Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P568-cd

Extending safety control structures: a knowledge graph for STAMP

Francesco Simone

Department of Mechanical and Aerospace Engineering, University of Rome "La Sapienza", Italy. E-mail: francesco.simone@uniroma1.it

Antonio Javier Nakhal Akel

Department of Mechanical and Aerospace Engineering, University of Rome "La Sapienza", Italy. E-mail: antonio.nakhal@uniroma1.it

Antonello Alvino

Department of Technological Innovation and Safety of Plants, Products and Anthropic Settlements, INAIL (National Institute for insurance Against Accidents at Work), Italy. E-mail: a.alvino@inail.it

Silvia Maria Ansaldi

Department of Technological Innovation and Safety of Plants, Products and Anthropic Settlements, INAIL (National Institute for insurance Against Accidents at Work), Italy. E-mail: s.ansaldi@inail.it

Patrizia Agnello

Department of Technological Innovation and Safety of Plants, Products and Anthropic Settlements, INAIL (National Institute for insurance Against Accidents at Work), Italy. E-mail: p.agnello@inail.it

Maria Francesca Milazzo

Department of Engineering, University of Messina, Italy. E-mail: mariafrancesca.milazzo@unime.it

Giulio Di Gravio

Department of Mechanical and Aerospace Engineering, University of Rome "La Sapienza", Italy. E-mail: giulio.digravio@uniroma1.it

Riccardo Patriarca

Department of Mechanical and Aerospace Engineering, University of Rome "La Sapienza", Italy. E-mail: riccardo.patriarca@uniroma1.it

The increasing interactions among technical components and human agents in modern industrial systems poses new challenges for safety management, demanding for novel approaches to extend techno-centric investigations with social-oriented analyses. In these scenarios, it becomes crucial the usage of a detailed accident analysis beyond immediate failures, to encompass physical, cyber and social aspects. The Systems-Theoretic Accident Model and Processes (STAMP) was developed as an accident model that makes use of systems theory to arrange a causality model focusing on system hazards. The inner nature of a STAMP model, which maps connections (feedbacks and control actions) among system elements, matches the principles of a graph representation, which are made up of vertices and edges mapping connections. This correspondence may then enable the exploitation of a STAMP-driven graph to guide safety assessments by systematic graph analyses. This paper explores the possibility of deriving a knowledge graph from a STAMP safety control structure and use it as a key element for subsequent hazard analyses. The study is instantiated on case study related to the inspection (based upon Seveso III directive) of a Seveso establishment. The analysis is meant to highlight the safety requirements to adapt the inspection procedure to possible future changes, as promoted by an energy transition. The preliminary results show the potential of such tools to empower - or possibly update – modern Safety Management System.

Keywords: Knowledge management, Inspection management, Systems theory, Risk management, Industrial plants.

1. Introduction

Industrial processes are characterized by a high number of components, tight exchange interactions, and dynamicity, which make them complex systems prone to unexpected variability (Righi, Saurin, and Wachs 2015; Pasman 2009). As a consequence, industrial systems have become highly interdependent, and any action of an agent may influence the entire system (Dekker 2019). These feed-forward loops highlight the socio-technical dimension of industrial systems, which must be studied in terms of the number of interactions, diversity of elements, variability, uncertainty, awareness situations, and resilience (Baxter and Sommerville 2011). As socio-technical industrial processes become increasingly complex, it is necessary to use safety analysis methods to manage risks and to improve safety. In this context, the Systems Theoretic Accident Modelling and Processes (STAMP) model (Leveson 2004) relies on control theory to study system complexity through hierarchical safety control structures. While the value of this methodology has been proven by many scientific research and industrial applications (Patriarca et al. 2022), a limitation results in the difficulty of modelling large complex systems and analyzing them to extract relevant safety information. A solution might be represented by the usage of a knowledge graph, that is a graph-structured data model based on the semantic rules of an ontology (Newman 2010). The knowledge graph structure enables analysts to perform big data investigations that are empowered by mathematical tools and constructs as network theory-based calculation may be.

