DETECTING AND CHARACTERIZING PATTERNS OF FAILURE IN COMPLEX ENGINEERED SYSTEMS: AN ONTOLOGY DEVELOPMENT AND CLUSTERING APPROACH

Walsh, Hannah Scharline (1);
Dong, Andy (2);
Tumer, Irem (2);
Brat, Guillaume (1)

1: NASA Ames Research Center;
2: Oregon State University

ABSTRACT

While the causes of failures in complex engineered systems are often clear in hindsight, it can be challenging to predict failures proactively during the design of novel engineered products or systems. Identifying patterns can be useful for capturing common characteristics that may lead to failure. In this paper, we present a methodology for identifying patterns of failure from NASA's publicly available Lessons Learned Information System (LLIS). We apply an ontology development and clustering approach to identify representative patterns leading to failures in historical lessons learned. A joint inductive-deductive approach reveals the key themes in lessons that lead to failure, which are formalized and recorded as an ontology of complex systems failure causes. Documents from the LLIS are manually tagged with relevant characteristics from the ontology. From the tagged set, clustering is used to capture co-occurring sets of characteristics that lead to failure. The primary contribution of this work is a method for extracting a set of generic failure patterns in complex engineered systems and characteristics for these patterns that can be identified at design time, knowledge of which can be used to plan mitigation strategies.

Keywords: Complexity, Risk management, Ontologies

Contact:
Walsh, Hannah Scharline
NASA Ames Research Center
United States of America
hannah.s.walsh@nasa.gov

Cite this article: Walsh, H. S., Dong, A., Tumer, I., Brat, G. (2023) 'Detecting and Characterizing Patterns of Failure in Complex Engineered Systems: An Ontology Development and Clustering Approach', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.143

1 INTRODUCTION

In spite of rigorous analysis and testing, complex engineered systems are often prone to unexpected failures. As a result, it is often challenging to fully specify a complex system (including all subsystems, interactions, and necessary external factors in sufficient detail) in a way that enables complete predictions of failure behavior. While the causes of failures are often clear in hindsight, it can be challenging to predict failures proactively during the design of complex engineered systems. To combat this, expert input alongside a formal approach to identifying failures is typically performed. Expert analysis can be aided by a process such as Diogenes (Bahill, 2012), but human analysts (while invaluable) are subject to bounded rationality (Gurnani & Lewis, 2008) and biases. Formal approaches including function-based failure assessment methods (Stone et al., 2005) can reduce these biases, but are nonetheless limited by the imagination of the analyst.

When enumerating likely failures for a system under development, it is often useful to reference existing and related knowledge available. Some approaches abstract prior knowledge into ontologies or formalizations of common templates describing phenomena. Failure taxonomies present one format in which similar, known failures can be grouped (O'Halloran et al., 2012). Patterns of failure in complex systems can be expressed in several formats. A well-known format is system dynamics archetypes, which can be applied to specific needs such as system safety (Marais et al., 2006) or represented more generically (Wolstenholme, 2003). Such resources can be referenced to aid expert analysis of likely failures in a new system under development. However, one shortcoming is that it is labour intensive to compile and update these artifacts. Complex system designs that use novel technologies for which knowledge may only be gained or updated rapidly after deployment further increase labour intensity.

A possible solution is the application of text mining and machine learning approaches to extract failure patterns from knowledge repositories. Knowledge repositories are valuable tools to aid experts in identifying likely failures in a system. NASA's Lessons Learned Information System (LLIS) (NASA, n.d.), for instance, stores a rich history of failures, their causes, and recommendations for preventing these failures in the future, which can aid such processes. Knowledge management is one such way in which this knowledge can be leveraged to reduce the space of "unknown knowns" in system design. Such a system may include knowledge discovery tools, which help a user discover meaningful and useful knowledge in a dataset. As a part of knowledge discovery, it is useful to understand the major characteristics in the data. While there are automated or semi-automated ways to extract themes from documents (Andrade & Walsh, 2022), it is typically useful for an analyst to have familiarity with the dataset before applying such techniques (e.g., if an analyst is required to label documents to train a machine learning model, familiarity with the dataset is required). To this end, characterizing themes in documents provides a means to explore and gain familiarity with the dataset as well as provide guidance on feature extraction for machine learning tasks. Documents can be characterized based on specific low-level characteristics present as well as which characteristics tend to co-occur, i.e., higher-level patterns that emerge. While useful for the knowledge discovery process as discussed, these results can be useful on their own as descriptions of failures documented in NASA's LLIS.

