporal interval (see the right-hand plot in Figure 5), requires the additional time series portion that corresponds to the duration of the event itself: l_E^{dur}. We employed the maximum mean discrepancy algorithm (Gretton, 2006) to perform such testing.

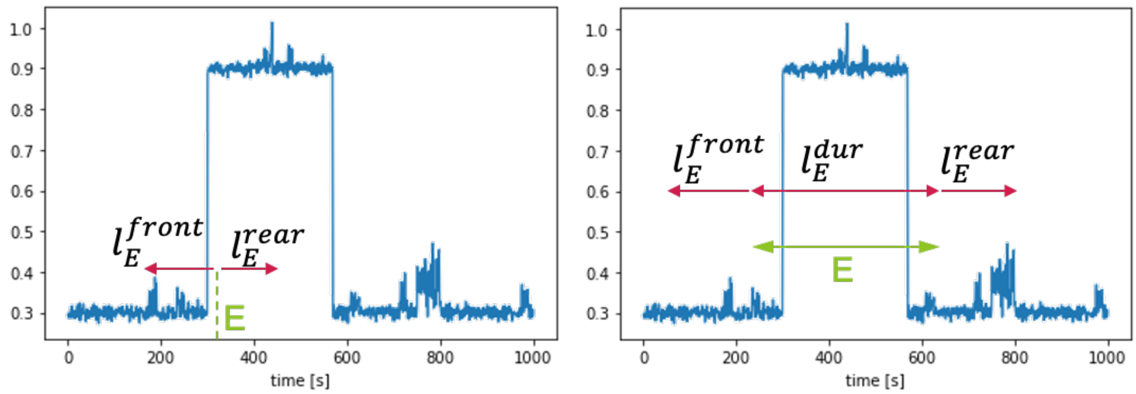

Figure 5. Temporal correlation of a time series with an instantaneous (left plot) and interval (right plot) event E.

Algorithm 1: Identification of the temporal relation between event and time series—time instant case
Input: Event (E, t_E), time series $S = (s_1, s_2, \dots, s_m)$
Output: Temporal correlation flag R, direction D

1. Initialize Θ
2. Determine l_E^{front}, l_E^{rear} from t_E
3. Test $T\left(\Theta, l_E^{front}\right)$, and $T\left(\Theta, l_E^{rear}\right)$
   - Results are denoted as: D_f, D_r
4. Test $T\left(\Theta, l_E^{front} \cup l_E^{rear}\right)$
   - Result is denoted as D_{fr}
5. Test $T\left(l_E^{rear}, l_E^{front}\right)$
   - Result is denoted as d_{fr}
6. If $D_r = True$ & $D_f = False$: #E1
   - Return $R = True$ and $D = E \rightarrow S$
7. Elif $D_r = False$ & $D_f = True$: #E4
   - Return $R = True$ and $D = S \rightarrow E$
8. Elif $D_r = True$ & $D_f = True$: #E2 and E3
   - Return $R = True$ and $D = S;E$
9. If $D_{fr} = True$:
   - Return $R = True$ and $D = S;E$
10. Elif $d_{fr} = True$:
    - Return $R = False$ and $D = S?E$
11. Else $d_{fr} = False$:
    - Return $R = False$ and $D = S!E$

The plots in Figure 5 were adapted from those by Luo (2014) and provide an overview of the set of cases observable when testing the temporal correlation between time series and events. When indicating the time series with S, we can look at the left-hand plot in Figure 5 and intuitively infer that $E_1 \rightarrow S$, $S \rightarrow E_2$, $E_3 \rightarrow S$, and $S \rightarrow E_4$. Note that the symbol $\rightarrow$ here indicates a temporal relationship between an event E and S but does not necessarily imply a causal relationship between the two. Without a loss of generality, let us consider an event E—defined over either a time instant (E, t_E) or time interval $(E, t_E, \Delta t_E)$—and a time series S (either univariate or multivariate). Algorithm 1 presents the identification of the temporal relation between E and S is presented in detail.

An example of a correlation analysis of events and time series is shown in Figure 6 where a monitored variable is correlated to a set of events processed in Section 7. The identified events that have a temporal correlation with the time series are indicated red, black, yellow.

Lastly, note that the reported time of occurrence of an event is assumed to reflect the actual temporal occurrence of that event. More specifically, the reported occurrence of an event (e.g., sudden bearing failure of a pump) is logged when the event is first observed; however, the actual event may have occurred prior to the logged date (i.e., a temporal delay may exist between the actual and the observed occurrence of an event). In such situations, the analysis of the temporal correlation between events and time series may be biased by such delays. This situation is currently the subject of study.