In this manuscript, we explore the use of STAMP as a systemic model for building a socio-technical safety control structure that is then used as a basis for building a knowledge graph. A similar problem has been investigated by Pereira, Hirata, and Nadjm-Tehrani (2019). Nevertheless they mainly focused on the results gained after applying STAMP, requiring the identification of such element and data to use their ontology. This solution make it impossible to gather such information from the graph analysis. On the contrary, in our ontology, system elements are classified in a SCS perspective. This point of view is translated on a graph which enables systemic (semi-)quantitative safety analyses.

This approach is contextualized in the Oil & Gas sector by modelling a Natural Gas Storage process. The process is meant to treat raw wellhead gas into clean sales gas for sustainable power generation that can be delivered to customers with less environmental impact. The results show the potential of integrating STAMP model data into a knowledge graph to enable data and metrics analysis, systematically investigating system elements (graph nodes) and relationships among them.

The remainder of this paper is organized as it follows. Section 2 presents the foundational aspects of the methodology we propose. Section 3 includes a demonstrative use case to prove the applicability of the methodology. In Section 4 conclusions and possible future developments are discussed.

2. Methodology

The methodology we propose integrates the STAMP method with the knowledge graph technology. Accordingly, the STAMP is used to highlight the interactions between system components and to build the Safety Control Structure (SCS) of the system, this latter is then translated into a knowledge graph through an ontology model. The following section presents the theoretical foundation of the proposed methodology.

2.1. *Systems-Theoretic Accident Model and Process*

Systems-Theoretic Accident Model and Process (STAMP) is a model that allows to map and to investigate the interactions among the system components at different socio-technical levels (Leveson 2012). In STAMP, accidents result from inadequate or inappropriate control and enforcement of safety-related constraints on the system development, design, operation, and organization (Mannan 2012). Accordingly, new levels to study complexity in terms of technological and human factors must fit into reliability-oriented safety approaches. The STAMP model considers the system as a whole, not on parts taken separately. Emergent properties

can only be treated adequately, considering all social and technical aspects. The STAMP model is compound of two elements: (i) system components (represented by blocks in the SCS) and (ii) system interactions (represented by arrows in SCS) (Leveson 2018). The system components are (Nakhal A. et al. 2022):

- Controlled process: these blocks are the model lowest hierarchical level, representing the processes investigated in the safety analysis. A block depicts a controlled process if it has arrows entering from its upper or left edge, and arrows exiting from its upper or right edge.
- Controller: they represent the highest hierarchical level of the system. The controlled processes are controlled or modified through them. A controller imposes this modification through two inherent aspects: (i) a model of the process it is controlling, (ii) a control algorithm. In this paper we define as model (e.g., process model, mental model) the set of all variables and data that the controller owns to perform its control. Please note that the controller model can be updated based upon the feedback it receives. On the other hand, we define as algorithm (e.g., control algorithm) the set of rules the controller owns to combine the variables in the model and to generate a control action. A block depicts a controller if it has arrows entering and exiting from its bottom edge.
- Intermediate elements: the elements between the higher-level controller and the controlled process. These elements act as both controller for lower-level elements, and controlled process for higher-level elements. They can be represented by controllers (e.g., human controller, automated controller) or converters (e.g., actuators, sensors).

Conversely, system interactions may be defined as:

- Control actions: are the actions and tasks which manage, command, direct, or regulate the behavior of the system process. In the SCS representation they are depicted by arrows entering blocks from their upper edge.
- Feedbacks: are the results of information, data, and report from the action of the system process reflected on itself to correct, modify or inform its performance and behavior. In the SCS representation they are depicted by arrows entering blocks from their bottom edge, and arrows exiting blocks from their upper edge.
- Input/Output: these are the data, documents, and process a system component requires to work and design the industrial process. In the SCS representation, they are depicted by arrows entering blocks from their left edge, and exiting blocks from their right edge, respectively.

The information flow shall model the possibility for a data change within the process. For example, it is appropriate to consider the impossibility of getting a perfect measurement from a process. Accordingly, an output of the process which becomes a feedback may change its information if passing through a sensor (e.g., an "inaccessible" process data may be different from the feedback reported by the sensor), or, conversely, the output control of an algorithm may change with respect with the condition imposed by an actuator.