In this paper, we build a taxonomy of characteristics leading to failure in complex engineered systems and identify failure patterns in a large set of documents describing lessons learned in complex engineered systems. Specifically, we identify failure characteristics in NASA's publicly available LLIS. We apply a taxonomy development and a clustering approach to identify representative patterns leading to failures in historical lessons learned. A joint inductive-deductive approach reveals the key themes in lessons that lead to failure, which are formalized and recorded as a taxonomy of complex systems failure causes. A subset of LLIS documents is manually reviewed multiple times during this process in order to both develop the taxonomy and label the documents according to which ontological elements they describe. A clustering approach is then applied to capture co-occurring sets of characteristics that lead to failure. The results provide a description of common trends in complex systems failures, which can be identified at design stage and can be used to plan appropriate mitigation strategies.

2 BACKGROUND

As an initial step in many analyses, including and especially natural language processing, it is important for an analyst to be familiar with the dataset. Moreover, when performing any sort of document labelling, an analyst must follow logical conventions that are consistent with the literature and empirical evidence available. Domain-independent entities (date, person, place, etc.) generalize well, but any application that uses domain-dependent entities must often be carefully defined and developed. The biomedical domain especially is increasingly recognizing the importance of developing appropriate ontologies for such tasks (Wang et al., 2021), in addition to increasing recognition in software-specific named entity recognition (Ye et al., 2016). In engineering design, there is some precedent for using Failure Modes and Effects Analysis (FMEA)-style labels (Andrade & Walsh, 2022). Cause, failure, effect, and other common labels from an FMEA-style analysis are useful in many contexts for organizing common failures; however, ontologies in a style more similar to that of Wang et al (2021), which expresses common sources of failure causes such as those related to the organization, development process or operator, may be needed for to characterize non-technical factors that contribute to failure.

More broadly, ontological representations of existing design knowledge have been used to support decision-making (Sarkar, 2016; Ming et al., 2018; Zhang et al., 2018). Other relevant ontologies include the taxonomy of error types proposed by (Sutcliffe & Rugg, 1998), which covers operator errors as well as those stemming from social/organizational challenges and design errors, and the taxonomy of information technology project failures proposed by Al-Ahmad et al. (2009). Another example of taxonomy usage in engineering design is for information retrieval (Li et al., 2014), which can also be useful as part of a knowledge management system. Such techniques can utilize relationships between entities and may be constructed using machine learning based techniques (Shi et al., 2017).

In addition to developing ontologies of characteristics leading to failure, identifying patterns in characteristics can be useful. For instance, Weking et al. (2020) document archetypal blockchain business models using literature review and cluster analysis. Similarly, Hermes et al. (2020) used qualitative cluster analysis to understand platform envelopment. In the design literature, guidelines have been extracted and classified for disassembly knowledge (Favi et al., 2016). This research synthesizes these ideas to demonstrate into a method to extract a taxonomy for complex systems failures.

3 METHODOLOGY

The methodology is divided into two parts: (1) taxonomy development and (2) clustering. A joint inductive-deductive approach is used to obtain the taxonomy. During this process, documents are manually labelled according to the ontological elements (characteristics) present. Clustering is used to find common patterns of failure in the corpus based on characteristics in the documents.

3.1 NASA lessons learned information system

NASA's Lessons Learned Information System (LLIS) is used as the source of documents in this study. The LLIS contains documents describing lessons learned in NASA projects. The LLIS provides a suitable corpus for this study because it describes a wide variety of projects as well as issues occurring throughout system design, development, and operation. The earliest entries date to the 1970s and entries are continually added today. Documents are semi-structured in that there are sections such as Abstract, Lesson(s) Learned, and Recommendation(s), but the text is unstructured within these sections. Which sections are filled, and to what extent, depends on the discretion of the contributors. In previous work, we used a term frequency-inverse document frequency (TF-IDF) classifier to filter out lessons that are not relevant to failure analysis (Andrade & Walsh, 2022); this abbreviated set containing 1639 relevant lessons is also used in this study. From this abbreviated set, a subset of one-hundred-fifty lessons is randomly selected. These are read and evaluated in full by a human analyst. Because of the filtering approach, the lessons studied discuss a failure, but the range of issues covered in the dataset are varied. Topics covered in the LLIS have been extensively covered by Andrade and Walsh (2022).