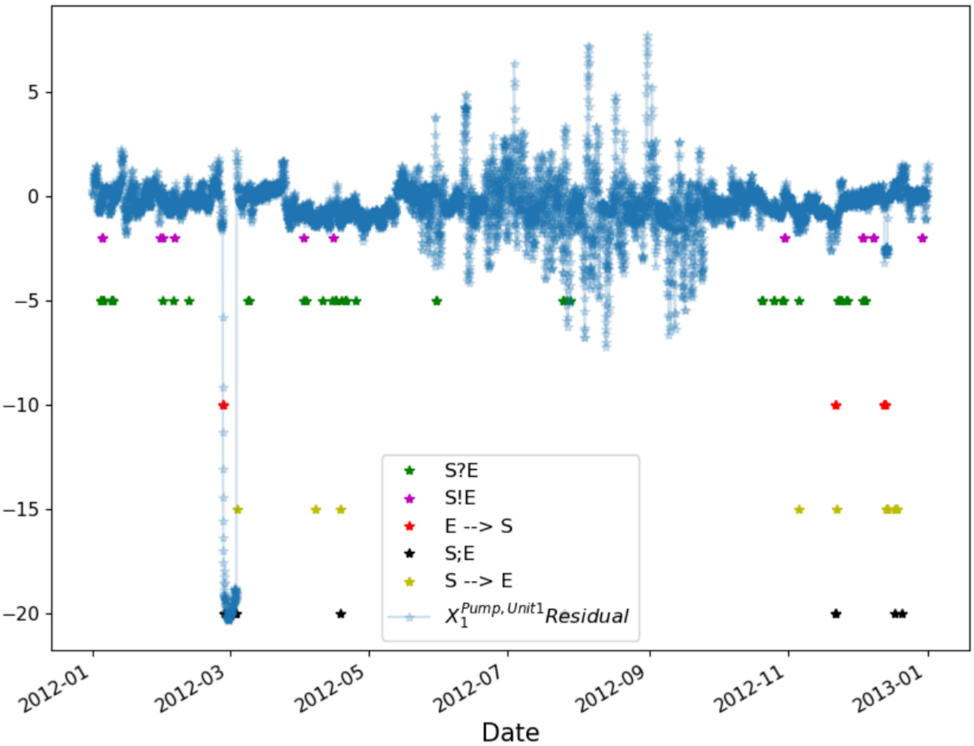

Figure 6. Example of correlation analysis of events and time series using notation shown in Algorithm 1.

9 KNOWLEDGE GRAPH CONSTRUCTION

Provided the set of processed ER data elements—either textual (see Section 4) or numeric (see Section 5)—the goal becomes to organize each element into a graph structure that captures the cause-effect relations (logical and temporal) identified in Section 6. Our approach began with the graph structure derived from the MBSE models of the system and assets under consideration (see Section 3), then progressed through the following steps:

1. Associate an ER textual data element with one (or more) MBSE entity.
2. Identify ER numeric data elements that have a logical path to the ER textual data element identified in Step 1.
3. Determine whether there is a temporal relation between the ER textual data element identified in Step 1 and the ER numeric data elements identified in Step 2 (see Section 6).
4. If both the temporal and logical relation have been identified in Step 3:

   a. Link the portion of the ER numeric data element to its corresponding MBSE element.
   b. Link the data element identified in Step 4a to the ER textual data element identified in Step 1.

5. Repeat Steps 1–4 for each ER textual data element.

The resulting relational database will take the form of a graph structure reflecting the links between the data elements associated with a particular MBSE entity. Again, the actual skeleton of the graph structure is directly derived from the MBSE diagram of the system and assets under consideration. In this respect, Figure 7 shows the CWS graph structure directly generated from the provided MBSE diagram. Note that the graph nodes can reflect different data types (form or function), and the same applies to edges.

For the present article, we focused on the textual portion of the available ER dataset for the considered CWS over a 10 year lifespan. The knowledge extraction methods presented in the past sections were employed to analyze all shift logs, WOs, and IRs, enabling us to identify the nature of textual elements and the MBSE elements associated with them. As an example, Figure 7 graphically shows how the knowledge graph is populated by first obtaining the graph from the system MBSE model; then, anomalies identified using the methods indicated in Section 6 and the events processed using the TLP methods shown in Section 7 are associated to one or more MBSE entity.