2.2. *Knowledge graph representation of the STAMP model*

A knowledge graph is a structured representation of data that includes entities (in the form of nodes), and their relationships (in the form of edges). It is meant to organize information in a graph format that helps to connect and link different pieces of information. Knowledge graphs can be used to model complex systems and to build intelligent systems that can analyse large amounts of data. To organize the information inside a knowledge graph, there is the need to label nodes and relationships by setting common rules on data. An ontology model can be used for this purpose. If a knowledge graph enables a structured representation of knowledge, an ontology model provides a formal specification of the concepts and relationships that make up the knowledge graph itself. Accordingly, an ontology defines a set of categories, properties, and relationships that describe the entities and concepts in a domain of interest (Lin and Harding 2007).

2.2.1. *The ontology model for STAMP*

Based upon the STAMP model, we propose an ontology model to translate the information contained in a SCS into a knowledge graph. The proposed ontology only spans over categories (classes) and relationships between them, no specification about categories' properties (e.g., subclasses) is detailed in this paper. Elements and interaction of the STAMP model (cf. Section 2.1) becomes the entities in the graph. A simplified representation of the proposed ontology is presented in Figure 1. Please note that category labels are denoted with italic capital letters, relationship labels are reported with italic lower case font.

The information about a process that is an element of control are denoted with the *CONTROLLED PROCESS* label. For example, a tank in an industrial plant that is object of control is labelled in this way. The *CONTROLLED PROCESS* label does not refer to characteristic data values of the process, as the tank internal pressure may be. This data is labelled as *FEEDBACK*, instead. A *FEEDBACK* node contains the data related to a process dimension. Accordingly, a node with label *CONTROLLED PROCESS* can be *defined_by* a node with label *FEEDBACK*. A feedback may be transferred by a converter element. It is labelled through the *SENSOR* label, and it is *defined_by* a *FEEDBACK*. *SENSOR* label does not refer to data values the elements receive or generate (e.g., a barometer is labelled as *SENSOR*, but the pressure it measures will be a *FEEDBACK*). Information about a system element that controls a controlled process are denoted by the *CONTROLLER* label. The *CONTROLLER* labelled node *has* a *MODEL* and *has* an *ALGORITHM*. These two entities serve as hubs for the *FEEDBACK* nodes that: (i) update the controller model, and (ii) are used by the control algorithm. *MODEL* and *ALGORITHM* nodes only contains information about the set of all variables that defines the controller model, and the rules to combine such variables in a *CONTROL ACTION*, respectively. Accordingly, both *MODEL* and *ALGORITHM* nodes are *defined_by* one (or a set of) *FEEDBACK* node(s), and a *CONTROL ACTION* is *defined_by* an *ALGORITHM*. As for the *FEEDBACK*, a *CONTROL ACTION* node contains data related to a process dimension, that, at this stage, must be

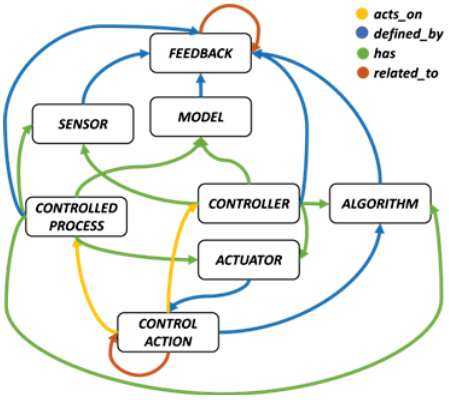

Fig. 1. Simplified version of the ontology model originally proposed by the authors to transform a knowledge graph into a STAMP model. Blocks represent category labels (classes), the colour code on arrows represent the relationship labels.

modified. *ACTUATOR* nodes mirror the *FEEDBACK* nodes, but they are *defined_by* *CONTROL ACTION* nodes instead. Relationships with label *related_to* are used to map the feedbacks and control actions transformation within the system. For example, a tank (*CONTROLLED PROCESS*) may be *defined_by* its internal pressure (*FEEDBACK*) and its. The tank *has* a barometer (*SENSOR*) that will be *defined_by* the actual pressure it measures (*FEEDBACK*). It is clear that the two feedbacks will be connected by a *related_to* relationship. Lastly, an additional relationship (i.e., *acts_on*) is used to close the loop and connect back a *CONTROL ACTION* to the *CONTROLLED PROCESS* it should modify.