3.2 Taxonomy development

To develop the taxonomy, we use an approach inspired by content analysis and the method of Nickerson et al. (2013), which alternates inductive (empiricism) and deductive (conceptualization) approaches to taxonomy development while applying the concepts of meta-characteristics, which ensure the taxonomy is consistent with its intended purpose, and ending conditions, which ensure the taxonomy is sufficiently mature. This approach has been applied widely, including in blockchain systems (Sai et al., 2021), smart machines in the mechanical engineering industry (Scharfe & Wiener, 2020), digital twins (Mazzurco et al., 2018), to name a few examples. We also tune the taxonomy development process specifically to conventions of failure taxonomies in the literature, which is noted throughout the methodology.

3.2.1 Meta-characteristics

Two concepts must be defined before beginning the taxonomy development process: the meta-characteristic and ending conditions. The meta-characteristic should be consistent with the purpose of the taxonomy and all discovered characteristics should be derivable from this meta-characteristic. The purpose of our taxonomy under development is to describe complex systems failure characteristics in a way that is useful to system designers. The intended users of this taxonomy are system designers during the early stages of system development. We do not want the taxonomy to be specific to any single system or even domain, so it is desirable that characteristics are not domain specific. However, we want to characterize specific sources of failure, such as those that are organizational or due to operator error, rather than simply specifying a cause versus an effect of failure (as in an FMEA-style analysis). Given this stated purpose and audience, we select the following meta-characteristic: characteristics of the system architecture, operation, environment, or development process that introduce a hazard.

3.2.2 Inductive approach

Either the deductive or inductive approach may be used to start, depending on factors such as the analyst's familiarity with the data set (Nickerson et al., 2013). The entire set of selected documents is used for each iteration. In this work, we started with the inductive approach because there is significant data available but few existing formal frameworks describing complex systems failure characteristics (that are appropriate for our stated purpose). The inductive approach follows the outline of conventional content analysis. This approach derives the key themes from a knowledge base; themes are learned from the data, rather than through existing theory or literature. The general steps for a conventional content analysis are as follows (Hsieh & Shannon, 2005): (1) read all documents to gain an understanding of the information captured; (2) highlight parts of text that appear to capture the key themes; and (3) make notes of initial thoughts and analysis. Repeat Steps 1-3 until an initial coding scheme emerges.

At this point, the coding scheme can be matured by grouping similar themes or organizing them hierarchically depending on the content and goals of the analysis. Nickerson et al. specify that characteristics proposed to describe documents must be logical consequences of the meta-characteristics of the taxonomy. Additionally, Nickerson et al. specify that these characteristics must not apply to all documents, as the purpose of the characteristics is to enable distinguishing features of certain documents. Groupings are also established, as in conventional content analysis. Nickerson et al. call these groupings "dimensions", although they have constraints in addition to containing similar characteristics. In Nickerson et al.'s taxonomy definition, characteristics in each dimension must be "mutually exclusive" (avoiding overlap) and "collectively exhaustive" (documents have one characteristic in each dimension).

3.2.3 Deductive approach

In the deductive approach, the researcher begins with their notions of the taxonomy's characteristics which, as with the inductive approach, must be logical consequences of the meta-characteristics (Nickerson et al., 2013). In this approach, once characteristics have been proposed, the researcher examines documents and determines whether there are documents that are described by these characteristics. Dimensions must follow the same rules as in the inductive approach (Nickerson et al., 2013). At the termination of the deductive or inductive approach, the researcher checks whether the ending conditions are met - if the ending conditions are met, the taxonomy is complete; if not, the researcher switches to the alternate (inductive or deductive) approach (Nickerson et al., 2013).

https://doi.org/10.1017/pds.2023.143 Published online by Cambridge University Press