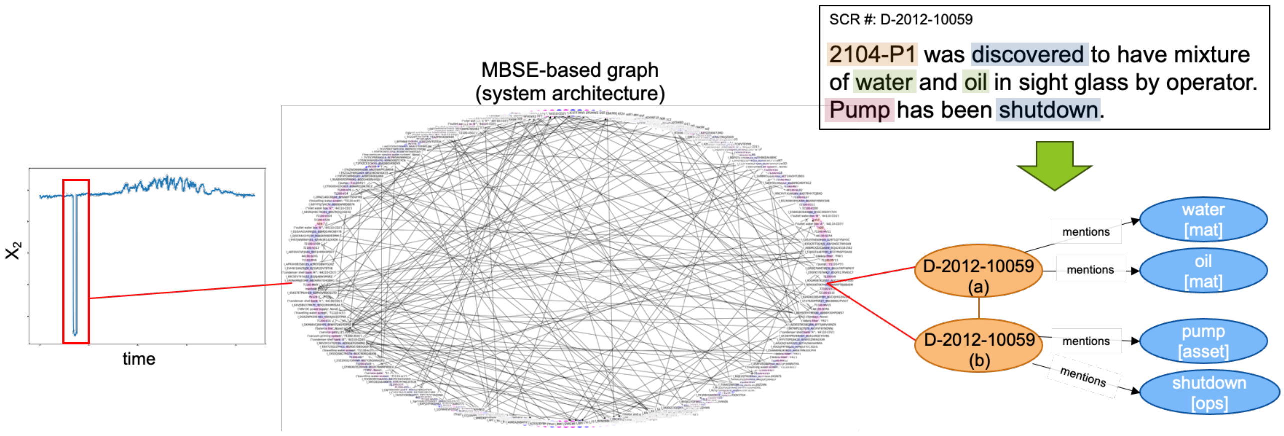

Figure 7. Graphical representation of the knowledge graph data structure.

10 CONCLUSION

This paper has presented an approach designed to holistically integrate ER data, both numeric and textual, into a single knowledge graph. We first employ MBSE models to capture the system architecture (form and functional representation based on diagrams), which are translated into graph structures. Then, monitoring data are analyzed to identify anomalies while textual data is parsed by TLP methods designed to extract knowledge from text and generate a data graph out of textual elements. We also include in our process methods designed to correlate events and anomalies with a time series, which capture causal relations (temporal and logical). The graph is then populated by associating anomalies and events to specific MBSE entities. The obtained knowledge graph merges system architecture and system ER data into a single digital structure. This data structure can be then employed to perform several tasks, including the identification of patterns of anomalies, diagnosis of causes from a set of anomalies across systems, assessment of historic asset health performances, and update of plant PRA models.

REFERENCES

Borky, J. M., and T. H. Bradley. 2018. *Effective Model-Based Systems Engineering*. Springer. https://doi.org/10.1007/978-3-319-95669-5.

C.-C. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. Furtado Silva, A. Mueen, E. Keogh (2016). *All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets*. IEEE ICDM 2016.

Coble, J.; Ramuhalli, P.; Bond, L.; Hines, J.W.; Upadhyaya, B. A review of prognostics and health management applications in nuclear power plants. Int. J. Progn. Heal. Manag. 2015, 6, 2271. https://doi.org/10.36001/ijphm.2015.v6i3.2271.

Dori, D., and E. Crawley. 2002. *Object-Process Methodology: A Holistic Systems Paradigm*. Heidelberg: Springer. https://doi.org/10.1007/978-3-642-56209-9.

Friedenthal, S., A. Moore, and R. Steiner. 2008. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann. https://doi.org/10.1016/C2013-0-14457-1.

Gretton, A., K. Borgwardt, M. Rasch, B. Scholkopf, and A. Smola. 2006. *A Kernel Method for the Two-Sample-Problem*. Advances in Neural Information Processing Systems, MIT Press, vol. 19. https://doi.org/10.48550/arXiv.0805.2368.

Lane, H., H. Hapke, and C. Howard. 2019. *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python*. Shelter Island, New York: Manning Publications.

LML steering committee. "Lifecycle Modelling Language (LML) Specification". https://22132398.fs1.hubspotusercontent-na1.net/hubfs/22132398/LML%20specification%201.4.pdf.

Luo, C., J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang. 2014. *Correlating Events with Time Series for Incident Diagnosis*. KDD'14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1583–1592. https://doi.org/10.1145/2623330.2623374.

Wang, C., D. Mandelli, and J. Cogliati (2024). "Technical Language Processing of Nuclear Power Plants Equipment Reliability Data." *Energies,* vol. 17, no. 7.

William, S. 2004. *The Object Primer: Agile Model Driven Development with UML 2.0*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511584077.

Xingang, Z., J. Kim, K. Warns, X. Wang, P. Ramuhalli, S. Cetiner, H. G. Kang, and M. Golay (2021). "Prognostics and Health Management in Nuclear Power Plants: An Updated Method-Centric Review with Special Focus on Da-ta-Driven Methods." Frontiers in Energy Research 9: 696785. https://doi.org/10.3389/fenrg.2021.696785.