Some clarifications are worth noticing with respect to the defined ontology model:

- In the SCS representation, a *CONTROLLER* can be also a *CONTROLLED PROCESS*. Accordingly, *CONTROLLER* and *CONTROLLED PROCESS* labels must share the same relationships with remaining entities (e.g., a CONTROL ACTION may *acts_on* a *CONTROLLER*, too, cf. Figure 1). On this basis, some nodes in the graph may be assigned to multiple labels.
- The ontology model does not admit the definition of a *CONTROLLER* if it is not connected through a *has* relationship with both a *MODEL* and an *ALGORITHM* node. It

is supposed that there is no capability of the controller in generating a control action if there is no information about what the controller knows (*MODEL*) and how it is using that information (*ALGORITHM*). Accordingly, both the controller model and algorithm must be connected in the SCS through (at least) an arrow representing the information that is "internally transferred" from the model to the process. This latter data is labelled as *FEEDBACK* and represents the process feedback as the controller perceived it.

- The ontology model allows for neglecting converter element if not relevant (i.e., *SENSOR* nodes and *ACTUATOR* nodes). The lack of a *SENSOR* node is relaxing the constraint of a directly inaccessible process parameter, that is, in this case, accessible and perfectly transferred to the controller. The lack of an *ACTUATOR* node is imposing the control action desired by the controller to be perfectly transferred to the controlled process.

Summarizing, eight different nodes labels (i.e., *CONTROLLED PROCESS*, *CONTROLLER*, *SENSOR*, *ACTUATOR*, *MODEL*, *ALGORITHM*, *FEEDBACK*, and *CONTROL ACTION*) and four relationships labels (i.e., *defined_by*, *has*, *related_to*, and *acts_on*) have been defined.

2.2.2. *Knowledge graph construction*

To construct a knowledge graph from the STAMP model, the labelled data must adhere to the ontology model outlined in Section 2.2.1. The resulting graph, denoted as G, is a data structure composed of a set of vertices (or nodes) and a set of edges connecting them (or representing relationships between nodes):

$$G(V, E) \tag{1}$$

Here, V represents the set of vertices V_n, where $1 < n < N$, and E represents the set of edges, where $1 < m < M$. Each vertex V_n is a multi-dimensional object with the following form:

$$V_n = (L_n^v, p_{in}^v) \ , \ 0 \leq i \leq I \tag{2}$$

In this form, L_n^v is the label of the n-th vertex (it can be *CONTROLLED PROCESS*, *CONTROLLER*, *SENSOR*, *ACTUATOR*, *MODEL*, *ALGORITHM*, *FEEDBACK*, or *CONTROL ACTION*), and $p_{1n}^v, p_{2n}^v, \dots, p_{In}^v$ are the properties assigned to the n-th vertex. The "v" in the apex is added to differentiate between vertices and edges. Since vertices can have multiple properties or none at all, i lies between 0 and I, in the ontology model presented in Section 2.2.1, $I = 0$. The set of all vertices V is defined as:

$$V = \{V_n\} \ , \ 1 \leq n \leq N \tag{3}$$

Similarly, a generic edge of the graph has the form:

$$E_m = (V_m' \ , V_m'', L_m^e, p_{jm}^e) \ , \ 0 \leq j \leq J \tag{4}$$

In this form, $V_m' \in V$ is the vertex from which the m-th edge starts, $V_m'' \in V$ is the vertex to which the m-th edge points, L_m^e is the label of the m-th edge (it can be *defined_by*, *has*, *related_to*, or *acts_on*), and $p_{1m}^e, p_{2m}^e, \dots, p_{jm}^e$ are the properties assigned to the m-th edge. Again, from the ontology model from Section 2.2.1, $J = 0$. The set of all edges E is defined as:

$$E = \{E_m\} \ , \ 1 \leq m \leq M \tag{5}$$

3. Case study

This section demonstrates the applicability of the ontological translation through a case study in the Oil & Gas industry. Even if for merely demonstration purposes, the case study is in line with the new directives of the European Commission (EU) for energy transition in the coming years. Specifically, Natural Gas processing has been used to apply the methodology as it is comparable to Crude Oil refining. The goal of gas processing is to convert raw wellhead gas into clean sales gas and sustainable power generation, which can be delivered to customers with minimal environmental impact. Additionally, the manuscript briefly describes the process of Natural Gas Storage, which involves storing natural gas in liquid or gaseous form in above-ground tanks. The suitability of each storage tank for specific applications is determined by its physical characteristics (porosity, permeability, retention capability) and economics (site preparation and maintenance costs, deliverability rates, and cycling ability). The process of the Natural Gas Storage plant works as follow (Mokhatab et al. 2014). The natural gas is