3.2.4 Ending conditions

Ending conditions define when the taxonomy is sufficiently mature to terminate the development process. The first ending condition, common to all taxonomy development, is that there must be "dimensions" with mutually exclusive and conceptually complete characteristics (Nickerson et al., 2013). There are eight additional objective ending conditions proposed by Nickerson et al., some of which were first described by Sowa and Zachman (1992). These are summarized as follows: (1) all documents (or a representative sample of documents) have been examined; (2) within the most recent iteration, no document merges or splits have occurred; (3) each characteristic is described in at least one document; (4) in the most recent iteration, no new characteristics have been added; (5) within the most recent iteration, no grouping or characteristic merges or splits have occurred; (6) each grouping is unique; (7) each characteristic is unique within its grouping; and (8) each combination of characteristics is unique. Application-specific conditions may be added as well.

Subjective ending conditions include that the taxonomy is concise, robust, comprehensive, extendable, and explanatory (Nickerson et al., 2013). The method is intended to produce a taxonomy that is useful, but not necessarily optimal (Nickerson et al., 2013). Depending on the ending conditions chosen by the researcher, different, valid ontologies may be produced by different researchers. For this study, we apply all objective and subjective ending conditions provided. Because lessons learned documents do not typically describe all dimensions of the system that did not lead to failure, a lack of discussion of a dimension is assumed to mean that dimension did not substantively contribute to the failure. A characteristic that describes this situation is defined for each dimension. For the clustering approach, only those characteristics that contribute to failure are considered.

3.3 Clustering

Clustering is performed to find failure patterns in the LLIS based on the ontological characteristics identified and developed in this work. After applying multiple clustering algorithms in the sklearn package in Python, the spectral clustering method is selected due to resulting in the highest quality clusters. Spectral clustering has roots in graph theory and utilizes the eigenvalues of a similarity matrix of the data in order to reduce its dimensionality prior to clustering. A clustering algorithm such as k-means can then be used to cluster the data (Pedregosa et al., 2011). Spectral clustering often outperforms more conventional methods and is relatively simple to implement (von Luxburg, 2007). Spectral clustering is especially useful for data that has non-convex clusters, making it a useful point of comparison to a standard k-means approach (Pedregosa et al., 2011). The number of clusters must be specified.

In this case, and in many clustering problems, we do not have a ground truth for the correct clustering, so evaluation metrics, e.g., those based on similarity of clusters, are used. The quality of the clusters can be evaluated using silhouette coefficient, s, which measures how well each sample fits into its assigned cluster. Higher values of s indicate higher quality clusters, specifically how well each sample matches its own cluster and how different it is from other clusters. Silhouette coefficient is calculated per sample, where a is the mean intra-cluster distance and b is the mean nearest-cluster distance, as in Eq. 1 (Pedregosa et al., 2011). The average and standard deviation of silhouette coefficient across samples can also be used to assess the overall performance of the clustering algorithm.

$$s = \frac{b-a}{\max(a,b)} \tag{1}$$

4 RESULTS

We present the developed taxonomy in Table 1. The process requires five full iterations of the taxonomy development process to reach the ending conditions. One-hundred-fifty documents from the LLIS are examined in detail by a human analyst to build the taxonomy. To show the relative importance of characteristics within the LLIS, we provide the total number of characteristics labelled in the dataset in a column of Table 1. The three most influential characteristics according to frequency of occurrence are Analysis: Development Process Missing Standard Analysis, System Environment: Interaction with Environment, and Subsystem Behaviour: Coupling between Subsystems. Monitoring: Failure of Monitor has the lowest count followed by Timing of Action: Delayed Action and Timing of Action: Premature Action. All counts being greater than zero means the condition from the taxonomy

https://doi.org/10.1017/pds.2023.143 Published online by Cambridge University Press

development method in which each characteristic must be present in at least one document studied is satisfied. All dimension totals being equal to the total document count means the collectively exhaustive criterion is satisfied.

Table 1. Taxonomy of failure in complex engineered systems, derived from the LLIS.