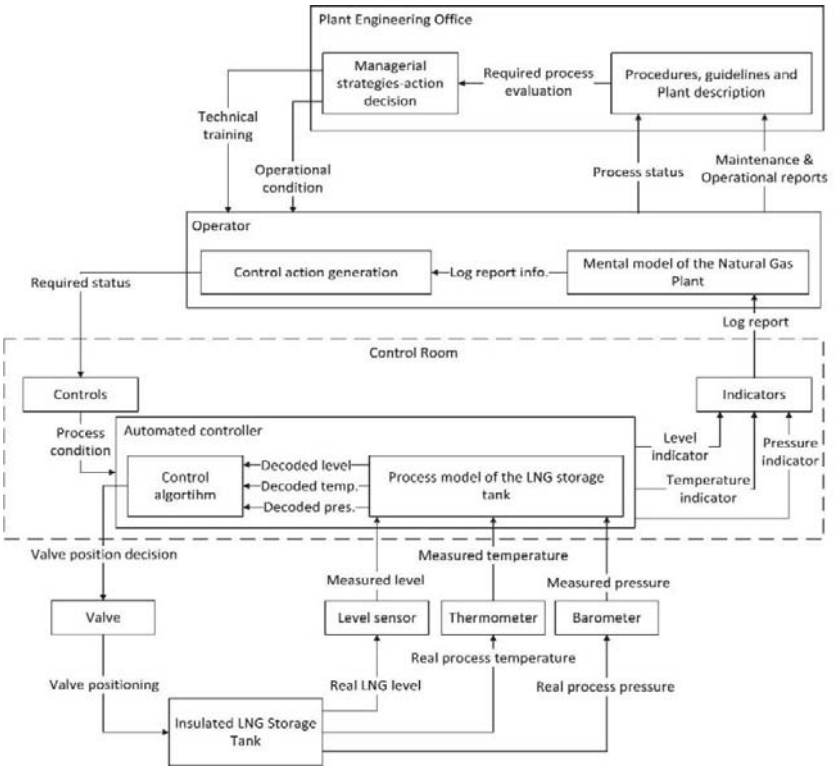

Fig. 2. STAMP model for the insulated LNG storage tank in the natural gas processing plant.

received at shallow temperatures while transferred to the storage tanks. Later, the gas passes through the pipelines that join the arms to the tanks and stored inside the tanks at a low temperature (between 105 – 115 K). Then, a compressor and a recondensing system collect the gas and convert it into liquid to be transferred in the pumping system. A pumping stage is used to transport the liquid natural gas from the storage tanks to the vaporized. Finally, the heat exchangers convert the liquid gas into gas to be pressurized (between 7 – 10 MPa) and deliver it into the supply pipeline.

The STAMP model has been limited to this process for exemplary purposes. Figure 2 depicts the STAMP model of an insulated tank and its components, along with its monitor and sensor technologies. The figure reports: (i) the Plant Engineering office responsible for setting and complying with the operational and technical aspects of the social and organizational requirements; (ii) Operators (defined by the control action generation and mental model); (iii) the Control Room, where controls, indicators, and automated controllers are included. The automated controller is programmed with a control algorithm for developing actions and it has a process model for understanding and knowing the system design and its processes. Finally, the industrial process includes (iv) the Valve, (v) the Insulated LNG storage Tank, (vi) the Level sensor, (vii) the Thermometer, and (viii) the Barometer. These system components are only an extracted part of the Natural Gas Storage organizational-equipment process.

3.1. *Results*

The SCS from Figure 2 is processed to tag each element (blocks and arrows) with the ontology model in Section 2.2.1, following the SCS connections, in order to obtain the following fields:

- From_node_label: containing the label assigned to the node from which the edge starts (i.e., L_n^v of V_m').