Dimension	Characteristic	Definition	Example	Count
Analysis	Development Process Missing Standard Analysis	An analysis that is widely considered standard is not performed	"departures from the… robust design practices"	48
	Standard Analysis Not Performed on Final Configuration	Analysis is performed, but too early, so the final configuration is not analysed	"change in weld penetration warranted the need to perform a second round of fatigue analysis"	16
	Analysis is Adequate for System Operation	Standard analysis is performed at an appropriate time in design	Not available	86
Decision-Making Structure	Too Centralized	Single decision-maker does not communicate sufficiently	"stress analysis team was not made aware of the change"	23
	Too Decentralized	Multiple decision-makers are not well coordinated	"issues … team having common understanding of individual roles"	20
	Decision-Making Structure is Effective	Decision-maker(s) and communications are suited to decision	Not available	107
Human-System Relationship	Inadequate Human Factors	Human factors not adequately considered in design	"control panel height being placed too low"	12
	Adequate Human Factors	Human factors adequately considered in design	Not available	138
Monitoring	Failure of Monitor	Monitor is present, but failed	"…technology is … ineffective in providing gas leak detection"	1
	Omission of Monitor	Monitor is needed, but not present	"assure sufficient downlink telemetry to ascertain their health"	8
	Monitors Adequate and Functional	Monitor is either not needed or present and functional	Not available	141
Procedures	Intrusion or Replacement of Procedure	Addition of new procedure or replacement of a correct step with an incorrect step	"battery was deep charged while mounted in the robot, without a BMS, instead of on the test stand"	7
	Omission of Procedure	Operator skips a step or procedure	"ground support crew did not follow all of the steps"	14
	Procedures Adequate and Appropriate	Only correct procedures are followed if any are required	Not available	129
Safety Equipment	Inadequate Safety Equipment	Safety equipment or safeguards are inadequate	"issues like… use of protective goggles"	6
	Adequate Safety Equipment	Safety equipment or safeguards are adequate	Not available	144

https://doi.org/10.1017/pds.2023.143 Published online by Cambridge University Press

Table 1. Taxonomy of failure in complex engineered systems, derived from the LLIS, continued.

Subsystem Behaviour	Coupling between Subsystems	Small deviation in one subsystem affects another subsystem	"potential failure scenarios of other subsystems that interface with…"	30
	Isolated Subsystems	Subsystems difficult to integrate	"developed solution cannot be transitioned to other similar hardware"	8
	Subsystem Interaction Effective	Subsystem interaction is effective	Not available	112
System Behaviour	Unaccounted for Non-Linear Behavioural Relationships	A small deviation or fault has large impacts on system-level behaviour	"small errors in the on-board model of local gravity… can cause GN&C to induce unintended accelerations"	10
	Accounted for Behavioural Relationships	If non-linear relationships exist, they are accounted for in design or operation	Not available	140
System Environment	Unaccounted for Interaction with Environment	Interactions between the system and its environment are not adequately isolated or protected	"radiation-induced damage"	31
	Accounted for Interaction with Environment	Interactions between the system and its environment are adequately isolated or protected	Not available	119
Timing of Action	Premature Action	An action in design or operation is made prematurely	"Do not prematurely reduce the scope and depth of post flight … assessment"	5
	Delayed Action	An action in design or operation is made late	"if mock-up activities had begun earlier … problems could have been addressed sooner"	2
	Action Appropriately Timed	An action in design or operation is made at the correct time if one is needed	Not available	123

The clustering method yields clusters with an average silhouette score of 0.7704 and a standard deviation of 0.3969. The largest cluster contains twenty-five lessons, while the majority of clusters contain five or fewer lessons. The distribution is skewed, with much larger clusters being present but uncommon. These larger clusters represent more common patterns of failure since they occur in more lessons in the LLIS. Clusters are defined by, and can be characterized according to, common dimensions. We show the top ten (by size) failure patterns (clusters) identified in Table 2 from a total of twenty-seven. We provide the dimension(s) present in the cluster as well as text from a lesson from that cluster that describes the dimension in order to provide a real example of that dimension in the dataset. These are selected at random from the documents in which the characteristics are detected. The Analysis dimension is prominent in the failure patterns. This is likely due to the negative characteristics of the Analysis dimension having relatively high counts, and therefore the overall dimension score being high. The clusters contain between one and six characteristics. In most cases, all or a large majority of lessons within a cluster contain the same characteristics. In some cases, a small number of lessons within a cluster contained an additional characteristic. In these cases, when the additional characteristic is in a minority (<50%) of lessons within the cluster, the characteristic is not included in the cluster description.

https://doi.org/10.1017/pds.2023.143 Published online by Cambridge University Press

Table 2. Top ten clusters (failure patterns) in the LLIS.

No.	Size	Lesson	Characteristics
1	25	26703	(1) Analysis: "Knowledge of the 1-g physics of large bubbles in the condenser legs would have allowed the limits of ground testing to be better understood."
2	23	24503	(1) Decision-Making Structure: "flight software has been developed by a consortium… The definition of the software coding standards was non-specific."
3	15	24403	(1) System Environment: "Bird strikes are known hazards during aircraft takeoff and landing operations."
4	9	28101	(1) Analysis: "the redundancy and failure analysis for LRS was incomplete", (2) Subsystem Relationships: "include all relevant interfacing subsystems"
5	9	27003	(1) Subsystem Relationships: "have been found to have sensitivity to noise that could lead to inadvertent firing"
6	6	27701	(1) Analysis: "robust design practices and integrated set of engineering checks and balanced that may have prevented…", (2) System Environment: "radiation-induced damage"
7	5	1710	(1) Procedures: "did not follow all of the steps in a complex checklist", (2) Decision-Making Structure: "Less than adequate communications between…", (3) Human-System Relationship: "…caused the pilot to mismanage the UAV's energy"
8	4	28105	(1) Subsystem Relationships: "propulsion subsystem should ensure positive isolation of…"
9	4	4999	(1) Procedures: "active leveling should be employed to accommodate center of gravity offsets"
10	4	28202	(1) Analysis: "warranted the need to perform a second round of fatigue analysis", (2) Decision-Making Structure: "stress analysis team was not made aware of the change"

5 DISCUSSION

The proposed method for defining a failure taxonomy captures failure cases as they are described by contributors. Since only the factors that are expressed by the contributors can be captured in the method, there may be failure causes or contributing factors that are missed. Increasing the size of the data set can offset this problem. Compared to existing ontologies, the taxonomy developed in this paper shares characteristics, providing a measure of validity to the results, but with differing scope and structure. For instance, the taxonomy from Sutcliffe et al. (1998) includes timing delays and premature actions taken by operators as well as procedural errors, such as omissions and intrusion. Other portions Sutcliffe and Rugg's taxonomy are lower level than what is characterized in our taxonomy and fall broadly under the Human-System Relationship characteristic. Al-Ahmad et al. (2009) proposed an information technology project failure taxonomy which includes several of the top-level themes present in our taxonomy, including complexity factors, management factors, organizational factors, and process factors (decomposed differently in our taxonomy as Analysis, Decision-Making Structures, Procedures, Subsystem Behavior, System Behavior, and System Environment). Because our taxonomy has a unique set of aims and scope compared to other ontologies and is developed from a finite set of documents, it is expected that it may not capture all factors addressed in other ontologies. By increasing the size of the document set as well as using multiple ontologies, a more complete model of the factors contributing to failure in complex systems may be developed.

6 CONCLUSIONS AND FUTURE WORK

This work identified twenty-seven patterns of failure in the subset of lessons examined in the LLIS and a taxonomy for characterizing complex systems failures. The failure patterns were derived by clustering labelled lessons in the LLIS. The database contents exhibited a moderate to strong cluster structure (s=0.7704). The results will guide further exploration of the dataset and provide a conceptual model to guide labelling processes for training natural language processing (NLP) models for tasks such as named entity recognition. The method can be applied to other datasets to characterize failures and failure

https://doi.org/10.1017/pds.2023.143 Published online by Cambridge University Press

characteristics in other domains. This represents an important step in supporting developing efforts for improving knowledge management and fully utilizing knowledge repositories available to designers to prevent failure in early design. The goal is to reduce the space of unknown knowns, improving designers' ability to anticipate issues before they cause safety issues or require expensive and time-consuming redesign. Future work will apply the taxonomy development method to other datasets to test its extensibility and improve the breadth of failures characterized. Finally, while the purpose of this research is to develop the method for extracting the taxonomy and finding failure patterns using automatic clustering, the broader project will require interpretation of results by domain experts. Multiple experts will likely be required to interpret failure patterns from a dataset as broad as the LLIS.