- From_node_value: containing the information in blocks and arrows of the SCS related to nodes from which relationship starts. This information is then inserted in nodes as a property p_{in}^{v} named "value".
- Relationship_label: containing the label of the relationship connecting the two nodes (i.e., L_m^e of E_m).
- To_node_label: containing the label assigned to the node to which the edge ends (i.e., L_n^v of V_m'').
- To_node_value: containing the information in blocks and arrows of the SCS related to nodes to which relationship ends. Similarly to From_node_value, this information is managed as a property p_{in}^{v} named "value".

An import algorithm enables the construction of the knowledge graph of the Oil & Gas process STAMP model (Simone et al. 2023). The resulting graph is reported in Figure 3. The knowledge graph representation of the STAMP model enables semi-quantitative and quantitative analyses based on network theory (Hernandez and Van Mieghem 2011). For example, one can highlight the possible paths leading to a modification of a specific *CONTROLLED PROCESS* node, or calculate network metrics to point at the most connected (may be most critical) nodes. If properly queried, the knowledge graph can also be used as a baseline for different safety analyses: e.g., by translating the SCS into a fault tree, or by highlighting cause-effect relationships to be evaluated semi-quantitatively through a Bayesian Network. The knowledge graph representation of the system can also be updated with real time process data and serve as a digital twin of the system itself, permitting the continuous monitoring of the process in a STAMP-based perspective.

4. Conclusion

This manuscript presents a methodology for analyzing system interactions and identifying safety hazards using the STAMP model as a instrument to deploy knowledge graphs. This methodology has practical implications for the Oil & Gas sector and other industries where safety is crucial. In this case, ontology models are key in achieving a shared understanding of a domain, reducing ambiguity and promoting consistency in

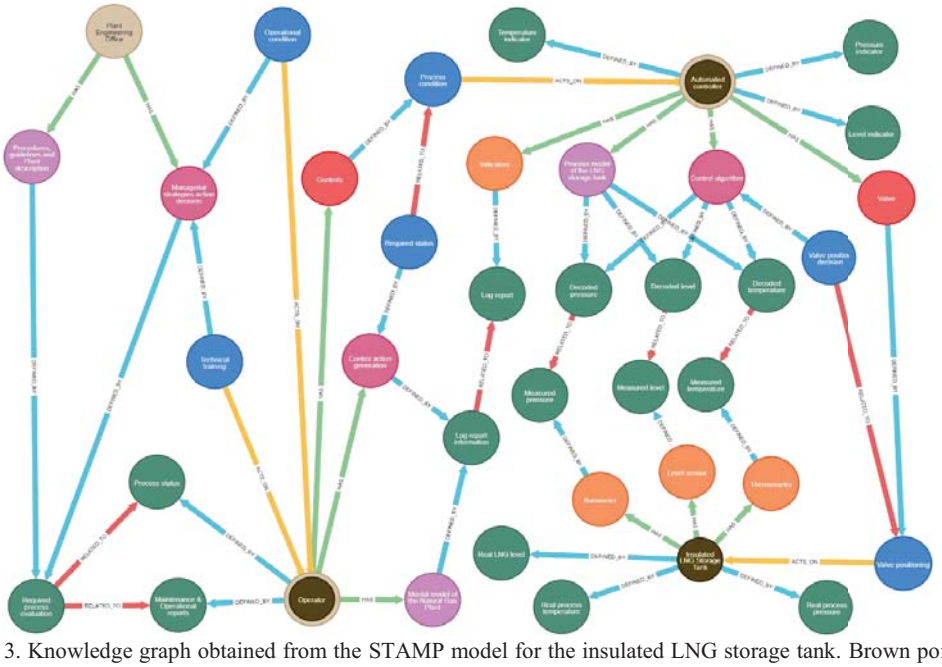

Fig. 3. Knowledge graph obtained from the STAMP model for the insulated LNG storage tank. Brown points represent *CONTROLLED PROCESS*, beige points represent *CONTROLLER* nodes, dark green points represent *FEEDBACK* nodes, dark blue points represent *CONTROL ACTION* nodes, purple points represent *MODEL* nodes, pink points represent *ALGORITHM* nodes, orange points represent *SENSOR* nodes, and red points represent *ACTUATOR* nodes. The edges colours follow the same scheme as the one reported in Figure 1.

information representation, and facilitating better communication and interoperability between different systems. The main drawbacks of this approach are related to the gathering of data. The knowledge graph needs to be populated with data to make analyses possible. Anyway, a knowledge graph is indeed by definition incomplete, i.e., it is not able to capture the entire knowledge on the system, but it can be progressively revised to this extent. Further discussion should consider different possibilities for its validation (Huaman, Tauqeer, and Fensel 2021).