ACKNOWLEDGMENTS

This research is supported by the System-Wide Safety (SWS) project in the Airspace Operations & Safety program (AOSP) in the NASA Aeronautics Research & Mission Directorate (ARMD). Any opinions or findings of this work are the responsibility of the authors and do not necessarily reflect the views of the sponsors or collaborators.

REFERENCES

Al-Ahmad, W., Al-Fagih, K., Khanfar, K., Alsamara, K., Abuleil, S., and Abu-Salem, H., (2009), "A taxonomy of an IT project failure: root causes", *International Management Review*, Vol. 5 No. 1, pp. 93-104, 106.

Andrade, S. R. and Walsh, H. S. (2022), "Discovering a failure taxonomy for early design of complex engineered systems using natural language processing", *Journal of Computing and Information Science in Engineering*, Vol. 23 No. 3, p. 031001. https://dx.doi.org/10.1115/1.4054688.

Andrade, S. R. and Walsh, H. S. (2022), "What went wrong: a survey of wildfire UAS mishaps using named entity recognition", 2022 *IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, IEEE. https://dx.doi.org/10.1109/DASC52595.2021.9594501.

Bahill, T. (2012), "Diogenes, a process for identifying unintended consequences", *Systems Engineering*, Vol. 15 No. 3, pp. 287-306. https://dx.doi.org/10.1002/sys.20208.

Favi, C., Germani, M., Mandolini, M. and Marconi, M. (2016), "Disassembly knowledge classification and potential application: a preliminary analysis on a washing machine", *ASME 2016 International Design Engineering Technical Conferences and Computers & Information in Engineering Conference (IDETC/CIE 2016)*, Charlotte, North Carolina, USA, ASME, pp. V004T05A011. https://dx.doi.org/10.1115/detc2016-59514.

Gurnani, A. and Lewis, K. (2008), "Collaborative, decentralized engineering design at the edge of rationality", *Journal of Mechanical Design*, Vol. 130 No. 12, p. 121101. https://dx.doi.org/10.1115/1.2988479.

Hermes, S., Kaufmann-Ludwig, J., Schreieck, M., Weking, J., and Böhm, M., (2020), "A taxonomy of platform envelopment: revealing patterns and particularities", *AMCIS 2020 Proceedings*.

Hsieh, H.-F. and Shannon, S. E. (2005), "Three approaches to qualitative content analysis", *Qualitative Health Research*, Vol. 15 No. 9, pp. 1277-1288. https://dx.doi.org/10.1177/1049732305276687.

Li, L., Qin, F., Gao, S. and Liu, Y. (2014), "An approach for design rationale retrieval using ontology-aided indexing", *Journal of Engineering Design*, Vol. 25 No. 7-9, pp. 259-279. https://dx.doi.org/10.1115/detc2013-12522.

Marais, K., Saleh, J. and Leveson, N. (2006), "Archetypes for organizational safety", *Safety Science*, Vol. 44 No. 7, pp. 565-582. https://dx.doi.org/10.1016/j.ssci.2005.12.004.

Mazzurco, A., Leydens, J. A. and Jesiek, B. K. (2018), "Passive, consultative, and coconstructive methods: a framework to facilitate community participation in design for development", *Journal of Mechanical Design*, Vol. 140 No. 12, p. 121401. https://dx.doi.org/10.1115/1.4041171.

Ming, Z., Wang, G., Yan, Y.; Panchal, J. H., Goh, C., Allen, J. K., and Mistree, F. (2018), "Ontology-based representation of design decision hierarchies", *Journal of Computing and Information Systems in Engineering*, Vol. 18 No. 1, p. 011001. https://dx.doi.org/10.1115/1.4037934.

NASA, *NASA Public Lessons Learned Information System.* [online] Available at: https://llis.nasa.gov/ [Accessed 2022].

Nickerson, R. C., Varshney, U. and Muntermann, J. (2013), "A method for taxonomy development and its application in information systems", *European Journal of Information Systems*, Vol. 22, pp. 336-359. https://dx.doi.org/10.1057/ejis.2012.26.

O'Halloran, B. M., Stone, R. B. and Tumer, I. Y. (2012), "A failure modes and mechanisms naming taxonomy", IEEE 2012 Proceedings Annual Reliability and Maintainability Symposium, IEEE, pp. 1-6. https://dx.doi.org/10.1109/rams.2012.6175455.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011), "Scikit-learn: machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830.

Sai, A. R., Buckley, J., Fitzgerald, B. and Andrew, L. G. (2021), "Taxonomy of centralization in public blockchain systems: A systematic literature review", *Information Processing & Management*, Vol. 58 No. 4, p. 102584. https://dx.doi.org/10.1016/j.ipm.2021.102584.

Sarkar, A. a. S. D. (2016), "Foundation ontology for distributed manufacturing process planning", *ASME 2016 International Design Engineering Technical Conferences and Computers & Information in Engineering Conference (IDETC/CIE 2016)*, Charlotte, North Carolina, USA, pp. V01BT02A031. https://dx.doi.org/10.1115/detc2016-60159.

Scharfe, P. and Wiener, M. (2020), "A taxonomy of smart machines in the mechanical engineering industry: toward structuring the design solution space", *ICIS 2020 Proceedings*, p. 1139.

Shi, F., Chen, L., Han, J. and Childs, P. (2017), "A data-driven text mining and semantic network analysis for design information retrieval", *Journal of Mechanical Design*, Vol. 139 No. 11, p. 111402. https://dx.doi.org/10.1115/1.4037649.

Sowa, J. F. and Zachman, J. A. (1992), "Extending and formalizing the framework for information systems architecture", *IBM Systems Journal*, Vol. 31 No. 3, pp. 590-616. https://dx.doi.org/10.1147/sj.313.0590.

Stone, R. B., Tumer, I. Y. and Van Wie, M. (2005), "The function-failure design method", *Journal of Mechanical Design*, Vol. 127, pp. 397-407. https://dx.doi.org/10.1115/1.1862678.

Sutcliffe, A. and Rugg, G. (1998), "A taxonomy of error types for failure analysis and risk assessment", *Int. J. Hum. Comput. Interaction*, Vol. 10 No. 4, pp. 381-405. https://dx.doi.org/10.1207/s15327590ijhc1004_5.

von Luxburg, U. (2007), "A tutorial on spectral clustering", *Statistics and Computing*, Vol. 17 No. 4, pp. 395-416. https://dx.doi.org/10.1007/s11222-007-9033-z.

Wang, K., Stevens, R., Alachram, H., Li, Y., Soldatova, L., King, R., Ananiadou, S., Schoene, A. M., Li, Mao., and Christopoulou, F. (2021), "NERO: a biomedical named-entity (recognition) ontology with a large, annotated corpus reveals meaningful associations through text embedding", *NPJ systems biology and applications*, Vol. 7 No. 1, pp. 1-8. https://dx.doi.org/10.1101/2020.11.05.368969.

Weking, J., Mandalenakis, M., Hein, A., Hermes, S., Böhm, M., and Krcmar, H. (2020), "The impact of blockchain technology on business models –- a taxonomy and archetypal patterns", *Electronic Markets*, Vol. 30 No. 2, pp. 285-305. https://dx.doi.org/10.1007/s12525-019-00386-3.

Wolstenholme, E. F. (2003), "Towards the definition and use of a core set of archetypal structures in system dynamics", *System Dynamics Review*, Vol. 9 No. 1, pp. 7-26. https://dx.doi.org/10.1002/sdr.259.

Ye, D., Xing, Z., Foo, C., Ang, Z., Li, J., and Kapre, N. (2016), "Software-specific named entity recognition in software engineering social content", 2016 *IEEE 23rd International Conference on Software Analysis, Evolution and Reengineering (SANER)*, IEEE, pp. 90-101. https://dx.doi.org/10.1109/saner.2016.10.

Zhang, C., Zhou, G., Bai, Q., Lu, Q., and Chang, F. (2018), "HEKM: a high-end equipment knowledge management system for supporting knowledge-driven decision-making in new product development", *ASME 2018 International Design Engineering Technical Conferences and Computers & Information in Engineering Conference (IDETC/CIE 2018)*, ASME, Quebec City, Quebec, Canada, pp. V01BT02A014. https://dx.doi.org/10.1115/detc2018-85151.