Although the proposed ontology model does not include subclasses, it serves as a basis for further detailing nodes through other ontologies, enabling the creation of a multi-layer knowledge graph that promotes more effective data integration and knowledge discovery. Future works may connect the proposed ontology with industry standards (e.g., UML) by considering the entities defined in this paper as higher level classes of other ontologies. Relying on a formal structure for representing knowledge, it becomes possible to link and analyze data from multiple sources, uncovering new insights usually discoverable only in retrospect.

Acknowledgement

The authors would like to acknowledge the support of INAIL (National Institute for insurance Against Accidents at Work). This work is partially funded through the project "RE-SET: Resilience Engineering for Safe Energy Transition" under the INAIL BRIC 2022 funding scheme.

References

Baxter, G, and I Sommerville. 2011. "Socio-Technical Systems: From Design Methods to Systems Engineering." *Interacting with Computers* 23 (1): 4–17.

Dekker, S. 2019. *Foundations of Safety Science: A Century of Understanding Accidents and Disasters*. *Routledge*.

Hernandez, Javier Martin, and Piet Van Mieghem. 2011. "Classification of Graph Metrics." *Delft University of Technology: Mekelweg, The Netherlands*, 1–20.

Huaman, E, A Tauqeer, and A Fensel. 2021. "Towards Knowledge Graphs Validation Through Weighted Knowledge Sources." *Communications in Computer and Information Science* 1459 CCIS: 47–60.

Leveson, Nancy. 2004. "A New Accident Model for Engineering Safer Systems." *Safety Science* 42 (4): 237–270.

Leveson, Nancy. 2012. *Engineering a Safer World: Systems Thinking Applied to Safety*. *Choice Reviews Online*. The MIT Press.

Leveson, Nancy. 2018. *STPA Handbook*. The MIT Press.

Lin, H K, and J A Harding. 2007. "A Manufacturing System Engineering Ontology Model on the Semantic Web for Inter-Enterprise Collaboration." *Computers in Industry* 58 (5): 428–437.

Mannan, S. 2012. *Lees' Loss Prevention in the Process Industries: Hazard Identification, Assessment And Control: Fourth Edition*. *Lees' Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control: Fourth Edition*. Vol. 1–2.

Mokhatab, S., J. Mak, J. Valappil, and D. Wood. 2014. *Advances and Innovations in LNG Industry*. *Handbook of Liquefied Natural Gas*.

Nakhal A., A J, R Patriarca, M Tronci, P Agnello, S M Ansaldi, and A Ledda. 2022. "A STAMP Model for Safety Analysis in Industrial Plants." *Chemical Engineering Transactions* 91: 403–408.

Newman, Mark. 2010. *Networks: An Introduction*. *Networks: An Introduction*. Oxford University Press.

Pasman, H J. 2009. "Learning from the Past and Knowledge Management: Are We Making Progress?" *Journal of Loss Prevention in the Process Industries* 22 (6): 672–679.

Patriarca, Riccardo, Mikela Chatzimichailidou, Nektarios Karanikas, and Giulio Di Gravio. 2022. "The Past and Present of System-Theoretic Accident Model And Processes (STAMP) and Its Associated Techniques: A Scoping Review." *Safety Science* 146 (105566). Elsevier Ltd.

Pereira, Daniel Patrick, Celso Hirata, and Simin Nadjm-Tehrani. 2019. "A STAMP-Based Ontology Approach to Support Safety and Security Analyses." *Journal of Information Security and Applications* 47. Elsevier Ltd: 302–319.

Righi, A W, T A Saurin, and P Wachs. 2015. "A Systematic Literature Review of Resilience Engineering: Research Areas and a Research Agenda Proposal." *Reliability Engineering and System Safety* 141: 142–152.

Simone, Francesco, Silvia Maria Ansaldi, Patrizia Agnello, and Riccardo Patriarca. 2023. "Industrial Safety Management in the Digital Era: Constructing a Knowledge Graph from near Misses." *Computers in Industry* 146: 103849.